Department of Electrical & Computer Engineering
University of Central Florida

LED Globe: Persistence of Vision

Group #8

Viet Nguyen
Rhonal Soto
Omar Miguel Vazquez
8/1/13
Senior Design I
EEL 4914

Table of Contents

Appendices

Appendix A: Works Cited

[1]”RGB Globe.” Solderlab.com. Web.
<http://www.solderlab.de/index.php/led-projects/rgb-globe.>

[2]”LED Globe.” Leyanda. Web.
<http://leyanda.de/light/povglobe.php>

[3]”LED: 101 Identifying Different Types Of LEDs.” Electronic Products. Web.
<http://www.electronicproducts.com/Optoelectronics/LEDs/LED_101_Identifying_different_types_of_LEDs.aspx>

[4] “Home Page.” CadSoft. Web.
<http://www.cadsoftusa.com>

[5] “IPC-2221A.” IPC. Web.
<http://www.ipc.org/toc/ipc-2221a.pdf>

[6] “Cost-Pricing Strategy.” Yahoo Small Business. Web.
<http://smallbusiness.yahoo.com/advisor/cost-plus-pricing-strategy-120017163.html>

Appendix B: Figures

Appendix C: Tables

1 Executive Summary

Persistence of vision is the phenomenon of the eye by which an image is thought to persist for approximately 1/25 of a second on the retina after the image is removed. Many classic and modern film use the theory that the human perception of motion, which is brain centered, is the direct result of persistence of vision.

Although this theory was later disproven by Max Wertheimer in 1912*, classic and modern film still use this theory today. A more credible theory to explain the perception of motion is two distinct perceptual illusions, which are phi phenomenon and beta movement. The phi phenomenon is the perception of a continuous motion between separate objects viewed very fast in succession such as a series of LEDs blinking on and off in succession to give the illusion that something is covering the LEDs and is moving in front of them, were as the beta movement is similar to the phi phenomenon in which both cause a sensation of motion but the beta movement is different because it give the optical illusion that the light itself is moving although the lights themselves are stationary.

An example of phi phenomenon will be like having 10 lamps all lined up and turned on. If you turn off the first lamp and turn it back on but at the same time you turn it back on and you turn off the second lamp and at the same time you turn on the second lamp you turn off the third lamp and so forth, the illusion will be as if something or somebody was walking in front of each lamp and giving you the perception of movement. An example of beta movement will be like having all 10 lamps turned off and having the first lamp turn on and at the same time you turn off the first lamp you turn on the second lamp, at the same time you turn off the second lamp you turn on the third lamp and so forth thus giving you the illusion that the light itself is moving in a straight line.

The goal of this project is to design a persistence of vision spinning LED Globe, which uses a combination of phi phenomenon and beta movement. The LEDs will be mounted on a ring like structure, which will be spun at a certain angular velocity. The series of LEDs will be controlled by a microcontroller, which will be programmed to individually turn on and off the LEDs at a certain time in space to give the illusion of an image, which seems to be suspended in midair in the shape of a globe. The LED globe will be interfaced with Bluetooth technology so that a user can input a text or image and have it displayed on the LED globe.

2 Project Description

2.1 Motivation and Goals

The LED globe is a display that from the eyes perception looks like a solid sphere of light when rotating. Nowadays, the people are bombarded with information constantly and the most of the time the best way to deliver this information is visually. There are lots of different types of information needed for people on the daily basics but when people go out to the street and get on the road they rely on signs to provide the directions and at the same time they see banners with information everywhere they go.

On the same token, the way the information is displayed determines the amount of attention is drawn from people; therefore, design plays a very important role in information delivery. Regular banners, LED signs, and TV's display information in one plane and only one direction and depending the type of displayer the can be very expensive.

The idea of the LED Globe is to build a device that is able to display information in three planes and depending of the design the resolution can be very high. Just imagine driving on the highway or any dark road with poor visibility conditions and one sees this luminous sphere displaying direction or advertising businesses or products; there is no way that someone can miss this. When the application can be apply to deliver information that can be see horizontally from two opposite sides on the horizontal plane and from the top of the vertical plane, so if we retake the example of the highway exposed above, people on the same highway or on any horizontal plane will be able to see the information displayed by the same device when the travel in both direction of the highway.

The LED Globe is able to display any type of information, such as written messages and images, and the information sequence the information delivered can be changed easily. Moreover, besides the advantage of displaying information in two opposite directions, the device's power consumption is very low and on the same way the production costs are very low also. Also, using this device for commercial purposes could set a new way to deliver advertisements because the LED Globe has the capability of implementing any kind of designs that includes written information and images and combining them in any sequence; therefore, our goal is to build an small version of this devise capable of perform the functions mentioned above.

2.2 Objectives

The primary objective is to have a small-scaled, energy efficient, environmental friendly, low maintenance and user friendly LED Globe. All elements within the system, such as the microcontrollers, shift registers, and LED's, should consume the lowest amount of power as possible.

2.2.1 Scalability

The LED Globe will be within an 8" x 8" x 3' space, and will operate to be as quiet as possible.

2.2.2 Efficiency

LED global will be designed to use the lease amount of energy as possible. This can be accomplished by having a smaller motor which will operate at about 3.5 V, and having the LED ring structure and support shaft as light as possible will help reduce the amount of torque on the motor thus consuming less energy. The microprocessors and the LEDs will be powered by a separate power supply, which will consume no more than a few milliwatts of power.

2.2.3 Environmentally Friendly

The LED Globe does not use any chemicals related to batteries or chemicals in general and should therefore not produce any harmful chemical reactions which could be harmful to humans or animals; since the motor that we are using is brushless, and the power supply will be shielded and grounded, there should be little to no electromagnetic radiation which could be harmful to humans or animals; the LED globe is environmentally friendly and safe.

