LED Brush Pro

Customizable LED for Light Painting Photography

Kyoungyeon Cho
Seungjun Cho
Jihong Min

Design Review for ECE 445, Senior Design, Spring 2017
TA: Eric Clark

24 February 2017
Project No. 42
**Table of Contents**

1. Introduction .......................................................................................................................... 3
   1.1. Objective ......................................................................................................................... 3
   1.2. Background ..................................................................................................................... 3
   1.3. High Level Requirements ............................................................................................... 3

2. Design .................................................................................................................................. 4
   2.1. Block Diagram ................................................................................................................ 4
   2.2. Physical Design .............................................................................................................. 5
   2.3. Block Design .................................................................................................................. 5
       2.3.1. Functional Overview ............................................................................................... 5
       2.3.2. Requirements and Verification ............................................................................... 9
       2.3.3. Supporting Material .............................................................................................. 14
   2.4. Power Consumption Analysis ....................................................................................... 21
   2.5. Tolerance Analysis ........................................................................................................ 22

3. Cost and Schedule .............................................................................................................. 22
   3.1. Cost Analysis ................................................................................................................ 22
   Adafruit NeoPixel Digital RGB LED Strip 144 LED ............................................................. 22
   3.2. Schedule ....................................................................................................................... 24

4. Ethics and Safety .................................................................................................................. 25

5. References ............................................................................................................................ 25
1. Introduction

1.1. Objective

Light painting is a photographic technique in which a photographer manipulates a hand-held light source, as if it were a paintbrush, to illuminate a subject during a long exposure photograph. This photography practice can produce masterpiece artworks for various purposes.

However, in order to excel in light painting, the photographer must expose just the right among of light in different places for aesthetic effects [1]. As with many other photography tools, the equipment used in traditional light painting are often excessively expensive and ungainly large.

Our client Rick wants us to design a cost effective, portable LED paint brush that can alter its light settings in delicate detail. The requested functions include: the ability to control the fade/gradient of the light, brightness of the light, and location of the light within the case. Some additional functionalities we would like to add to the project are light temperature variation, and remote controllability. We envision our product to be utilized by many photographers throughout the world to generate jaw dropping light paintings efficiently and cheaply.

1.2. Background

Our client Rick is an engineering minded photographer who likes to solve real world problems. When he started to light paint cars, he ran into a problem because he couldn't find an affordable light source with a light gradient. In an effort to solve the problem, he bought a 4ft long LED light for $40. Then he took a non-technical approach to creating a gradient; he piled a number of diffusion papers between the LED strip and the outer plexi-glass to improvise a fade.

The prototype was effective in producing several quality automobile photographs. However, the prototype was lacking in many ways. In order to change the fade of the LED, the user had to manually allocate multiple layers of diffusion papers in the desired locations. Also, the brightness was not adjustable. Rick also said the power cable could be cumbersome, especially in extremely dark environments where light painting takes place. Also, the power cable limits outdoor light painting practices. Our group will put our head together in order to solve these real-world problems for our client, Rick.

1.3. High Level Requirements

- We must be able to control the brightness and color of each segment of the LED strip in precise detail.
- The LED state updates must be fast and smooth enough such that the blinking is not visible to the human eye, or a camera during a long exposure photograph.
- We must be able to power the LED system using a rechargeable battery for at least 15 minutes.

2. Design

2.1. Block Diagram

Figure 2.1.1 shows the block diagram of the overall lighting system. In high-level, the system is designed into four main subsystems: microcontroller, power system, user interface, and light system. Each subsystem communicate data and transfers or receive energy other subsystems to perform specific tasks necessary to allow the user to easily and effectively control the lighting system.

The power system is responsible for regulating the input voltage and to provide power to various components in the overall system. It also stores energy in NiCd battery and charges the battery if the battery charging circuit is connected to the 110V wall outlet. Furthermore, the user interface
subsystem reads user input through various sensors and Android software interface, and it converts the data into the form that the microcontroller can interpret.

The microcontroller subsystem contains the Atmega328p chip and Bluetooth transceiver module, and it interacts with user interface and light system to read user input, process the input data, and control the digital LED light strips accordingly. The light system receives power from power system and is controlled mainly by the microcontroller to display appropriate pattern, color, and brightness of the light, specified by the user input.

### 2.2. Physical Design

The light casing for photography module shown in figure 2.1.1 corresponds to the physical design of the light system, as it will contain the LED strips and will diffract lights, so that the individual pixels will not be distinguishable.

