Persistence of Vision LED Sphere

Design Review

TA: Vivian Hou

ECE 445

Group 26
Michael Ling
Lunan Li
Jiale Quan

Table of Contents

1.0 Introduction

Our group plans to create a persistence of vision (POV) display. Our POV display will create a 3D hologram by utilizing a single string of LEDs rotating with two degrees of freedom. Consumer electronic companies have created one degree of freedom POV displays and usually sell them as clocks. Engineers have created 3D holograms using devices that utilize only one degree of freedom. Others have created two degree of freedom POV display, but they have not been able to create a smooth holographic image. We plan to take these POV displays to the next level of complexity. Our design will include a spinning ring of radially-facing-out LEDs that will spin about an axis parallel to the XY-plane. We will then attach this spinning ring to another motor that will spin about the Z-axis. The ring will have a spinning frequency of at least 30Hz to convey the persistence of vision concept. In order for the POV to take on full effect, a considerable amount of timing will have to be taken into consideration when coding how and when the LEDs will light up. Creating precise spinning frequencies poses mechanical and electrical challenges that will require extensive engineering to resolve. Overall, this project involves great mechanical, hardware, and software related challenges that we think will help us get the most out of ECE 445 and prepare us for our future engineering careers.

1.1 Purpose and Motivation:

We have selected this project because it presents a good balance of hardware, software, and mechanical problems are we are excited to face and solve. Programming the LEDs to spin and light up with precise timing is something all three of us are particularly excited to do. Beyond the challenges, the project will be incredibly fun to present. We think our project has the capability of providing a type of entertainment a lot of the other projects lack. Overall, we are extremely excited to work on this project

1.2 Objectives:

1.2.1 Goal

- Provide the user with a 3D holographic experience via persistence of vision

1.2.2 Functions & Features

- Measure RPM of each motor to ensure animations maintain proper position on sphere
- Balanced physical design for safety and minimal vibration
- Display pictures, patterns, and text at 30 frames per second (FPS)

- LEDs will spin at a rate of at least 30 Hz for smooth animation
- LEDs have a refresh rate of over 5 KHz for smooth animation
- LEDs have 24-bit RGB color capability