2.2.4 Low Maintenance and User Friendly

The LED globe is lightweight portable and can be used anywhere there is a 120 V outlet. The LED globe will have a simple on off switch which will power up the microcontroller and the LEDs, and a separate switch to turn on the motor. The Bluetooth user interface will be very easy to use, and there will be a blue LED mounted on the housing to indicate Bluetooth connection.

2.2.5 Input 1: Power switch

The power switch will be the main switch that will turn on both the power supply to the motor and the power supply to the microcontrollers. A green LED will indicate that the motor is getting power and a red light will indicate the LED's on the ring are getting power. When the microcontrollers receive the power, the ring will rotate 360 degrees and each light will turn one is a random sequence to indicate these two components (ring and LED's) are working.

The power will be coming from the outlet on the wall delivering 120 Volts to our power supply unit, which is going to eliminate spikes and surges common in most electrical systems. From our power supply it will go to the power regulator and AC/DC converter which is going to pulls the require amount of electricity and delivery to the motor and microcontrollers. There will be two micro controllers, one to control the LED's and another one to control the motor. Between the AC/DC power regulator and microcontroller a switch will be placed and when the power gets to the motor, the microcontroller will receive analog signal through one of its pins, and after the signal is received, the microcontroller will turn the green light on. Likewise, when the microcontroller controlling the LED's receive the power from the AC/DC converter, the microcontroller will received analog signal through one of its pins, and will turn the red light to indicate the LED's are powered. The implementation of this user interface is depicted in Figure 2.1.

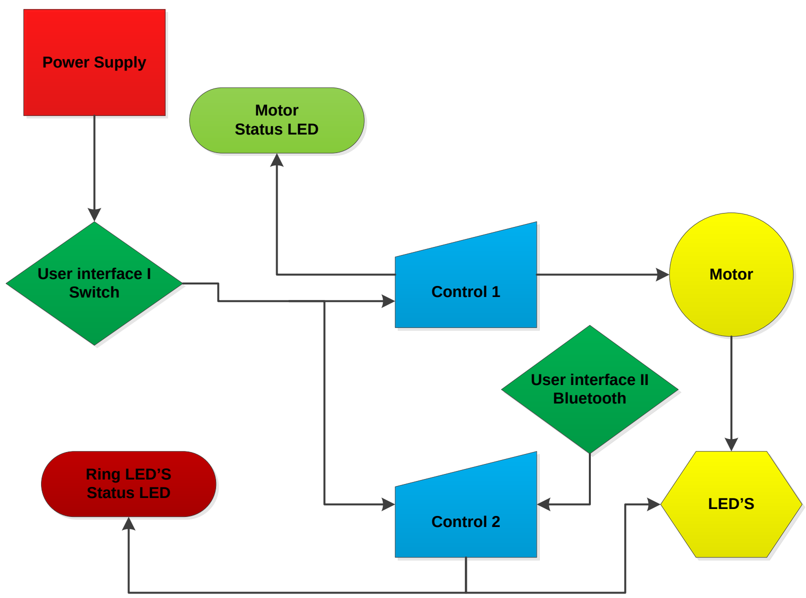


Figure 2.1: User Interface (Switch)

The microcontrollers for the motor and LED's will be programmed with a simple conditional statement, so if they received each of the voltages going to these two devices the corresponding LED will turn on, and as it was explained before two analogs, pins are going to be used to detect the power going to the devices and when this power is detected, two digital pins will be used to turn on the respective green LED for the motor and the red LED for the LED's ring.

2.2.6 Input 2: Bluetooth

After a power connection is established, a Bluetooth module will be used to control the lights sequences and to start the motor. The JY-MCU Bluetooth Wireless Serial Port is the module is going to be used to establish the connection between the user interface and the LED Globe. This module has four pins: RX, DX, VCC, and GND. The module requires an input voltage between 3.6 volts and 6 volt; the RX and TX pins of the module are going to be connected to the TX and RX pins of the ATMEGA respectively. The device used to send the command can be any cell phone that has Bluetooth capability and to receive the commands the ATMEGA will be programmed with serial communications.

The ATEMA used for this project has a serial port or also known as Universal Asynchronous Receiver Transmitter port (UART). Through this transmitter and receiver pins we can establish communication between the Bluetooth and the ATMEGA, so the Bluetooth will receive information from the cell phone after devices, the cell phone and Bluetooth are paired. More details about this communication are covered in the microcontroller section of the project.

Moreover, the Bluetooth is module is going to receive three commands: A, B, C.

A. In this command an upper case letter ‘A’ is going to be sent from the cell phone to the ATMEGA to start a sequence which will display, “University of Central Florida Senior Design Fall 2013 Omar Vazquez Rhonal Soto Viet Nguyen”
B. In this command an upper case letter ‘B is going to be sent from the cell phone to the ATMEGA to start a sequence which will display a earth globe.
C. In this command an upper case letter ‘B is going to be sent from the cell phone to the ATMEGA to start a sequence which will display a random sequence of lights with different colors on and off.

The Bluetooth connection to the microcontroller is depicted in Figure 2.2, and more details about the LED’s sequences and design are given in the output section, LED section, and programming section of the project.