### 2.3. Block Design

#### 2.3.1. Functional Overview

#### 2.3.1.1. Light System

The light System is composed of the digital LED strip module and the LED case module. Our client’s prototype LED strip was analog and therefore would be constantly on full brightness. The fading effect was met by blocking out the light with multiple layers of diffusion papers. Also, the number of modes of the system was limited and changing gradient options required manual labor of moving around diffusion paper. Using an individually addressable digital LED strip allows
us to enjoy more lighting modes while being more power efficient by turning off unnecessary LEDs.

**Digital LED Strip Module**

The microcontroller determines the temperature, gradient, location, and brightness of the densely packed WS2812B digital LED strip. In order to make the gradient changes as smooth and specific as possible, we chose the most densely packed LED with 144 LEDs/m. Since our system will be 1.2192m (4ft) long, 175 LEDs will fit into our system.

**LED Case Module**

For light painting, it is crucial that the lighting is smooth and not pixelated, and therefore the LED case will have diffusion papers and plexiglass to diffuse the lights. Also our client wanted the case to have a handle at one end such that a wheeled LED holder can clamp on it.

2.3.1.2. Power System

**NiCd Battery Charger Module**

Using the MAX713 NiCd Battery fast-charge controller, we will be able to fully charge our NiCd batteries in 2.5 hours. Our Battery fast charges at a standard charge rate of C/4. Nonetheless, overcharging the NiCd battery can still be hazardous. However, the MAX713 chip can terminate fast charge by detecting when the voltage slope becomes negative.

**NiCd Battery Module**

Our light system consumes about 60W total as calculated in the power consumption analysis section below. Therefore our high-rate NiCd battery must supply 6V at a maximum current of 10A for at least 15 minutes (6V 2.5Ah). We chose the NiCd battery over the lead-acid battery and the lithium-ion battery because NiCd batteries are more stable and sturdy than li-ion batteries. Also, the NiCd battery is more charge dense than the lead-acid battery, making it smaller and lighter, perfect for our application.

**DC/DC Buck Converter Module**

The TPS53353 buck converter takes 6V input from the NiCd battery and supplies a steady 5V to the LED strip and a few other modules. The buck converter can handle a maximum output current of 20A, but for our application, we only need to draw 11A. We used a buck converter instead of a linear regulator so that this photography lighting tool can be used steadily for a long time.

**Linear Regulator Module**

The TPS79901 linear regulator takes a regulated 5V input from the buck converter and outputs a regulated 3.3V to the microcontroller and the Bluetooth transceiver. The regulator can output up to 200mA, but we only need a maximum output current of 60mA. We used a linear regulator
instead of a buck converter in this case because the overall power needed for the microcontroller system is very low.

2.3.1.3. Microcontroller

The main functionality of the microcontroller subsystem is to read the input data from physical sensors and Bluetooth transceiver and process the input data using software to update the state of the overall system. After updating the state in the software, the microcontroller communicates with the digital LED strip to control the pattern, brightness, and color of the displayed light. This subsystem contains Atmega328p microcontroller & software module and Bluetooth transceiver module.

**Atmega328p Microcontroller & Software**

This module consists of the Atmega328p microcontroller chip and the software encoded on the microcontroller. The main functionality of this module is to read digital and analog data from other modules and process them appropriately using software and send serial digital data to the digital LED strip to control the properties of the light. The state of the system will be organized into a struct that contains the starting/ending location of the light gradient, overall brightness, color, current light pattern, and other variables necessary to control the LED strip. The software will update the values of these state variables according to the user input data read from the analog and digital pins. Then, it constructs and calculates an array of RGB input values for each LED driver chip using these state variables, and sends these calculated values to the LED strip.

Another functionality of the microcontroller is to store the state of the light system by interfacing with the on-board flash memory, so when the user restarts the machine, the microcontroller can set its state to the saved state memory to allow user to continue using the latest used setting.

**HC-05 Bluetooth Transceiver Module**

This module is connected to and is controlled by the microcontroller chip and the software. When this module is turned on, this module will be able to communicate with other devices using Bluetooth LE signal wirelessly, allowing the user to communicate remotely with the light system via Android device.

The Bluetooth transceiver module mainly communicates data with Android device via TCP. In the Bluetooth protocol, the Android device will serve as the master device and this transceiver will be the slave device. While this device is not connected to any Android device, it will become
discoverable for any Android device to request connection. Once it is connected to the Android device, this module will be used to send system’s current state to the Android application and to receive the user input data generated in the application.

2.3.1.4. User Interface

This subsystem includes any component that involves direct interaction with the user. The types of sensors and other modules included in this subsystem were chosen to promote straightforward and effortless interaction with the system to control the displayed light.

