Final Design Report

Jessie Almon
jeal227@g.uky.edu

Hao Bai
hba237@g.uky.edu

David Carpenter
david.carpenter@uky.edu

Ben Ragusa
benragusa@uky.edu

Matt Ruffner
mpru223@g.uky.edu

Bill Williams
brwill3@g.uky.edu

December 5, 2016

Team 14: OrbIT

Mentors
Dr. Samson Cheung
Dr. Kevin Donohue
Dr. Regina Hannemann

Sponsors
Matt Ruffner

Contents

List of Figures

List of Tables

Abstract

Traditional print ads are durable and distributable but can be too detailed and easily overlooked, not to mention a waste of paper. We propose a connected, innovative and affordable advertising solution which employs visually intriguing display techniques to draw attention. Usable by storefronts, restaurants and businesses it lends a modern approach to mountable signage with wireless and online content management. This device will have WiFi connectivity for ease of reprogramming, and at least be able to display a textual message. We propose a lightweight rotating ring structure which is enclosed in an acrylic globe, protecting it. Using small components allows for a minimal design.

1 Problem Statement

1.1 Need

Most people by now are familiar with the billboards that change advertisements over time. This sort of interconnected marketability is certainly the future of advertising, since central control over displayed media is superior from a management aspect. These sorts of advertisements are showing up in much simpler forms, from TVs being utilized as digital menus in restaurants to smaller screens implemented as less-frequently changed promos in other establishments.

A reprogrammable sign for storefronts and restaurants engineered such that its ease of content management, versatility and price point make it comparable to ordinary storefront signs would have excellent marketability. Not only would the display be inexplicably more eye catching, but a design approach utilizing the Internet of Things (IoT) would lead to a more easily connectable and scalable product.[1]

1.2 Objective

Our objective with this project is to design, build, test and implement a persistence of vision based advertising solution. This could then be mounted above the entrance to the Electrical and Computer Engineering Commons on the fourth floor of F.Paul Anderson Tower to display the time and a welcome message to students. Pricing the globe clock under $500 would make it more appealing to potential customers since the starting retail price of competition is $700. By requiring the device to support wireless communication, its functionality has expanded to provide a more versatile, valuable and useful product to the end user. Eliminating the need for physical interaction with the device also introduces more mounting opportunities: ceilings, unreachably high walls, et. al.

1.3 Background

Digital signs are being rapidly adopted in eateries, shops and stores as a lower long-term cost advertising solution.[3] Billboards are certainly the large scale picture of this advertising technique, with smaller scale versions for store fronts appearing as dedicated PC monitors or LED signs that scroll predetermined text. These are both eye catching means of advertising, though the key limiting factor of them is their lack of physical movement; animated lights can only draw so much attraction. However, by utilizing a trick of the eye known as persistence of vision (POV as it shall be referred to), moving modulated LEDs are capable of simulating an image. This physical movement of the lights makes the sign inherently more eye-catching to a passerby. An example of this 'simulated' image can be seen in Fig. 1.

Persistence of Vision (POV) signs are not new; there are a few solutions available to today's market. Although outdated, they are certainly attention grabbing, even if they do have limited functionality, size and color replication.

One such example of the existing POV sign technology is the SpaceWriter iBall [4] seen in Figure. 1. It is limited by 3 bit color (8 total colors) as well as an 8 inch vertical display height. This might be sufficient for simple graphics and ad animations, but there is only so much one can do with 8 colors. The interface for uploading new content to the iBall is a microSD card slot. The user must use archaic PC software to create 'frames' from provided images, or to create text. The instructional manual for this software can be found in one place [5], which is linked to by none of their advertising pages. The only available place to purchase such a device is eBay.

Figure 1: The basic SpaceWriter iBall [2], to give an idea of the concept.

By minimizing the energy consumption of the motor that spins the device, POV signs can be made to be more efficient than traditional scrolling text rectangular signs. Since the total number of LEDs required for a POV sign is a fraction of that required for a rectangle of LEDs with the same surface area, energy savings are possible. The reduction in LED usage in POV signs also helps cut production costs due to the fraction of LEDs required.

Sean Kelley, general manager of Buffalo Wild Wings on South Broadway, was consulted to see what sort of characteristics an advertising device would need in order to be noticed and thrive in already media-rich environments. According to Sean, certain TVs in the restaurant were set to run predetermined ads and logos several times per hour. When told about an ad display that would catch the customers eye more effectively, he agreed that there would certainly be a use for it. He also explained limitations; it would not be a table top device since room is already limited. Enclosing the device was also discussed, it was deemed dependent on size and height of mounting position.[6]

2 Requirements Specification

The device's functional goals were segmented into marketing requirements based on the importance put on specific qualities of operation. The goals represented with the word *shall* are necessary core functions for the globe clock while the goals represented with the word *should* are goals that while not necessary would improve the marketability and appeal of the globe. Mandating some details of design while still including lofty goals allowed us to set core goals for us to not only succeed, but to potentially go above and beyond by creating less-likely features.

2.1 Marketing Requirements

The concept of visualizing ad material on this emulated surface is quite scalable. Care was taken to design marketing requirements that do not specify a size, but rather a specific resolution for the device to achieve. Letting text and image refresh rate specifications imply the necessary windows in which a LED should pulse to generate a pixel allows for variously sized Globes to be feasible. The features which are definite are denoted by the italicized *shall.* Those that would deem us as having gone above and beyond, if they are to be completed, are denoted by *should.* The marketing requirements as we have formed them are shown below.

Requirement	**Status**
1. The device *shall* be able to display a 32 letter message (16 characters by 2 lines) around the circumference of the Globe (3360 bytes). This is the *standard configuration.*	**IN PROGRESS**
2. The device *shall* be priced competitively (currently less than $500).	**COMPLETE**
3. The device *shall* not be affected by people touching it or surrounding areas.	**COMPLETE**
4. The device *shall* be reprogrammable without physical interaction with the device.	**COMPLETE**
5. The device *shall* be easily mountable.	**IN PROGRESS**
6. The device *shall* be powered from the socket with which it is mounted.	**IN PROGRESS**
7. The device *shall* be more energy efficient than competition.	**COMPLETE**
8. The device *should* be able to store 1 *standard configuration* of text.	**COMPLETE**
9. The device *should* be able to be mounted on the ceiling with clearance for bystanders.	**IN PROGRESS**
10. The device *should* be able to display 1 image.	**IN PROGRESS**
11. The device *should* be able to display content clearly	**IN PROGRESS**
12. The device *should* be able to operate outside in the elements.	**IN PROGRESS**

2.2 Objective Tree

The objective tree found in Fig. 2 is based on the marketing requirements which are outlined below. The tree shows the high importance put on the devices advertising effectiveness. Safety can be encompassed in one chunk as the main foreseen danger is high speed spinning parts. Keeping both the upfront and long-term costs low is of equal importance to the device. This also ties into the device's need to be energy efficient.

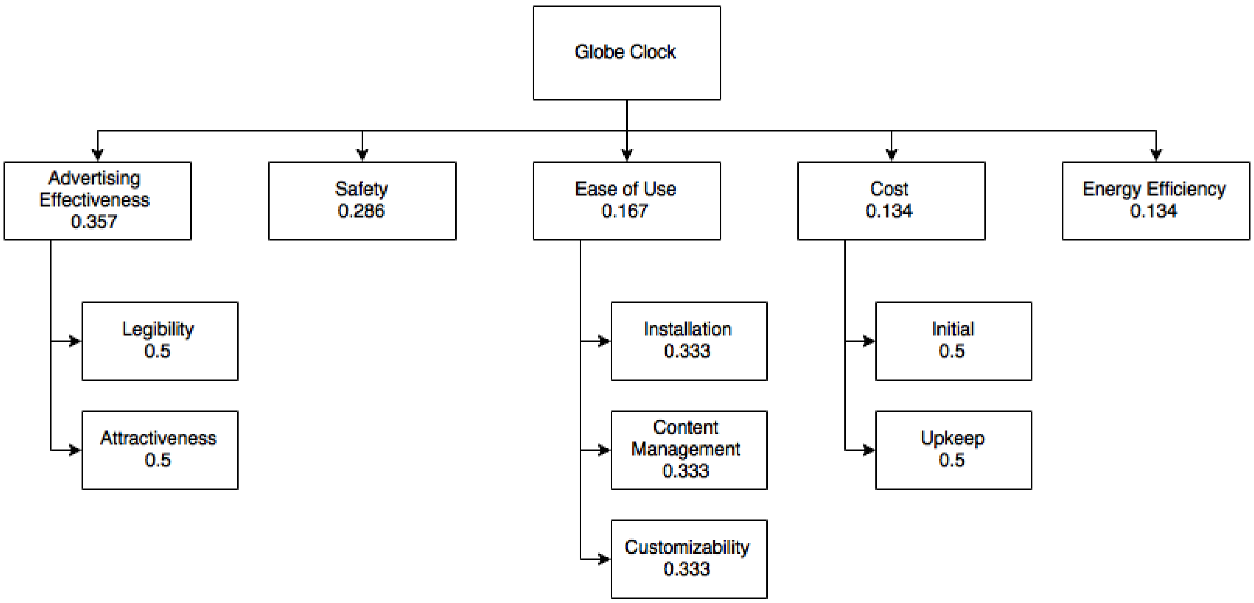


Figure 2: The objective tree with weights found using AHP method. This method originates from the mandatory class textbook[7]. The weight calculations for tree's various criteria can be found in Appendix A

2.3 Engineering Requirements

Requirement	Rel. Marketing Req.
1. The device *shall* be able to connect to 2.4Ghz 802.11b/g networks.	4
1.1. The device *shall* be able to store 1 *standard configuration* of uploaded text.	8,4
1.2. The device *shall* be able to open a TCP or UDP connection to transfer the data to display.	4
1.3. The device *should* be able to broadcast its own 802.11b/g network with a preset SSID.	4
1.4. The device *should* be able to connect to 2.4Ghz 802.11n networks.	4
1.5. The device *should* be able to store 1 uploaded image.	
2. The device *shall* spin at a minimum of 900 RPM to achieve at least 15 fps.	1,11
2.1. The pixels *shall* be able to turn on and off within 833μs to achieve 15 fps. (to meet text display specifications)	1,11
2.2. The resolution per character *shall* be greater than 5x7.	1,11
2.3. The pixel spacing must be close enough such that 14 pixels fit within the 45° section in the middle of the globe.	
3. The device *should* display a distinguishable image.	10,11
3.1. The pixels *should* be able to turn on and off within 521μs to achieve 24fps (to meet image display specifications).	7,11
3.2. Distortion along the edges of images *should* be minimized.	9,11
3.3. The image *should* be at least 14x80.	7,10,11
4. The device *shall* be able to replicate at least 256 colors using RGB LEDs.	1,7
5. The device shall consume less than 45W of power when displaying text.	7
6. Moving parts pertaining the creation of the POV image *shall* be enclosed by a physical barrier with minimal visual degradation.	3,9,12

2.3.1 Justification and Testability

Table 1 shows the sub-specifications of Engineering Requirement 1.

Table 1: The device *shall* be able to connect to 802.11b/g networks, Ch. 1-11. (Marketing Requirement 4)

	Engineering Requirements	Justification	Testing	Status
1.1	The device *shall* be able to store 1 *standard configuration* of uploaded text.	The displayed text will be an advertisement or other relevant information for the viewer	Look at the globe clock and see if it displays text correctly.	Complete
1.2	The device *shall* be able to open a TCP or UDP connection.	This allows for data transfer from a computer to the display so that text and images can be changed	Transfer known data over TCP connection; dump over serial from μController and verify similarity.	Complete
1.3	The device *should* be able to broadcast its own 802.11b/g network with a preset SSID.	This give the globe clock the ability to receive data from a computer.	Check and see if a 802.11b connected device shows the globe clock's broadcast.	Complete
1.4	The device *should* be able to connect to 24 GHz 802.11n networks.	This lets the globe clock connect to a computer	Successfully connect the globe clock to various other 24 GHz 802.11n enabled devices	Complete
1.5	The device *should* be able to store 1 uploaded image.	This will allow the owner to display an advertisement or other relevant information as an image	Upload an image and look at the globe clock to see if it displays correctly	Complete

Table 2 shows the sub-specifications of Engineering Requirement 2.