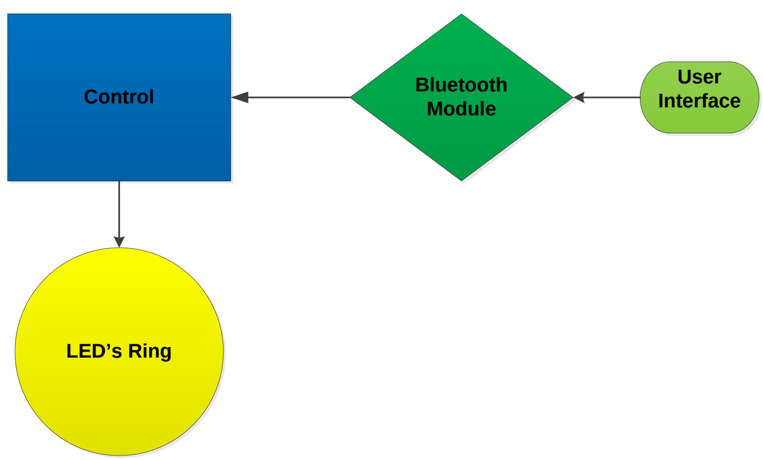


Figure 2.2: Bluetooth Module

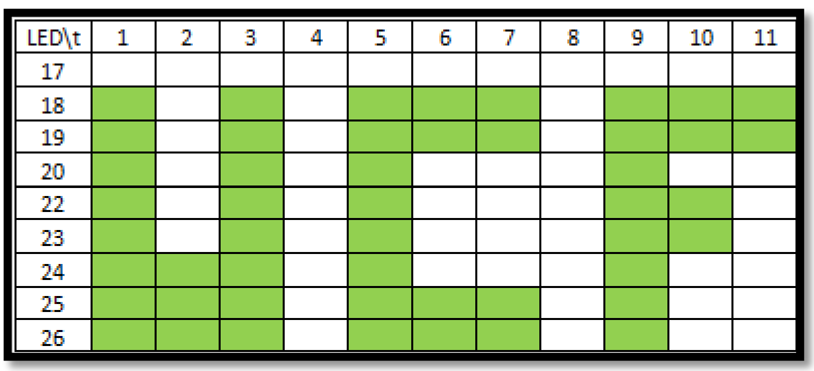
2.2.7 Control Box.

Control box is where the power supply AC/DC convertor and the microcontroller to drive the motor is going to be located this unit to be knowledgeable about system status. The control box is one half of the brain of the system with the responsibility of regulate and send power to the motor, LED ring and most importantly to the other half of the brain which is the microcontroller controlling the LED’s on the ring.

2.2.8 Output

The output of the system will be the LED display of any preprogrammed images in the microcontroller or an uploaded text message from the user using the Bluetooth interface. The type of image that is going to be displayed depends on the program uploaded to the microcontroller so far; as it was mention before we had established three possible sequences: A, B, and C.

Furthermore, sequence A is going to display the university name or logo in three possible colors red, green, or red. The output displayed in this sequence is depicted in Figure 2.3.



LED\t	1	2	3	4	5	6	7	8	9	10	11
17											
18											
19											
20											
22											
23											
24											
25											
26											

Figure 2.3: UCF Output

The sequence B as it was also mention before will display an earth globe. For this sequence the blue LED’s are going to be on during the entire sequence. The green lights will be moved from right to left to simulate the rotation of the earth (Figure 2.4 and 2.5). Moreover, the date and time will be displayed in the center of the globe with the red LED’s, and this one is going to be moving from left to right as depicted in Figure 2.6.