Rotary Encoder

The rotary encoder module is a circuit that combines the rotary encoder and other components to prepare the encoder data to be readable by the microcontroller module. This 12-step rotary encoder features the rotation without a limit on the maximum turning angle. There will be four total rotary encoder modules in the overall system. Each one is used to control the overall brightness, color temperature, and gradient starting/ending locations.

One alternative to using rotary encoder was to use the rotary potentiometer to sense how much the user has turned the knob. However, this option limited the maximum rotation angle. Since our system uses other inputs, such as Bluetooth signal, to change the system’s state, a sensor that can rotate without limit was desirable.

Power Switch & Buttons

This module includes circuits involving the power switch and buttons that allow the microcontroller to read the states of the power switch and buttons via voltage value. The power switch will be used to turn the light on and off based on the location of the switch. When the power switch indicates that the light should turn off, the microcontroller will immediately save the current state of the system. Also, if the switch indicates that the LED’s should turn on, the microcontroller will load the most recently used state. Furthermore, the buttons will mainly be used to switch the light temperature. There will be three buttons total, where each button will indicate specific temperature such as natural light, etc.

Android Smartphone & Applications

The android application will feature UI’s such as multiple sliders, battery status indicator, load/save buttons to allow the user to interact with the light system via richer and intuitive control, remotely. The application will collect data from these UI elements and sends the data to
the microcontroller module using Bluetooth socket, so the microcontroller can control the LED strip accordingly.

This module will also periodically receive the current state of the light system from the microcontroller. It will use this information to update items on the application screen, including the battery status indicator and the light pattern expected to be displayed by the light system module.

**Battery Status Bar**

The LED battery status bar indicates the remaining charge in the battery. Total five LED’s will be used to display the battery status. For example, when the battery is fully charges, all five LED’s will turn on, and it will turn on less number of LED’s as the remaining charge decreases. The circuit used to detect the remaining charge takes the output voltage of battery as the input. Then, it will use five comparators to check whether the output voltage is greater than certain values, such as 0, 20%, 40%, etc. Then, the output of these comparators will be connected to the digital input pins of the microprocessors, which will control the LED status bar based on these input values. The color of the LED’s will be green if the remaining charge on the battery is greater than 60% of capacity, or when the battery is charging, and it will turn red if it is less than 40% and yellow otherwise.

However, the lights from the status may disturb the photo as the camera might pick up the light during the light painting process. So, one of the buttons from the power switch & buttons module will be used to allow the user to turn the battery status bar on and off.