2.0 Design

2.1 Block diagram

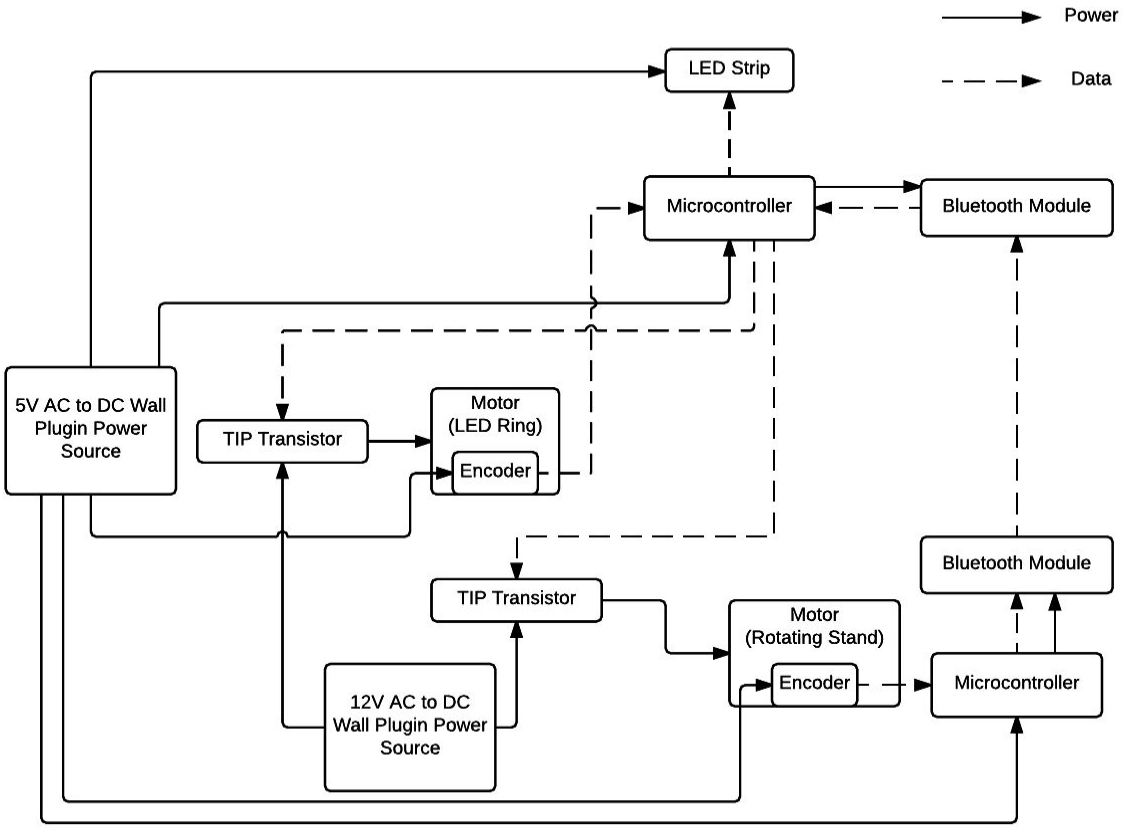


Figure 2.1:Top level electrical overview

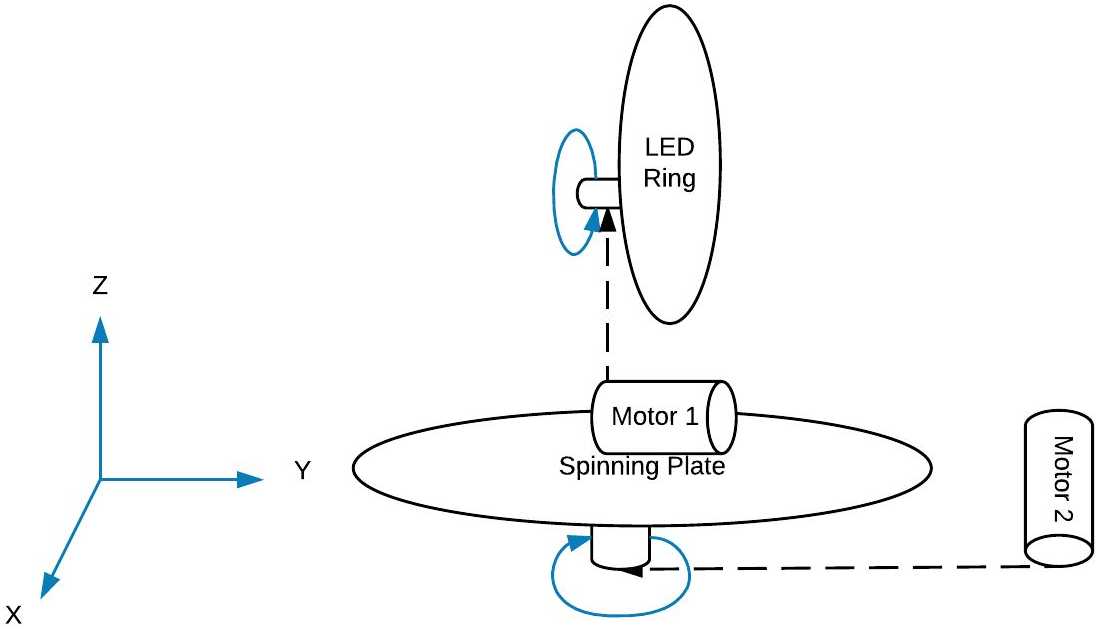


Figure 2.2: Mechanical Component

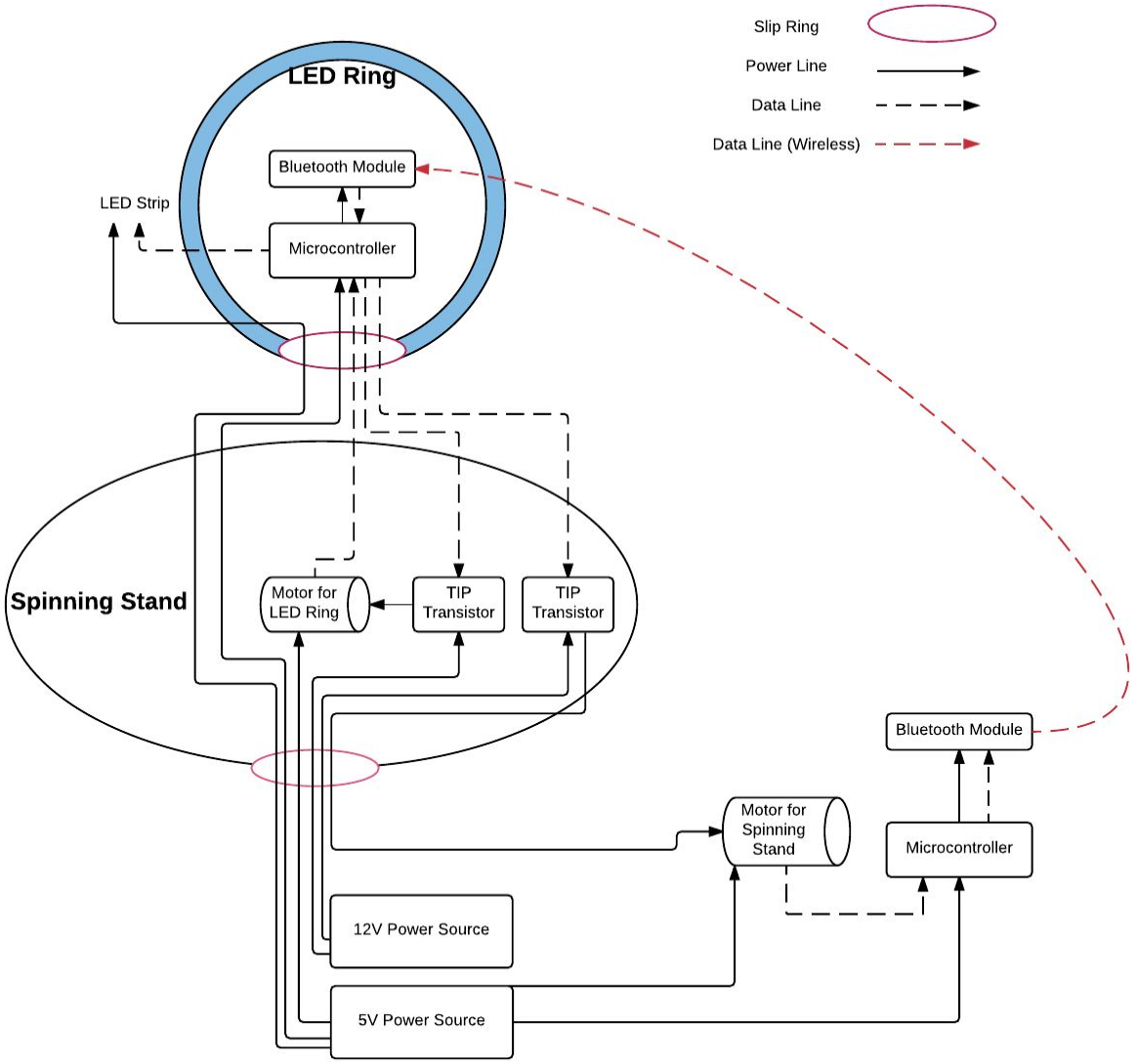


Figure 2.3: System Overview

2.2 Block Descriptions

2.2.1 Mechanical Component

Figure 2.1 is a simple, yet informative diagram that demonstrates *how* the two motors in our design will create two degrees of freedom for our project. Motor 2 will be connected to the Spinning Plate via a rubber engine belt along with a 1:4 gear ratio couplet. This means the Spinning Plate will spin four times every time the output shaft of Motor 2 completes one rotation. Motor 2 allows our display to rotate about the z-axis. This is one degree of freedom.

The other degree of freedom is made possible by Motor 1, which will be mounted on the Spinning Plate. This motor will be connected to the LED Ring, which will be attached to the Spinning Plate via two small towers (not shown in the diagram). The ring and the motor will also be attached to a 1:4 gear ratio couplet, and the ring will spin in about the xy-plane. This is the second degree of freedom in our project.

Finally, we will mount an RGB LED strip on the LED Ring such that the LEDs are facing *radially outward*. Our wheel will be 7 inches in diameter with a circumference of about 22 inches. Since

our LED strip has a pixel density of four LEDs/inch, so our LED Ring will have 88 LEDs total. When both motors are spinning, the LED Ring will be spinning with two degrees of freedom. This will allow us to create a spherical surface.

2.2.1.1 Spinning Stand

The spinning stand will be the mechanical portion of our project. It will hold the LED ring (which will be spinning in the XY-plane) and microcontroller unit while spinning about the Z-axis. This creates the two degrees of freedom we need to achieve the 3D effect of our POV display.

2.2.1.2 LED Ring

The LED Ring is the 7 inch diameter spinning wheel with the LEDs attached to it, and it will be attached to the spinning stand in such a way that it will spin around axis in XY plane while utilizing a slip ring for power and wires for data transmission. We are refraining from using the slip ring to transmit data because it is not very reliable when transmitting data at the speeds we require.

2.2.1.3 Motors and Gears

In order for the persistence of vision phenomenon to take effect, the ring of LEDs must spin at a relatively fast rate. Our motors are rated to spin at 500 RPM at 12V, but we need the LED ring to spin at 900 RPM (Equation 4.1) if we want an FPS of 30. Buying a motor that rotates at the speeds we require is too expensive. So, we will be using a gear ratio of 1:4 to increase the RPM of the LED ring while keeping the motor's RPM at a manageable value.

2.2.2 LED Strip

The LEDs we have chosen for our project have high enough refresh rates for our POV display to run smoothly and have 24-bit RGB capabilities. The LEDs rely on serial data transmission, which allows for a low profile when it comes to wiring. In fact, only six wires are required to drive all 100 LEDs we plan to use in our project. We plan to create software that will precisely time when these LEDs turn on and off to create smooth images for our POV display.

2.2.3 Control Unit

The control unit will be consist of a PCB board with the Atmega328P and a Bluetooth module.The Atmega328P chip will transmit designated voltage to drive the motor to the base of TIP transistors which will then accordingly control the voltage to the motor thus control the spinning speed of the motor,. The microcontroller will monitor the RPM of the horizontal axles through slip ring connection and monitor the RPM of the vertical axles through Bluetooth module since the motor driving vertical axle is not on the plate, and we decide to transmit data by Bluetooth to avoid transmit data through two slip ring which may cause much noise while transmitting. . It will then use this data and manipulate the input of the PWM transistors wired to each motor in order to control the voltage supplied to the motors. The microcontroller will also transmit data via wires to the LEDs for timing and display purposes.

2.2.3.1 Motor Encoder

The motors we purchased for our project have built-in encoders. The encoders will be able communicate the motors' RPM with the microcontroller. The encoders use the Hall Effect to measure the rate at which the motor is spinning. Equipped with 16 magnets placed equally on the rotor, each time the magnet pass through the sensor, the sensor generates an impulse, and the encoder will return the value of total magnets pass through the sensor in one second. The motors can perform 64 counts per rotation of the main shaft. With a gear ratio of 18.75:1, this translates to 1200 counts per rotation of the motor's output shaft. So recording the number of magnet passing per second and the real RPM can be calculated by:

$$RPM = \frac{Number\ of\ Magnet\ passings\ \ per\ sec}{Number\ of\ Magnet\ passing\ per\ rotation \times Gear\ Ratio} \times 60\ seconds \quad (2.1)$$

The accuracy of the encoders will allow us to accurately control the speed of the motor shafts in our project.

2.2.3.2 Bluetooth Module

We will be using a Bluetooth 2.0 module to transmit the data from vertical motor encoder to microcontroller units because slip rings are not reliable for high-speed data transmission. The module can also has an adequate data transfer rate, 2 Mbit/s, which is enough for our 20 kbps requirement through the calculation:

$$kbits/s = \frac{Max\ RPM \times number\ of\ magnets \times gear\ ratio}{60 sec\ per\ minutes} * 8\ bits\ per\ character \quad (2.2)$$

We will use a Bluetooth module to transmit data wirelessly to the microcontroller, which will be wired to an identical Bluetooth module that will act as a receiver.