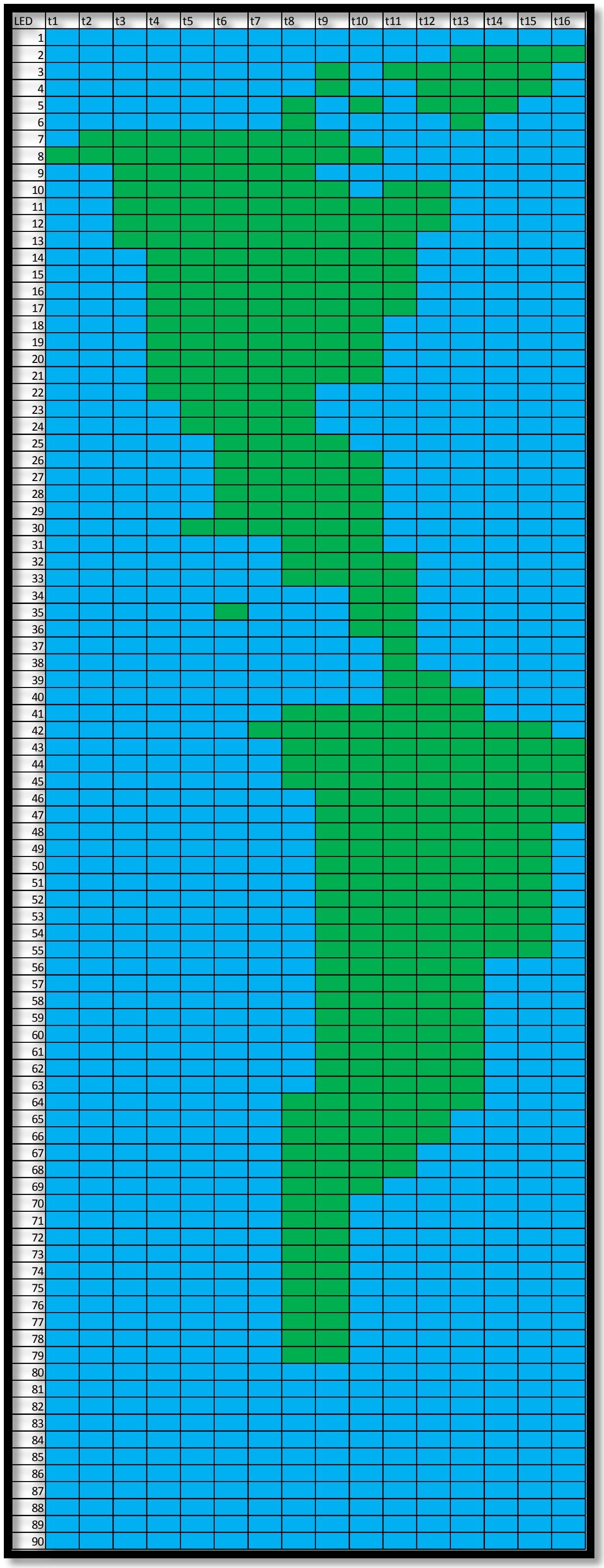


Figure 2.4: Earth Output 1

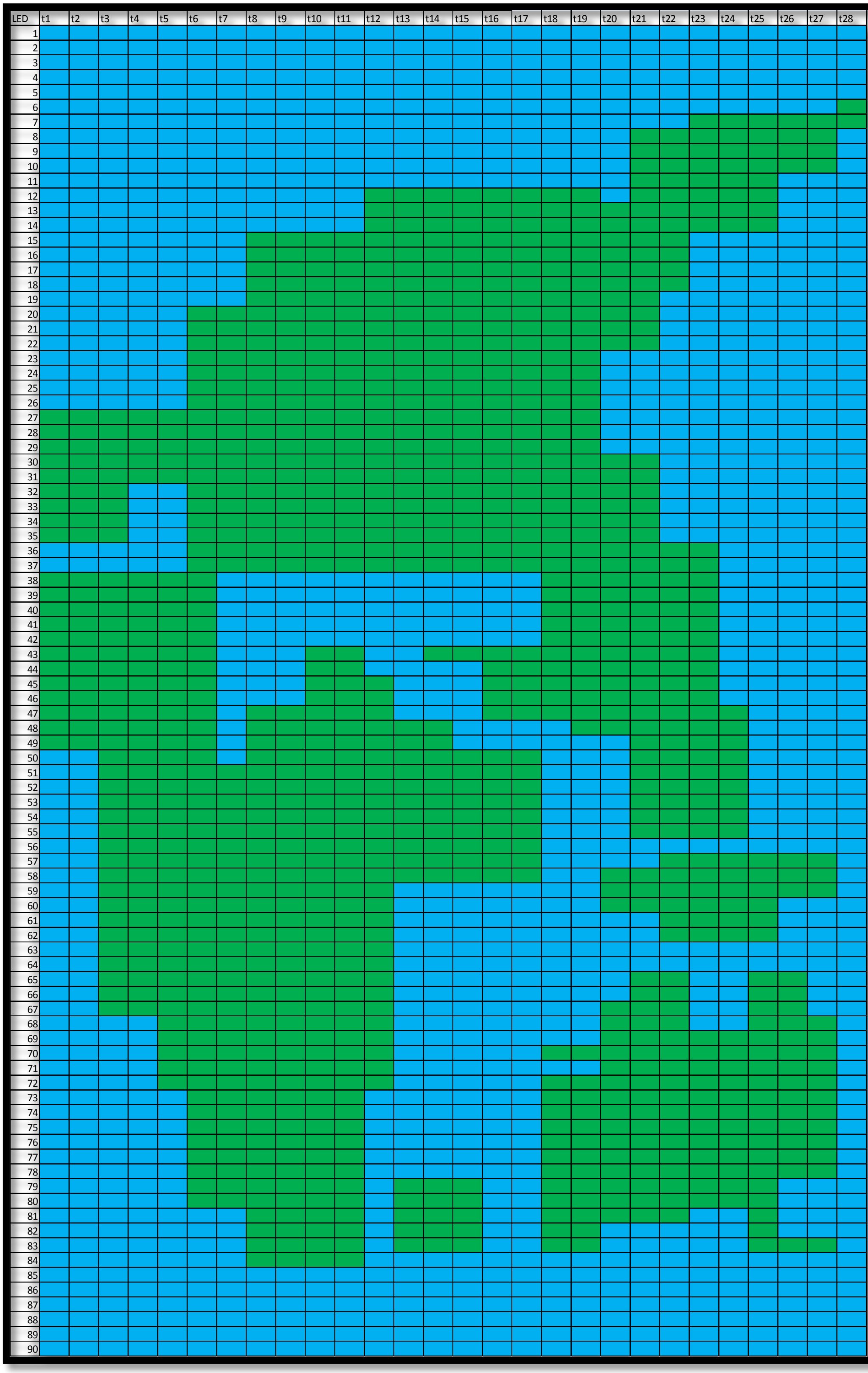


Figure 2.5: Earth Output 2

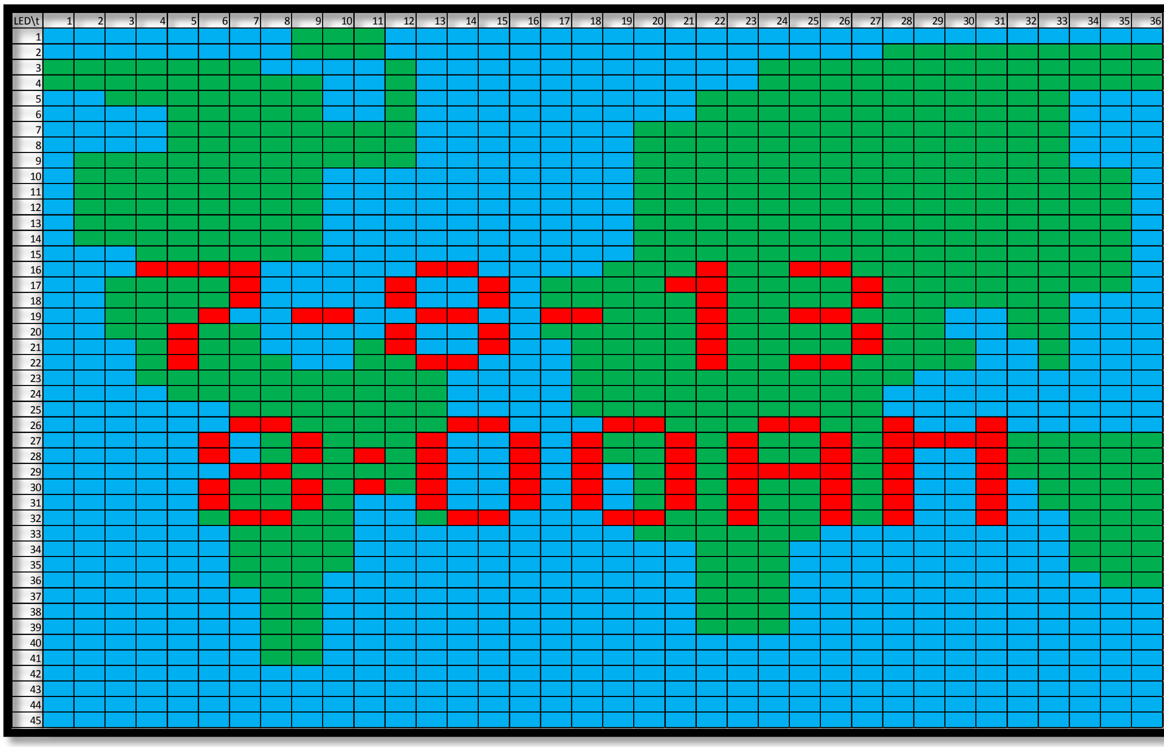


Figure 2.6: Earth Output Date and Time

As it can be seen, if we increase the number of LED's the definition also increases. For Figure 1 and Figure 2, 90 LED's were used and for Figure 3, 45 LED's were used. Keep in mind the green and red portions in Figure 6 are going to be rotating in opposite directions.

2.3 Project Requirements and Specifications

The required operation of the LED globe is to be able to display a preprogrammed image or text from the user using the Bluetooth interface by lighting the LEDs at a certain space and time. All of the internal components should consume as little power as possible. Moreover, the system must be safe both for the user and the environment. There are three categories of specifications: 1) power supplies 2) Voltage regulators 3) and the microcontroller. The specifications are shown in tables 2.1, 2.2, and 2.3 as follows.

Motor Power Supply	
Input Voltage	120V AC
Output Voltage	9V DC
Output current	<2A
Weight	< 2lb
Microcontroller Power Supply	
Output Voltage	9V
Output Power	<1A
Weight	<.1lb

Table 2.1 Power Supply Specifications

Microcontroller	
Clock Frequency	Low
Serial Ports	Yes
Programming Language	High level similar to C
Programming Memory	≥16K
Analog Pins	Yes
Digital input/output Pins	Yes
PWM Output Pins	Yes
Programming Debugging	Yes
Power consumption	Low, good sleep mode

Table 2.2 Microcontroller Specifications

Voltage Regulator for Motor	
Maximum Voltage	32V
Output Voltage	9V
Voltage Regulator for microcontroller	
Input Voltage	32V
Output Voltage	9V

Table 2.3 Voltage Regulator Specifications

LEDs	
Number of LEDs	80~100 units
Number of colors per LED	256 colors
Refresh Rate	60 Hz
Required speed for POV	25 rotations/s or 1500 rpm

Table 2.4 LED Specifications

3 Research

Without proper research, constructing any project would be a pain-staking task. Even a little brainstorming will go far when it comes to research. There are plenty of methods and procedures that can be looked up to get to the same solution and even then, there might be many different types of solutions to a problem. Research is necessary to plan out a project step by step and provide the appropriate information in order to work on the project more efficiently. The LED globe is based on the persistence of vision theory

