FINAL REPORT

Voxel: A Spinning, 3D LED Persistence of Vision Display

ECE 4872 Senior Design Project

Voxel
sd22p45
Faculty Advisor: Richard Kyle Bryan

Anna Carow; Computer Engineering; acarow3@gatech.edu
Marcus Chan; Electrical Engineering; mchan79@gatech.edu
Ryan Chen; Electrical Engineering; rchen390@gatech.edu
Akash Harapanahalli; Computer Engineering; aharapan@gatech.edu
Andrew Hellrigel; Electrical Engineering; ahellrigel3@gatech.edu

Submitted
May 3, 2022

Table of Contents

Executive Summary

Voxel is a spinning 3D LED display that generates a 360 degree visual in three dimensional space by spinning a panel of densely packed LEDs about a central axis. By spinning at a fast enough rate and rapidly flashing each individually addressable LED on and off, persistence of vision (POV) will take effect. POV is a phenomenon in which a rapid succession of images becomes blended into a persistent, moving image when processed by the human brain [1]. By flashing the LEDs in a specific pattern over time, Voxel will be able to display a smooth, sharp 3D video.

Persistence of vision has already been exploited in the design of many 2D LED displays. Most monitors use this phenomenon by turning on and off individual rows of LEDs at high frequencies to create the illusion of having the entire display on [2]. More recently, LED 'fans' have emerged on the commercial market. These fans have two to four blades lined with LEDs that create smooth 2D images when spun at high enough speeds. These commercial products are generally used for either entertainment or advertising and cost $300 on the low end [3] and upwards of $10,000 on the high end [4].

Instead of spinning individual lines of LEDs to create a 2D display, the premise of Voxel is to spin a grid of LEDs to create a truly 3D LED display. Three dimensional displays that appear to be holographic would be useful in applications such as home entertainment or commercial advertising. If production of Voxel devices is scaled up to 1,000 a year, the unit cost would be $609.78. It could be sold at $999 for a profit margin of 61% - making it economically viable to market for home design and commercial advertising uses.

Voxel: A Spinning, 3D LED Persistence of Vision Display

1.0 Introduction

Voxel is a persistence of vision display that uses a rapidly spinning panel of RGB LEDs to create a moving, three dimensional image. The product uses HDMI and FPGA technology to enable high-speed data transfer to the LED display. The team used a $2000 budget to design a Voxel prototype and demonstrate its feasibility and capabilities.

1.1 Objective

The premise of Voxel was to create a device that makes use of the persistence of vision (POV) phenomenon so that a relatively small number of LEDs can create a smooth, clear, and complex image [1]. Voxel's LED display is composed of an RGB LED matrix panel. The panel is outfitted with a densely packed array of LEDs and mounted on a base that spins the panel at a high rate about its central axis. As the 2D LED panel spins, its pixels are updated regularly to generate a 3D image. The team wrote software to control when individual LEDs light up, allowing Voxel to "draw" with discrete points in a 3D point cloud cylinder. The POV phenomenon enables the audience to see a smooth 3D image generated from these rapidly flashing LEDs.

1.2 Motivation

The use case for a spinning LED display is for entertainment and home design purposes. If commercialized, Voxel would be sold to households or companies to put up small-scale displays that add to the design element of a room. The product would appeal to companies looking to advertise in a public space because bright, non-stationary displays like Voxel help catch consumers' eyes. Voxel would be great for Walt Disney Imagineering which develops novel immersive attractions at Disney's theme parks. Similarly, home design enthusiasts would appreciate how they can constantly change up what is on the display so that it can be made to fit any season or occasion. In the future, Voxel's technology could even serve as an alternative to virtual reality technology since it wouldn't require a user to wear a headset to display 3D images.

If commercialized, Voxel would be the first high resolution, truly 3D display on the market. All previous technologies have made compromises: 2D displays have been used to create an illusion of depth, and truly 3D displays lack the resolution to view anything concrete [5]. The novelty of this project was part of its motivation; it's impossible to know all the different applications high resolution 3D displays could have until one has been made.

1.3 Background

Persistence of vision is the phenomenon whereby the brain processes a rapid succession of images as smooth, continuous movement. The phenomenon is a result of the fact that when light strikes the retina, the brain retains an impression of that light for about 0.125 seconds after the light source is removed [1]. Therefore, humans cannot register distinct changes in lighting that occur more than 8 times a second. Instead, people perceive these lighting changes as a continuous transition [1]. This is why videos, which are in fact a rapid succession of distinct images, appear as smooth and continuous.

Recently, POV LED displays have made their way onto the consumer market. The majority of these displays take the form of a rapidly spinning fan that has a dense strip of LEDs lining each blade [6]. When turned on, these fans create a two dimensional POV display by rapidly switching the LEDs on and off in a software-coordinated pattern to produce a certain image. By creating a 3D POV display that uses a rapidly spinning LED matrix instead of LED-lined fan blades, Voxel is a significant upgrade from traditional two dimensional displays.

2.0 Project Description and Goals

2.1 Project Description

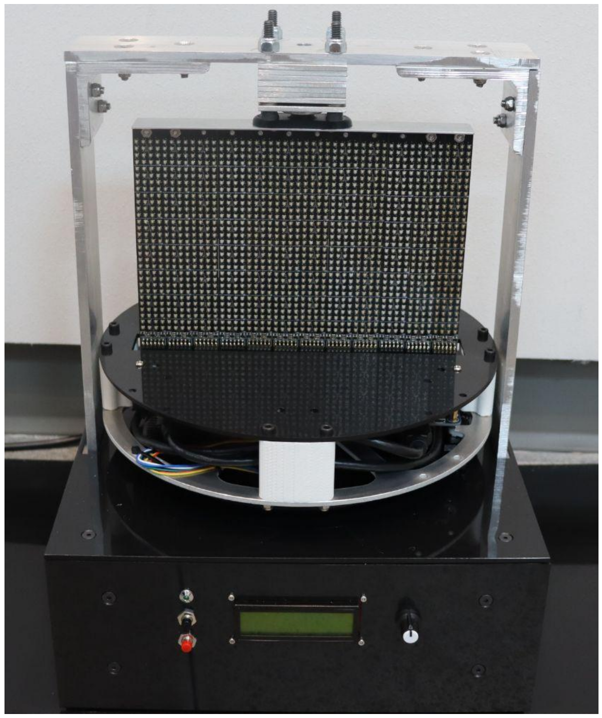


Figure 1. Voxel assembly.

Voxel can be divided up into four components: the mechanical assembly, the LED matrix, the main power and control electronics, and the software to create the image to project on the LED display. These four components were developed in parallel and then integrated to create the final prototype.

The first component is the physical mount that the LED panels are attached to. This mount is designed in CAD, and has various parts that are 3D printed, laser cut, or machined out of stock aluminum, allowing for simple manufacturing and rapid prototyping. The mount also contains the motor which drives the display's rotational motion. The mount assembly holding the LED matrix can be seen in Figure 1. The second component is the LED matrix. The LED matrix is composed of 12 separate PCBs, each with an array of 120 LEDs, meaning the display has 1,440 RGB LEDs in total. Third, to support high refresh rates for the LEDs (on the order of 5400 Hz), the system has a control board with high-speed communication between an FPGA and the LEDs. This was accomplished through usage of fast LED driver chips that are able to drive 16 RGB LEDs each. The final component is the embedded software. The team wrote software to control all of the embedded systems and to accurately translate a 3D cartesian point cloud of a 3D image into cylindrical coordinates.

2.2 Stakeholders

When it comes to satisfying customers, there are many user requirements that Voxel had to meet. The most important of these requirements are cost, size, ease of use, and refresh rate. Voxel should cost $1,000 or less to keep the price in line with what people are currently paying for 2D LED displays and other types of 3D visual displays. The dimensions of the display should be small enough that it can fit on a coffee table in someone's home, but is still large enough that the visual can be seen from across a room. It should be fast and easy for customers to select what video they want to see displayed and upload it to their Voxel. Finally, Voxel must have a refresh rate high enough so that the POV effect will take place and viewers will see one smooth, high resolution, continuous image. We still have plans to improve the process for users to select what video they want to see on Voxel to make it easier and faster, but our final prototype met all of the other requirements.

3.0 Technical Specification

In order to meet customers' needs, Voxel was designed to meet a wide range of requirements. A comprehensive listing of design requirements can be found in the QFD chart in Appendix A. Tables 1 and 2 highlight some of the most critical technical specifications that Voxel meets. Note that 'voxel' is a unit of measurement for a 3D pixel.

Table 1. Specifications for LED Visual Display.

Item	**Specification**
Refresh Rate	15 Hz
2D Matrix Resolution	6 dpi, 36 pixels per in^2
Rotational Speed	15 RPS
Angular Resolution	360 2D updates per revolution
3D Image Resolution	2063 voxels per in^3
2D Matrix Size	8 x 5 inches
HDMI Standard	HDMI Specification 2.1
LED Operating Voltage	3.3V
Computer Operating Voltage	5V
Nominal Individual LED Current Draw	3mA

Table 2. Specifications for the Display Mount.