**2.3.2. Requirements and Verification**

<table>
<thead>
<tr>
<th>Component</th>
<th>Requirement</th>
<th>Verification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Battery Charger</td>
<td>Requirement 1: The temperature of the circuit will be between 0°C and 40°C while charging the batteries at a charge rate of 0.4C.</td>
<td>Verification 1: We will use a thermometer to measure the temperature of the IC during a full charge cycle and make sure the temperature never exceeds 40°C.</td>
</tr>
<tr>
<td></td>
<td>Requirement 2: The charger must fully charge the battery within 2.5 hours.</td>
<td>Verification 2: After fully discharging the batteries, we will charge the batteries for 2.5 hours. We will probe the voltage of the battery</td>
</tr>
<tr>
<td>Component</td>
<td>Requirement 1</td>
<td>Verification 1</td>
</tr>
<tr>
<td>----------------------------</td>
<td>--------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NiCd Battery</td>
<td><strong>Requirement 1:</strong> The 6V battery must store at least 2.5Ah of charge to keep the system continuously powered.</td>
<td><strong>Verification 1:</strong> We will discharge the battery at 10A for 15 minutes and ensure that battery voltage never drops below 5V.</td>
</tr>
<tr>
<td></td>
<td><strong>Requirement 2:</strong> The battery pack must be mobile enough to be mounted on the led case. It must weigh less than 1kg.</td>
<td></td>
</tr>
<tr>
<td>Linear Voltage Regulator</td>
<td>Requirement 1: The voltage regulator must provide 3.3V (+/-10%) from a 5V buck converter output.</td>
<td><strong>Verification 1:</strong> Draw 60mA from the linear voltage regulator from a 5V input for 10 minutes. Make sure that the output voltage stays between 2.97V and 3.63V.</td>
</tr>
<tr>
<td></td>
<td>Requirement 2: The voltage regulator must maintain a thermal stability below 125°C.</td>
<td></td>
</tr>
<tr>
<td>DC/DC Buck Converter</td>
<td>Requirement 1: The buck converter must provide 5V (+/-10%) from a source that varies from 7V ~ 5V.</td>
<td><strong>Verification 1:</strong> Draw 11A from the buck converter from inputs 5V and 7V for 10 minutes each. Make sure that the output voltage stays between 4.5V and 5.5V.</td>
</tr>
<tr>
<td></td>
<td>Requirement 2: The buck converter must maintain a thermal stability of below 85°C.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Requirement 3: The buck converter must handle a peak current draw of 11A</td>
<td></td>
</tr>
<tr>
<td>Atmega328p Microcontroller &amp; Software</td>
<td>Requirement 1: The delay between the change in user input and microcontroller starting to write data to the LED strip must be less than 400ms for all user input including the android application.</td>
<td><strong>Verification 1:</strong> Program the microcontroller program to timestamp the time at which it reads a specific message from the smartphone and the time at which it starts to write data to the LED strip. Also, program it to send the measured time difference between the timestamps to the smartphone</td>
</tr>
<tr>
<td><strong>HC-05 Bluetooth Transceiver</strong></td>
<td>Requirement 2: When changing the LED’s state, the time it takes for the overall LED strip to fully change its state should be less than 100ms.</td>
<td>via Bluetooth. Then, using the smartphone, send the specified message, and once it receives the data, check to see if the time is less than 400ms. Verification 2: Program the microcontroller to timestamp immediately before it starts to write data to LED strip and immediately after it finishes writing the data. Also, program it to send the difference between the two timestamps to the smartphone once. Then, check to see if the measured time difference is less than 100ms.</td>
</tr>
<tr>
<td><strong>Rotary Encoder</strong></td>
<td>Requirement 1: The communication with the Android smartphone should not fail within the distance of 30ft.</td>
<td>Verification 1: Write microcontroller program to send a test message every second to the Android application, which should display the test message on screen. While the program is running, slowly move the smartphone away from the transceiver until the smartphone fails to receive a message. Check if the distance at which the transmission failed is greater than 30ft. Verification 2: Turn on the system. Then, turn the gradient endpoint encoder exactly 10 degrees to the right. Check to see if the gradient endpoint has moved to the right by exactly 1 pixel. Verification 2: Tape long string on to the encoder cap. Then, using forcemeter, determine the force required to rotate the encoder. Check to see if the measured value is less than 5N.</td>
</tr>
<tr>
<td>Requirement 2: Turning 10 degrees in the gradient location encoder should move the starting or endpoint of the gradient by one pixel. Requirement 2: To allow effortless control of rotary encoder, the minimum force necessary to turn the rotary encoder needs to be low, and less than 5N.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Power Switch & Buttons | **Requirement 1**: The power switch and buttons circuits should output LOW (0V) signal when pressed or switched on, and output digital HIGH (5V) signal otherwise.  

**Requirement 2**: The switch and buttons should be debounced, and once the push button is pressed, the changed value should be kept for at least 0.75ms (see calculations) so that the microcontroller is guaranteed to sense the changed state of the button. | **Verification 1**: Using the multimeter, measure the voltage output of the power switch circuit and button circuit. Check that the voltage output is 0V if power switch is on and button is pressed. Also, check that the voltage output is 5V otherwise.  

**Verification 2**: Connect the switch and button circuits to the 5V power source. Using oscilloscope, measure the voltage curve over time when the switch is triggered and button is pressed. Check that for both sensors, there is no bouncing. Also, check that the changed value of the button’s output is kept for at least 0.75ms. |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Android Smartphone &amp; Application</td>
<td><strong>Requirement 1</strong>: The android application must be able to correctly display the battery status and state of the light system, such as gradient pattern, color, and brightness.</td>
<td><strong>Verification 1</strong>: Run the application, and set the light to arbitrary pattern. Then, compare the expected displayed pattern on the application and the actual pattern displayed to check that the patterns, including color, gradient, and brightness, match. Also check to see if the battery status indicator shown in the application matches the LED battery status bar on the light system.</td>
</tr>
<tr>
<td>Battery Status Bar</td>
<td><strong>Requirement 1</strong>: The circuit needs to be able to map the voltage output of the NiCd battery into a value in 0-5V range, where 5V should indicate fully charged battery, while 0 indicates fully drained battery, with a tolerance of ±10%.</td>
<td><strong>Verification 1</strong>: Fully charge the battery. Then, connect the battery status circuit to the battery, which should connected to the overall light system to power the LED light at maximum intensity. Using the oscilloscope, measure the output voltage of the circuit’s over time until the battery is fully drained. Lastly, check that the shape of the voltage curve matches voltage</td>
</tr>
</tbody>
</table>
**Requirement 2:** The number of LED’s turned on must be proportional to the remaining charge in the battery. That is, if some percentage $x$ of the battery is remaining, only the rounded integer value of $x$ percent of the total number of LED’s should be turned on to indicate how much battery is actually remaining.