Persistence of vision is the theory. This theory states that the eye will retain an image for approximately 1/25 of a second on the retina after the image has been removed. This slight delay of 1/25 of the second can be advantageous for certain applications including film, television and advertising. There has been other persistence of vision related projects designed by engineering students and hobbyists.

3.1 Other Persistence of vision related projects

Research into a POV LED Globe was quite easy. There are many examples on the Internet. Their websites detail their materials, processes, design and even have videos of their model tests.

Most of the research we did was about what kind of components we wanted to use. Decisions that had to be made on what power supply we wanted to use or how were we going to transfer data between our input and output. The first issue we must face is how to power our LED globe. Do we want to make it portable or have it plugged into an outlet? If portable, then we need to find a battery module to power our device. If we are using the outlet, then all we need is a power adapter that plugs into the wall. The second issue would be communication between input and output. Do we want USB connection, WiFi connection or a Bluetooth connection? USB connection will require a USB port in our device and also for the device to be always connected to our input device (computer). WiFi connection will have our device's connection to the input be wireless but we will have to use a Wifi module in our device and also have a router of some sort with our input device. Bluetooth connection only requires one part and that is the Bluetooth module on our microcontroller.

Moreover, SolderLab has also made a LED Globe as seen on their website [1]. Their first version had 40 RGB LEDs arranged at the outside of a rotating ring with a diameter of about 25 cm. The LEDs are controlled by one ATmega328 with 15 8bit shift registers (74HC595) sitting on two PCBs in the center of the globe. Their power was supplied via slip ring. Their design is a bit similar to what we are going for so this is good reference material.

Also, this is also very similar in design to what we want to make [2]. They used SMD RGB LEDs (5050, PLCC60). They also had shift registers. Their electric motor was relatively small but powerful and could provide the rotational frequency that was required to maintain POV. They used an ATMega644 to program their globe.