2.2.4 TIP120 - Pulse Width Modulation

Pulse-width modulation (PWM) is a modulation technique used to control the voltage a load receives. We decided to use the TIP120 transistor to control the voltage supplied to the motor. For the PD5 and PD6 port on the ATmega328P chip, it supports PWM function, we connect each port to the base of the transistor to control the voltage delivered to the motor. Our microcontroller will use the encoders to measure the speed of the motors, and control the rotation speed of the LED ring using the TIP120 accordingly. The TIP transistors we will use can handle voltages up to 60 V and currents up to 5 A.

2.2.5 Power Supply

We will use two power sources for different parts : a 5V power supply with maximum current of 10A and a 12V power with maximum current of 5A. Each LED can draw a maximum of 60 mA when all three color channels (RGB) are at full brightness, which results in white light. Adafruit suggests using the One-Third Model[1]. Assuming 1/3 brightness, it will be 20 mA for each unit, as they are all connected to the power supply parallelly, the total current will be 100*0.02 A=2 A. The control unit will also be powered by the 5V source. For the motor, since the motor needs 12V voltage to run at its full speed, we will supply 12V voltage to assure the rotation speed, and because the stall current or maximum current is 5A, using a 12V with 5A maximum current is adequate. While considering using a voltage regulator to provide both 12V and 5V power using a single 12V AC to DC converter, it is risky to burden all current to one power source, with LED drawing 2A, motor drawing 4 to 5 A, it passed the maximum current of

12V source, furthermore, it will trigger the circuit breaker in the power outlet if the current reaches 10A.So we decided to keep our two voltage source instead of just one.