Verification 2: Generate the voltage curve of the battery status circuit using the method described in verification 1. While generating the voltage curve, mark the time at which the number of LED’s turned on changes, along with the number of LED’s turned on. Then, check that the ratio of number of LED’s turned on immediately after the marked time to total number of LED’s matches the ratio of the marked time to the time it takes to fully discharge the battery.

<table>
<thead>
<tr>
<th>Digital LED Strip</th>
<th>Requirement 1: We must be able to control the white light temperature from 3000k to 5600k.</th>
<th>Verification 1: We will set our LEDs to different light temperatures and use a mobile phone application that measures light temperature to verify that our LED system can output the desired light temperatures.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Light Casing</td>
<td>Requirement 1: The lights produced by the LEDs must be fully diffused when it reaches the surface of the housing such that it is impossible to spot a single LED. Requirement 2: The case should be 4ft long, and 4 inches wide, and lighter than 5kg.</td>
<td>Verification 1: We will make sure that we can’t see the separation between any individual pixels when all the LEDs are turned on. Verification 2: We will use a measure tape to measure the dimensions of our LED case. Also, we will use a weight scale weight scale to make sure that our case is lighter than 5kg.</td>
</tr>
</tbody>
</table>
2.3.3. Supporting Material

Calculations for TPS53353 Buck Converter Circuit

According to [2], the ripple current for the circuit shown in figure 2.3.3.4 can range from \( \frac{1}{4} \) to \( \frac{1}{2} \) and is determined by the inductance \( L \). Also, increase in ripple current decreases the SNR (signal-to-noise ratio) but increases the energy loss. The inductance value is related to the output voltage by

\[
L = \frac{1}{L_{\text{IND}} f_{\text{SW}}} \cdot \frac{(V_{\text{IN}} - V_{\text{OUT}}) V_{\text{OUT}}}{V_{\text{IN}}}.
\]

In order to achieve both adequate SNR and adequate energy loss, the ripple current is chosen to be \( \frac{1}{3} \). Since \( I_{\text{IND}} = \frac{I_{\text{OUT}}}{3} \), the value of the inductance becomes

\[
L = \frac{3}{I_{\text{OUT}} f_{\text{SW}}} \cdot \frac{(V_{\text{IN}} - V_{\text{OUT}}) V_{\text{OUT}}}{V_{\text{IN}}}.
\]

According to the data sheet [2], \( f_{\text{SW}} = 500\text{kHz} \). Also, the desired output voltage needed for other components in the overall light system is 5V, while the input voltage is 6V, specified by the battery voltage. Also, since the output current needed is \( I_{\text{OUT}} = 11\text{A} \), the inductance can be calculated

\[
L = \frac{3}{11\text{A} \cdot 500\text{kHz}} \cdot \frac{(6\text{V} - 5\text{V}) \cdot 5\text{V}}{6\text{V}} = 0.000455\text{mH}.
\]

Then, we can calculate the ESR using

\[
ESR = \frac{V_{\text{OUT}} \cdot 10\text{mV} \cdot (1 - D)}{0.6\text{V} \cdot I_{\text{IND}}} = \frac{10\text{mV} \cdot L \cdot f_{\text{SW}}}{0.6\text{V}} = \frac{L \cdot f_{\text{SW}}}{60},
\]

where \( D \) is the duty factor.

Since \( L = 0.000455\text{mH} \), and \( f_{\text{SW}} = 500\text{kHz} \), we obtain

\[
ESR = \frac{0.000455\text{mH} \cdot 500\text{kHz}}{60} = 0.00379\Omega.
\]

According to [2], the recommended value for \( R_2 \) is between 1k\( \Omega \) and 20k\( \Omega \). So, 10k\( \Omega \) was chosen as the value for \( R_2 \). Furthermore, \( R_1 \) can be calculated using the equation
\[ R_1 = \frac{V_{OUT} - I_{IND} \cdot ESR}{0.6} \cdot R_2. \]

Since \( V_{OUT} = 5V \), \( I_{IND} = \frac{I_{OUT}}{3} = \frac{11A}{3} = 3.667A \), and \( ESR = 0.00379\Omega \), the \( R_1 \) is calculated as

\[ R_1 = \frac{5V - 3.667A \cdot 0.00379\Omega}{0.6} \cdot 10k\Omega = 73.218k\Omega. \]