Item	**Specification**
Power Draw	< 120W
Weight	~14lbs
Size	11" x 11" x 15"
Voltage Input	110V AC

4.0 Design Approach and Details

4.1 Design Approach

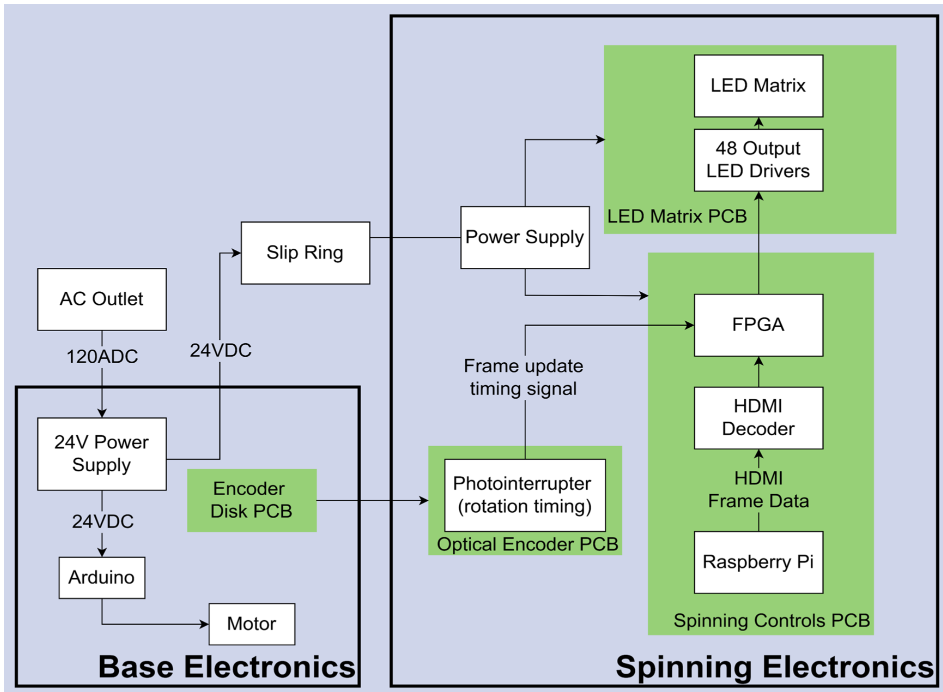


Figure 2. High-level electronics system overview.

Voxel's electronics are divided into two sections based on the mechanical assembly: base electronics and spinning base electronics. The stationary base of Voxel houses the DC brushed motor and motor controller that spins the spinning base, as well as the encoder disk that is used to track the rotational speed of the spinning base.

The second section, the spinning base electronics, contains all of the electronics used to display a 3D image. The Raspberry Pi mounted to the spinning base PCB generates the data for every voxel of the 3D image. This data is then transmitted over HDMI to an FPGA. The FPGA uses the data generated by the Raspberry Pi to control the color and brightness of each RGB LED on the LED matrix PCB by sending signals to the LED drivers. Optical encoders are used to time and sync the rotation of the spinning base to the 3D image.

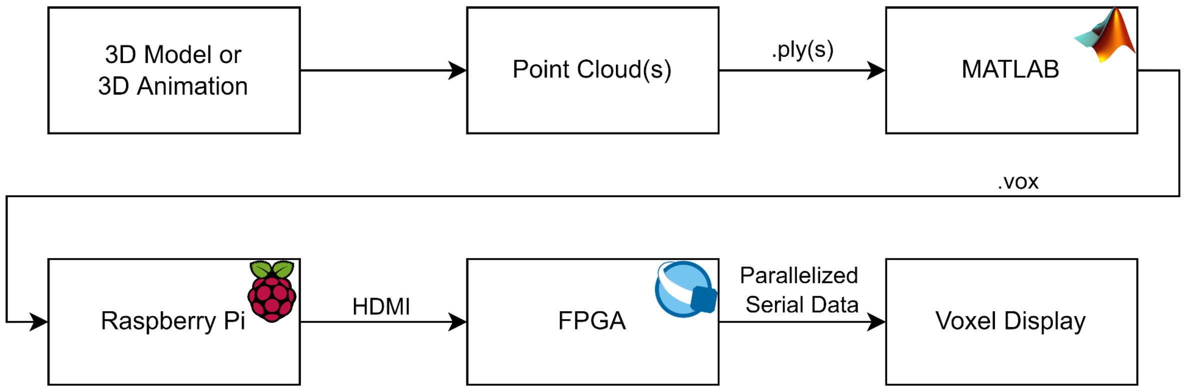


Figure 3. High-level software system overview.

In order to display a new graphic to Voxel, a 3D model or animation is converted into a point cloud. The point cloud is then converted to a custom .vox file format using a MATLAB script. The .vox file is uploaded to the Raspberry Pi, which decodes the .vox file and sends the data over HDMI to the FPGA. The FPGA then stores the graphic's decoded data in SDRAM. The stored graphic data is then sent over serial to all LED drivers on Voxel's LED matrix to display the graphic.

4.1.1 Mechanical Design

The design of the physical mechanics that house the electrics and make the LED display spin was made to be robust and easy to manufacture and assemble.

The base that serves as the support structure for the spinning LED matrix is made of an aluminum base plate with aluminum bars extending upwards to support a horizontal aluminum bar which houses a bearing to support the top of the spinning assembly. The aluminum plate and bars are held together with 90° aluminum brackets and countersunk bolts. The aluminum in the CAD is colored light gray.

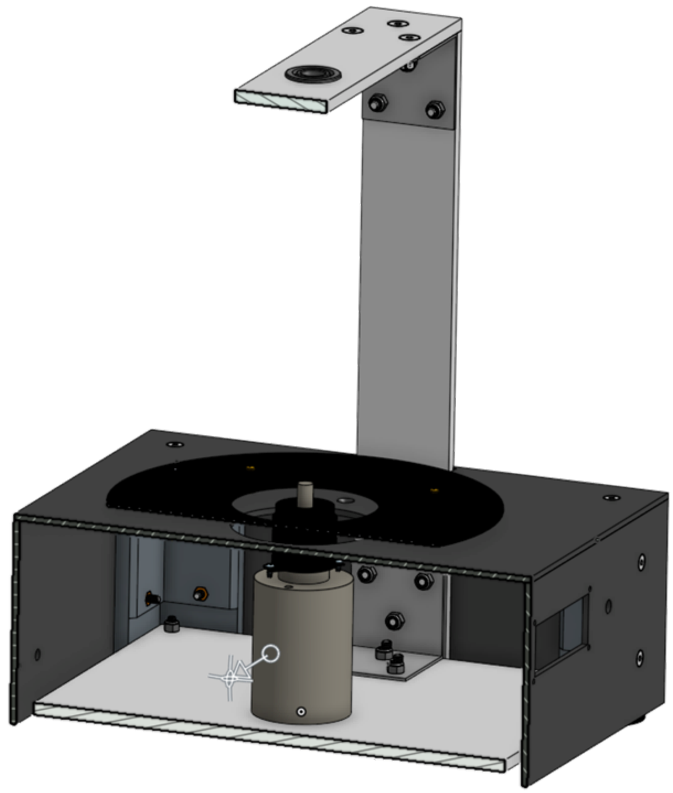


Figure 4. Section view of Voxel base.

All of the structural components on the spinning assembly are also made of aluminum. The structural parts were made with optimized geometry to achieve a lightweight design that could be manufactured through water jetting aluminum plate stock.

Everything else in the design is either made of acetal — a durable and easy to laser cut plastic — or is 3D printed. Everything that is acetal in the CAD is colored black. The 3D printed parts are gray with a blue hue to them. The spinning assembly has a base to hold electronics with an acetal top plate and an aluminum bottom plate. The plates are separated by 3D printed supports which have slots in them to hold balancing weights. Balancing the spinning assembly with small iron weights proved to be critical for Voxel's stability during operation.

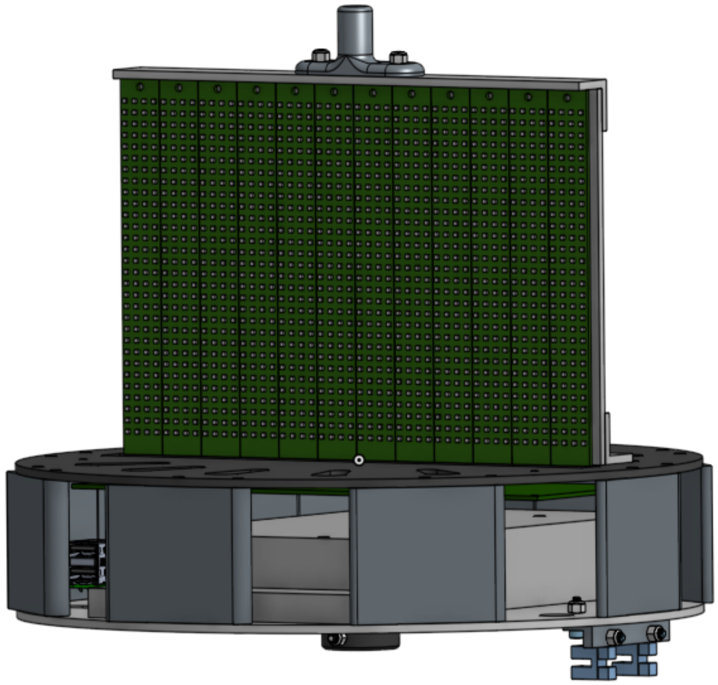


Figure 5. Assembly of spinning electronics and LED Matrix.

4.1.2 LED Matrix Design

4.1.2.1 Overview

The LED matrix is subdivided into 12 identical 4-layer subpanel PCBs that are mounted side by side. Each subpanel consists of 120 RGB LEDs arranged in a 30 x 4 grid on the top layer and eight 48-channel TLC5955DCAR LED drivers on the bottom layer [7].