Table 2: The device *shall* spin at a minimum of 900 RPM to achieve at least 15 fps. (Marketing Requirement 1)

	Engineering Requirements	Justification	Testing	Status
2.1	The pixels *shall* be able to turn on and off within 8.33 ms to achieve 15 fps. (to meet text display spec)	These values are necessary for the transitions and movements to appear fluid to the human eye.	Monitor LED data lines with logic analyzer. [1]	Complete
2.2	The resolution per character *shall* be greater than 5x7.	This ensures that letters don't appear grainy to the viewer.	Measure the number of pixels across characters and verify that its greater than 5x7.	In Progress
2.3	The pixel spacing must be close enough such that 14 LEDs fit within the 45° section in the middle of the globe	The pixel density needs to be high enough so that an image can be displayed clearly on the globe clock.	Count the number of pixels around the clock, divide by 4 and check to see if this number is 14 or greater.	Complete

Table 3 shows the sub-specifications of Engineering Requirement 3.

Table 3: The globe clock *should* display a distinguishable image. (Marketing Requirement 7).

	Engineering Requirements	Justification	Testing	Status
3.1	The pixels *should* be able to turn on and off within 521μs ms to achieve 24fps.	This will make the image appear clear to the viewer while the globe clock is spinning.	Monitor LED line with logic analyzer	In Progress
3.2	Distortion along the edges of the image *should* be minimized.	The image should be displayed clearly on the surface of the sphere.	Display different test images onto the globe clock and ensure they are mapped clearly.	In Progress
3.3	The image *should* be at least 14x80.	This ensures that an image won't appear grainy to the viewer.	Count the number of pixels displayed on the image displaying portion of the globe clock.	In Progress

Table 4 shows the rest of the engineering requirements outlined in 2.3.

Table 4: Other Engineering Requirements as seen in 2.3.

	Engineering Requirements	Justification	Testing	Status
4	The device *shall* be able to replicate at least 256 colors using RGB LEDs	To have greater color reproduction capabilities than the competition. [2]	Check LED spec sheet to verify PWM output is more than 7bits/color.	Complete
5	The device *shall* consume less than 45W of power when displaying text.	To be more efficient than the competition	Calculate and measure current consumed by circuitry and motor.	Complete
6	Moving parts pertaining the creation of the POV image *shall* be enclosed by a physical barrier with minimal visual degradation	To protect bystanders, consumers and the circuitry.	Subject the enclosed globe to jostling forces.	Complete

2.4 Impact

2.4.1 Economic

The globe clock is economically viable, because it allows the buyer to earn consistent ad revenue without spending much on capital or other reoccurring costs. The globe clock is built on energy efficient technologies such as LED lights so it will consume around 45 Watts of power which is less than a typical incandescent light bulb. Additionally, the globe clock will be sold for $500 which is a low start up cost compared to alternative means of advertising such as billboards or radio ads. The start up cost will be recouped quickly by the profits made from advertising thus making the globe clock an economically viable advertising solution.

2.4.2 Social

In order to meet the social need the globe clock will be more attractive, effective and impressionable. We should make the information displayed with rotating ring of LEDs become the great bright spot compared to the traditional print ads while benefiting the users by creating an advertising form that both attracts attention and stimulates customer response. In addition, it should have widespread use in society by being easy to install and use.

2.4.3 Safety

The globe clock will be widely used in storefronts and businesses, so the safety of customers is very important. The Globe Clock will not interfere with its environment due to its enclosure. This will ensure that no harm would be incurred on any person should they so happen to come in contact with it.

2.4.4 Manufacturability

Despite the flashing lights and mesmerizing nature of the globe clock, the underpinnings are built simply. The components are straightforward to assemble which will increase the scalability of mass manufacturing. Additionally, much of the underpinnings of the globe clock is software, which is exceptionally easy to mass produce, and furthers scalability.

2.4.5 Environmental

The globe clock is built to be environmentally sustainable as it encompasses technologies to keep energy consumption minimal. This not only keeps the operating costs of the globe clock down but also reduces the carbon footprint by not causing the power plants to burn as much fuel. The globe clock consumes only 45 watts of power which is substantially less than other advertising mediums such as billboards or television. Additionally, despite the advances of modern technology, many ads are still in print whether it be in the form of fliers, mail, or newspapers. If companies advertised on the globe clock instead of in print, less paper could be used and trees could be saved. Thus, advertising with the globe clock would be much more environmentally friendly than alternative advertising platforms.

3 Design

3.1 Systems Interaction

The globe clock harnesses many systems working in unison to provide an efficient, minimalist design. The mechanical portion of the Globe, responsible for rotating

To provide a global overview of how all of the subsystems interact, we have provided Figure 3. In this project, hardware and software design are very closely intertwined. The mechanical hardware was designed to effectively power the electrical hardware while not being overly bulky or visually obstructive. The top priority to ensure proper functionality is balance. Setting up something with minimal slack means that something pre-machined would be ideal. A stainless steel shaft with mated gear, nylon belt and motor were salvaged from an inkjet printer. This ensured tightly coupled gear interactions and provided a motor with predetermined mounting holes.

Design files, source code and referenced libraries can be found on the projects repository page, `https://github.com/ruffner/globe-clock`.

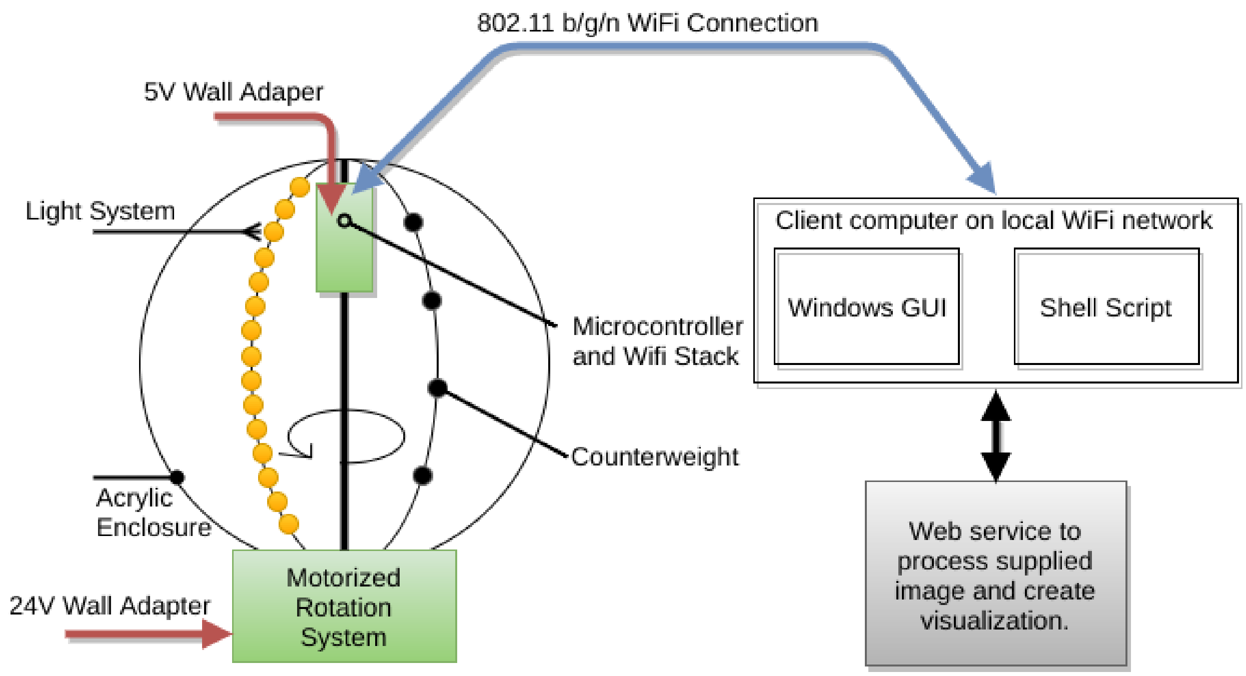


Figure 3: A high level overview of subsystem interaction

3.2 Hardware Design

It is necessary to create a rotating ring that would be able to be inserted and taken out of an acrylic enclosure. Without this, the acrylic globe seen in Figure 25 would have to be severed. This would create an unsightly ring around it that would degrade the quality of the perceived image. To avoid this, a concentric shaft system was devised which allows the spring steel rods which hold the LED strip to deform into an oblong shape which will fit into the acrylic sphere more easily.

Figure 4: The acrylic sphere and partial view of rotating ring.

3.2.1 Level 0 Hardware

Our top level design consists of affixing springed steel bands onto a solid metal rod. The concentric shaft that attaches the steel bands to the rod is extendable which allows the globe to easily slide in and out of the acrylic sphere. This is shown in figure 12.

In this section, physical parts needed to create this contraption will be outlined, including mechanical operational schematics and dimensional drawings for parts that need to be 3D printed.

3.2.2 Level 1 Hardware

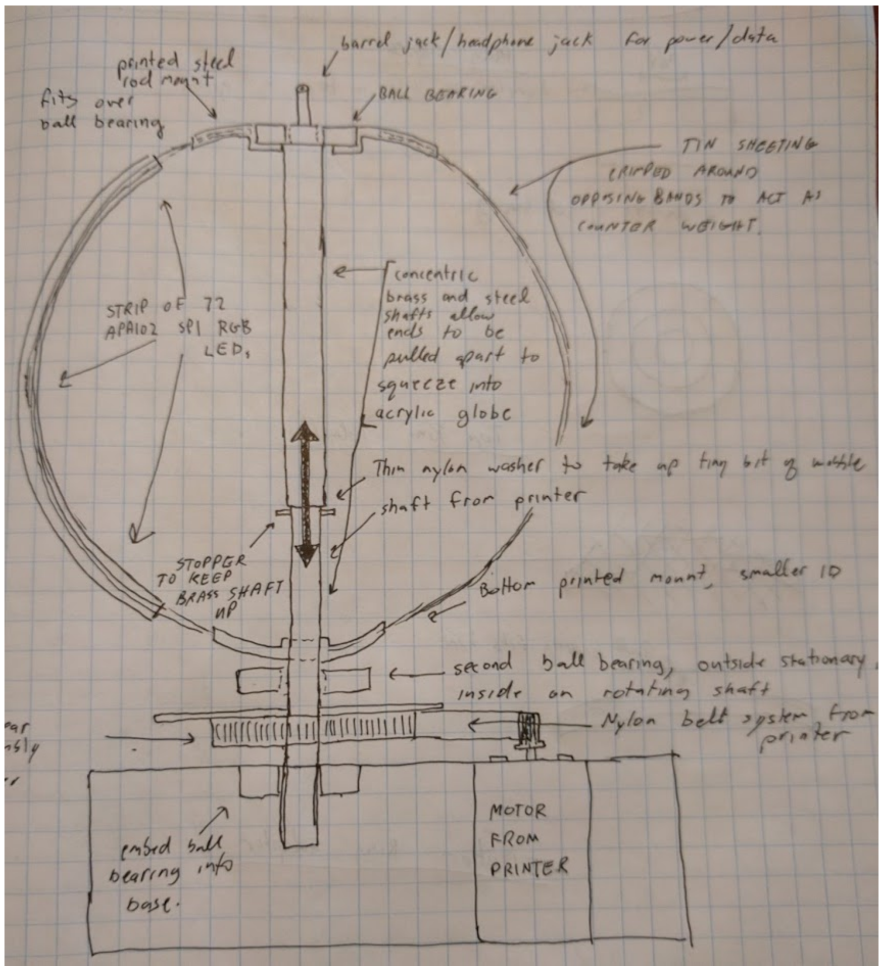


Figure 5: The scavenged motor mating with the main drive gear. There will be a nylon drive belt attaching the two when the connection is implemented.