3.2 LED Globe Mechanical Structures

3.2.1 LED Ring Structure

The LED ring structure is a 12-inch diameter by 1 inch wide by .12-inch thick ring made out of balsa wood. Other materials were considered including aluminum and carbon fiber to make the LED ring structure, but carbon fiber was hard to find and tricky to make properly. Aluminum is heavier, which is about 395 g/m^3, it is harder to mold, and it is more expensive than balsa wood. Balsa wood is very lightweight that is about .16g/m^3, which is about 99.9% lighter than aluminum. Thus balsa wood was the best choice for this application. The light weight of the balsa wood will help to decrease the loading on the motor which in turn would allow us to design a more energy-efficient LED globe. Another advantage of the balsa wood is that it can bend easily, as opposed to the heavier aluminum. To make the balsa wood flexible, the balsa wood was first soaked in a half water half ammonia solution for about two hours. This solution helps to break down its molecular structure, which allowed the wood to be bent around a 12-inch diameter disk without splintering.

The ring then was clamped down and allowed to dry in an environment of about 50% relative humidity for about 48 hours. After the 48 hours has passed, the ring's ends where overlapped at about 2 inches, then the overlap was secured using two quarter inch screws which were tightened down with two locknuts. The whole ring structure weighs about 1.3 ounces. A half-inch diameter hole was drilled on the bottom of the LED ring structure to allow the ring support shaft to fit in. A quarter inch diameter hole was drilled on the top of the LED structure to allow a quarter inch bolt to be fitted in and screwed into the ring support shaft.

3.2.2 Ring Support Shaft

The support rod going through the center of the ring is a half-inch diameter by 27 inches long hollow aluminum rod. A half-inch diameter threaded insert, which is used for wood furniture, is inserted into one of the rods ends. A 0.25" diameter x 1.6" long bolt is to be inserted through the ring and screwed into the insert to hold the ring in place. Near the head of the bolt, a quarter of its length is smoothed down in order to insert a 8.6" x .25 diameter with a quarter inch cutout cylindrical ball bearing. This ball bearing will be held in place by the support shaft. At the other end of the support shaft, a 1.12-inch diameter cylindrical ball bearing with a half-inch diameter cutout in the center is inserted 1 inch up from the rod’s end; this will be used to help with the rotation of the LED ring structure. Two quarter inch copper rings will be wrapped around the outside of the support shaft with insulation underneath. These two copper rings, will be the slip rings to provide power to the microcontroller by means of copper brushes; the microcontroller will be placed inside of the ring structure.

A wire will be soldered onto the upper copper slip ring towards the top to provide positive power to the microcontroller, and the bottom ring will likewise have a black wire soldered onto the copper slip ring to provide ground for the microcontroller. A small hole will be drilled on the support shaft between the two copper slip rings. This drilled hole is necessary because the bottom copper slip ring wire, which will be the negative terminal for the microcontroller, will lift up the upper copper slip ring’s brush if allowed to rotate, and thus this wire will disconnect power to the microcontroller. So to avoid this, the lower copper slip ring wire will be inserted through the hole and go underneath the positive terminal and come out above the ring’s bottom through another hole in the support shaft, to power up the microcontroller and thus avoid disconnecting power to the microcontroller.

In the center of the support shaft, two aluminum plates, 5" x 2", will be screwed onto the support shaft. These two aluminum plates will mount the PC boards. The two aluminum plates will be insulated to prevent any short-circuiting on the PCBs. Mounting the two aluminum plates in the center of the support shaft helps with decreasing rotational inertia, wind resistance and torque which can load down the motor, and consume more power.

3.2.3 Housing Unit Frame

The housing unit frame will be the support needed for the plexiglass housing encapsulation. The housing frame dimension will be 6" x 6" x 6". The housing frame will be made out of 1"x0.08" x 6" aluminum bars connecting each corner of the box thus creating a hollow cube. The reason for choosing the aluminum bars was because it was cheaper than Iron and lighter than iron. The density of iron is about 787.4g/m^3. Please see Figure 3.1 on the next page.

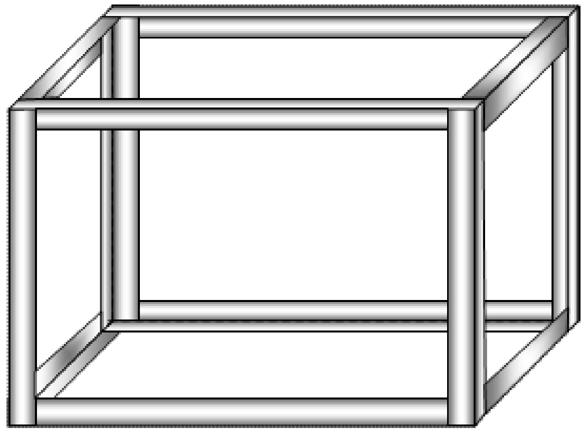


Figure 3.1 Housing unit frame

3.2.4 Plexiglass housing encapsulation

This will be the foundation for our LED Globe. The Plexiglass housing encapsulation will be a clear housing encapsulation to show the electronic circuits of the power supply plus the motor. The Plexiglass housing encapsulation will be made up of six 6" x 6" x .5" clear Plexiglass boards that will fit into the housing unit frame. The board on the very back will be perforated to provide ventilation for the motor and the power supply and will secure the support bar. In the center of the top board a 1.13 inch in diameter hole will be cut out to fit in the support shaft rotating cylindrical ball bearings; a .43 inch diameter hole will be cutout to secure the on-off power switch; a .19 inch diameter hole will be cut out for the power switch LED and an additional 19 inch diameter hole will be cut out for the Bluetooth LED. Underneath the 1.12-inch diameter hole, a four-inch by 2.067 inch by .5 inch Plexiglass board will be nailed perpendicular to the top board. This board would be used to hold the motor and to erect it vertically in place. The motor will be fastened by two U bolts, which will protrude through the 4 inch by 2.067 inch by .5 inch Plexiglass board and bolted in place. A 2.067inch by 1.409 inch clear by .5inch Plexiglass board will be nailed perpendicular to the 4 inch by 2.067 inch by .5 inch Plexiglass board at the very bottom to help support the motor in place. The very bottom board will have for rubber pads on each corner to provide friction and prevent slippage on a smooth surface.