4.1.2.2 PCB Layer Designations

The top and bottom layers act as a 3.3V plane and a ground plane, respectively. The inner top layer is dedicated to the serial input, clock, and latch signals to allow the signals to take the most direct path from the FPGA. Vias between the top and bottom layer are required to connect each channel on the LED driver to the LEDs. As a result, the inner bottom layer is used to route the traces between these connections to minimize interruption of the power and ground planes.

4.1.2.3 PCB Component Schematics and Layout

Figure 6 shows the schematic for a single LED driver. Figure 7 shows the schematic for the 24 LEDs controlled by a single LED driver. These two schematics are duplicated eight times to create the schematic for the entire LED matrix subpanel PCB. Each matrix subpanel has a buffer that strengthens the three signals common to all LED drivers: GSCLK, LAT, and SCLK.

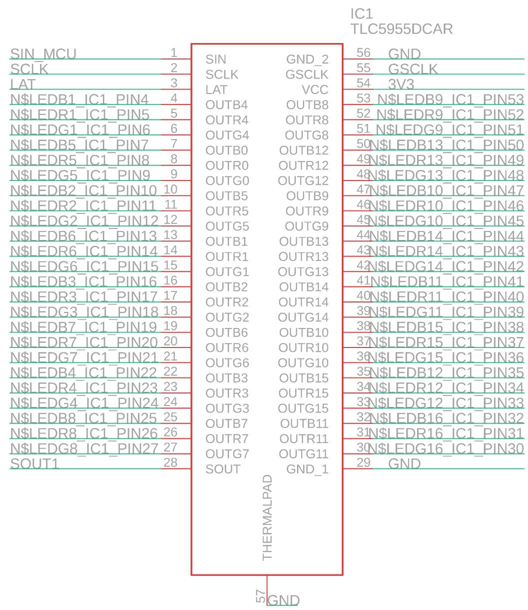


Figure 6. Schematic for LED driver.

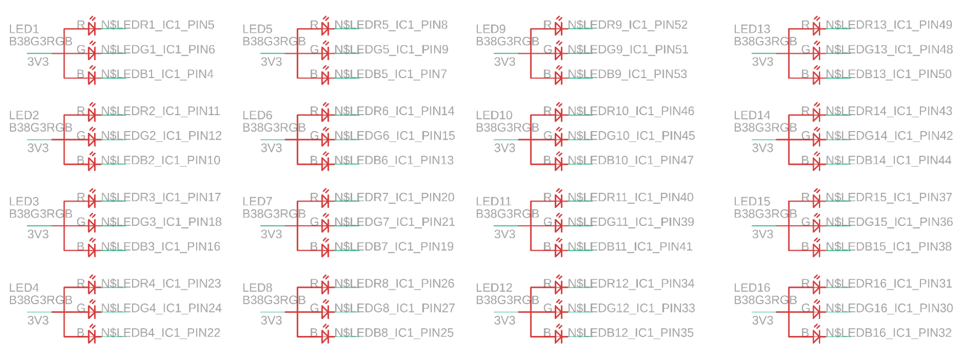


Figure 7. Schematic for one LED driver’s worth of RGB LEDs.

The block in Figure 8 below showcases the routing for LEDs for one LED driver. This same layout is repeated eight times as shown in Figure 9.

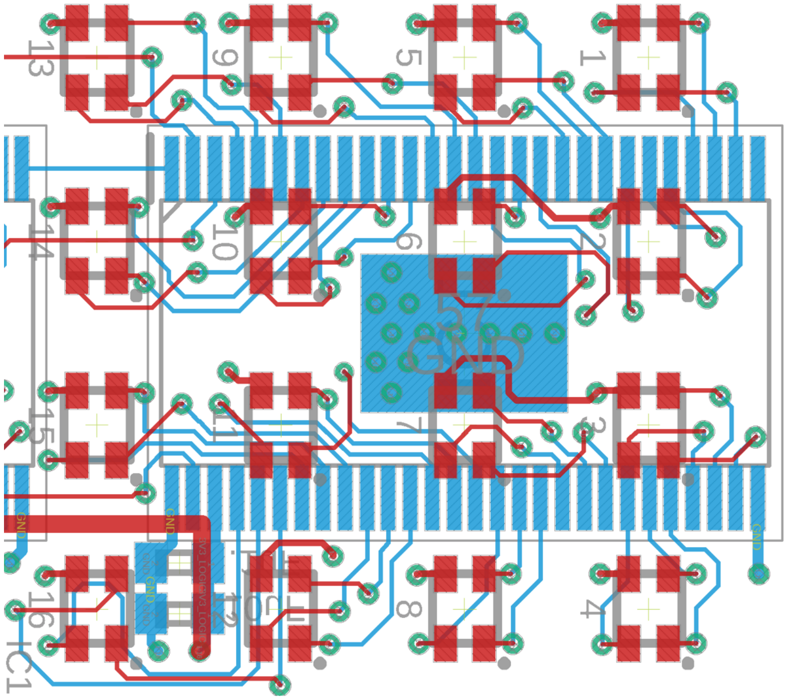


Figure 8. PCB layout for one LED driver's worth of LEDs.

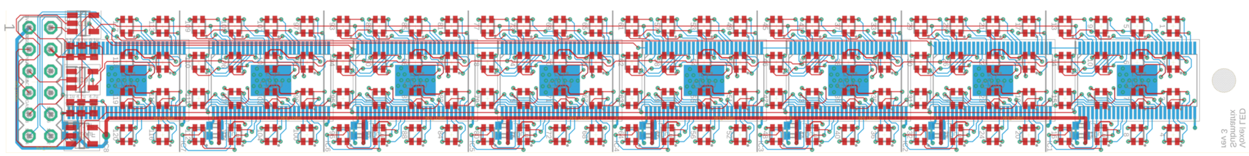


Figure 9. PCB layout for LED matrix subpanel.

In order to connect each LED matrix subpanel to the FPGA, there are right angle male headers at the bottom of the subpanel, which plugs into female headers on the FPGA development board that the team is using. See Section 4.1.5.2 for more details on the FPGA Voxel makes use of.

4.1.3 Power Electronics Design

The power for Voxel is sourced from wall outlets. Refer to Figure 2 to follow the path of power for Voxel. On the stationary base, AC power is converted into 24VDC using an AC-DC power supply to power a DC motor that rotates the spinning base. The motor is controlled by a PWM signal from the Arduino.

To deliver power to the spinning base, 24VDC is passed through a slip ring. DC voltage is passed rather than AC voltage because it is comparably safer when using sliding metal contacts. The 24VDC is then regulated to 5V and 3V to power all of the electronics on the spinning base. In the worst case scenario, each of the three colors on an LED can source up to 8mA per color. Each LED matrix subpanel (120 LEDs each) must be able to handle 2.7A. Voxel has 1440 LEDs overall, requiring 32.4A capacity total. As a result, the 3.3VDC power supply must supply at least 110W. When in use, the full matrix of LEDs will not be drawing current. Each color of each LED will turn on and off depending on the model, which will significantly lower the power draw of the LED matrix.

4.1.4 Control Electronics Design

The overall refresh rate of Voxel determines the target speed for the spinning base. For example, 15 Hz directly corresponds with 15 revolutions per second, or 900 rpm. The desired rotational velocity of the base is achieved via user-input through the menu interface. A power percentage is selected and the DC motor runs accordingly. A closed loop controller could be used to precisely set the spinning base speed with optical encoders, but for simplicity, the slices of Voxel's display are simply updated based on encoder position.

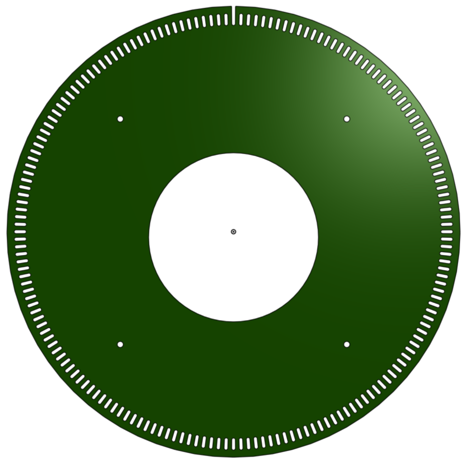


Figure 10. Custom optical encoder disk with 180 slots and one zeroing slot at the top.

A custom optical encoder disk has been designed for this purpose. One optical encoder is used with 180 discrete slots for 360 pulses per revolution (measuring rising and falling edges). This signal correlates with the 2D LED matrix update, leading to a max of 360 LED updates per revolution, or one update per degree. Another adjacent optical encoder is used on the same wheel with one discrete step per revolution, allowing for a “zero”, or “home” position. The secondary encoder ensures that the 3D image is always oriented the same at startup and throughout animations.

Each encoder outputs to an FPGA pin. A digital high corresponds to a blocked encoder (e.g., optical encoder disk interrupting the light) and a digital low corresponds to an unblocked encoder (e.g., encoder passing through a slot). To avoid mis-triggers, the encoder signal only registers a rising or falling edge after a consecutive window of values. In other words, a rising edge is only perceived if 100 consecutive 1’s are read in, immediately after a steady digital low. Similarly, a falling edge only occurs when 100 consecutive 0’s are read in after a steady digital high. Since the FPGA samples the encoder at 50 MHz such that each slot is seen as approximately 9260 samples, the window size of 100 samples is appropriately large to ensure true triggers and edges, while not being too large to miss any. Each encoder edge is a signal for a new slice update.