2.3 Schematic

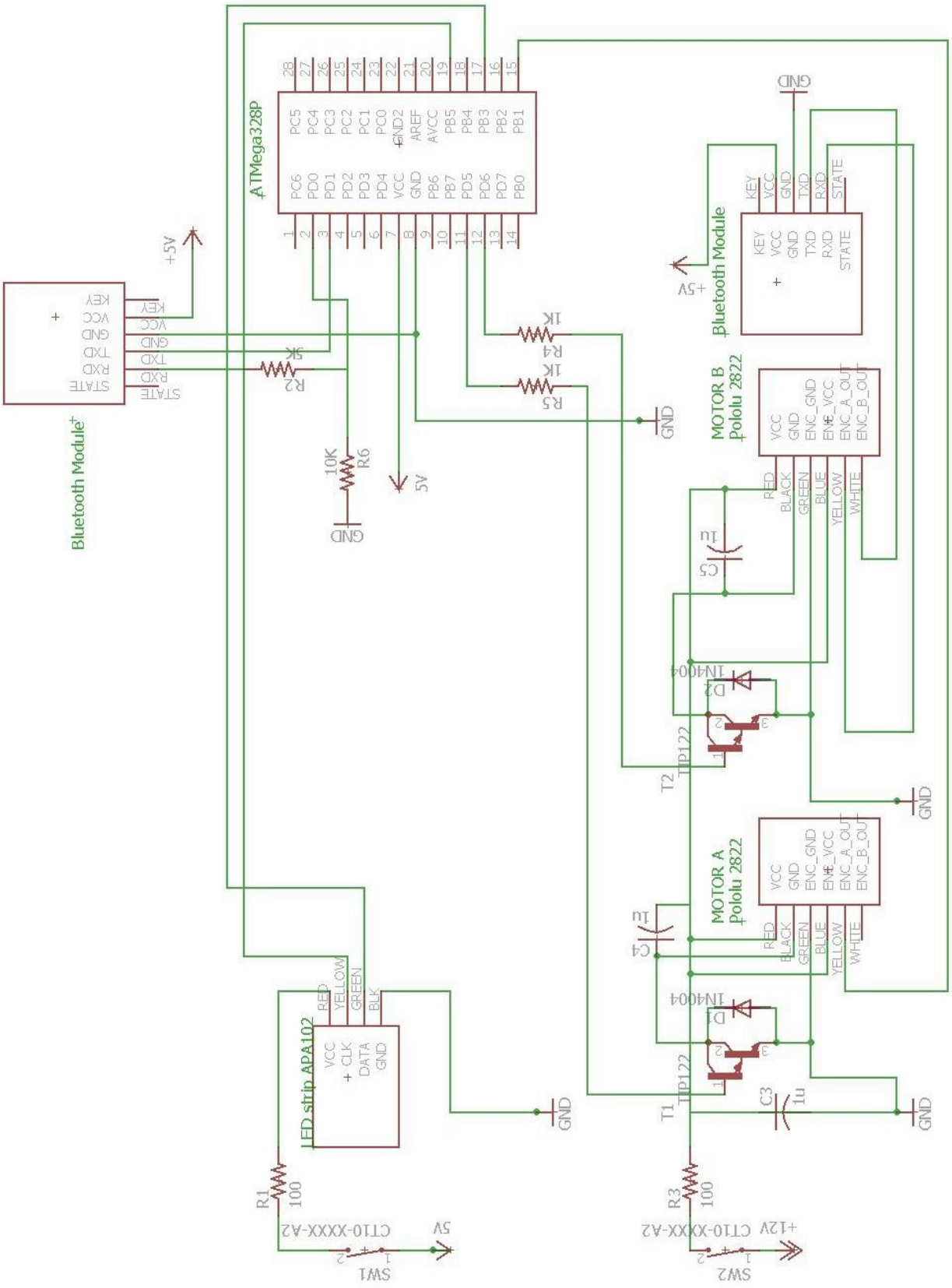


Figure 2.4: Schematic of Complete Circuit

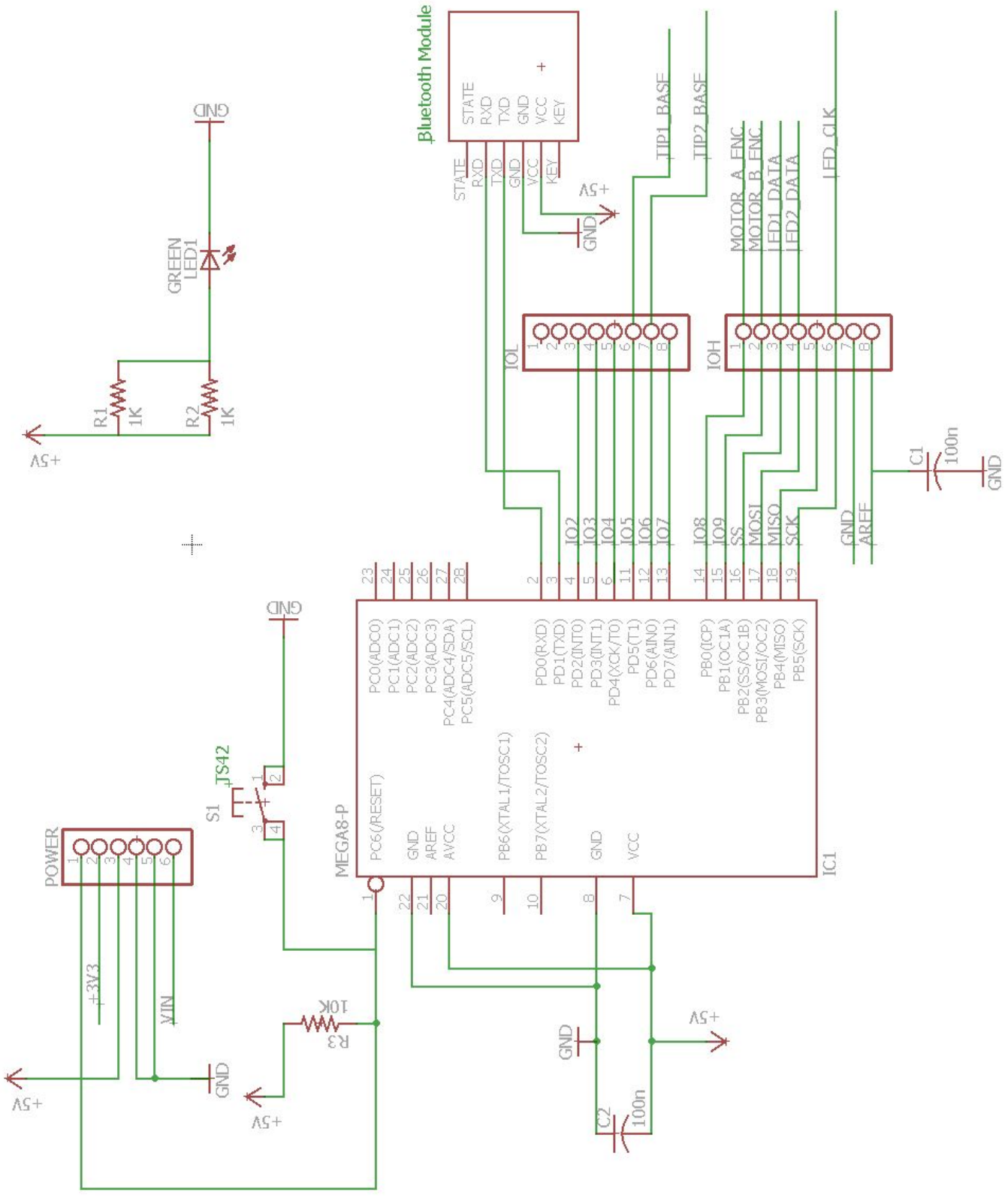


Figure 2.5: Schematic of Microcontroller PCB

3.0 Requirements and Verification

Requirement	Verification	Points
12 V Power Source a. V_{out} = 12 V +/- 5% b. I_{out} = 5 A +\- 5%	12 V Power Source Voltage Verification Process: 1. Connect the 12 V power source to a 2.5 Ω resistor. 2. Place a digital multimeter in parallel with the 2.5 Ω to measure the voltage difference across it. 3. Turn on the 12 V power source. 4. Ensure output voltage is within 11.4V and 12.6V. 12 V Power Source Current Verification Process: 1. Connect the 12 V power source to a 2.5 Ω resistor. 2. Place a digital multimeter in series with the 2.5 Ω to measure the current through it. 3. Turn on the 12 V power source. 4. Ensure output voltage is within 4.5 A and 5.5A.	5
5 V Power Source a. V_{out} = 5 V +/- 5% b. I_{out} = 10 A +\- 5%	5 V Power Source Verification Process: 1. Connect the 5 V power source to a 0.5 Ω resistor. 2. Place a digital multimeter in parallel with the 0.5 Ω to measure the voltage difference across it. 3. Turn on the 5 V power source. 4. Ensure output voltage is within 4.75V and 5.25V. 5 V Power Source Current Verification Process: 5. Connect the 5 V power source to a 0.5 Ω resistor. 6. Place a digital multimeter in series with the 0.5 Ω to measure the current through it. 7. Turn on the 5 V power source. 8. Ensure output voltage is within 9.5 A and 10.5A.	5
Motors a. V_{out} = 12 V +/- 5% b. I_{out} = 5 A +\- 5% c. Motor Encoder for vertical rotation must output 20 RPM +/- 5%. d. Motor Encoder for horizontal rotation must output 225 RPM +/- 5%.	Motor Voltage Verification Process: 1. Connect the motor to a 2.5 Ω resistor and a 12V power source in series. 2. Place a digital multimeter in parallel with the motor to measure the voltage difference across it. 3. Turn on the 12 V power source. 4. Ensure output voltage is within 11.4V and 12.6V. Motor Current Verification Process: 1. Connect the motor to a 2.5 Ω resistor and a 12V power source in series. 2. Place a digital multimeter in series with the motor to measure the current through it. 3. Turn on the 12 V power source. 4. Ensure output current is within 4.75A and 5.25A.	20

	Motor Encoder for Vertical Rotation Process: 1. Connect the motor to a 12V power source in series. 2. Connect the output of the Encoder to Pin 2 of Arduino UNO. 3. Run motorEncoder() to measure the current RPM of the motor and observe RPM number in the Serial Monitor. 4. Adjusting the voltage to change the RPM. 5. Ensure the output of the encoder is 20 +/- 5% RPM. Motor Encoder for Horizontal Rotation Process: 1. Connect the motor to a 12V power source in series. 2. Connect the output of the encoder to pin 2 of Arduino UNO. 3. Run motorEncoder() to measure the current RPM of the motor and observe RPM number in the Serial Monitor. 4. Adjusting the voltage to change the RPM. 5. Ensure the output of the encoder is 225 +/- 5% RPM.	
LED Strip a. Supply voltage must be 5V +/- 5% b. Max current should between 2A +/- 5% c. Data package must be 32bits each. 31th - 29th bits must all be 1. 28th - 24th bits are brightness settings. The rest 24 bits are divided into three consecutive 8-bit chunk, representing BLUE, GREEN, and RED from right to left.	LED Voltage Verification Process: 1. Connect the LED Strip to a 2.5 Ω resistor and a 5V power source in series. 2. Place a digital multimeter in parallel with the LED Strip to measure the voltage difference across it. 3. Turn on the 5 V power source. 4. Ensure output voltage is within 4.75V and 5.25V. LED Current Verification Process: 1. Connect the LED Strip to a 2.5 Ω resistor and a 5V power source in series. 2. Place a digital multimeter in series with the motor to measure the current through it. 3. Turn on the 5 V power source. 4. Ensure output current is within 4.75A and 5.25A. LED Strip Data Verification Process 1. Connect the microcontroller to your personal computer via USB 2. Run program to enable microcontroller and start to send data packages(32 bits) to computer 3. Run setBrightness() to change the brightness of the LED. 4. Ensure the LED strip looks brigher when setting a larger 5-bit binary. 5. Run setColor() to change the color of the LED. 6. Ensure the LED strip looks red when 7th-0th bits are set to 1 and 23th - 8th bits are set 0. 7. Ensure the LED strip looks green when 15th-8th bits are set to 1 and 23th - 16th bits and 7th -0th bits are	10

	set 0. 8. Ensure the LED strip looks red when 23th-16th bits are set to 1 and 15th - 0th bits are set 0.	
Microcontroller a. Speed of Data transmission must be no less than 1.6 Mbit/s	Microcontroller Speed Verification Process: 1. Connect the microcontroller to your personal computer via USB 2. Run program to enable microcontroller and start to send data packages(32 bits) to computer using SPI (Serial Periferal Interface) 3. Run countPackage() to counter the number of data packages that your computer received in one sec 4. Ensure your computer receive no fewer than 52800 packages	10
Motor Encoders a. Provide 64 counts per rotation of main motor shaft	Motor Encoders-Hall Effect Sensors 1. Use an oscilloscope to measure the time between peaks. 2. Use an Arduino program to measure the RPM of the motor. 3. Divide 1.0 sec by the amount of time between each peak of the encoder's signal, and the result should be the RPM measured by the Arduino program multiplied by 1200.	20
Bluetooth Module a. Speed of data transmission must be at least 20 kbit/s.	Bluetooth Module Speed Verification Process: 1. Put the bluetooth module into PAIRABLE state. 2. Search the bluetooth module on your computer and enter the pair code to connect. 3. Write a countSpeed() function in any language that continues to send 8-bit data package to the bluetooth module. 4. Run countSpeed() to enable bluetooth module and start to receive data packages(32 bits) from your computer. 5. Use any Serial Port Monitor software to monitor the number of data packages receiving by the bluetooth module. 6. Ensure your bluetooth module receive no fewer than 625 packages.	10
Microcontroller I/O a. Bluetooth Module RXD connect to PD0 (Receiving signal from the microcontroller) Bluetooth Module TXD connect to PD1(Transmitting signal to the microcontroller)	Bluetooth I/O Verification: 1. Connect bluetooth RXD port to the PD0 (RXD port of microcontroller) and bluetooth TXD port to the PD1 (TXD port of microcontroller), TIP transistor base connect to PD5 , 2. Use Oscilloscope to monitor the signals between the microcontroller 3. Use bluetooth function on computer to pair with the bluetooth module 4. Send AT commands through bluetooth communication	5

b. TIP transistor Base connect to PD5(transmitting desired rotation speed through voltage to the TIP transistor) c. LED strip Data port connect to PB3 (receiving LED signals from the microcontroller), Clk port connect to PB5 (receiving clock signal from microcontroller) d. Motor encoder A connect to PB0(transmitting motor A rotation speed to microcontroller), motor encoder B connect to PB1(transmitting motor B rotation speed to microcontroller)	5. Ensure there are signals on the pins while computer sending commands to the bluetooth module, and the bluetooth module response to the commands accordingly.	
	LED strip I/O Verification: 1. LED strip data port connect to microcontroller PB3 pin, clk port to microcontroller PB5 pin. 2. Connect Arduino to computer and upload test program “Blink”(a program sending commands to LED to let it blink) to Arduino. 3. Run the test program and see if the LED strip blinks accordingly.	5
	Motor Encoder I/O Verification: 1. Connect motor VCC to DC power supply, GND to ground, encoder_VCC to 5V power, and encoder_GND to ground, encoder output to PB0 port on microcontroller. 2. Connect Arduino to computer. 3. Monitor the value that motor encoder returns to the Arduino on the serial monitor, and change the voltage between the motor to verify the rotation speed value changes correspond to the voltage change.	5
	TIP transistor I/O Verification: 1. Connect base pin of TIP transistor to PD5 port (PWM port of microcontroller) on microcontroller, 2. Connect Arduino to computer and send PWM signal to the microcontroller to control the rotation speed of the motor through the voltage change. 3. Along with the motor encoder connected to the microcontroller PB0 pin, monitor the rotation speed change on the serial monitor.	5

4.0 Tolerance Analysis

4.1.1 Critical Component

The critical component we will do the tolerance analysis on is the motors and how we can achieve a desired FPS using our 12V DC motors and gear ratios.