Packaging For the actual enclosure for the spinning LEDs, there were two thought processes; cover each individual LED bulb and let the strip of LEDs spin freely, or cover the entire unit inside of what can be thought of as a classic fish bowl type enclosure. After further research, it quickly became evident that simply protecting each individual LED would not aid in the protection of the overall structure. This solution is based on engineering requirements 6, 6.1.The strip of LED lights is placed atop a thin plastic ring and is very flimsy. Even if this strip was reinforced with a stronger material, it would add little to the overall integrity of the structure and the globe itself would still be vulnerable to damage. Covering just the LEDs would also do nothing for the overall product pertaining to the weather elements if it was mounted outside. Therefore, the best plan of action is to encapsulate the parts inside of the globe.

The next step of the research process pertains to what material must be used to act as our encapsulating globe. The following table shows possible globe materials and their characteristics. Note that these evaluations are based on known similarly sized samples of the individual materials [8]

Table 5: Different enclosure materials and their characteristics.

Material	Price	Durability	Visibility	Weight
Inj. Molded Plastic	Lowest	Weakest	Low visibility, severe discoloration	Lightest
Polycarbonate	Highest	Strongest	Good visibility, some discoloration	Light
Inj. Molded Acrylic	Medium	Moderate	Good visibility, slight discoloration	Light
Hand Crafted Acrylic	High	Moderate	No discoloration	Light
Glass	High	Strong	Slight discoloration	Heaviest

Because the clock is always put at somewhere close to people and it rotates at a very fast speed, a shelter covering the whole clock is necessary to keep it safe for customers. A good shelter should not decrease the sharpness of the display and protect the customer from the high speed rotating clock. The shape of the shelter should also be considered and it must be suitable for the clock. Size and material are the parameters should be concerned and also the weight. On the other hand, shelter should be suitable to carry and the clock will be easy to remove from and place into it. With these criteria in mind it seem obvious that a spherical, transparent material would be idea. Deciding the ideal material for this task, between acrylic, glass and others can be seen in Table. 20. From this it may be seen that the most important aspects in choosing an enclosure are temperature resistance, transparency, solidity and weight. Temperature resistance of a shelter is seen as moderately more important than solidity transparency and weight (factor of 3). Because the clock will be placed indoor or outdoor, the shelter should be suitable for varying temperatures. Additionally, these factors cannot be changed. While the transparency of the shelter is moderately more important than both solidity and weight, it is mainly used to determine the attractiveness and effectiveness of the device. Since the shelter can be fixed in place, the weight of a shelter is not as important as any others. All of these criteria depend on the shelter material. Main material chooses for shelter can be acrylic sphere, plastic or metal cage. The metal cage is solid but it will disturb sight and heat dissipation. Plastic is available because it light and adiabatic but the cage affects aesthetic outcomes and the ability to see the LEDs.

Motor As you can see in Figure 6 there is a perfect match of gear teeth. This would not have been possible without salvaging these parts from an old printer.

Improper speed will affect stability and safety. So a miniature high quality motor is necessary to keep customers safe. The rotating speed should not be too high or too low. The motor should operate at a stable condition and keep a low temperature. The material should be waterproof and make the motor durable. The miniature motor should also be charged at a low voltage and current to make it safer.

Figure 6: Image showing the stainless steel rotating shaft with attached drive gear being mated with the teeth of the motor.

As you can see from Table **??** for a motor the safety criteria determined in order of importance are speed, heat dissipation and voltage/current. Heat dissipation and voltage/current are deemed to be equally important, while speed is seen as much more important than either of them. A highly rated motor with higher speed will increase the heat generation rate and power consumption. A lower motor may not be appropriate for applications as well as the balance when the device is spinning. Since the operating speed is always slower than the synchronous speed, a motor with around 1000 rpm will be considered to select [9]. The performance of heat dissipation and voltage/current design play the same role as these in LEDs for the clock safety (mentioned before). From [10], one can derive a maximum possible power usage for 72 LEDs to be 30W, Thus, motor selection can be deemed a lesser important decision as much power can be spared and the globe will still be within E.R. 5.

Display Area Optimization One design decision that needed to be accounted for was what should be the vertical height of an image mapped onto the globe clock. It would be ideal to have a larger image height because this would make the image legible from farther away. However, the larger the image, the more it will be distorted along the top and bottom edges. To make referencing simple, we referred to the vertical height as an angle above center. The vertical heights considered were 30°, 45°, and 60° which would make the heights for the globe clock (with 6 inch radius) 6.28, 9.42, and 12.57 inches respectively. While a height of 12.57 inches would be ideal, there would be a lot of distortion on the image at the top and bottom edges. Therefore, the 45° angle (9.42 inches) would display the image at a large enough height while keeping distortion minimal.

Another design decision is the horizontal length of the image. Like vertical height, size and distortion are a factor in optimal horizontal height. Another factor to consider is that the globe clock will be spinning in a vertical direction. The vertical distance is chosen by the angle from the left edge to the right edge of the image and a 90° and 120° angles were compared with lengths of 9.42 and 12.57 inches respectively. The 90° angle was chosen because it would minimize distortion while allowing the image to remain visible to the viewer for a longer period of time. This is the optimal way to fulfill E.R. 3.2. This also makes the programming easier, because the image could be sketched from the x plane to the y plane. Thus, the ideal image mapping size would be 9.42 by 9.42 inches for a globe clock with a radius of 6 inches. The display area is shown in figures 7 and 8.

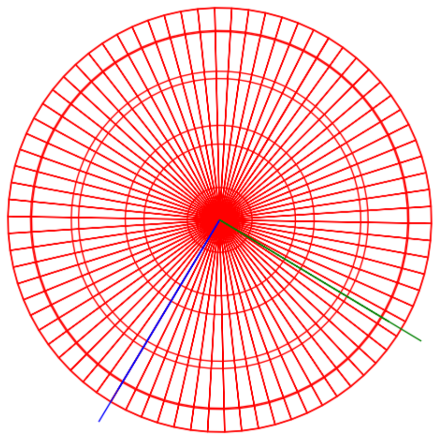


Figure 7: A top-down view of the globe with a 90° angle shown between the green and blue lines

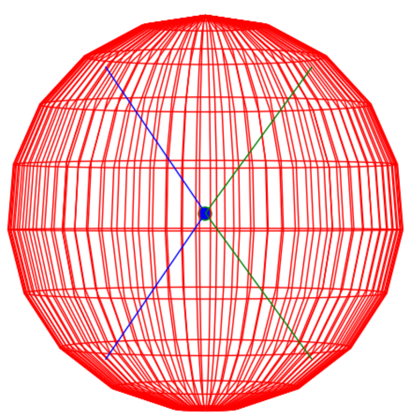


Figure 8: A side view of the globe with a 90° angle shown between the green and blue lines

3.2.3 Level 2 Hardware

Physical Construction Many of the parts necessary to create this device had to be instantiated from imagination. A MakerBot Replicator 5th Generation 3D printer was used to print the parts seen in the following figures. All dimensional annotations on drawings shown of printed parts are in *inches.*

Top Shaft Anchor This part affixes the spring steel bands to the stationary brass shaft. It does this by having an internal diameter which is that of the OD of the ball bearing. The brass tube then fits through the oversized hole in the other end with plenty of clearance.

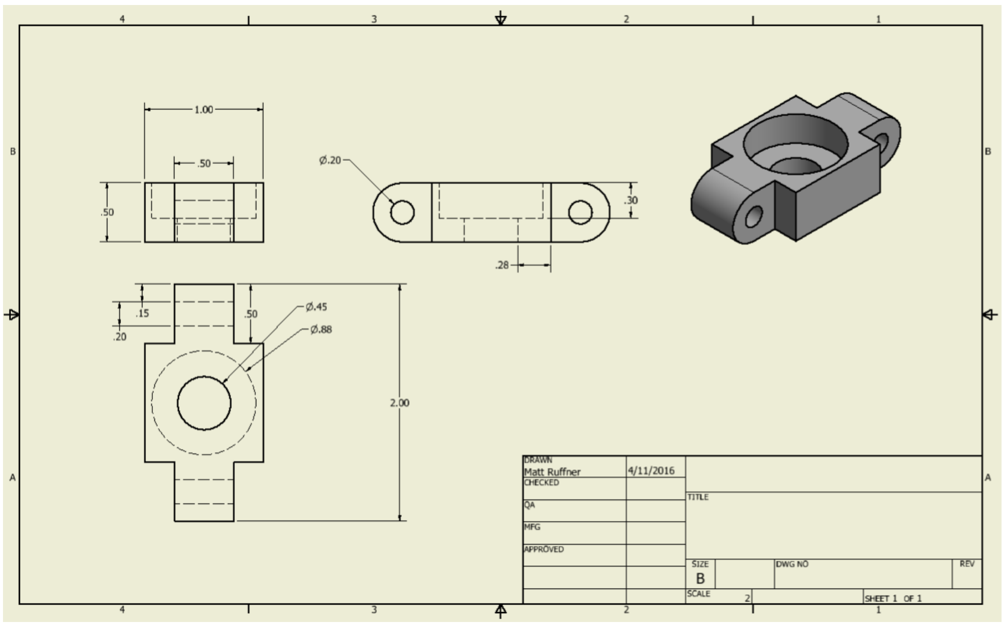


Figure 9: Top Shaft Anchor. Drawing showing top mounting point of stationary shaft to ball bearing to steel rods. Dimensions in inches.

Bottom Shaft Anchor This part attaches firmly to the stainless steel shaft salvaged from the printer. It utilized a stop pin already on the shaft to ensure that this part rotates with the shaft. Holes are in the sides of this piece as well so that the spring steel which forms the band may be attached. It is the bands which synchronize the top and bottom parts, as the top anchor is independent of any moving parts.

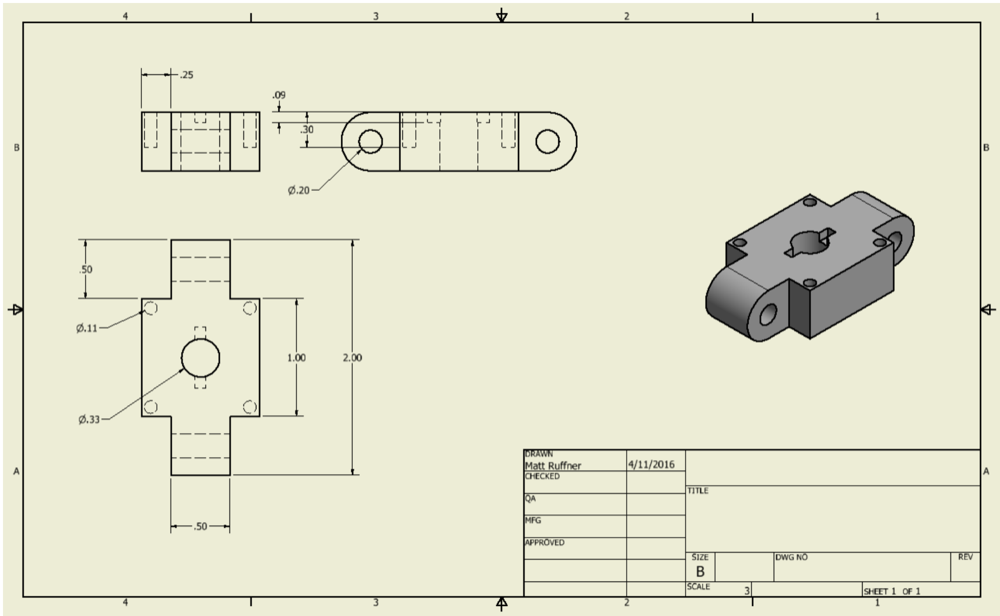


Figure 10: Bottom Shaft Anchor. Drawing showing bottom mounting point that affixes steel rods to driven shaft. Dimensions in inches.