4.1.5 Embedded System Design

4.1.5.1 Raspberry Pi

The project requires a central system inside the spinning base to generate the 3D images in real-time. A Raspberry Pi Zero W offered enough computing power to output the images while meeting the constraints. Wireless capabilities are crucial for programming the device while mounted in the assembly. As far as outputting the image, the Raspberry Pi includes several peripheral options, including General Purpose I/O (GPIO), USB, and HDMI outputs. The team used HDMI, which is a common choice for displays due to its high data rate. While HDMI ports are usually used to connect to displays as a user interface, they were reconfigured for the project's needs.

4.1.5.2 FPGA System Design

Since the Raspberry Pi outputs a serial HDMI signal, this signal has to be decoded, processed and parallelized into the various inputs on the LED drivers. The TFP401 HDMI decoder converts the serial stream of data into individual pixels [8]. A frame buffer is implemented using external memory to store an incoming HDMI frame in memory since the output of the LED drivers is timed separately from the HDMI input. An optical encoder is used to regulate exactly when the matrix updates.

This system has to interact with multiple different high frequency clock domains, while simultaneously reading from memory and outputting to several different digital outputs. As a result, microcontrollers were ill-suited for this task, and the flexibility of an FPGA was a better fit [9]. The design uses a DE0-Nano Development board, which significantly decreases the amount of design overhead by breaking out a memory module, serial communications devices, ROM for bootup configuration, and the desired pins [10]. The DE0-Nano has a total of 106 GPIO, which is enough for the project with its specifications: 48 pins for serial outputs to LED drivers, two pins for clock outputs for LED drivers, one pin for the latch for LED drivers, 36 pins for the HDMI decoder, and two pins for the speed sensor input. The team used System Verilog to design controllers for each of these systems to integrate together. The FPGA HDL was primarily composed of two state machines – one to control the LED driver output and one to read in HDMI data – that synchronized with each other to ensure proper system timing requirements were met.

4.1.5.3 HDMI Data Transfer Using the .vox File Format

Since there doesn't yet exist a 3D video file format, the team created its own file format for storing 3D animations, the .vox file. The .vox file format is explained further in the Animation design section. Voxel uses HDMI to transfer video data encoded in a .vox file to the dispaly. HDMI is an interface for transmitting uncompressed video data between two HDMI compliant devices [11]. The most recent HDMI Specification, HDMI 2.1, has a bandwidth of 4.95 Gbps and a 30Hz refresh rate when operating at 720p, the resolution we use [12]. This means that roughly 25 million cubic pixels, or 'voxels,'can be updated every second in the display.

4.1.6 Animation Design

4.1.6.1 Simulations in Python and Matlab

Simulation is vital to generating any reliable image on the actual device. The team explored several options for simulation. For starters, a Python script was written for visualizing the 3D cylinder using the 3D plotting capabilities of PyQtGraph (below).

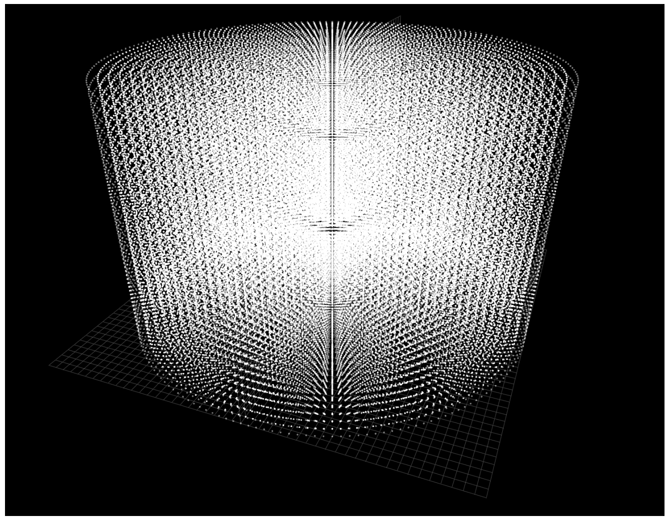


Figure 11. Voxel Cylinder Visualization using PyQtGraph with OpenGL.

A simple demo of a time decaying oscillating 3D sinc function was created using this simple 3D plotting utility. However, another layer would be required before trying to display arbitrary 3D models. The team switched to MATLAB for arbitrary mesh visualization due to its robust point cloud library. In MATLAB, any mesh point cloud (or series of mesh point clouds for animations) is converted into point clouds sampled at the 3D voxels of our cylinder.

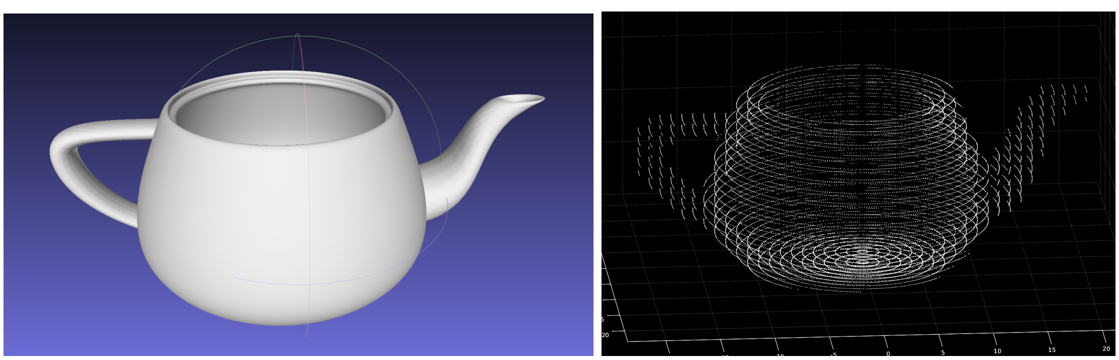


Figure 12. Teapot Mesh File Sampled Into a 3D Point Cloud. Displayed on Voxel in Figure 14.

4.1.6.2 The .vox 3D Video File Format

Regardless of the generation method, Voxel needs some unified way of storing 3D stills and videos. However, there does not already exist a lossless compressive 3D file format. The team decided to create its own format, which could be generated by any simulation method.

While the format was created with the purpose of this project in mind, it is generalized enough to work with any sized spinning LED matrix, as well as any number of slices per revolution or overall framerate. These parameters are set in the header of the file. The Raspberry Pi or other decoding interface handles reading this generalized file format and outputting whatever it needs to over HDMI.

4.1.6.3 Animation Generation Pipeline

To streamline the process of generating .vox files, a flexible and robust animation generation pipeline was created. This generation first begins with a 3D model or animation, downloaded from popular sites such as Thingiverse or Free3D. If the file is an STL, it can immediately be translated into the .vox file format.

While STL's are a very common 3D model file format, they have the disadvantage of not storing color data. Instead, a colored 3D model or animation can be imported into Blender (3D computer graphics software). Then, a Python script can be run in Blender to automatically generate an animated and colored .vox file.

The stretch goal of being able to visualize and interact with 3D mesh files from CAD design was partially realized in our streamlined back-end animation generation pipeline that allowed for rapid 3D animation and object generation, sent over WiFi to the RaspberryPi. However, we were unable to interact and visualize CAD models live on Voxel.

4.2 Codes and Standards

During the design phase of the 3D LED display, there were a variety of standards and codes that had to be kept in mind. The most important of these were one, safety certifications for the high-voltage device, two, FCC certification for frequencies in the range the display will operate at and, three, adhering to standardized HDMI encoding and decoding protocols to ensure the device is compatible with other systems.

The most important design standards to focus on are those dealing with safety. If Voxel were to be commercialized in the US, it would first need to get UL certified. UL certification is important due to the high voltages the device operates at (~110AC). Several UL design standards can be used to certify a device has sufficient high-voltage isolation - Voxel would have to adhere to those standards which are geared towards consumer devices [13]. Another relevant UL standard is UL 796, which is for printed wiring boards [14]. Although PCB manufacturing was outsourced, it would still be necessary to certify the display PCB design meets UL standards and that the PCB manufacturer has been certified by a respected safety organization.

Due to the high frequencies Voxel operates at, it would be necessary to get an FCC certification before commercializing the product. The FCC, or Federal Communication Commission, regulates the usage of high frequencies in commercial devices because certain frequencies are reserved for government applications [15]. Since Voxel operates in the megahertz range, it would be necessary to report to the FCC what range of frequencies Voxel operates in and get approval to have the device communicate at a certain frequency level.

Another standard Voxel adheres to is the HDMI 2.1 Specification. The device uses HDMI 2.1 to transmit image and video data from a laptop to an FPGA that sends out graphical data to populate the display. All HDMI compliant devices must use a standardized type of cable and port to send/receive data from [12]. Voxel is an HDMI receiver device. Therefore, after receiving data, Voxel has to follow the standardized HDMI decoding algorithm so that accurate data is derived from the encoding stream coming in over HDMI.

4.3 Constraints, Alternatives, and Tradeoffs

One important constraint for the team to properly manage was the team's limited mechanical design knowledge. With this constraint in mind, the team opted for a relatively simple mechanical mount for the display, with a primary focus on stability. Although adding more complexity to the mechanical design could have made Voxel more aesthetic and made electrical integration easier, the team decided to make the mechanical design component as minimal as possible so that it did not create a bottleneck early on in the manufacturing process that hindered the development and programming of the actual display.