4.1.2 Tolerance Analysis

The key for persistence of vision is the coordination between the spinning LED strip and the timing LEDs shining. The minimum FPS humans see continuous motion is 24 FPS[2]. We calculate how fast the motors will spin with the following equation:

$$60\ [sec/min]\ \times \frac{Desired\ FPS}{2\ [LED\ Passes/Ring\ Revolution]} = LED\ Ring\ RPM \tag{4.1}$$

Since we will be using a ring of LEDs for our display, one rotation of the ring results in two passings of LEDs. The motors in our system ultimately determine the RPM of the LED Ring. We plan on using a gear ratio of 1:4, with the motor have the smaller of the two gears. So, to determine the necessary motor RPM we use the following equation:

$$\frac{LED\ Ring\ RPM}{4} = Motor\ RPM \tag{4.2}$$

Again, the minimum 24 FPS. Using the two equations listed above, the following figure has been created:

Desired FPS	Desired LED Ring RPM	Necessary Motor RPM
24	720	180
30	900	225
36	1080	270

Figure 4.1. Table containing necessary motor RPM based off desired LED Ring RPM

We will use a 12V +/- 5% power supply to drive the two motors. To ensure we can drive our motors at a sufficient RPM, we can test our motors' RPM using a tachometer.

4.1.3 Testing Procedure

1. Supply the motors with the maximum voltage our 12 V battery could supply: 12.6 V.
2. Measure the motors' RPM to ensure they are faster than the necessary motor RPM.
3. Supply the motors with the minimum voltage our 12 V battery could supply: 11.4 V.
4. Measure the motors' RPM to ensure they are faster than the necessary motor RPM.

5.0 Power Analysis

5.1 Power analysis for single element

5.1.1 LED Strip

88 LEDs will be mounted on the spinning ring in our project. For each LED, it can draw up to 60 mA. Since all LEDs are connected in parallel, and Adafruit suggests using the One-Third Mode as described in section 2.2.5, the total current for all LEDs would be 20 mA * 88 LEDs = 1.76 A. According to the datasheet of the Dotstar LED Strip, a 5V supply voltage is needed.

5.1.2 Bluetooth Module

We will be using the HC-06 Bluetooth Module is used in our project to handle data transmission between motor encoders and the microcontroller. The Bluetooth Module requires 3.6V - 6V for Vcc, 3.3V for serial operating voltage(RXD/TXD) and 40 mA working current. A 5V supply voltage will be used to provide power for the Bluetooth Module. A 5kΩ and a 10kΩ resistor will be used to divide the 5V and generate a 3.3V input for RXD/TXD as shown in Figure 3.

5.1.3 Motor

A 19:1 Metal Gearmotor is used in our projects to spin the LED ring. The motor accepts maximum 12V voltage and 5A current. The actual voltage of the motor depends on the RPM that we want to use.

5.2 Power analysis for the whole project

According to Fig.3 in section 2.3, the LED and the two bluetooth modules are connected in parallel. Thus, we need a total of 1.76A + 40mA*2 = 1.84A. Both of the LED Strip and Bluetooth modules need a 5V supply voltage. Two motors are also connected in parallel. By experimenting, one motor requires 0.27A when 12V is supplied under no torque-load. We calculate the Current vs. Torque relationship using the following equation:

$$Torque = \frac{StallCurrent - FreeRunCurrent}{StallTorque} \times T \times FreeRunSpeed \qquad (9.1)$$

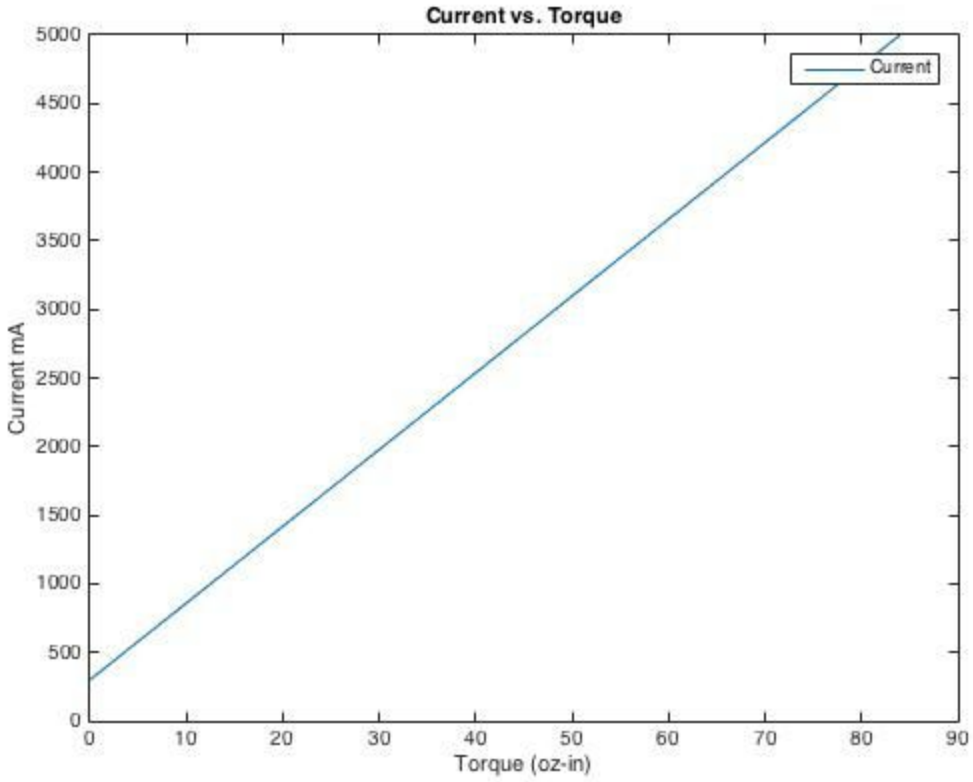


Figure 5.1: Torque vs. Current

6.0 Algorithm and Flowchart for 2DOF

6.1 Algorithm for 2DOF

We decide to have 30 frames per second(FPS) for both horizontal and vertical LED refresh rate. We will consider each LED unit as single frame, and we need to pass 30 LED units on a static point on the spherical surface every second.