Lastly, to calculate component values for the circuit shown in figure 2.3.3.4 we need to calculate the value of \( R_{TRIP} \) to specify the overcurrent value at which the controller restarts after a hiccup delay. The resistance \( R_{TRIP} \) can be calculated using

\[ R_{TRIP} = \frac{\left( I_{OCP} - \frac{1}{2L_f SW} \right) \cdot \left( \frac{V_{IN} - V_{OUT}}{V_{IN}} \right) \cdot 32 \cdot R_{DS}}{I_{TRIP}}. \]

In order to prevent the current from damaging the components of the circuit, the overcurrent value, \( I_{OCP} \) was chosen to be 20A. According to [2], for \( I_{OCP} = 20A \), \( I_{TRIP} = 10\mu A \), and \( R_{DS} = 1.6m\Omega \). Using these values, \( R_{TRIP} \) can be calculated as

\[ R_{TRIP} = \frac{20A - \frac{1}{2 \cdot 0.000455mH \cdot 500kHz} \cdot \left( \frac{6V - 5V}{6V} \cdot 5V \right) \cdot 32 \cdot 1.6m\Omega}{10\mu A} = 93k\Omega. \]

**Calculation for Resistance Values in Battery Status Circuit**

Using the 6V NiCd battery’s voltage discharge curve, the table 2.3.3.1 shown below is constructed.

<table>
<thead>
<tr>
<th>Charge Remaining (%)</th>
<th>Battery Voltage (V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 25</td>
<td>6.0</td>
</tr>
<tr>
<td>&lt; 50</td>
<td>6.25</td>
</tr>
<tr>
<td>&lt; 75</td>
<td>6.4</td>
</tr>
<tr>
<td>&lt; 100</td>
<td>7.0</td>
</tr>
</tbody>
</table>

Table 2.3.3.1 Battery Voltage over Range of Charge Remaining
Figure 2.3.3.6 shows the circuit used to detect the remaining charge of the battery by comparing current voltage output of the battery to some preset voltage values. Input voltages for $V_{in2}$ pins of the LM399D comparator IC circuits are the voltage values at various charge remaining in battery. Since these pins are connected to the voltage divider circuit, the resistance values need to be appropriately adjusted to match the values shown in table 2.3.3.1. Let $V_i$ be the $V_{in2}$ value of the i-th LM399D chip from the bottom of figure 2.3.3.6.

The figure 2.3.3.7 below is a PSPICE simulation showing at which battery voltage each LED turns on. As presented in the graph, all four LEDs are on when the battery output is 7 volts and they slowly switch off one by one as the Battery voltage decreases.

**Calculation for Atmega328p Expected Software Execution Time per Loop**

Atmega328p microcontroller uses the AVR-RISC architecture. According to [3], the AVR architecture uses 16-bit instruction at minimum, and many of the instructions used are 16-bit. Since the program memory is 32KB [4], the microcontroller can store at maximum

$$\text{Max. # of Instructions} = \frac{\text{Memory Size}}{\text{Bytes per instruction}} = \frac{30\text{KB}}{2\text{B per instruction}} = 15000 \text{ instructions}.$$  

This calculation assumes that there is no branch/loop instruction in the code. However, even if there is a loop, it will usually be a small portion of the code, and the effect will likely be negligible. Also, since the microcontroller software using more than 70% portion of the program memory for this application is highly unlikely, this value serves as an adequate upper bound on the number of total instructions.

Also, [4] states that the CPU speed of Atmega328p is 20 MIPS. So, the maximum time the program can take per loop is

$$T_{max} = \frac{\text{Number of Instructions}}{\text{CPU Speed}} = \frac{15000 \text{ instructions}}{20\text{MIPS}} = 0.75\text{ms}.$$  

**NiCd Battery Charger Circuit Parameter Calculations**

The calculations below are based on the guidelines presented in the MAX713 NiCd Batter charger datasheet [5]. We need to charge 5 1.2V NiCd battery cells and therefore the minimum input voltage is

$$V_i = \# of cells \times 1.9 + 1.5V = 5 \times 1.9 + 1.5V = 11V.$$
R1 in the schematic powers the chip and it is

\[ R_1 = \frac{V_{i,\text{min}} - 5V}{0.005A} = \frac{11V - 5V}{0.005A} = 1k\Omega. \]

In order to decide on the fast charging current, we need to take into consideration the battery capacity and the desired charge time. Our battery capacity is 2500mAh and we want to charge it in 2.5 hours, at a fast charge rate of 0.4C.

\[ I_{\text{fast}} = \frac{\text{battery capacity}}{\text{charge time}} = \frac{2500\text{mAh}}{2.5\text{hours}} = 1A. \]