A key tradeoff that the team had to keep in mind throughout the prototyping process was the overall resolution of the 3D display versus update frequency. Data is transmitted over HDMI, which introduces an upper bound for the maximum amount of data Voxel can transmit. The amount of data needed is the 3D resolution of the displayed image multiplied by the frame rate. The team decided to focus on maximizing vertical and horizontal resolution, at the expense of a slightly lower framerate. This decision was made because the degree of image quality is very noticeable to those viewing the display, and therefore of high importance. So long as the framerate meets the minimum required value for the POV effect to take place, further increases in framerate will have little visual impact for most animations. The horizontal and vertical resolution are fixed, but if the team decided later on that higher framerate is desirable, some angular resolution could be sacrificed to achieve that end.

5.0 Schedule, Tasks, and Milestones

The team's original plan was to get the mechanical and electrical parts of the system working by mid-December so that January through April could be spent on software and animation design. Getting the mechanical system working and the electrical systems tested were on the critical path and limits the ability to test software which is why it was the team's main objective to complete by December. Unfortunately, we are unable to order parts until the start of January, which made our actual schedule diverge from the one in the GANTT chart in Appendix B. Table 3 highlights the approximate date when major tasks were actually completed.

Table 3. Task Completion Dates

Goal	**Completion Date**
Project proposal submitted	Nov. 3
Complete mechanical assembly	Nov. 28
PCB design review	Jan. 15
Order PCBs	Feb. 1
Begin FPGA code	Feb. 5
Complete mechanical manufacturing	Feb. 14
Begin testing PCBs	Feb. 18
Begin animation design	Mar. 1
Complete full system test	Apr. 23

Given how far the team progressed in the first semester of Senior Design, the team was optimistic on being able to complete Voxel on time. We initially estimated we had a 95% chance of finishing Voxel at least one week before the Senior Design Expo. The inability to order parts until January did set the team's schedule back, but fortunately we were able to finish everything we had originally set out to do in our original proposal before the Senior Design Expo.

6.0 Final Project Demonstration

The team primarily used the Senior Design Expo as an opportunity to demonstrate the full capabilities of Voxel. Voxel was paused and started up multiple times to help the audience realize what the product is and how it works. During each startup, the display was spun at a rate below the persistence of vision threshold. This showcased why rapid speeds and high refresh rates, which are the most technically challenging aspects of the project, are necessary. Next, the team operated the spinning display at full speed and to showcase a variety of different graphics. The 3D graphics ranged from still objects with simple geometries (e.g., a cube) to complex animations (e.g., people fighting). With sufficient update rates and high 3D pixel resolution, all attempted images and motions were displayed successfully.

With a streamlined animation-processing backend, part of our expo demo was to take requests from the audience and utilize open source 3D graphics sites (including Thingiverse and Free3D) to quickly generate and display any objects or animations. By displaying a variety of graphics with high resolution and quick turnaround, the team proved that Voxel can indeed create a novel 3D POV effect and with potential for a wide range of use cases.

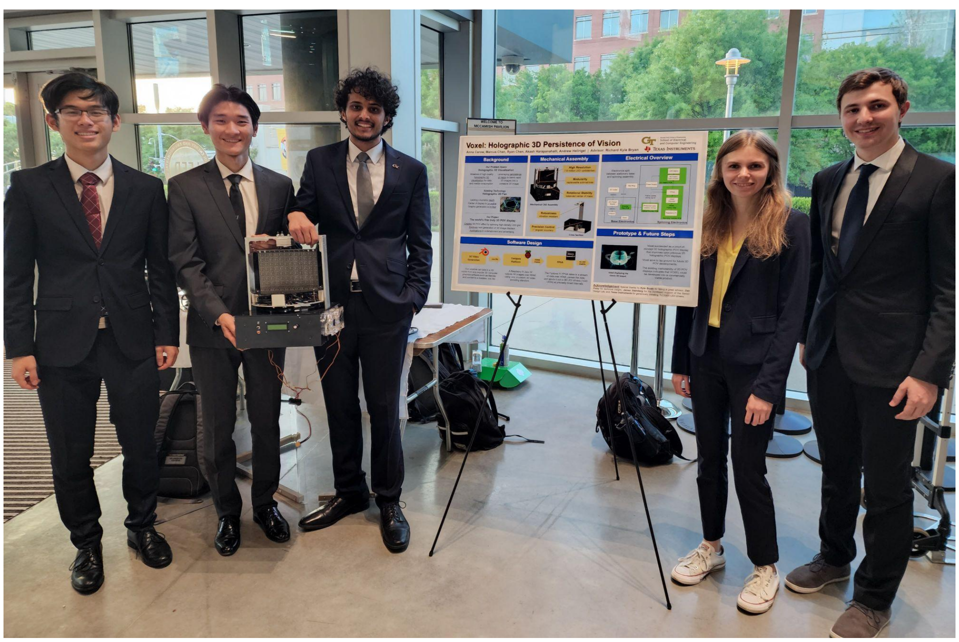


Figure 13. Team Voxel at GATech Senior Design Expo

Further detailed breakdown of prototyping and intermediate demonstrations are covered here. Project specifications that were met include:

- Sufficient power supplied to the motors and circuits for necessary speeds and brightness
- Sufficient update rates of the LEDs for appropriate angular resolution
- Sufficient RGB color space to distinguish between different shades in the 3D graphic

The specification parameters that differ from the proposal are the (1) rotational speed, dropping from 20 Hz to 15 Hz, (2) the operating current, needing significantly less amperage when realizing the LEDs need only a fraction of their full brightness for visibility, (3) the angular resolution, which dropped from 720 slices to 360 due to encoder optical encoder disk manufacturing capabilities, and (4) product weight which halved the expected specification.

System modularity testing was broken into mechanical testing and electrical testing. Mechanical assembly went smoothly. Integrating the motor and spinning base required CAD modifications and balancing to reduce vibrations and to stabilize the overall system. Electrical testing was broken into LED submatrix, FPGA controls, and user-interface/motor speed control testing. LED submatrices were individually plugged into a breadboard and verified with an Arduino script, then further debugged and fixed for any hardware or manufacturing defects. The FPGA controls required probing of signal lines, modifications of the PCB, and partial-to-full integration with the LED submatrices for signal integrity issues. The UI/motor speed control was integrated with ease, only needing to modify and iterate on the Arduino Nano code that interfaced user-input with motor control. Ultimately, all hardware (i.e., mechanical and electrical) system modules were fully integrated and tested with some animations, as seen in Figure 13 below. The software backend was then developed from front-to-back end, allowing for seamless and continuous testing and operation.

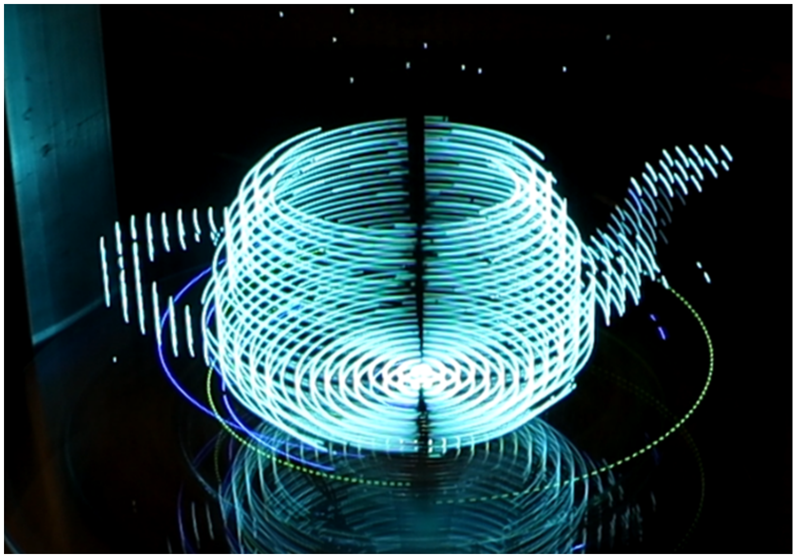


Figure 14. Voxel 3D Teapot Graphic Display

7.0 Marketing and Cost Analysis

7.1 Marketing Analysis

If brought to the market, Voxel would be able to provide a variety of features that do not exist in commercial image displays. Although the concept of seeing a moving image is of course nothing new on the market, a three dimensional image display is quite novel. Product designers have been working on creating the illusion of 3D images for years, and this has found its way into a variety of popular products such as 3D movies and VR headsets. Voxel brings display technology to a new level by providing a truly 3D image - not just the illusion of one.

Commercialized products that serve a similar function such as a 2D persistence of vision (POV) fan [1] or virtual reality goggles [2] can be found on the market for $300-$400 on the low end, and some products such as a holographic glass display [3] can be found for $17500. Voxel not only competes with these existing technologies, but also takes some of them to the next level. These existing technologies serve as a good reference point for cost. Voxel could competitively be marketed at $1000 due to its increased component costs and complexity compared to lower end products.

7.2 Cost Analysis

To determine a reasonable selling point for Voxel, an estimate can be made for the cost of a production run of 1 ku. It is estimated that raw stock and fasteners will take up 1/3 and 1/10 of the prototyping cost respectively. Most of the electronics should cost about 1/5 at scale and the main expenses such as the LED drivers, LEDs, and PCBs will be quoted directly.

Labor costs will also need to be factored into the selling price of Voxel. These include PCB fabrication, assembly and testing costs, material machining and manufacturing costs, overall assembly costs, quality control costs, and finally the costs for logistics, management, and overhead. Unskilled labor will be billed at $15/hr, medium skill labor will be billed at $25/hr, and skilled labor will be billed at $40/hr. Fringe benefits will be added at 30%.