Steel Band Hinge The function of this part is to mate with top and bottom shaft anchor. This creats a hinging system that allows the steel bands to straighten such that they are parallel to the center shaft. This is visualized in Fig. 15. This way, the strip is narrow for insertion into the globe, and is able to be compressed via the concentric center shafts. This compressing action forces the hinges to pivot and steel bands to form a circular shape.

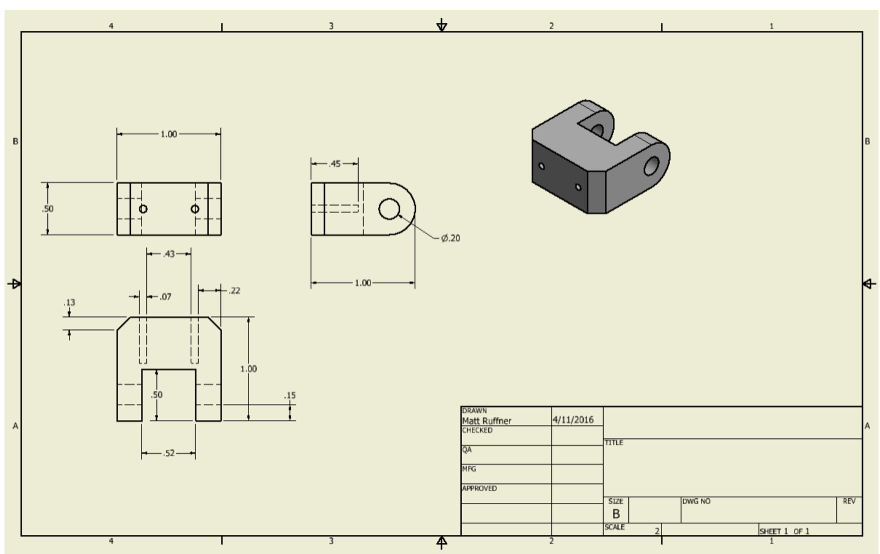


Figure 11: Bottom Shaft Anchor. Drawing showing bottom mounting point that affixes steel rods to driven shaft. Dimensions in inches.

Figure 12 shows the top shaft anchor combined with two steel band hinges. Two hinge ends were also used with the bottom shaft anchor in this manner. The ball bearing assembly can also be seen in Fig. 12. This allows the brass center shaft to stay stationary for its role in transferring power to the rotating ESP8266 controller.

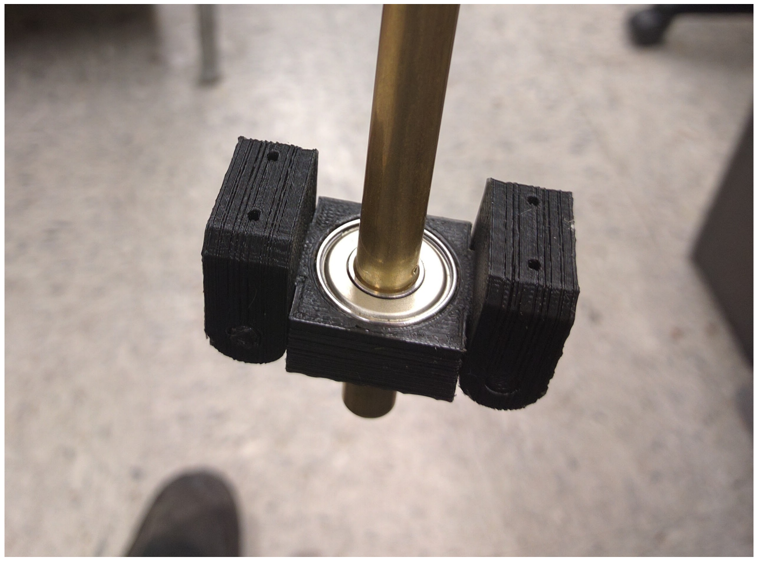


Figure 12: Image showing the top shaft anchor on the stationary brass shaft complete with ball bearing and steel rods affixed.

Stationary Shaft Mount In order for the rotating ring to stay centered within the globe, there must be an anchor point at the top. There was a nipple in the acrylic from where it had been extruded and pinched off. This was taken to be the center of the top and a 1 inch hole was drilled here. The stationary shaft mount then serves as a plug for this hole. Since the ball bearing isolates the brass shaft from the rotating ring, this connection point can be made to be very sturdy. The $\frac{1}{16}$ inch lip at the bottom serves as a standoff so that the ball bearing does not come into contact with the plug.

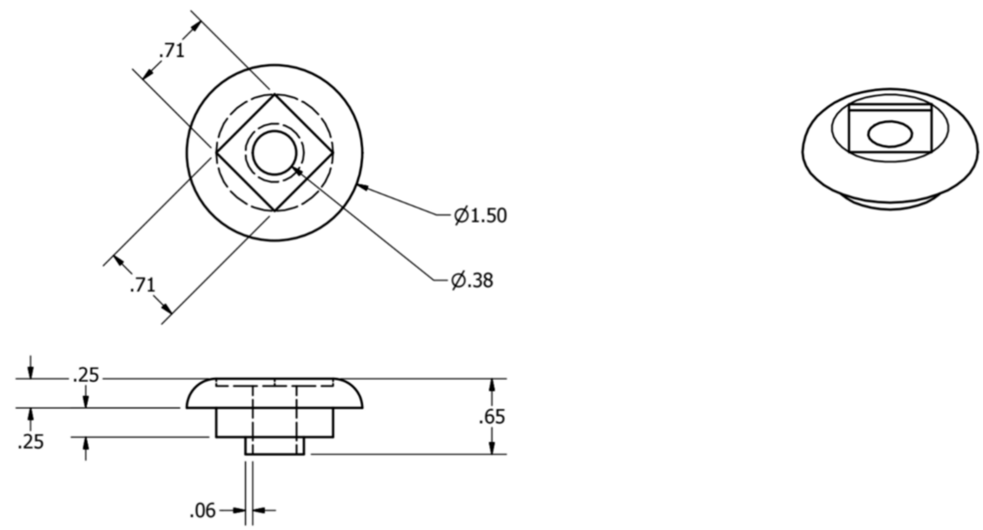


Figure 13: Stationary Shaft Mount. Drawing showing part that centers stationary shaft in top of acrylic globe. Dimensions in inches.

Base Segment The following part, seen in Fig. 14 was created to be a modular solution for positioning the motor, shaft and opening of the acrylic sphere. By breaking the design down like this, it allows for easier repairs as well as an assured success when 3D printing the object. Pieces that are more than 3-4" in diameter have the tendency to start to peel up from the print bed while printing. There are several methods to mitigate this such as heating the build plate on which the part is printed as well as coating the surface with hairspray before printing.

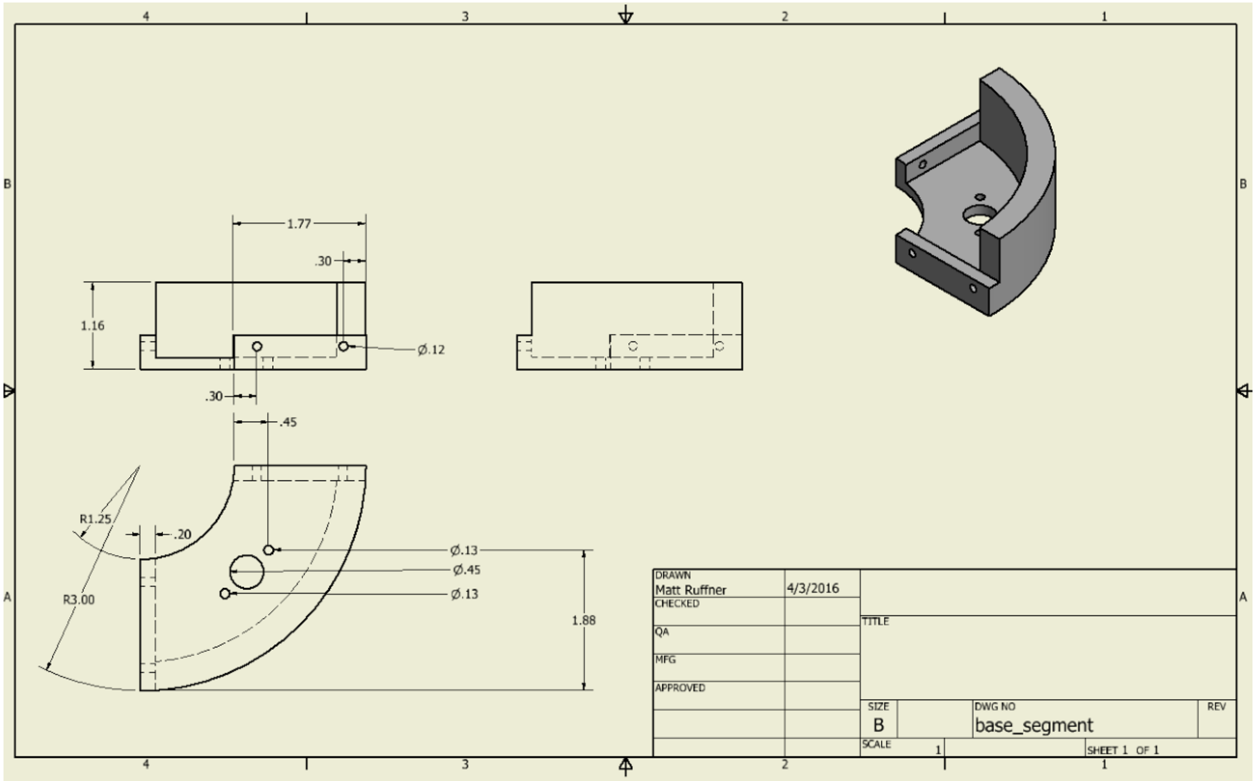


Figure 14: The modular base design.

Seen in Fig. 15, the APA102 LED strip has been affixed to the supportive steel bands with the 3D printed clips holding it in place. The clips secure the strip while still allowing it to shift over the steel skeleton. The base segments have also been assembled and bolted together. Since the plates are identical, the motor may be positioned in any quadrant.

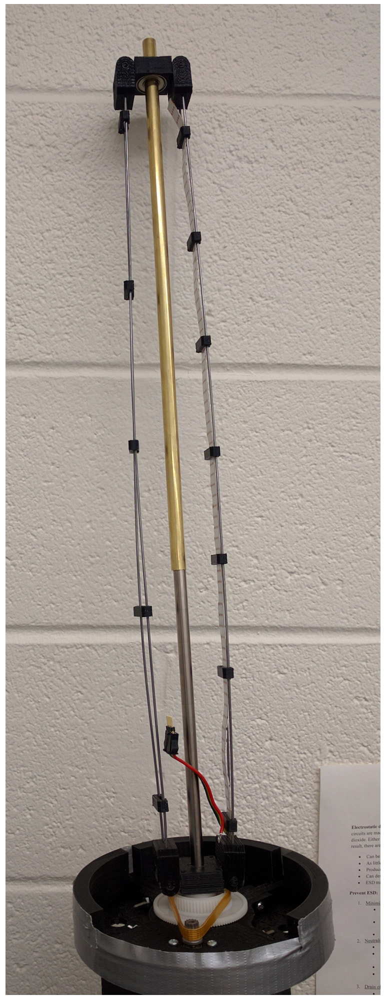


Figure 15: The assembled LED strip and base segments with motor and drive belt attached.