3.3 Electric motor

The motor is possibly the most important part of this project. The motor is one of the key components for creating the illusion of persistence of vision. We will need a motor that can spin at a certain-constant rpm in order to create POV. We need a strong motor that will consume as little power as possible in order to run and minimal distraction due to noise.

The human eye can only see motion at a rate of about twenty-four frames per second. So we need our motor to spin at about the same rate of twenty-four rotations per second at the very least. That is the minimum speed for the motor.

Research into this has shown that in order to get the torque required to spin the LED ring structure, the motor will need at least half to one horsepower for the initial startup. After that, the motor will need to push out a constant torque in order to keep the LED ring spinning at a constant rate. We need the motor to achieve a constant rpm to get POV to happen. Let's analyze now the LED ring structure and determine the type of motor we will need. Let's start with the speed of the motor.

In order to calculate the amount of speed needed for the motor to rotate the LED ring effectively, we must start with simple geometry and rotation physics. Consider Figure 3.2 below.

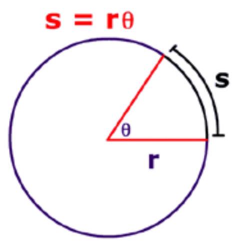


Figure 3.2 Arc Length Definition

In order to perceive one continuous motion according to the persistence of vision theory, the time lapse between images must be one in 24 seconds or less. Thus:

$\Theta = \frac{S}{r}$ and we need an average angular velocity of ωavg = $\frac{2\pi \text{ rad}}{\frac{1}{24} sec} = \frac{150.79 \text{rad}}{sec}$.

Convert this into revolutions per sec:

$$\frac{150.79 \text{ rad}}{sec} * \frac{1rev}{2\pi \text{ rad}} = \frac{24rev}{sec}$$

Convert into revolutions per minute:

$$\frac{24rev}{sec} * \frac{60sec}{1min} = 1{,}440rpm$$

Now that we have the angular velocity and the RPM for persistence of vision, now we need the time it takes for each array of LEDs to turn on or off in sequence to the preprogrammed image in accordance with its angular displacement as the following Figure 3.3 will explain.

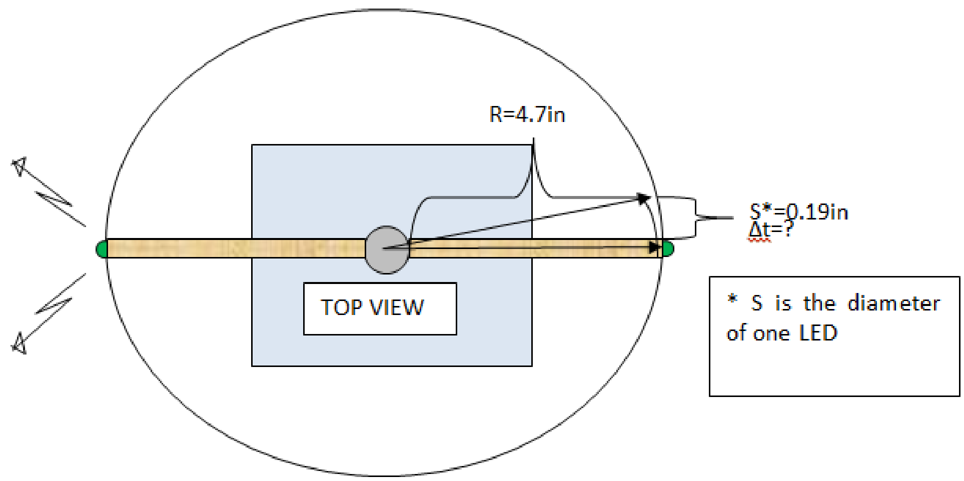


Figure 3.3 Top View of LED Globe

Since S=θR, θ=S/R then θ = 0.0404Rad and 0.0404=ωt. Solve for 't' we have:

$$\frac{0.0404}{2\pi/(\frac{1}{24})} = .26808ms$$

Now let's calculate the torque and the horsepower required for the motor.

If we were to take the weight of the ring structure plus the ring support shaft plus the shift registers and the LEDs and add them all up, the total weight will be 15.8 ounces. The radius of the ring is about 4.7 inches. If we were to take the sum total weight of 15.8 ounces and take the perpendicular distance from the ring shaft to the edge of the ring structure, which again is the radius of the circle, then this will be the lever arm to calculate the torque on the motor. See Figure3.4 below.