The total cost will be increased by 5% to account for products that do not pass quality control. After that a reasonable selling price can be determined by aiming for a profit margin of 60%. With the costs listed in the tables below, and accounting for 5% defective products, a selling price of $999 gives a profit margin of 61.0%.

Table 4. Production Run Material Costs at 1ku

Item	**Total Cost**
Raw stock material (steel, aluminum, etc.)	$85.41
Fasteners	$18.05
Electronics	$74.88
LED Drivers	$223.00
LEDs	$103.42
PCBs	$3.97
Total Cost	$509.03

Table 5. Labor Costs at 1ku

Item	Hours	Rate	Total Cost
PCB fabrication, assembly, testing	1	$25	$25.00
Manufacturing costs	0.5	$25	$12.50
Assembly costs	1.5	$15	$22.50
Product testing / quality control	0.5	$15	$7.50
Logistics / overhead	0.25	$40	$10.00
Fringe benefits	n/a	30%	$23.25
		Total Cost	$100.75

The prototyping costs of this project are the sum of the electrical components, hardware components, research, and labor required for the prototype. These costs are broken out below. The total electrical cost for prototyping the electronics totals up to $2673.86. The total cost for the mechanical components totals up to $469.61. The total labor costs for the project would add up to $41340 if engineers were hired at $40/hr and manufacturers were hired at $20/hr.

The Voxel team is requested $2000 to complete the project. The total raw cost ended up being $3143.47 (see Appendix C). The team went over budget due to ordering delays that caused parts to become inavabilable before they were ordered. This increased the cost of our LEDs to about $1400 vs. $140 because they were the only suitable replacements that we could find. Had we not had this issue, we would've been approximately on budget.

8.0 Current Status

Voxel was able to be completed according to many of our design specifications. Our goals for the future of Voxel are to clean up some of the software and firmware to make it slightly more user friendly and understandable by people that wish to use our project for reference or people tha wish to use our software for 3D animation file generation.

9.0 Leadership Roles

Team Lead: Andrew Hellrigel

- Use the team-developed GANTT chart to set deadlines for tasks, and adjust the team's timeline as necessary, communicating any timeline changes (and the reasoning behind it) to the team.
- Manage the mechanical design and assembly of the product working with the rest of the team to make sure all of the subsystems will be able to work with each other and fit together.

Financial Lead: Anna Carow

- Keep track of all purchases the team makes, ensuring that things are ordered far enough in advance to deal with long lead times and that the team can request a budget increase from Senior Design if it becomes necessary.
- Focus on the presentation of Voxel, ensuring that the team has necessary presentation materials acting as the Expo coordinator.

Software Lead: Akash Harapanahalli

- Manage the design process of all of the software systems present in the design including both software and firmware.
- Make system level diagrams and flowcharts of the software systems along with any other documentation that is needed to keep the software systems organized.

Electrical and Documentation Lead: Ryan Chen

- Act as the documentation coordinator by working on creating technical designs, such as wiring layouts, to use as visuals in the final project proposal and to submit to other companies (such as a PCB manufacturer) for parts of the design being outsourced.
- Manage the design process of all the mechanical subsystems within the product.

Web Master: Marcus Chan

- Keep track of all deadlines for class assignments and ensure the group completes and submits them on time.
- Take meeting minutes during group discussions and during advisor meetings; schedule team and advisor meetings as needed throughout the semester.
- Act as the Webmaster to manage the team website, ensuring that the appropriate information gets added to the website.

10.0 References

[1] McKinney, Michael, “The Persistence of Vision,” *Vision*, 2015. [Online]. Available: https://www.vision.org/persistence-vision-1179. [Accessed Oct. 21, 2021].

[2] Tyson, Jeff, “How Computer Monitors Work,” *howstuffworks.com*, 2016. [Online]. Available: https://computer.howstuffworks.com/monitor1.htm. [Accessed Nov. 1, 2021]

[3] Vevor, “Vevor 65cm 3d Holographic Fan 768 Led Beads Wifi Hologram Advertising Projector,” *vevor.com,* [Online]. Available: https://www.vevor.com/products/vevor-65cm-3d-holographic-fan-768-led-beads-wifi-hologram-advertising-projector?gclid=CjwKCAjwwsmLBhACEiwANq-tXDxhqTerSDhXxFBKdVBF9SiTPpmNMBBUUjrhc0a1t-a6K50XplDIFhoCLHgQAvD_BwE. [Accessed Nov. 1, 2021].

[4] Oculus, “Quest 2,” *oculus.com*, [Online]. Available: https://www.oculus.com/quest-2/. [Accessed Nov. 1, 2021].

[5] Kozacki, Tomasz, “On resolution and viewing of holographic image generated by 3D holographic display,” *Optica Publishing Group*, vol. 18, issue 26, pp. 27118-27129, Dec. 9, 2010. [Online]. Available: https://www.osapublishing.org/oe/fulltext.cfm?uri=oe-18-26-27118&id=208729. [Accessed Nov. 1, 2021].

[6] Hammerschmidt, Christoph, “LEDs go for a Spin in POV Displays,” Power Electronics, Jul. 1 2011. [Online]. Available: https://www.powerelectronics.com/markets/article/21852498/leds-go-for-a-spin-in-pov-displays [Accessed Oct. 21, 2021].

[7] Texas Instruments, “TLC5955DCAR PWM LED Driver,” TLC5955 48-Channel, 16-Bit, PWM LED Driver Datasheet, Mar. 2014.

[8] Texas Instruments, "TFP401x TI PanelBusTM Digital Receiver," TFP401x Datasheet, Mar. 2000 [Revised May 2016].

[9] AF Inventions, "FPGA advantages," *AF Inventions*, 2021. [Online]. Available: https://af-inventions.de/en/services/fpga-development/fpga-advantages.html . [Accessed Nov. 1, 2021]

[10] Terasic, “DE0-Nano Development board Manual,” terasic.com, 2021. [Online]. Available:http://www.terasic.com.tw/cgi-bin/page/archive_download.pl?Language=English&No=593&FID=75023fa36c9bf8639384f942e65a46f3. [Accessed Nov.1, 2021].

[11] T. Wilson, “How HDMI Works,” *HowStuffWorks*, n.d. [Online]. Available: https://electronics.howstuffworks.com/hdmi.htm. [Accessed: Oct. 3, 2021].

[12] HDMI® Licensing Administrator, Inc, *HDMI® 2.1 Specification*, Aug. 1 2018. [Online]. Available: https://www.hdmi.org/spec/hdmi2_1. [Accessed Oct. 3, 2021].

[13] C3 Controls, “What is UL Certification,” c3controls.com, n.d. [Online]. Available: https://www.c3controls.com/white-paper/difference-between-ul-recognized-ul-listed/. [Accessed Oct. 21, 2021].

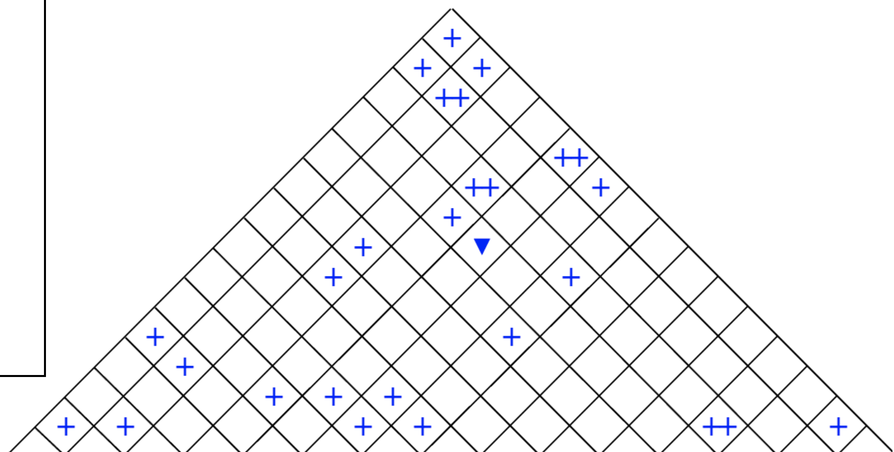
[14] Document Center, “UL-796,” documentcenter.com, Aug. 4, 2021. [Online]. Available: https://www.document-center.com/standards/show/UL-796. [Accessed Oct. 21, 2021].

[15] Federal Communications Commision, “Radio Spectrum Allocation,” *United States Federal Communications Commision*, n.d. [Online]. Available: https://www.fcc.gov/engineering-technology/policy-and-rules-division/general/radio-spectrum-allocation. [Accessed Oct. 21, 2021].

[16] “Looking Glass 8K Gen2,” *lookingglassfactory.com*, [Online]. Available: https://lookingglassfactory.com/product/8k. [Accessed Nov. 1, 2021].