Connectivity Working to achieve goals set under Engineering Requirement 1, the Espressif ™ESP8266 to be the most logical embedded WiFi and processor solution.Tables showing the AHP process for criteria and device selection can be seen in section 6.2. The ESP8266 is a combination WiFi stack and reprogrammable application processor that can run on a variety of firmwares. By default, the device runs an interpreter in the language of Lua, based on the open source NodeMCU firmware.[11]

A wide code base availability for this device (SDK from Espressif ™, Arduino IDE plugin, μPython port, NodeMCU interpreter and more) makes it an effective all-in-one controller solution. Break out boards are available from a variety of retailers. The one used here, shown in Fig. 16, is from Adafruit and offers on board voltage regulation for the 3.3V the ESP8266 requires, as well as level shifting on the TX/RX pins for reprogramming. The ESP8266 is also capable of connecting to 802.11b/g/n networks, satisfying E.R. 1.3 and 1.4.

This configuration of the device is best suited for the project as it has a form factor and mounting holes that are sized to mount on the top shaft anchor part. This sets up the orientation for power commutation from the stationary brass shaft.

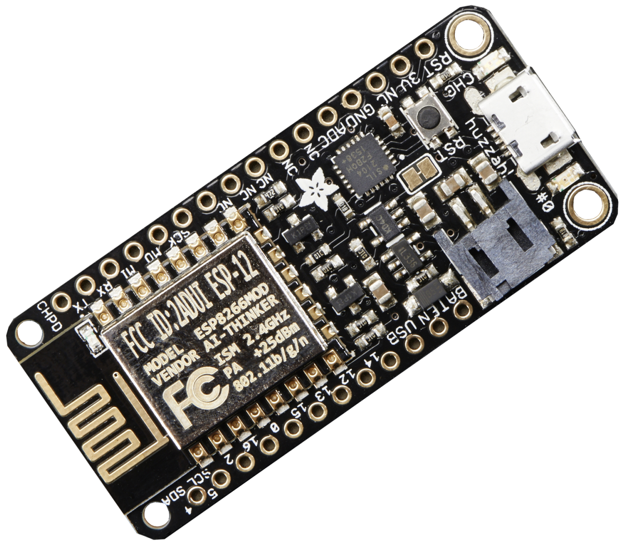


Figure 16: The ESP8266 standalone module from Adafruit.[12]

The Espressif ™ESP8266 WiFi stack was determined to be the best 802.11b/g network interface, the block diagram for this IC can be seen in Fig. 17. This exemplifies various interface components of the IC as well as visualize separation of wireless interface and command execution hardware.

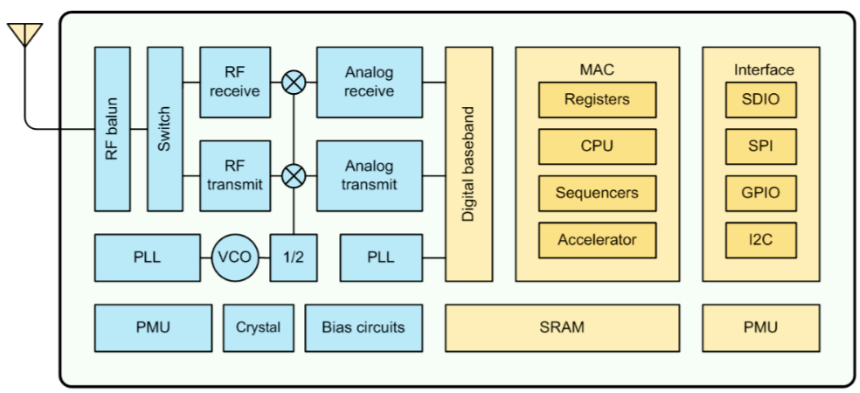


Figure 17: Block diagram for the ESP8266 WiFi stack and combined application processor. [13]

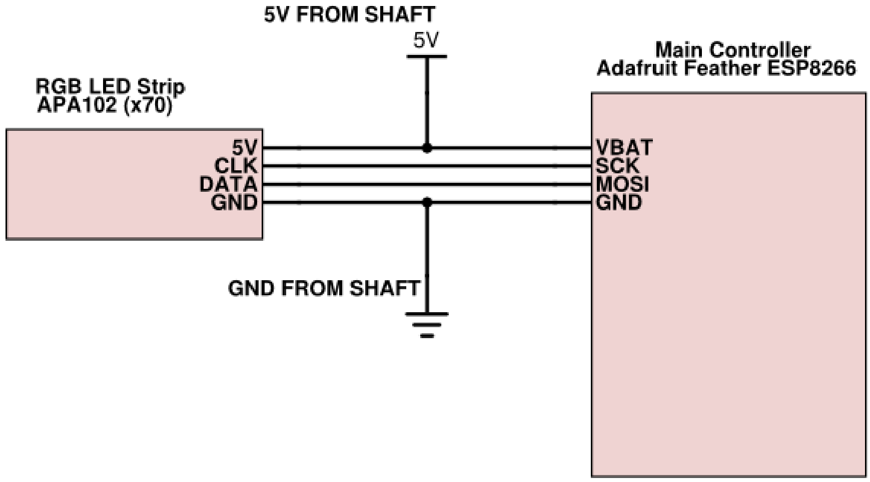


Figure 18: Schematic showing the electrical connections of the rotating circuitry.

As you can see in Fig. 18, both the LEDs and ESP8266 get power from the shaft. The intricacies of this operation are detailed in Fig. 19. Detailed in Fig. 20, a simple 4 pin male header is used to attach the LED strip, interfacing with the strip's stock connection. This positioning allows leeway so that excess stress is not endured by the headers and lead wires when the frame changes shape upon insertion into the acrylic sphere.

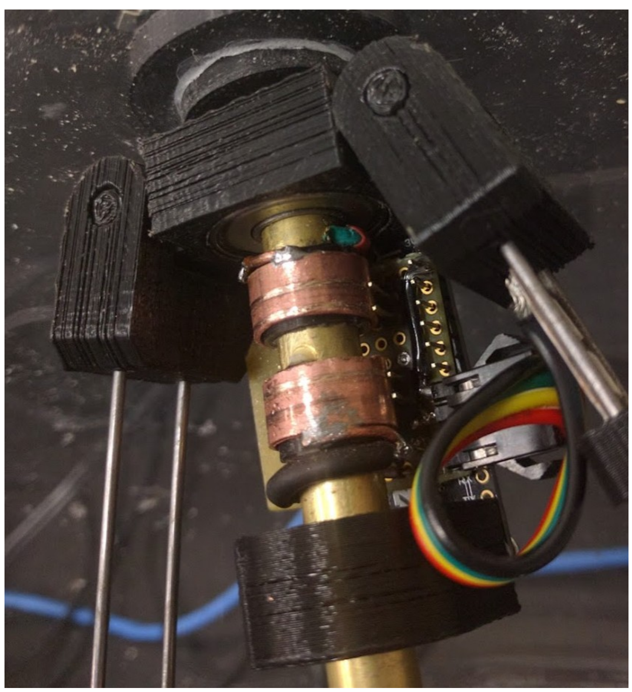


Figure 19: The copper rings used to transfer power from wires inside the brass shaft to the rotating ESP8266 PCB.

In Figs. 19 and 20 you can also see the top shaft anchor printed part.

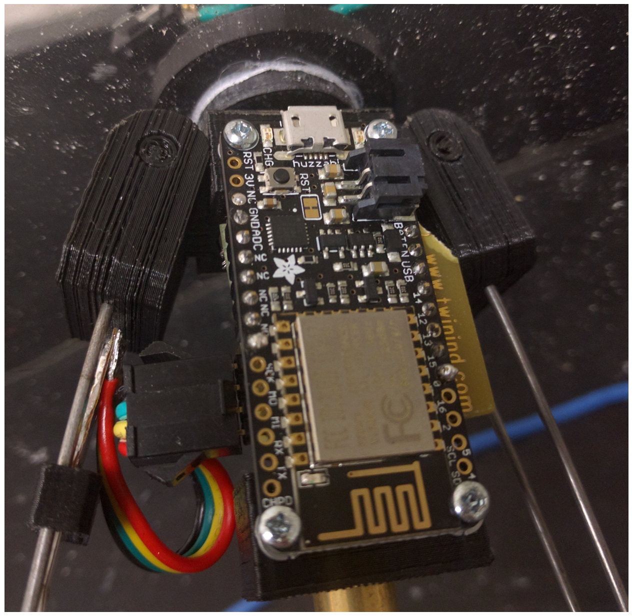


Figure 20: The power transfer and ESP8266 assembly. The 4 pin header connection to the APA102 LED strip is also shown.

Pixels A major selection point for this project was the type of LEDs to use. To start, we had to choose between individual and strip mounted. A disadvantage of using individual LEDs would've been the inherent complexity of addressing them, as well as separating the control wires on the rotating ring. For this reason, strip based LEDs were chosen, specifically, the APA102. It is a daisy-chainable 2 wire SPI interfaced LED that comes in a variety of strips with different densities (LEDs/inch). Using a strip LED adds overhead depending on the number of LEDs.

Other factors that went into the choosing of the LEDs include size, number of reproducible colors, control interface and cost. Again, the AHP selection process was used to compare properties. Using the AHP table, seen in Fig. 24, we were able to find the importance of each property. The weights chosen were based on engineering requirements 2.2, 2.3, 3.1, 3.2, 4.

Figure 21: The .5 meter strip of APA102 SPI RGB LEDs used in the project.

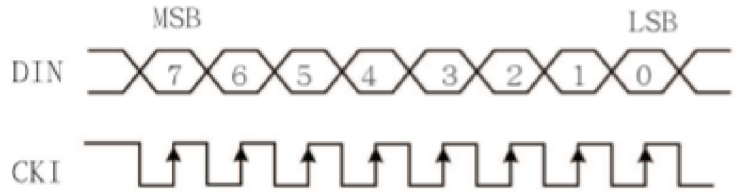
The APA102 is capable of reproducing 8 bit Pulse Width Modulation on all three color channels, yielding 256^3=16,777,216 total possible colors fulfilling E.R. 4. This also reinforces the importance of our most important node on the objective tree: Advertising Effectiveness.

As can be seen from table 19, for LED parameter the criteria determined in order of importance are heat dissipation, voltage/design current, durability and light angle safety. From the last row heat dissipation of a LED is seen as strongly more important than durability (factor of 5), moderately more important than light angle and voltage/current (factor of 3). This is the most important criteria because overheated LEDs can damage the device or cause a fire to break out. A forced cooling system will be needed if the temperature exceeds 60 degrees centigrade [14]. Durability makes the clock usable long term, but it is not necessary. The hazard of using a limited durable bulb can be avoided by changing it according to its lifetime. LEDs are current-driven devices, exceeding the maximum current rating will cause excessive power dissipation which will increase the heat production and then endanger the safety of the device [14]. Most people will not stare at the globe for a long time, so eye safety isn't as important as other considerations.

Control of these LEDs from the ESP8266 is straight forward. There are 5 bits of current control for overall pixel brightness and 8 bits per color of duty cycle variation for PWM control of each red, green and blue diode. An excerpt from the APA102 datasheet[10] can be seen in Figure 22.

DATA MSB↔LSB	Driving Current
00000	0/31
00001	1/31
00010	2/31
…	
11110	30/31
11111	31/31(max)

PWM input and output signals Relations



Data MSB—	Duty Cycle
00000000	0/256(min)
00000001	1/256
00000010	2/256
…	
11111101	253/256
11111110	254/256
11111111	255/256(max)

Figure 22: Excerpt from APA102 Spec Sheet [10] showing SPI timing.