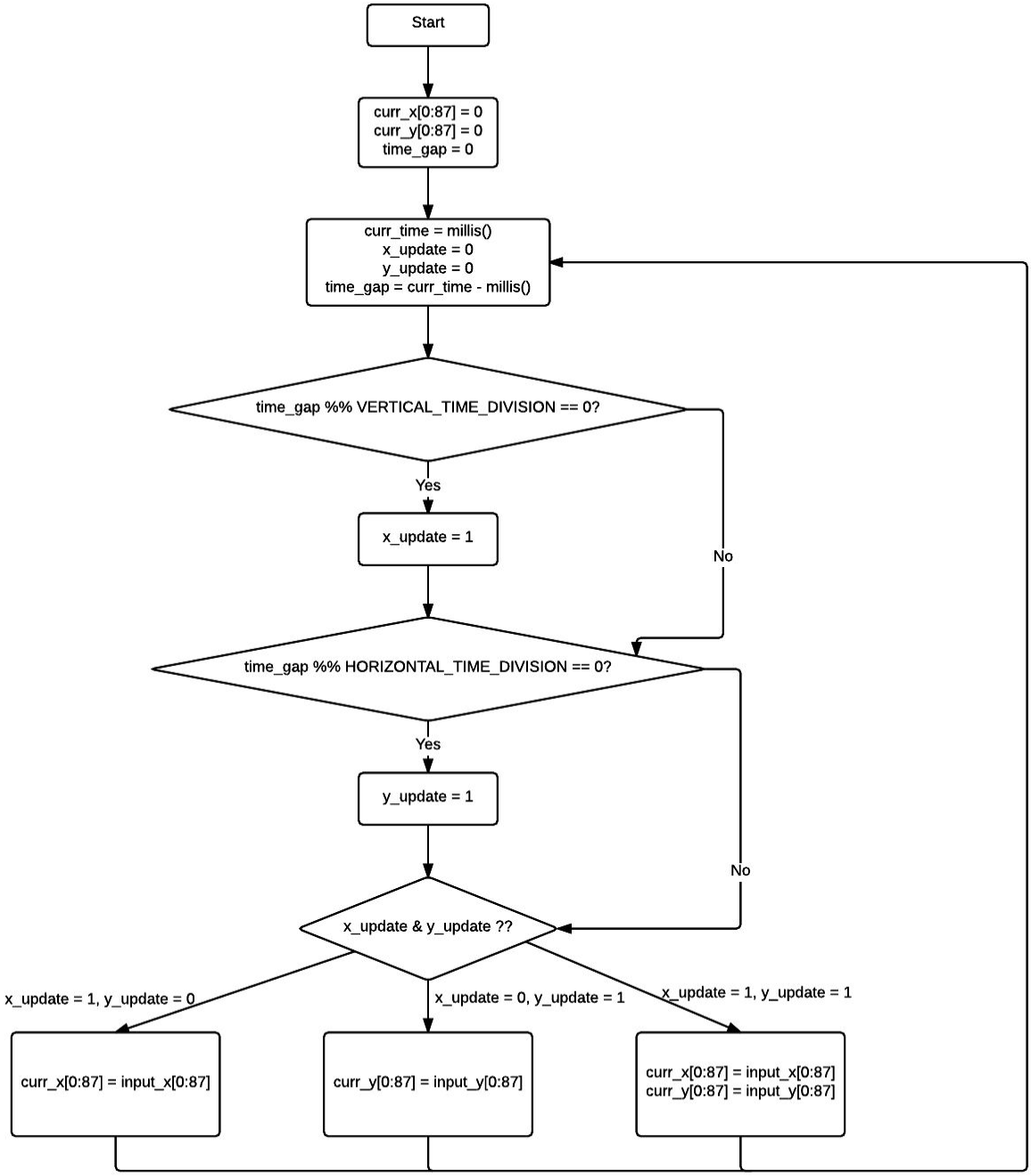
For the horizontal spinning direction, it has 44 levels of LEDs and 2 LEDs at each level. The radius of our spinning ring is 8.89cm (3.5”), and the perimeter of the ring would be 55.86 cm. Since there are only 2 LEDs passing a static point on one rotation, we need to have 15 rotations per second, which is 225 RPM due to the 1:4 gear ratio, so that each LED needs to spin 15 rotations, which is 837.86 cm. Since each LED takes 0.635 cm (including the gap between two LEDs), there is around $\frac{837.86}{2 \times 0.635} = 598.5$ available LED position on the ring per second. Thus, the refresh rate for horizontal direction(f_h) would be $\frac{1}{598} \approx 1.6$ milliseconds.

For the vertical spinning direction, it has 88 LEDs in total. To achieve 30 fps, each LED should pass approximately one-third on the ring, which is 18.619 cm and 20 RPM. Then,there are $\frac{18.619}{0.635} \approx 29$ available LED position per second. Thus, the refresh rate for vertical direction(f_v) is $\frac{1}{29} \approx 34$ milliseconds.

If there is a conflict of data update between vertical and horizontal direction, the vertical direction will be updated first. The horizontal direction will be updated after, and the delay will be neglected because because f_v is much less than f_h.

In the flow chart, millis() returns the time since the program starts running in milliseconds. curr_time stores the value of the current time at each loop, curr_x and curr_y store the switch information of the LED in specific division, and x_update, and y_update keep track of the updating status in the x, and y direction(horizontally, vertically). The system keeps track of the time_gap among different loops, and sets x_upate to 1 every 1.6 ms and y_update to 1 every 34 ms. If x_update is 1, LEDs in x-direction need to be updated to the specified state storing in input_x which are sent from the microcontroller. If y_update is 1, LEDs in y-direction need to be updated to the specified state storing in input_y which are also sent from the microcontroller. There are 3 conditions in total. ①Both x_update and y_update is 1, according to our calculation above, we will update the state in x direction then in y direction to minimize the conflict. ②Either x_update or y_update is 1, we just update the state accordingly. ③Both x_update and y_update are 0, we do not update the state of LEDs in this circumstance. Finally, reset x_update and y_update and enter a new rotation. VERTICAL_TIME_DIVISION is 1.6ms and HORIZONTAL_TIME_DIVISION is 34 ms.

6.2 Flowchart for 2DOF



7.0 User Interface

We plan on creating Java applet that will convert a JPEG or PNG image into an unsigned long array whose size is equal to the resolution we plan on displaying. Figure 7.1 shows the main code of this applet. Once this array is created, we will load it onto the Atmega328P chip that will be used in the MCU that's mounted on the LED Ring. This entails using an Arduino Uno board to program an Atmega328P chip, removing the chip from the Arduino, and physically mounting it onto the LED Ring. For now, that is the main method we plan on using to load images/animations onto our display

If everything goes smoothly, we will try to program the Atmega328P via Bluetooth. This will get rid of the need to repetitively removing and mounting the Atmega every time we want to reprogram it. We will already have a Bluetooth module connected to the PCB that holds the Atmega328P located on the LED Ring; however, we will be using it to receive data from the encoder attached to the motor that drives the Spinning Plate of our display.

```
//image information
Image image("image_test.jpg");
int width = image.columns();
int height = image.rows();

int range = pow(2, image.modulusDepth());

//error check
assert(range > 0);

// pixels array for all pixels of the image
PixelPacket *pixels = image.getPixels(0, 0, width, height);

int column = 0;
Color color = pixels[w * row + column];

//print out RGB for each pixel
for(row=0; row<height-1; row++){
    for(column=0; column<width-1; column++){
        Color color = pixels[w * row + column];
        cout << (color.redQuantum() / range) << endl;
        cout << (color.greenQuantum() / range) << endl;
        cout << (color.blueQuantum() / range) << endl;
    }
}
```

Figure 7.1 Pixel-by-Pixel Analysis Code

8.0 Theoretical Torque Analysis

This section is "theoretical" because we currently do not know exactly how much the mechanical portion of our project will weigh. So, we have split up the mechanical portion into two parts: the LED Ring and the Spinning Stand. We also have two motors in our project, each responsible for driving one of the mechanical components.

8.1 LED Ring Theoretical Torque Analysis

The following calculations were performed to calculate the theoretical torque our motor will have to supply in order to drive the LED ring:

If we want a refresh rate of 24 FPS, we have to spin the LED ring about once every three seconds:

$$\frac{88\ LEDS}{1\ Rotation} \times \frac{1\ Second}{24\ LEDs} = \frac{11\ Seconds}{3\ Rotations} \tag{11.1}$$

Inverting the result of equation 11.1 gives 3 rotations per 11 seconds. We can now calculate the desired angular velocity:

$$\frac{3\ Rotations}{11\ Seconds} \times 2\pi\ Radians\ =\ \omega = \frac{6\pi}{11}[rad/sec] \tag{11.2}$$

Now, we can calculate the theoretical desired angular acceleration (assuming we want to get the LED ring up to speed in five seconds):

$$\frac{\omega}{t} = \alpha = \frac{6\pi}{55}[rad/sec^2] \tag{11.3}$$

We assumed the LED ring would weigh 2 kg, which is extremely high; however, it makes for a more conservative calculation. We also assumed the LED ring can use the same moment of inertia equation as a hollow cylinder with:

$$I = \frac{1}{2}M(a^2 + b^2) \tag{11.4}$$

Our ring will be 7 inches in diameter (17.78 cm), and the LEDs are 4 mm tall. So, our *a* and *b* values are 0.1778 m and 0.1782 m, respectively. Plugging the correct values into equation 11.4, we calculate that the LED Ring has a moment of inertia of 0.0633 kg-m^2. We now have all of the components necessary to calculate the torque required to drive our LED ring:

$$\tau = I\alpha \tag{11.5}$$

Plugging in the appropriate numbers, we arrive at the necessary torque of 0.0217 N-m. Which converts to 3.0768 oz-in. We now take our gear ratio into consideration because it will affect the torque our motors need to provide to the LED ring itself:

$$\tau_0 = \tau_i \times \frac{R_0}{R_i} \tag{11.6}$$

Since our gear ratio is 1:4, the torque needed to drive the LED Ring becomes 12.3072 oz-in. This is well below our motors' 84 oz-in torque rating.