Appendix A - QFD Chart

Legend	
Θ	Strong Relationship
O	Moderate Relationship
▲	Weak Relationship
++	Strong Positive Correlation
+	Positive Correlation
—	Negative Correlation
▼	Strong Negative Correlation
▼	Objective Is To Minimize
▲	Objective Is To Maximize
x	Objective Is To Hit Target



Column #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Direction of Improvement: Minimize (▼), Maximize (▲), or Target (x)	▲	▲	▼	▼	▲	x	x	x	x	x	▲	x	▼	▲	▲
Quality Characteristics (a.k.a. "Functional Requirements" or "Hows") / Demanded Quality (a.k.a. "Customer Requirements" or "Whats")	LED refresh rate	Rotational speed of LED display	Power Draw	Weight	2D Matrix Resolution	Number of LEDs	Cost (for scaled up production)	Number of LED Panels	Input Voltage	Max LED Current Draw	Storage	Display Size	Overall Size	Angular Resolution	3D Image Resolution
Lightweight				O								O	O		
Inexpensive						O	O					O	O		
Typical Power Draw			O						O	O					
Ease of Operation												O			
Ability to load custom images											O				
Smooth, continuous image	O				Θ							O		Θ	Θ
Easily Visible (i.e. bright enough)			O			O						O	O		
Target or Limit Value	20 Hz	1200 rpm	<150W	25lb	36 pixels/inch^2	2880	$999	2	110V AC	32.4A	25mb	5in x 8in	11 x 11 x 15 inches	720 updates per revolution	3094 voxels per inch^3

Appendix B - GANTT Chart

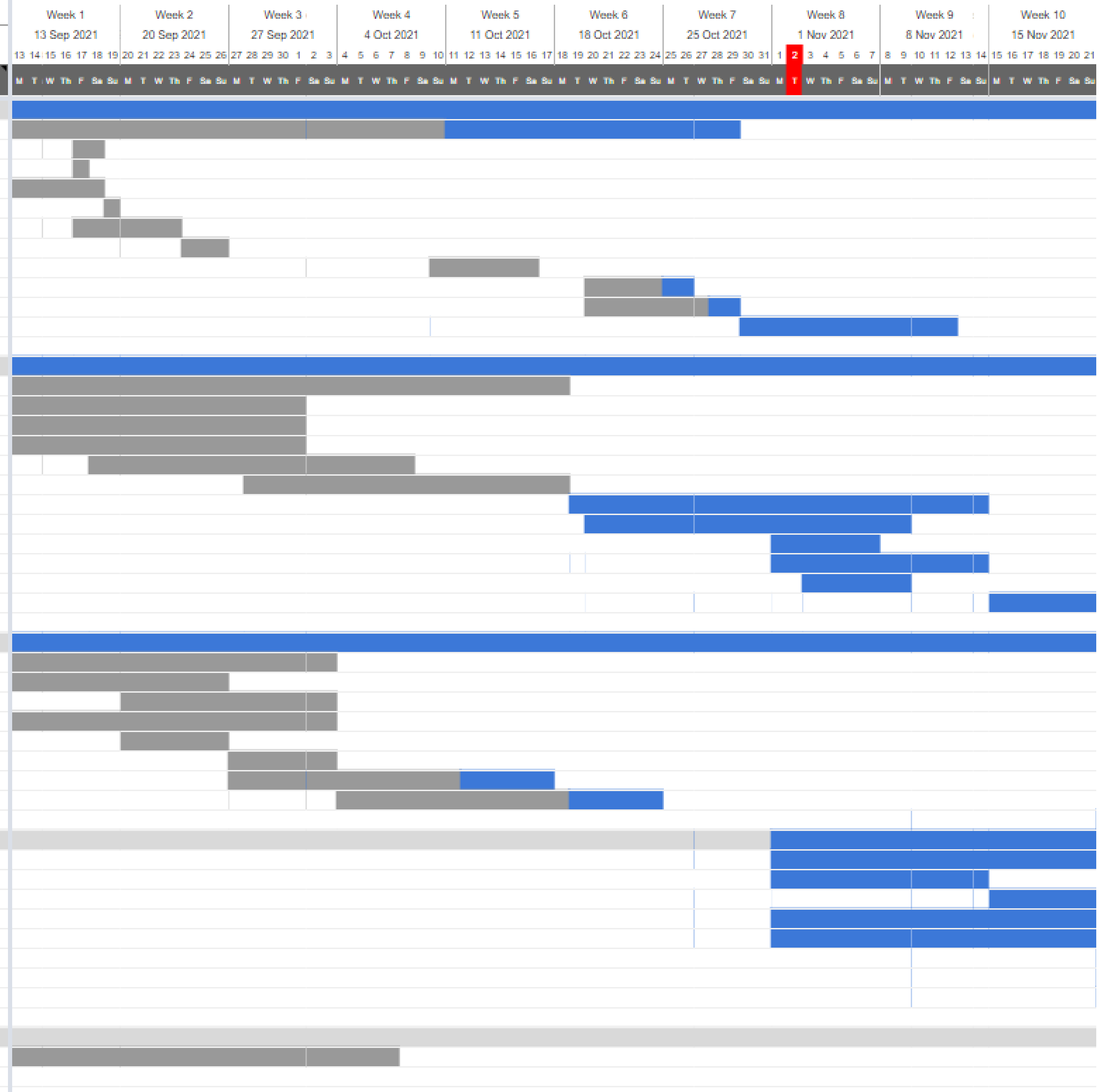
Spinning 3D LED Display Project Schedule

Project Start Date: 9/13/2021 (Monday) Display Week: 1

Project Manager: [Manager's Name]

[Go Pro](#)

WBS	Task	Lead	Start	End	Days	% Done	Work Days
1	**Logistics / Project Management**	Marcus	Sun 9/12/21	Thu 4/28/22	229		164
1.1	Class Projects	[Name]	Mon 9/13/21	Fri 10/29/21	47	60%	35
1.1.1	. Team Skills Matrix	Anna	Fri 9/17/21	Sat 9/18/21	2	100%	1
1.1.2	. Project QFD chart	Anna	Fri 9/17/21	Fri 9/17/21	1	100%	1
1.1.3	. GANTT Chart	Andrew	Sun 9/12/21	Sat 9/18/21	7	100%	5
1.1.4	. PERT Analysis	Marcus	Sun 9/19/21	Sun 9/19/21	1	100%	0
1.1.5	. Draft project summary form	Marcus	Fri 9/17/21	Thu 9/23/21	7	100%	5
1.1.6	. design standards and codes	Anna	Fri 9/24/21	Sun 9/26/21	3	100%	1
1.1.7	. project job budget and costing	Andrew	Sun 10/10/21	Sat 10/16/21	7	100%	5
1.1.8	. final project summary form	Andrew	Wed 10/20/21	Tue 10/26/21	7	80%	5
1.1.9	. project proposal	Andrew	Wed 10/20/21	Fri 10/29/21	10	80%	8
1.2	Finance / Parts Ordering		Sat 10/30/21	Fri 11/12/21	14	0%	10
1.3	Presentation		Fri 4/01/22	Thu 4/28/22	28	0%	20
2	**Mechanical Design**	Andrew	Sat 9/11/21	Sun 12/26/21	107		75
2.1	CAD Model Design		Sat 9/11/21	Mon 10/18/21	38	100%	26
2.1.1	. Assembly for LED matrix holder		Sat 9/11/21	Fri 10/01/21	21	100%	15
2.1.2	. Design base/power electronics storage		Sat 9/11/21	Fri 10/01/21	21	100%	15
2.1.3	. Design rotating base for control electronics storage		Sat 9/11/21	Fri 10/01/21	21	100%	15
2.1.4	. Design upper assmbly for LED matrix support		Sat 9/18/21	Fri 10/08/21	21	100%	15
2.1.5	. Integrate sub-assemblies		Tue 9/28/21	Mon 10/18/21	21	100%	15
2.2	Manufacturing		Tue 10/19/21	Sun 11/14/21	27	0%	19
2.2.1	. 3D Printing parts		Wed 10/20/21	Tue 11/09/21	21	0%	15
2.2.2	. Laser cut components		Mon 11/01/21	Sun 11/07/21	7	0%	5
2.2.3	. Machine/cutting aluminum support		Mon 11/01/21	Sun 11/14/21	14	0%	10
2.2.4	. Applying finish/sanding/cleaning		Wed 11/03/21	Tue 11/09/21	7	0%	5
2.3	Assembly		Mon 11/15/21	Sun 11/28/21	14	0%	10
2.4	Testing / Rework		Mon 11/29/21	Sun 12/26/21	28	0%	20
3	**Electrical Design**	Ryan	Mon 9/13/21	Sun 12/19/21	98		70
3.1	Design Full System Architecture		Mon 9/13/21	Sun 10/03/21	21	100%	15
3.1.1	. Choose microcontroller/processor		Mon 9/13/21	Sun 9/26/21	14	100%	10
3.1.2	. Choose LED Driver / Architecture		Mon 9/20/21	Sun 10/03/21	14	100%	10
3.2	Power Electronics		Mon 9/13/21	Sun 10/03/21	21	100%	15
3.2.1	. Power delivery system to LED matrix		Mon 9/20/21	Sun 9/26/21	7	100%	5
3.2.2	. Overall system power design		Mon 9/27/21	Sun 10/03/21	7	100%	5
3.3	LED Matrix Design		Mon 9/27/21	Sun 10/17/21	21	75%	15
3.4	LED Control Board		Mon 10/04/21	Sun 10/24/21	21	75%	15
3.5	Testing / Rework		Mon 11/22/21	Sun 12/19/21	28	0%	20
4	**Embedded Control Code**	Anna	Mon 11/01/21	Sun 1/16/22	77		55
4.1	Motor Control Code		Mon 11/01/21	Sun 11/28/21	28	0%	20
4.1.1	. Senser code to measure speed/position		Mon 11/01/21	Sun 11/14/21	14	0%	10
4.1.2	. Software for setting motor to a specifi speed (closed loop c		Mon 11/15/21	Sun 11/28/21	14	0%	10
4.2	LED Driver Code		Mon 11/01/21	Sun 1/16/22	77	0%	55
4.2.1	. Be able to interface with single LED driver at desired frequer		Mon 11/01/21	Sun 11/21/21	21	0%	15
4.2.2	. Parellelize LED update code		Mon 11/22/21	Sun 1/16/22	56	0%	40
4.3	Data Management Code		Mon 11/22/21	Sun 1/16/22	56	0%	40
4.3.1	. Design communication method for transmitting 3D data from		Mon 11/22/21	Sun 12/19/21	28	0%	20
4.3.2	. Increase data rates		Mon 12/20/21	Sun 1/16/22	28	0%	20
5	**Software Design**	Akash	Mon 12/20/21	Sun 4/10/22	112		80
5.1	Simulation of Visual Effect		Fri 9/10/21	Thu 10/07/21	28	100%	20
5.2	Image Creation Software		Mon 12/20/21	Sun 2/13/22	56	0%	40
5.3	Animation Design		Mon 2/14/22	Sun 4/10/22	56	0%	40