In the Figure 23, the following commands were sent to the ESP8266 controller:

```
apa102.write(0x90, string.rep(string.char(0x50, 0x50, 0x50), 70))
apa102.write(0x90, string.rep(string.char(0x00, 0x00, 0x00), 70))
```

The first argument is the brightness, the rest of each command is repeating the Red,Gree,Blue values of 75/255, 75/255, 75/255 and 0,0,0 , respectively.

You can see the top and bottom traces (SPI clock and MOSI (Master In Slave Out), respectively) are outlined by the cursors. The total change in time for writing the string of pixels 'on' and 'off' is therefor roughly 1.36ms. This is writing 70 LED's worth of data each command. To meet Engineering Requirement 2.1; that the pixels are able to turn on and off within 833μs, 14 vertical pixels is that associated resolution. 14 vertical pixels only requires that 14 sets of red, green and blue values (pixels) are written out. This can be done in about a quarter of the time, around 400μs. This fulfills Engineering Requirement 2.1.

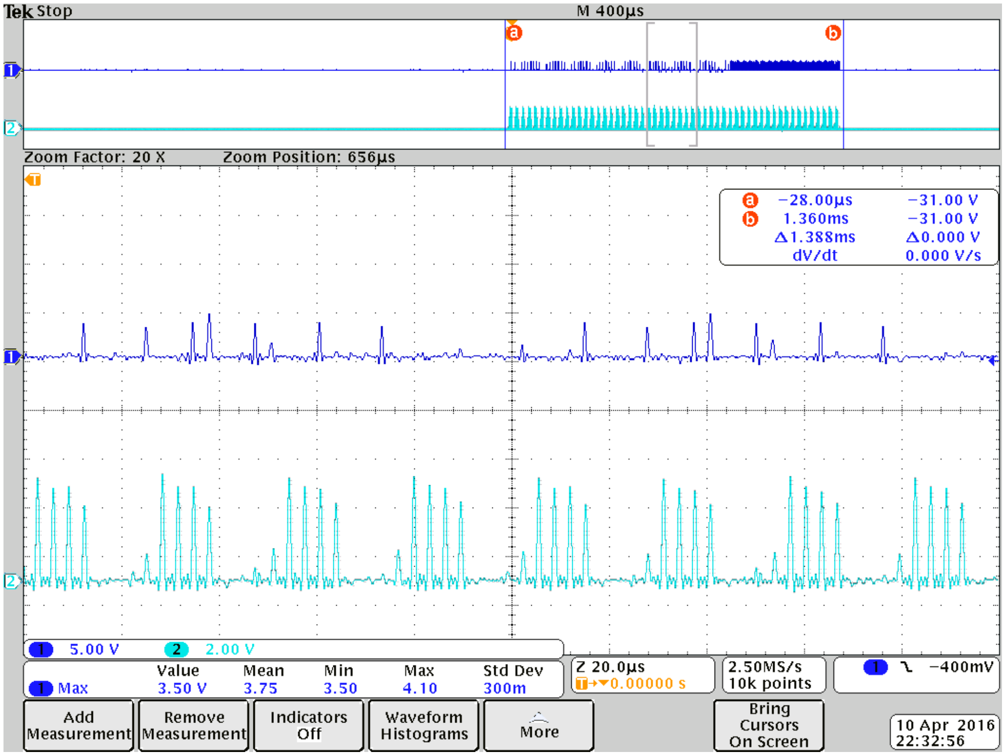


Figure 23: The clock and data lines from a 140 pixel data transmission

3.3 Software Design

In order to make this device as versatile as possible, it is desirable to have a variety of ways to upload media. Our initial necessary requirement for this device was for it to be able to display text and possibly an image.

Looking at it from an optimization standpoint, the easiest execution routine for the on-board ESP8266 would be to simply relay predetermined pixel values to the 70 LED strip. This way, all that is necessary to upload to the globe is a binary file that contains the pixel values concatenated in vertical sections that can be written out to the LED strip.

3.3.1 Level 0 Software

Major functional components of the software side of the Globe Clock are image conversion, image visualization and image uploading. The aforementioned `globegen.cc` program handles the conversion to both binary globe files and structured light point cloud format for web visualization. For gathering the color values from the input image's pixels a free bitmap C++ header library was used.[19]

A point cloud was a logical format for visualization since it in a simple `X, Y, Z, R, G, B` for each point to be displayed. This allows a basic coordinate transformation to be preformed on the input image and be written out to a `.asc` file.

Visualization of the structured light point cloud is done on a web page, dubbed the OrbIT portal. Since point clouds are becoming a very popular format with the advent of 3D laser scanners, there are multitudes of free visualization tools on the web. We chose to use asalga's XB-PointStream library, available on GitHub.[20]

A C++ program to convert a 24bit bitmap file to this format was created. This code can be seen in Section 7.1. As described in the Point Generation Section, 3.3.3. The process which an image goes through to be prepared for the Globe is shown in Fig. ??

3.3.2 Level 1 Software

A TCP connection is used to create the transfer path between the host computer and the globe. This is done on the globe by creating a server like so:

```
s=net.createServer(net.TCP, 20)

saved = true

s:listen(1234, function(c)
   c:on("receive", function(c, l)
      if( string.sub(l, 1, 4) == "data") then
        if(saved) then saved=not saved end

        if (not saved) then
            print("saving new globe image")
            file.open("image", "w")
            file.write(string.sub(l,5,-1))
        end
```

```
      elseif (l and not saved) then
            print("adding to file")
            file.write(l)
      end

   end)
   c:on("disconnection", function(c)
      print("connection complete")

      total = 0
      if file.open("image", "r") then
        repeat
            local line=file.read(210)

            if line then
                apa102.write(0xff,line)
                total = total + string.len(line)
            end
        until not line

        file.close()
      end

   end)
end)
```

As you can see it is very straight forward to create callback functions upon various socket events. To avoid errors spawned from lossy network connections, a continue boolean flag is created so that the read operation may continue to add to the currently downloading file instead of breaking up the files content.

Since the interfacing is as simple as sending the binary .gi globe image file to the device on port 1234, other host applications may written to preform this operation. For example, the windows GUI that can be seen in Fig. 24

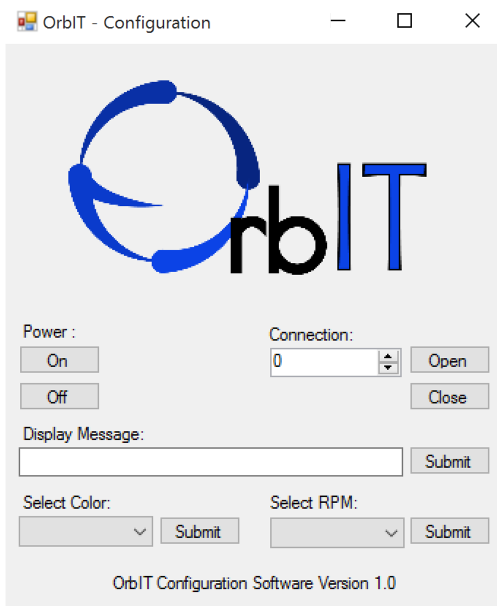


Figure 24: Preliminary Graphical User Interface for control by a Windows machine.

3.3.3 Level 2 Software

Representative Representation In order to perform as little unnecessary processing and calculation as possible, the points are generated to be in an arrangement similar to that of the LED strip. This means generating points from bottom to top, along one horizontal division, or $\Delta\theta$. This vertical meridian of points translates to the LED strip on the Globe Clock. All that must be done to make the two identical is set the number of points to be generated equal to the number of LEDs on the strip, in our case 72. The image shown in Figure 25 visualizes these points.

Point Generation

```
rmat = [[1.,0,0],[0,0,-1.],[0,1.,0]]

def gen_points(w,h,tex_map):
    points = VtkPointCloud()

    for x in range(w):
        percx = ((x+(1./w)) / w)
        lat = 2 * np.pi * percx - np.pi

        for y in range(h):
            percy = ((y+(1./h)) / h)
            lon = np.pi * percy

            p = [np.cos(lat)*np.sin(lon),
                 -1*np.cos(lon),
```

```
             -1* np.sin(lat)*np.sin(lon)]

        new_p = []
        for row in rmat:
            new_p.append(np.dot(row, p))

        c = tex_map.getpixel((percx*tex_map.size[0],tex_map.size[1]-percy*tex_map.size[1]))

        points.addPoint(new_p, c)

return points
```

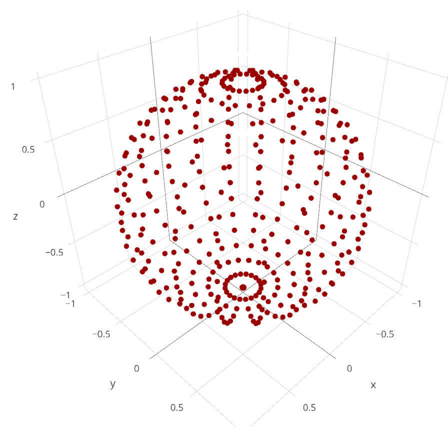


Figure 25: Points generated on a 2D grid and mapped onto the surface of a unit sphere.

Point Cloud Visualization Using the aforementioned WebGL point cloud rendering library, `.asc` files generated by `globegen.cc` can be realized. An example of this web interface displaying some sample text can be seen in Fig. 26.

Figure 26: An example rendering of a generated structured light point cloud, based on an input image.

Adding Color The beauty in approaching the problem this way is this: by starting with a two dimensional grid and mapping that to 3D space, the function is both one-to-one and onto. In other words, for every point that gets mapped to the sphere, there is exactly one *identifiable* point that corresponds to it on the original 2D grid. This creates a texture map. By taking the RGB color value in 2D space and transferring that to the mapped meridian in 3D space, a 144x72 image can be directly overlayed onto the sphere. Higher resolution images may also be mapped by finding ratios of that images width and height to 2:1 ratio of the goal.

Utilizing the Python Imaging Library, this pixel mapping can be done in only a few extra lines of code. Extra steps necessary were to flip the image horizontally and vertically prior to mapping to account for the inverse image that would appear normally. A 90 degree matrix rotation also was needed to account for longitude and latitude differences between the generated and goal image (We require a vertical axis, defaults to horizontal.)

An example of pixels from a simple "Hello World!" image being mapped to the points on the sphere can be seen in Figure 27.

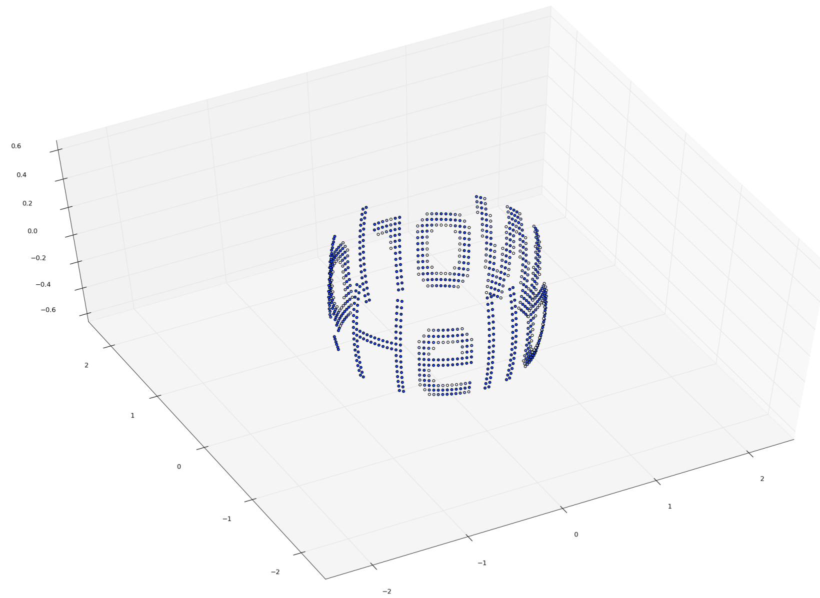


Figure 27: Hello World 144x72 bitmap translated onto the surface points of a unit sphere via the code seen above.