\( R_{\text{sense}} \) sets the fast charge current into the battery.

\[ R_{\text{sense}} = \frac{0.25V}{I_{\text{fast}}} = \frac{0.25V}{1A} = 0.25\Omega. \]

Since there are 5 cells, PGM1 should be connected to V+ and PGM0 should be open. Also, our charge time is 2.5 hours, and according to the datasheet, the voltage-slope cut off is enabled, then disabled when the voltage increase halts [5]. Next, the PNP BJT power dissipation is

\[ PD_{\text{pnp}} = (V_{i,\text{max}} - V_{\text{minimum battery}}) \times I_{\text{fast}} = (11V - 5V) \times 1A = 6W. \]

According to the 2N6109 PNP transistor datasheet, the maximum power dissipation of the transistor is 40W [5]. Since the maximum power dissipation is well above the power dissipation of our PNP transistor, the PNP transistor will operate well.

Figure 2.3.3.1. Schematics of Linear Voltage Regulator Circuit
Figure 2.3.3.2. Schematics of Atmega328p Chip Layout

Figure 2.3.3.3. Schematics of the Rotary Encoder Circuit
Figure 2.3.3.4. Schematics of DC/DC Buck Converter Circuit

Figure 2.3.3.5. Schematics of NiCd Battery Charging Circuit
Figure 2.3.3.6. Schematics of the Battery Status Bar Circuit
2.4. Power Consumption Analysis

**Digital LED Strip Module**

There are 175 LEDs in the LED strip and they each require a maximum of 60mA/LED. Therefore the total required current is

\[
\text{Total Current} = \# \text{ of LEDs} \times \text{current per LED} = 10.5A.
\]

The digital LED strip runs on 5V. At full brightness, the maximum power consumption of the LED strip is

\[
P_{\text{consumed}} = V \times I = 5V \times 10.5A = 52.5W
\]

**DC/DC Buck Converter Module**

The TPS53353 buck converter circuit must output 5V at a maximum of 11A. According to the datasheet, the efficiency of the buck converter is higher than 92% and therefore the maximum power consumption of the buck converter is

\[
P_D = V_o \times I_o \times (1 - \eta) = 5V \times 11A \times (1 - 0.92) = 4.4W.
\]

**Linear Regulator Module**

The TPS79901 linear regulator takes a regulated input voltage of 5V from the output of the buck converter and outputs 3.3V. The max current drawn is about 60mA. Therefore the maximum power consumption is

\[
P_{\text{consumed}} = (V_i - V_o) \times I_o = (5V - 3.3V) \times 60mA = 102mW.
\]

**Microcontroller Module**

The Atmega328P micro controller will be powered with 3.3V and it draws a maximum current of 21.12mA. Therefore the maximum power consumption is

\[
P_{\text{consumed}} = V \times I = 3.3V \times 21.12mA = 69.7mW.
\]

**Bluetooth Transceiver Module**

The HC-05 Bluetooth chip will be powered with 3.3V and it draws a maximum current of 40mA. Therefore the maximum power consumption is

\[
P_{\text{consumed}} = V \times I = 3.3V \times 40mA = 132mW.
\]
Total Power Consumption

Under the most conservative estimations, the maximum total power consumption of our system is below 60W.

2.5. Tolerance Analysis

One of the fundamental problems with using the PWM controlled LED’s for the photography is that the camera may catch the LED suddenly turning on and off, possibly causing unnecessary noise in the photo. The operating frequency of the WS2811 is 400kHz [6]. So, the maximum time period for which the camera may catch LED turning off is

\[ T = \frac{1}{f} = \frac{1}{400\text{kHz}} = 2.5\text{ns}. \]

So, in order to guarantee that the camera doesn’t sense the LED’s blinking, the exposure needs to be greater than \( T = 2.5\text{ns} \). However, one of the fundamental characteristics of the light painting technique is that the exposure of the camera is set to be very long, and it is usually much longer than 2.5ns, so it is highly unlikely that it the exposure will be shorter than 2.5ns. Therefore, for light painting application, the period of time during which the LED’s turn off is tolerated by the long exposure of the camera.

The digital LED requires a maximum minimum voltage amongst our components to operate, which is between 4.5V and 5.5V. Also, it operates between -20° C and 70° C. We are using 5 sealed NiCd batteries in series and according to the PS-CX NiCd battery cell data sheet, each cell has a bound output voltage range between 1.4V and 1V. This means that our battery pack outputs a voltage between 7V and 5V. Thus our buck converter outputs a steady 5V with a max dropout voltage of 230mV and a peak to peak ripple voltage of 20mV. Therefore, our LED voltage input requirements will be met at all times.