8.2 Spinning Stand Theoretical Torque Analysis

The process through which we performed our theoretical torque analysis of the spinning stand is nearly identical to the LED Ring process, except for we assumed a different weight and moment of inertia equation. We assumed the spinning stand would weigh 3 kg, take 15 seconds to reach the appropriate speed, and use the same moment of inertia equation as a disk:

$$I = \frac{1}{2}MR^2 \tag{11.7}$$

Performing the same calculations as the LED Ring, we arrive at a desired torque of 53.8125 oz-in. This is also below our motors' 84 oz-in torque rating.

8.3 Theoretical Torque Analysis Conclusion

Overall, we are confident our motors will be able to drive our display. We used relatively conservative numbers when estimating how much the mechanical portion will weigh, so the actual desired torques will most likely be lower than the ones we calculated in the above two sections.

Below are two graphs that depict how the weight of our mechanical portion affects the amount of torque the motors need to provide as well as how much current the motor will draw:

$$Current = \frac{5}{0.59317} \times Inertia \times \frac{FPS}{Seconds\ to\ Reach\ Desired\ RPM}\pi \times 4 \tag{11.8}$$

The "Inertia" variable in equation 11.8 is not a discrete equation because it depends on the shape of the mechanical portion. For example, if you wanted to calculate the current drawn by the motor driving the LED Ring, you would use equation 11.4. The "Seconds To Reach Desired RPM" is the amount of time we would like our motor to take to get our display up to the proper RPM.

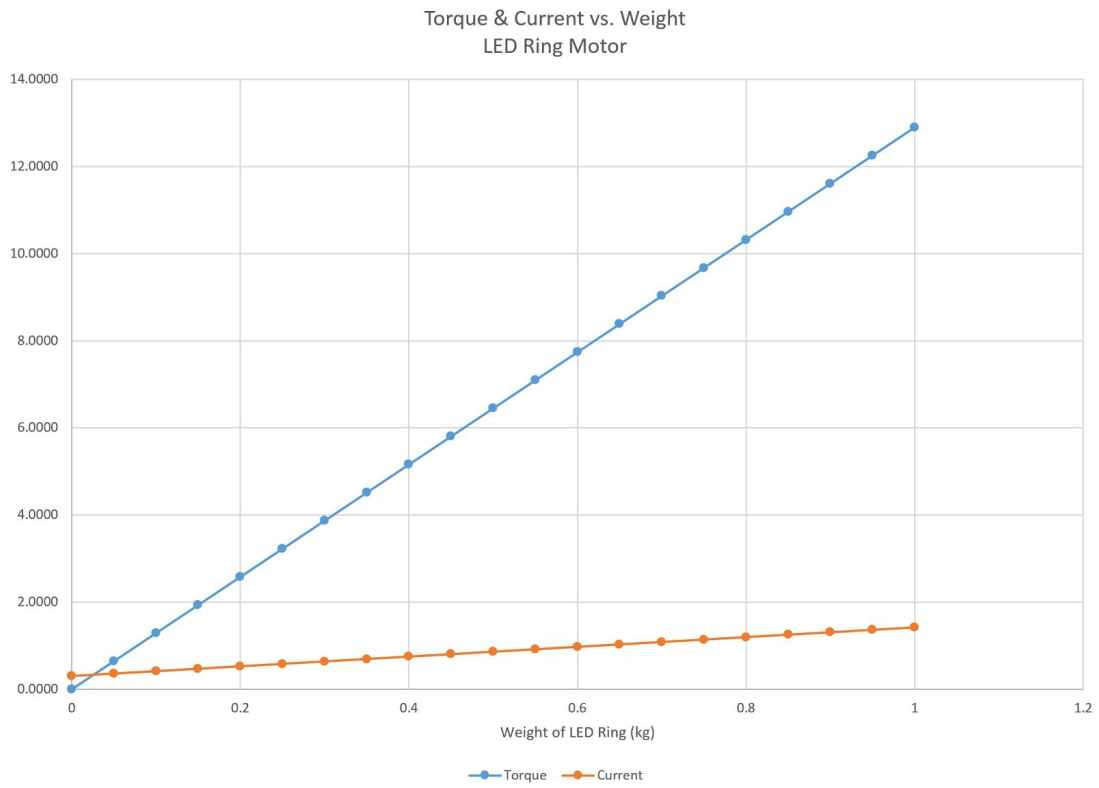


Figure 5.2: Torque & Current vs. Weight of Load for the Motor Driving the LED Ring

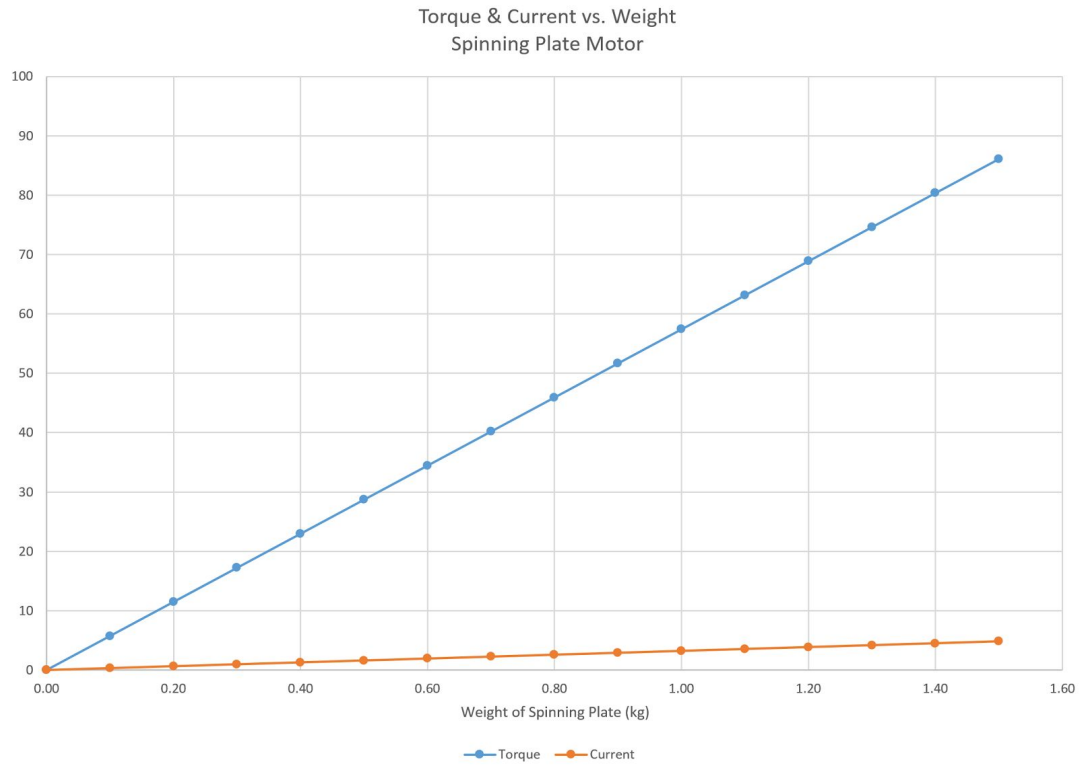


Figure 5.3: Torque & Current vs. Weight of Load for the Motor Driving the Spinning Plate

9.0 Safety Statement

Our project has two main components that pose some safety hazards: higher current and mechanical parts spinning at dangerously high rates. We plan on using 100 RGB LEDs for our project; each capable of drawing 60 mA. Plugging these numbers into Equation (1.1):

$$Number\ of\ LEDs \times Current\ Drawn\ per\ LED = Total\ Current\ Drawn \quad (6.1)$$

You will get a Total Current Drawn of 6 Amps. This is an extremely high amount of current to be dealing with. Refer to Figure (1.1). A lethal level of current ranges between 0.1 to 0.2 Amps. We are dealing with components that could draw up to 6 Amps. To ensure our safety, we must make sure to follow these precautions[3]:

- Avoid water all all times when working with live wires
- Never use equipment with frayed cords, damaged insulation, or broken plugs
- Always use insulated tools while working
- Never try repairing energized equipment
- Always communicate to others when power sources are turned on

1.0	Severe Burns and Breathing Stops
0.2	
0.1	Death
0.01	Extreme Difficulties Breathing
0.0001	Severe Shock Muscular Paralysis Painful Shock Mild Sensation Threshold of Sensation

Figure 9.1[4] Range of Certain Current Levels and the Dangers They Pose

It is important to note that the vendor of the LEDs, Adafruit Industries, suggests the actual amperage drawn by the LEDs can be estimated to be one-third of the maximum. This is referred to as the "One-Third Rule"[1]. Using this approximation, our LEDs will, on average, draw approximately 2 Amps. Although this is not nearly as high as the maximum amount of current the LEDs can draw, it is still quite dangerous and and safety should not be taken lightly.