Globe Code For interfacing with the APA102 LED strip, an open source library was used. [21] As stated before there are a multitude of modules available for the NodeMCU Lua firmware. This was no exception, the provided Lua code allowed us to focus on receiving the image data instead of how to display it.

The Lua interpreter on running on the ESP8266 allows for a startup scipt, saved in flash, to be run every time the device is powered on. The startup file that runs on our device can be seen below. As you can see file connects to the school's WiFi, initializes the APA102 driver, and starts the web socket handler listening for incoming connections.

```
wifi.setmode(wifi.STATION)
wifi.sta.config("ukyedu", "")

dofile("apa102.lua")

dofile("ledserver.lua")

apa102.init()
```

4 Project Plan

The progress thus far can be seen in picture documents herein. Further plans are documented below.

4.1 Breakdown Structure

The breakdown structure is sectioned into 3 tables, seen in Tables 6, 7 and 8. The tables are organized by parts acquired, with contingent sections referenced by ID in the far column. In the tables, μController is used for formatting sake.

Table 6: Breakdown Structure Part 1

Id	Activities	Description	Deliverables	Duration (days)	Member	Resources	Pred.
1.1	Order Parts	Order necessary parts	Working parts	10	Jessie		
1.2	Test indiv. parts	Connect to power	Testable connections	3	Jessie, Hao	multimeters	1.1, 2.1, 3.1, 4.1, 5.1
1.3	Testing power	Ensure proper voltages	Programmable system	3	Jessie, Hao	multimeter	1.1, 2.1, 3.1, 4.1, 5.1
2.0	LEDs						
2.1	Schematic for LEDs	Design digital logic for LEDs	Testable schematic for LEDs	6	Matt, Bill	PC	1.0
2.2	Order LEDs	Order LEDs and related parts	Determin LED arrangement	7	Matt		2.1
2.3	Solder LEDs	Solder LEDs to the digital logic	Testable schematic of LED logic	5	Matt, Bill	PC	1.0
2.4	Test LEDs	Test LEDs with μController	Working LEDs	2	David, Hao	multimeter	2.2
2.5	Attach LEDs	Attach LEDs to Ring	Ring ready to spin	4	David	Solder Iron	2.3
3.0	Motor				Bill		
3.1	Order Motor	Research, order, and receive motor	Acquire physical motor	7	Matt	PC	
3.2	Connect motor to LED ring	Attach motor to aparatus	A ring that can spin	3	Hao	wiring, soldering iron	1.0, 2.4
3.3	Test motor	Test motor itself and connected devices	Working motor and spinning LED ring	2	Bill, Hao	Multimeter	1.0, 2.4, 3.2
4.0	Microcontroller						
4.1	Order Microcontroller	Order and receive μController	Acquire μController	10	Matt	PC	
4.2	Program the Microcontroller	Connect μController to parts	Have a testable microcontroller	3	Matt, Ben	Wiring, soldering iron	1.0, 4.2
4.3	Test Microcontroller	Test basic programs	Have a working μController	3	Matt, Ben	PC	4.2

Table 7: Breakdown Structure Part 2

Id	Activities	Description	Deliverables	Duration (days)	People Responsible	Resources	Pred.
5.0	Wifi Module						
5.1	Order Wifi Module	Research, order, and recieve wifi module	Acquire wifi module	10	Matt	PC	
5.2	Connect and Test Wifi Module	Connect the wifi module with microcontroller to test	Have a working wifi module	3	Hao	Wiring, soldering iron	1.0, 4.3, 5.1
6.0	Base and Enclosure				Bill		
6.1	Order plastic base and enclosure	Research, order, and receive base and enclosure	Have a base and enclosure	13	Matt, Bill	PC	
6.2	Test base and enclosure	Perform safety tests such as breakage or waterproofing	Have a safe enclosure for the original parts	5	Bill, Hao	Blunt object water	6.1
6.3	Enclosing individual parts	Combine all parts to create the globe clock	Have entire globe clock intact	5	Bill, Hao	Wiring	1.1, 2.1, 3.1, 4.1, 5.1, 6.2
7.0	Computer Interface						
7.1	Design computer interface	Design program that allows user to change text and images on globe clock	Have interface to change text and images on LEDs	25	Ben, Matt	PC	
7.2	Testing Computer Interface	Find and correct bugs in program	Have a stable program	15	Ben, Matt	PC	7.1
7.3	Manual for program	Write a manual for using program	Have a complete user manual	16	Ben, Hao	Word	7.2

Table 8: Breakdown Structure Part 3

Id	Activities	Description	Deliverables	Duration (days)	People Responsible	Resources	Pred.
8.0	EE 490 Tasks						
8.1	Reports						
8.1.1	CDR Report	Write CDR report	Complete assignment	9	All	Word	
8.1.2	Final Design report	Write Final Design report	Complete assignment	9	All	Word	
8.2	Presentations						
8.2.1	CDR Presentation	Present CDR findings	Completed Assignment	3	All	Word and Powerpoint	
8.2.2	PDR Presentation	Present PDR findings	Complete assignment	3	All	Word and Powerpoint	
8.2.3	Status Update Presentation	Present Status Updates	Complete assignment	3	All	Word and Powerpoint	
8.2.4	Final Design Presentation	Present Final Design	Complete assignment	3	All	Word and Powerpoint	
8.2.5	Fall Poster/ Senior Design Day	Make poster and present it	Complete assignment	6	All	Word and Powerpoint, Posterboard	
8.2.6	Spring Poster/Senior Design Day	Make poster and present it	Complete assignment	6	All	Word and Powerpoint, Posterboard	

4.2 Gantt Chart

Shown in Fig. 28 is the projected time line of progress. Each cell contains one day's worth of progress. This is tentative as no dates for class future class reports or presentations are known to us at this time.

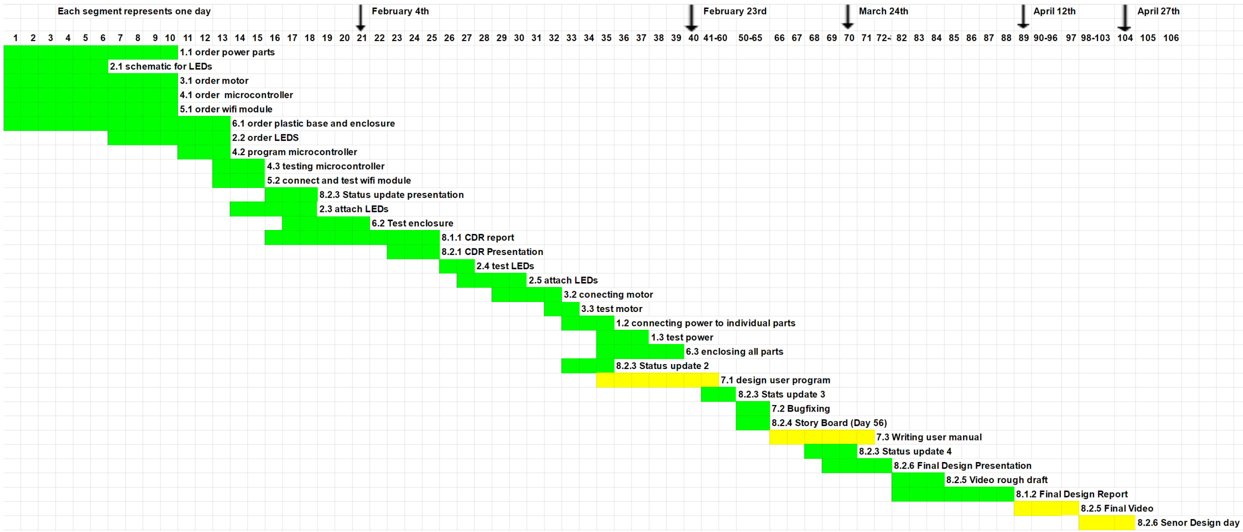


Figure 28: Projected time line where each cell represents 1 day.

4.3 Costs

The table below is the final cost of the project itemized.

Part	Description	Supplier	Amount	Cost
ESP8266	WiFi Stack/Processor	Adafruit [15]	1	$10.95
Clear Globe	12 inch Clear Acrylic Globe	Amazon	1	$41.51
Individually Addressable LED RGB Strip	.5 meter, 144 LEDs/meter	Adafruit [16]	1	$19.95
Brass Tube, Type K	20 in. long, $\frac{3}{4}$ in. width	Hobby King	1	$3.96
24 V Power Supply	DC 24 V @ 2A regulated power supply	Amazon	1	$9.99
Universal 5V 1.2A AC Power Supply	5V power supply	Amazon	1	$25.00
MAKERBOT REPLICATOR 2 BLACK PLA FILAMENT	1.75mm	Quickship	1	$ 16.72
Total Cost				$144.80

Table 9: Cost of parts

4.4 Team Responsibilities

This project was broken down into six major parts: power, LED's, motor, wifi and microcontroller chip, the outer enclosure and the user program. For each of these sections we had one primary person for each section with multiple secondaries. For power the main person responsible was Jessie. This was due to his main focus being power and he will have help from Hao. For the LED section, there was three people working on this section due to its importance for the overall project. Matt, Bill and David focused on this section. Hao was primarily working on the motor with Bill and Matt as secondaries. For the wifi and microcontroller, Matt was the main person responsible, due to his vast knowledge of electronics. Bill mainly focused on the enclosure with Hao helping as a secondary. Lastly Ben was the main person responsible for writing the user program with Matt helping. This will take the most time by a decent amount. Thus the majority will lend a hand with the programming at one point during the process. The group reports and presentations were worked on by everyone.

5 References

[1] Digital Signage Federation, How the Internet of Things is Reinventing Retail. June, 2015 http://www.digitalsignagefederation.org/Resources/Documents/A_O%20Guides/How%20IoT%20is%20Reinventing%20Retail.pdf

[2] SpaceWriter iBall ™ http://www.ishow360.com/products/pict/iball_class_3.php

[3] Digital Signage Federation, Primary Benefits of Digital Signage/Menu Boards for Restaurants and QSR Locations http://www.digitalsignagefederation.org/Resources/Documents/Articles%20and%20Whitepapers/Primary_Benefits_DSMenuBoards.pdf

[4] SpaceWriter iBall ™ http://www.ebay.com/itm/iBall-LED-Digital-Signage-Shop-Pub-Club-Spinning-Scrolling-360-Degree-POV-POS.

[5] SpaceWriter iBall ™Manual http://www.zaj.hu/pdf/sde_help.pdf

[6] Sean Kelley, Buffalo Wild Wings. Personal Interview. Lexington, KY. 15 September 2015.

[7] Ford and Coulston. *Design for Electrical and Computer Engineers*. McGraw-Hill Higher Education. 2008

[8] "The Many Forms of Clear Plastic." Displays 2 Go. N.p., 20 Mar. 2015. Web. 03 Nov. 2015. http://www.displays2go.com/Article/The-Many-Forms-Clear-Plastic-5.

[9] Keith Forsman, Motor Efficiency Selection and Management http://www.pge.com/includes/docs/pdfs/mybusiness/energysavingsrebates/incentivesbyindustry/a Accessed on Web. 02 Nov. 2015

[10] RGB Full Color LED Control IC https://www.adafruit.com/datasheets/APA102.pdf

[11] "Nodemcu/nodemcu-firmware." GitHub. Web. 11 Apr. 2016. https://github.com/nodemcu/nodemcu-firmware.

[12] "ESP8266 SMT Module." Adafruit Industries Blog RSS. Accessed November 16, 2015. http://www.adafruit.com/products/2491.

[13] "ESP8266EX Specification Sheet." Accessed December 2, 2015. https://www.adafruit.com/images/product-files/2471/0A-ESP8266__Datasheet__EN_v4.3.pdf.