3. Cost and Schedule

3.1. Cost Analysis

<table>
<thead>
<tr>
<th>Item</th>
<th>Part Number</th>
<th>Description</th>
<th>Unit Cost</th>
<th>Quantity</th>
<th>Total Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital LED Strip</td>
<td>Adafruit NeoPixel Digital RGB LED Strip 144 LED</td>
<td>1m digital LED strip consisting of 144 WS2812 based LEDs</td>
<td>$59.95</td>
<td>2</td>
<td>$119.9</td>
</tr>
<tr>
<td>Name</td>
<td>Hourly Rate</td>
<td>Hours Invested</td>
<td>Total Cost</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>-------------</td>
<td>----------------</td>
<td>-------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kyoungyeon Cho</td>
<td>$25</td>
<td>200</td>
<td>$12,500.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jihong Min</td>
<td>$25</td>
<td>200</td>
<td>$12,500.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seungjun Cho</td>
<td>$25</td>
<td>200</td>
<td>$12,500.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Total Cost</strong></td>
<td></td>
<td></td>
<td><strong>$37,500.00</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.2.2. Wages for Team Members
3.2. Schedule

<table>
<thead>
<tr>
<th>Week</th>
<th>Kyounyeon Cho</th>
<th>Jihong Min</th>
<th>Seungjun Cho</th>
</tr>
</thead>
<tbody>
<tr>
<td>2/13</td>
<td>Working on design review document (Battery)</td>
<td>Working on design review document (circuit diagram, requirement, and verification)</td>
<td>Working on design review document (Microcontroller, Bluetooth)</td>
</tr>
<tr>
<td>2/20</td>
<td>Prepare for Mock Design Review</td>
<td>Prepare for Mock Design Review</td>
<td>Prepare for Mock Design Review</td>
</tr>
<tr>
<td>2/27</td>
<td>Verify and build supply system</td>
<td>Perform Mock Review and review the document</td>
<td>Prepare for Design review, and finalize the document</td>
</tr>
<tr>
<td>3/13</td>
<td>Review the whole circuit diagram, and start to draw PCBs.</td>
<td>Review the whole circuit diagram, and start to draw PCBs</td>
<td>Set up controlling system, and write programming codes.</td>
</tr>
<tr>
<td>3/20</td>
<td>Spring Break</td>
<td>Spring Break</td>
<td>Spring Break</td>
</tr>
<tr>
<td>3/27</td>
<td>Verify and revise PCBs.</td>
<td>Review, and finish all installations.</td>
<td>Set up Bluetooth module.</td>
</tr>
<tr>
<td>4/3</td>
<td>Test PCB and Power supply system</td>
<td>Soldering PCB, and interface with the system</td>
<td>Test the whole controlling system including mobile application.</td>
</tr>
<tr>
<td>4/10</td>
<td>Work on the final paper</td>
<td>Test and review the functionality</td>
<td>Review and Debug programming code</td>
</tr>
<tr>
<td>4/24</td>
<td>Prepare for the final paper</td>
<td>Finalize the demonstration</td>
<td>Finalize the demonstration</td>
</tr>
<tr>
<td>5/1</td>
<td>Finalize the paper</td>
<td>Finalize presentation</td>
<td>Lab check out</td>
</tr>
</tbody>
</table>

Table 3.2.1. Proposed Schedule for Project
4. Ethics and Safety

Since our group is dealing with electrical and chemical materials, there are several potential dangers we need to take care of. Rechargeable Nickel-cadmium (Ni-Cd) batteries will be used in our design. Even though Nickel-cadmium is commonly used chemical for batteries, and not of significant risk, failure to carefully handle and manage battery can cause leakage of battery fluid (electrolyte), heat generation, bursting, and fire which could lead to minor and major personal injury.

Our design might be used outside or left unused most of time, and most electrical devices are vulnerable to water. And, it could result in bursting, false operation, and electric shock. In order to avoid rust and heat generation of battery, we ask users to keep the device out of humid places and water.

Like other rechargeable batteries, overcharging Nickel Cadmium batteries over the predetermined charging period specified by the battery charger’s instruction or indicator could lead the battery to be leaked and bursted [8].

Honesty is one of the most important factors to have a successful results at the end. Following the IEEE code of Ethics #3 and #7 [9] - the people have to be honest and realistic in stating claims, and seek for honest criticism of technical work to acknowledge and correct errors. Students are required to help partners to act properly and honestly. We promise to ask TA for help if there are any sorts of technical problems that we cannot solve during the project.

5. References