Spinning 3D LED Display Project Schedule

Project Start Date: 9/13/2021 (Monday) Display Week: 1

Project Manager: [Manager's Name]

WBS	Task	Lead	Start	End	Days	% Done	Work Days
1	**Logistics / Project Management**	Marcus	Sun 9/12/21	Thu 4/28/22	229		164
1.1	Class Projects	[Name]	Mon 9/13/21	Fri 10/29/21	47	60%	35
1.1.1	. Team Skills Matrix	Anna	Fri 9/17/21	Sat 9/18/21	2	100%	1
1.1.2	. Project QFD chart	Anna	Fri 9/17/21	Fri 9/17/21	1	100%	1
1.1.3	. GANTT Chart	Andrew	Sun 9/12/21	Sat 9/18/21	7	100%	5
1.1.4	. PERT Analysis	Marcus	Sun 9/19/21	Sun 9/19/21	1	100%	0
1.1.5	. Draft project summary form	Marcus	Fri 9/17/21	Thu 9/23/21	7	100%	5
1.1.6	. design standards and codes	Anna	Fri 9/24/21	Sun 9/26/21	3	100%	1
1.1.7	. project job budget and costing	Andrew	Sun 10/10/21	Sat 10/16/21	7	100%	5
1.1.8	. final project summary form	Andrew	Wed 10/20/21	Tue 10/26/21	7	80%	5
1.1.9	. project proposal	Andrew	Wed 10/20/21	Fri 10/29/21	10	80%	8
1.2	Finance / Parts Ordering		Sat 10/30/21	Fri 11/12/21	14	0%	10
1.3	Presentation		Fri 4/01/22	Thu 4/28/22	28	0%	20
2	**Mechanical Design**	Andrew	Sat 9/11/21	Sun 12/26/21	107		75
2.1	CAD Model Design		Sat 9/11/21	Mon 10/18/21	38	100%	26
2.1.1	. Assembly for LED matrix holder		Sat 9/11/21	Fri 10/01/21	21	100%	15
2.1.2	. Design base/power electronics storage		Sat 9/11/21	Fri 10/01/21	21	100%	15
2.1.3	. Design rotating base for control electronics storage		Sat 9/11/21	Fri 10/01/21	21	100%	15
2.1.4	. Design upper assmbly for LED matrix support		Sat 9/18/21	Fri 10/08/21	21	100%	15
2.1.5	. Integrate sub-assemblies		Tue 9/28/21	Mon 10/18/21	21	100%	15
2.2	Manufacturing		Tue 10/19/21	Sun 11/14/21	27	0%	19
2.2.1	. 3D Printing parts		Wed 10/20/21	Tue 11/09/21	21	0%	15
2.2.2	. Laser cut components		Mon 11/01/21	Sun 11/07/21	7	0%	5
2.2.3	. Machine/cutting aluminum support		Mon 11/01/21	Sun 11/14/21	14	0%	10
2.2.4	. Applying finish/sanding/cleaning		Wed 11/03/21	Tue 11/09/21	7	0%	5
2.3	Assembly		Mon 11/15/21	Sun 11/28/21	14	0%	10
2.4	Testing / Rework		Mon 11/29/21	Sun 12/26/21	28	0%	20
3	**Electrical Design**	Ryan	Mon 9/13/21	Sun 12/19/21	98		70
3.1	Design Full System Architecture		Mon 9/13/21	Sun 10/03/21	21	100%	15
3.1.1	. Choose microcontroller/processor		Mon 9/13/21	Sun 9/26/21	14	100%	10
3.1.2	. Choose LED Driver / Architecture		Mon 9/20/21	Sun 10/03/21	14	100%	10
3.2	Power Electronics		Mon 9/13/21	Sun 10/03/21	21	100%	15
3.2.1	. Power delivery system to LED matrix		Mon 9/20/21	Sun 9/26/21	7	100%	5
3.2.2	. Overall system power design		Mon 9/27/21	Sun 10/03/21	7	100%	5
3.3	LED Matrix Design		Mon 9/27/21	Sun 10/17/21	21	75%	15
3.4	LED Control Board		Mon 10/04/21	Sun 10/24/21	21	75%	15
3.5	Testing / Rework		Mon 11/22/21	Sun 12/19/21	28	0%	20
4	**Embedded Control Code**	Anna	Mon 11/01/21	Sun 1/16/22	77		55
4.1	Motor Control Code		Mon 11/01/21	Sun 11/28/21	28	0%	20
4.1.1	. Sensor code to measure speed/position		Mon 11/01/21	Sun 11/14/21	14	0%	10
4.1.2	. Software for setting motor to a specifici speed (closed loop c		Mon 11/15/21	Sun 11/28/21	14	0%	10
4.2	LED Driver Code		Mon 11/01/21	Sun 1/16/22	77	0%	55
4.2.1	. Be able to interface with single LED driver at desired frequer		Mon 11/01/21	Sun 11/21/21	21	0%	15
4.2.2	. Parellelize LED update code		Mon 11/22/21	Sun 1/16/22	56	0%	40
4.3	Data Management Code		Mon 11/22/21	Sun 1/16/22	56	0%	40
4.3.1	. Design communication method for transmitting 3D data from		Mon 11/22/21	Sun 12/19/21	28	0%	20
4.3.2	. Increase data rates		Mon 12/20/21	Sun 1/16/22	28	0%	20
5	**Software Design**	Akash	Mon 12/20/21	Sun 4/10/22	112		80
5.1	Simulation of Visual Effect		Fri 9/10/21	Thu 10/07/21	28	100%	20
5.2	Image Creation Software		Mon 12/20/21	Sun 2/13/22	56	0%	40
5.3	Animation Design		Mon 2/14/22	Sun 4/10/22	56	0%	40

Appendix C - Bill of Materials

Table 6. Electrical Components

Item	Unit Cost	Qty	Total Cost
LED Drivers	$3.49	150	$523.50
LEDs	$0.73	2000	$1460.00
Raspberry Pi Zero W	$10.00	1	$10.00
24V DC Motor	$28.09	2	$56.18
DC motor driver	$49.95	1	$49.95
24V AC-DC Power Supply	$18.95	2	$37.90
5V Buck Regulator	$19.95	1	$19.95
3.3V Buck Regulator	$22.95	1	$22.95
3.3V Linear Regulator	$2.72	2	$5.44
HDMI cable	$18.99	1	$18.99
Optical encoder	$4.73	4	$18.92
Right angle male headers 2x4pos	$0.21	20	$4.26
Female headers 2x4pos	$0.56	20	$11.28
PCBs	$170.00	1	$170.00
5V Power Supply for Raspberry Pi	$9.99	1	$9.99
DE0 Nano FPGA Eval Board	$73.00	1	$73.00
Miscellaneous electrical components	$181.55	1	$181.55
		Total Cost	$2673.86

Table 7. Hardware Components

Item	Unit Cost	Qty	Total Cost
Non Slip Pads	$7.98	1	$7.98
6 Wire Slip Ring 1000RPM	$54.98	2	$109.96
12"x24" Black Delrin Sheet	$36.98	3	$110.94
1"x1"x1/16" Aluminum Angle 4'	$12.95	1	$12.95

1'x1'x1/8" 6061 Aluminum Sheet	$32.90	2	$65.80
Aluminum Angle Brackets	$18.59	1	$18.59
M2.5 Standoff kit	$12.99	2	$25.98
1/4"x2" Aluminum Bar 3'	$32.92	1	$32.92
M3 Thread Forming Screws 5mm	$5.55	1	$5.55
M3 Locknuts	$3.82	1	$3.82
M4 Locknuts	$4.59	1	$4.59
M5 Locknuts	$4.96	1	$4.96
M4 Threaded Rod 300mm	$5.60	1	$5.60
M5 16mm Countersunk bolts	$10.56	1	$10.56
M4 20mm Countersunk bolts	$8.93	1	$8.93
8mm ID shaft hubs	$9.59	1	$7.99
M3 35mm Socket Head Screw	$4.50	1	$4.50
M4 Nut	$1.41	1	$1.41
M4 12mm Socket Head Screw	$4.60	1	$4.60
M2.5 10mm Button Head Screws	$5.13	1	$5.13
M3 12mm Socket Head Screw	$5.55	1	$5.55
1/2" Shaft 1-1/8" Housing Ball Bearing	$11.30	1	$11.30
		Total Cost	$469.61