[14] LED Application Notes - LED Basics http://www.theledlight.com/technical1.html Web. 01 Nov. 2015

[15] "Adafruit HUZZAH ESP8266 Breakout." Adafruit Industries Blog RSS. Accessed November 16, 2015. http://www.adafruit.com/products/2471.

[16] "Adafruit Industries, Unique & Fun DIY Electronics and Kits." Adafruit Industries Blog RSS. Accessed November 16, 2015. http://www.adafruit.com.

[17] "LED - SMD RGB (WS2812)." - COM-11821. Accessed November 16, 2015. https://www.sparkfun.com/products/11821.

[18] "LED RGB Strip - Addressable, Bare (1m)." - COM-12025. Accessed November 17, 2015. https://www.sparkfun.com/products/12025.

[19] "Arash Partow - 2002" http://partow.net/programming/bitmap/index.html

[20] "asalga/XB-PointStream" https://github.com/asalga/XB-PointStream

[21] "MisterRager/Nodestar" https://github.com/MisterRager/Nodestar

6 Appendix A

6.1 Marketing Requirement Weight Calculation

Table 10 shows how the weights for the main leaves of the objective tree were found using the Analytical Hierarchy Process defined in *Design for Electrical and Computer Engineers*[7]

	Ad. Eff.	E.O.U.	Efficiency	Cost	Safety	Geo. Mean	**Weights**
Add. Eff.	1	3	5	3	1	2.141	0.357
E.O.U.	$\frac{1}{3}$	1	3	1	1	0.999	0.167
Efficiency	$\frac{1}{5}$	$\frac{1}{3}$	1	$\frac{1}{3}$	$\frac{1}{5}$	0.338	0.564
Cost	$\frac{1}{3}$	1	3	1	$\frac{1}{3}$	0.802	0.134
Safety	1	1	5	3	1	1.719	0.286

Table 10: The AHP process outlined for the main leaves of the objective tree seen in Fig. 2

Table 11 shows how the weights were found for the Advertising Effectiveness category of the objective tree.

	Legibility	Attractiveness	Geo. Mean	**Weight**
Legibility	1	1	1	.5
Attractiveness	1	1	1	.5

Table 11: Geometric weights for advertising

Table 12 shows how the weights were found for the Ease of Use category of the objective tree.

	Installation	Content Management	Customizability	Geo. Mean	**Weight**
Installation	1	1	1	1	0.333
Content Management	1	1	1	1	0.333
Customizability	1	1	1	1	0.333

Table 12: Geometric weights for ease of use

Table 13 shows how the weights were found for the Cost category of the objective tree.

	Initial	Upkeep	Geometric Mean	**Weight**
Initial	1	1	1	0.5
Upkeep	1	1	1	0.5

Table 13: Geometric weights for cost

6.2 WiFi/Processor Criteria and Module Selection

	Power	Size	Interfaces	Reprogramability	Geo. Mean	**Weight**
Power	1	1	$\frac{1}{3}$	$\frac{1}{3}$	0.577	0.161
Size	3	1	1	$\frac{1}{5}$	0.880	0.246
Interfaces	$\frac{1}{5}$	$\frac{1}{3}$	1	$\frac{1}{3}$	0.386	0.108
Reprogrammability	3	3	1	1	1.732	0.484

Table 14: The AHP criteria weight selection process for values regarded when choosing the best WiFi module for the project.

	CC3000	ESP8266	RN-131G	Geo. Mean	**Weight**
CC3000	1	3	3	2.080	0.510
ESP8266	$\frac{1}{3}$	1	3	1.000	0.245
RN-131G	3	$\frac{1}{3}$	1	1.000	0.245

Table 15: The AHP process for Power Consumption.

	CC3000	ESP8266	RN-131G	Geo. Mean	**Weight**
CC3000	1	1	$\frac{1}{3}$	0.693	0.221
ESP8266	1	1	3	1.442	0.460
RN-131G	3	$\frac{1}{3}$	1	1.000	0.319

Table 16: The AHP process for Physical Dimensions.

	CC3000	ESP8266	RN-131G	Geo. Mean	**Weight**
CC3000	1	$\frac{1}{3}$	$\frac{1}{3}$	0.480	0.143
ESP8266	3	1	1	1.442	0.429
RN-131G	3	1	1	1.246	0.429

Table 17: The AHP process for peripheral interfaces.

	CC3000	ESP8266	RN-131G	Geo. Mean	**Weight**
CC3000	1	$\frac{1}{5}$	$\frac{1}{3}$	0.405	0.094
ESP8266	5	1	3	2.467	0.572
RN-131G	3	1	1	1.442	0.334

Table 18: The AHP process for reprogrammability.

6.3 Safety

Table 19: LED Safety Parameters AHP

	Durability	I/V Traits	Light Angle	Heat Diss.	Geo. Mean	**Weight**
Durability	1	$\frac{1}{3}$	3	$\frac{1}{5}$	0.67	0.13
I/V Traits	3	1	3	$\frac{1}{3}$	1.32	0.26
Light Angle	$\frac{1}{3}$	$\frac{1}{3}$	1	$\frac{1}{3}$	0.44	0.09
Heat Diss.	5	3	3	1	2.59	0.52

Table 20: Shelter AHP

	Solidity	Transparency	Temp Resistance	Weight	Geo. Mean	**Weight**
Solidity	1	$\frac{1}{3}$	$\frac{1}{3}$	3	0.76	0.16
Transparency	3	1	$\frac{1}{3}$	3	1.32	0.27
Temp Resistance	3	3	1	3	2.28	0.48
Weight	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	1	0.44	0.09

Table 21: Motor Selection AHP

	Speed	Heat Dissipation	Voltage/Current	Geo. Mean	**Weight**
Speed	1	5	5	2.92	0.72
Heat Dissipation	$\frac{1}{5}$	1	1	0.58	0.14
Voltage/Current	$\frac{1}{5}$	1	1	0.58	0.14

distributable

6.4 Image Mapping

The AHP for the vertical angles is shown in table 22. The weights showed that the 45° angle got a weight 0.8436 which was the highest score.

Table 22: The AHP process for the vertical angle from the center

	30°	45°	60°	**Weight**
30°	1	$\frac{1}{5}$	3	0.2023
45°	5	1	5	0.7004
60°	$\frac{1}{3}$	$\frac{1}{5}$	1	0.0973

The AHP for the horizontal angles is shown in table 23. The 90° angle garnered the largest weight here at 0.8333.

Table 23: The AHP process for the horizontal angle from the left edge

	90°	120°	**Weight**
90°	1	5	0.8333
120°	$\frac{1}{5}$	1	0.1667

6.5 LEDs

Table 24: AHP process for LED criteria.

	Size	Ease of Use	# Colors	Cost	Geo. Mean	**Weight**
Size	1	9	3	6	2.77	0.52
Ease of Use	$\frac{1}{9}$	1	$\frac{1}{6}$	$\frac{1}{6}$	0.315	0.06
# Colors	$\frac{1}{3}$	6	1	3	1.43	0.27
Cost	$\frac{1}{6}$	6	$\frac{1}{3}$	1	0.80	0.15

6.6 Power

Table 25: The AHP process for motor selection

	Speed	Heat Diss.	Op. Point	Geo. Mean	**Weight**
Speed	1	5	5	2.92	0.72
Heat Diss.	$\frac{1}{5}$	1	1	0.58	0.14
Op. Point	$\frac{1}{5}$	1	1	0.58	0.14

7 Appendix B

7.1 `globegen.cc`

```
/* Matt Ruffner
 * March 2 2016
 * C++ implementation of img2globe
 *
```

```
 * Thanks to http://www.partow.net/programming/bitmap/index.html
 * for the C++ Bitmap Library
 */

#include <iostream>
#include <fstream>
#include <string>
#include <cstdlib>
#include <cmath>

#include "bitmap_image.hpp"

#define PI 3.1415926535897

#define ASC 7
#define GI  8

using namespace std;

void usage(char *sig);

void outputASC(bitmap_image image, ostream *out );
void outputGI(bitmap_image image, ostream *out );

void printAsHex(unsigned char *data, size_t len);

int width = 0, height = 0;
int globeWidth = 144, globeHeight = 70;

int main(int argc, char * argv[])
{
    if(argc < 3) {
        usage(argv[0]);
    }

    int task = 0;
    if( argv[2][0] == 'a' ) {
        task = ASC;
    } else if( argv[2][0] == 'g' ) {
        task = GI;
    } else {
        usage(argv[0]);
    }

    bool toFile = false;
```

```
    ofstream *fout;

    if (argc == 4) {
        toFile = true;
        fout = new ofstream(argv[3]);

        if(!fout->is_open()) {
            cout << "[ERROR] - Could not open " << string(argv[3]) << " for writing." << endl;
            exit(1);
        }
    } else if(argc > 4) {
        usage(argv[0]);
    }


    string file_name(argv[1]);

    bitmap_image image;

    if( file_name.compare("-") == 0 )
        image = bitmap_image(cin);
    else
        image = bitmap_image(file_name);


    if (!image)    {
        cout << "[ERROR] - Failed to open " << file_name << endl;
        exit(1);
    }

    width = image.width();
    height = image.height();



    // TODO: support input image downscaling
    if (width != globeWidth || height != globeHeight) {
        cout << "Sorry, input image must be " << globeWidth << "x" << globeHeight << "." << enc
        exit(1);
    }


    switch(task) {
        case ASC:
            if( toFile ) {
```

```
                cout << "- Generating point cloud for visualization..." << endl;
                outputASC(image, fout);
                fout->close();
            } else {
                outputASC(image, &cout);
            }
            break;
        case GI:
            if( toFile) {
                cout << "- Generating file for GlobeClock" << endl;
                cout << "-- filesize: " << width*height*3 << " bytes" << endl;
                outputGI(image, fout);
                fout->close();
            } else {
                outputGI(image, &cout);
            }
            break;
        default:
            exit(1);
    }


    return 0;
}

void outputASC(bitmap_image image, ostream *out )
{
    unsigned char r, g, b;

    for (long x = 0; x < width; x++) {

        double dx = ((x+((double)1/width)) / width);
        double lat = 2 * PI * dx - PI;

        for (long y = 0; y < height; y++) {

            double dy = ((y+((double)1/height)) / height);
            double lon = PI * dy;
            double xc,yc,zc;

            image.get_pixel( (int)(dx*width), height - (int)(dy*height), r, g, b );

            xc = 7*cos(lat)*sin(lon);
            yc = -7*cos(lon);
            zc = -7* sin(lat)*sin(lon);
```

```
            (*out) << xc << "\t"
                << yc << "\t"
                << zc << "\t"
                << (int)r << "\t"
                << (int)g << "\t"
                << (int)b << "\n";

        }
    }

}

void outputGI(bitmap_image image, ostream *out )
{
    unsigned char r, g, b;

    for (long x = 0; x < width; x++) {

        double dx = ((x+((double)1/width)) / width);

        for (long y = 0; y < height; y++) {

            double dy = ((y+((double)1/height)) / height);

            image.get_pixel( (int)(dx*width), height - (int)(dy*height), r, g, b );

            (*out) << r << g << b;
        }
    }
}

void usage(char *exe)
{
    cout << "Globe Convert" << endl;
    cout << "  Utility for converting 24 bit (no alpha) bitmaps for GlobeClocks." << endl << en
    cout << "  Usage:" << endl;
    cout << "  " << string(exe) << " <input.bmp> [a|g] [outputfile]" << endl;
    cout << "    out.asc is the structured light point cloud for visualization in the OrbIT Poi
    cout << "            thanks to the XB-PointStream library." << endl;
    cout << "    out.gi is the globe image of arranged RGB values from the input bitmap." << en
    cout << "    - can be specified for the image input in which case it will be read from stdi
    exit(1);
}
```