The mechanical portion of our project also poses some safety concerns. Our ring of LEDs its support structure will be rotating at a rate of 900 RPM. The support structure will house a 4 oz motor. A 4 oz motor spinning at 900 RPM is not something to take lightly. We will advise the UIUC Machine Shop to create some sort of protective cover that we will put over our rotating device to protect those around it.

Lastly, the FPS aspect of our project can potentially affect those who have epilepsy and/or some sort of photosensitivity. Headaches and nausea caused by changing frames of light is a variation of motion sickness. The brain is struggling to sort out the mixed messages the eyes send to the brain (the body seems to be moving) and the messages the cerebellum's center of balance (the body isn't moving). Concerning epilepsy, flashing lights most likely to trigger seizures are between 5 Hz to 30 Hz. Our project aims to reach an FPS rate between 24 and 30 FPS, so those with epilepsy should take the following precautions:

- Watch the display in a well-lit room to decrease the contrast between the display and the ambient light in the room
- Do not watch the display if you are tired or exhausted
- Stand at least two feet from the screen
- Take frequent breaks from watching the display
- Stop watching the display IMMEDIATELY if strange or unusual feelings or body jerks develop

10.0 Ethics

We recognize the importance of committing ourselves to high ethical and professional conduct. There are a couple of specific points in the IEEE Code of Ethics that resonate with our project and the responsibilities it gives us:

- to accept responsibility in making decisions consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment
- to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others;
- to avoid injuring others, their property, reputation, or employment by false or malicious action;

Our project poses some safety precautions that me we must inform the audience of our display at *all* times in the event someone who could potentially be negatively affected by our project is present. it would be extremely unethical of us to ignore this issue because the safety, health, and welfare of the public is a huge responsibility for us.

Our project poses a lot of problems in field we are not well versed in, specifically the mechanical portion of our project. With this is mind, we have been very open to seeking and accepting any "mechanical" help we can get. We would especially like to thank the people at the UIUC ECEB Machine Shop for giving us some insight as to how to keep our project mechanically sound.

We also recognize it would be unethical of us not to protect our audience from the mechanical dangers of our project. We plan to have the Machine Shop create a barrier between the display and those using/watching it. If we were not cognisant of this issue, we would not be putting forth our best effort to avoid injuring others and ourselves. Our display will be spinning upwards of 900 RPM, so mechanical danger is not something to take lightly.

11.0 Cost and Schedule

11.1 Cost Analysis

11.1.1 Labor

Name	Hourly Rate	Total Work Hours	Total = Hourly Rate x 2.5 x Total Hours
Michael	$30	175	$13,125.00
Lunan	$30	175	$13,125.00
Jiale	$30	175	$13,125.00
Total		525	$39,375.00

11.1.2 Parts

Vendor	Part Number	Part Description	Cost Per Item	Quantity	Total Cost
Adafruit	DotStar	LED Strip 144/m	$74.95	1	$74.95
Adafruit	US5881LUA	Hall Effect Sensor	$2.00	2	$4.00
Adafruit	SRC022A-12	12-Wire Slip Ring (max 240V @ 2A)	$19.95	2	$39.90
Pololu	2822	19:1 Metal Gearmotor with 64 CPR Encoder	$39.95	2	$79.90
Adafruit	658	5V DC 10A Power Supply	$25.00	1	$25.00
Adafruit	352	12V DC 5A Power Supply	$25.00	1	$25.00
Atmel	ATmega328 P	Microcontroller	$4.00	1	$4.00
Amazon	BC417	KEDSUM Arduino Bluetooth Receiver/Transmitter	$10.00	2	$20.00
Adafruit	TIP120	NPN Darlington Transistor (pack of 3)	$2.50	1	$2.50
UIUC Machine Shop	N/A	8” Wheel	$10.00	1	$10.00
UIUC Machine Shop	N/A	Wheel Support Structure	$15.00	1	$15.00
UIUC Machine Shop	N/A	Spinning Platform	$30.00	1	$30.00
				Total	$285.25

11.1.3 Grand Total

Expense	Total
Labor	$39,375.00

Parts	$285.25
Total	$39,660.25

11.2 Schedule

Week	Task	Responsibility
1-Feb	Finalize Proposal	Michael
	Prepare Mock Design Review	Jiale
	Research POV Requirements	Lunan
8-Feb	Research,Purchase, Request Mechanical Parts	Michael
	Research & Purchase Motors,LEDs, and Microcontroller	Jiale
	Research & Purchase Data Transfer Equipment	Lunan
15-Feb	Prepare Design Review	Michael
	Prepare Microcontroller PCB Design	Jiale
	Test Data Transfer Rate of Microcontroller	Lunan
22-Feb	Assemble Mechanical Portion and Prepare Design Review	Michael
	Create Microcontroller PCB	Jiale
	Test LED and Sensor Functionality	Lunan
29-Feb	Finalize Design Review	Michael
	Create 1DOF Ring Animation Software	Lunan
	Solder Microcontroller and Test	Jiale
7-Mar	Test TIP120/Arduino/Motor interaction	Michael
	Mount Electronics onto 1DOF Display	Jiale
	Test 1DOF Ring Animation Software	Lunan
14-Mar	Prepare and Mount 2DOF Electrical Assembly	Michael
	Prepare and Mount 2DOF Electrical Assembly	Jiale
	Create 2DOF Animation Software	Lunan
21-Mar	Test 2DOF Stability	Michael
	Test 2DOF Electrical Components	Jiale
	Test 2DOF Animation Software	Lunan
28-Mar	Debug Mechanical System	Michael

	Debug Electrical System	Jiale
	Debug Software System	Lunan
4-Apr	Prepare Mock Demonstration	Michael
	Prepare Mock Demonstration	Jiale
	Prepare Mock Demonstration	Lunan
11-Apr	Maintain Mechanical Functionality	Michael
	Maintain Electrical Functionality	Jiale
	Fix Remaining Software Issues	Lunan
18-Apr	Prepare Presentation	Michael
	Prepare Demonstration	Jiale
	Prepare Demonstration	Lunan
25-Apr	Prepare Final Paper	Michael
	Finalize Demonstration	Jiale
	Finalize Demonstration	Lunan
2-May	Finalize Presentation	Michael
	Lab Checkout	Jiale
	Finalize Final Paper	Lunan

12.0 Citations

1. P.Burgess. (2013, Aug. 30). *Powering NeoPixels* [Online]. Available: https://learn.adafruit.com/adafruit-neopixel-uberguide/power

2. P. Bakaus. (2014, May 21). *The Illusion of Motion* [Online]. Available: https://paulbakaus.com/tutorials/performance/the-illusion-of-motion/

3. "Electrical Safety in the Lab," *Lab Manager*, 19-Nov-2009. [Online]. Available at: http://www.labmanager.com/lab-health-and-safety/2009/11/electrical-safety-in-the-lab?fw1pk=2. [Accessed: 15-Feb-2016].

4. P. Giovinazzo, "The Fatal Current," Feb-1987. [Online]. Available at: https://www.physics.ohio-state.edu/~p616/safety/fatal_current.html. [Accessed: 15-Feb-2016].

5. P. O. Shafer and J. I. Sirven, "Photosensitivity and Seizures," *Epilepsy Foundation*, Nov-2013. [Online]. Available at: http://www.epilepsy.com/learn/triggers-seizures/photosensitivity-and-seizures. [Accessed: 27-Feb-2016].