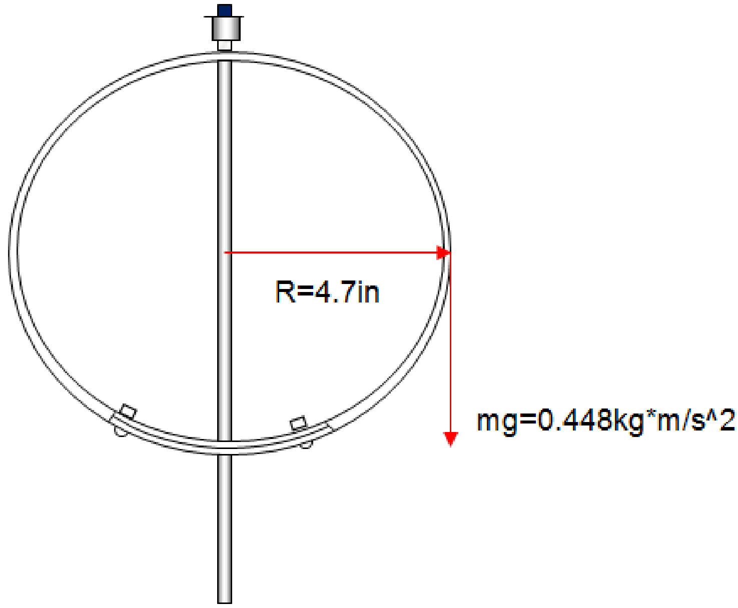


Figure 3.4 Torque Measurements

Given: 15.8oz=0.4479225kg, g=9.81m/s^2, R=4.7inches=0.11938meters

So we have: (0.4479225*9.81)kgm/s^2 = 4.39N

Torque = τ = F*D where F is Force in Newton and D is the perpendicular distance from the axis of rotation to the force being applied.

So F=4.394N and D= 0.11938m $\therefore$ $\tau = 4.39*0.11938 = 0.524N*m$

Motors are measured in ft-lb, so let's convert the torque to ft-lbs.

1N*m= 0.7375621 ft-lb $\therefore$ $\tau = 0.524\text{N} * \text{m} * \frac{0.7375261}{1\text{N}*\text{m}}$ = 0.38654ft-lb

Now let's convert 0.38654ft-lb to horse-power.

Hp=(Torque in lb-ft)*RPM/5253.5=(0.38654*1440)/5253.5 = 0.106 lb-ft*rev/min

Now let's convert horsepower to watts.

1 hp is equal to 746 Watts so: P(watts) = $0.106\ \text{lb} - \text{ft} * \frac{\text{rev}}{\text{min}} * \frac{746}{\text{lb}-\text{ft}*\frac{\text{rev}}{\text{min}}} \cong 80watts$

The electric motor that we are going to use is The HobbyWing XERUN 3.5T 3650-9100KV. This motor has a speed of 3650 RPM per voltage.

Let's calculate what would be the avg time lapse at this RPM.

$$\frac{3650rev}{1min} * \frac{1min}{60sec} * \frac{2\pi rad}{1rev} = 382.23\frac{rad}{sec} = \frac{2\pi rad}{tsec} \therefore t = .0164s = \frac{1}{60.833}sec < \frac{1}{24}sec$$

This meets our criteria for persistence of vision. This motor is also capable of supplying the 80watts that we need.

3.3.1 Specifications

The motor specifications are shown in table 3.1 below.

Model	KV	Diameter mm	Length mm W/O shaft	Shaft Diameter mm	Shaft Length mm	Weight grams
3.5T	9100	35.8	52.5	3.17	15	165

Table 3.1 Motor Specification

A special note about this motor is that it has a stepless timing adjustment. It has top quality materials like an aluminum shell case and high quality magnets within the motor. The copper wires inside the motor have high temperature endurance. The HobbyWing XERUN 3.5T 3650-9100KV has a speed of 9100 RPM per voltage. It also has the possibility to be configured with a PC USB connection which is possible with the HOBBYWING (2 In 1) LCD Program Box; this allows you to set every single detail of the ESC. The HobbyWing XERUN 3.5T 3650-9100KV is Compatible with all sensorless brushless motors and most sensored brushless motors such as LRP, Novak, Orion, Feigao, etc. Seamlessly changes to sensorless working mode when the sensor cable is broken. Lithium battery and NiMH battery supported. This motor has excellent start-up, acceleration and linearity features.

Firmware can be upgraded through an USB adapter on the Advanced Professional LCD Program Box (Optional equipment). It is user programmable, easily programmable with only one button on the ESC and compatible with the 3 Digital LEDs Program cards with the advanced professional LCD program box.

3.3.2 Motor Advantages and Limitations

There are many different types of motors for different jobs. For example, the AC motor is a cheap and simple motor used in familiar electrical appliances; however the AC motor can only operate on a fixed speed. To achieve different speeds will require complex control systems, and multiple windings or gearboxes which all of this in return cost extra money.

For the LED globe, we will use an electric DC motor. There are two types of DC motors: brushed DC motors and brushless DC motors. The brushed DC motor is capable of providing various speeds without the need of an extra control system, which is the advantage over to AC motor, however the disadvantage of the brushed DC motor is that the brushes wear out and need replacement, which can be expensive and undesirable. The brushes can also create dust, which is also undesirable. The brushed DC motor is subject to arcing, which can produce radiofrequency interference that can interfere with the operation of the microcontroller or even the power supply, and finally the arcing presents a safety hazard.

The motor that we are using for the LED globe is a brushless DC motor. The brushless DC motor gives us the advantage of a brushed DC motor, that is, the brushless DC motor is capable of providing various speeds without the need of an extra control system. On top of that, the brushless DC motor, like the name implies, does not have any brushes, which eliminates the need of replacing the brushes due to wear and tear which is one less expense; the brushless DC motor will not admit radiofrequency interference due to the fact that it does not have any brushes.

The motor that were using, the HobbyWing XERUN 3.5T 3650-9100KV, is a brushless motor. The main disadvantage of this motor is its high cost and lack of torque if the load is too high which can reduce the speed of the motor.

3.4 Microcontroller Unit

Microcontroller units will be the brain of the LED Globe since they will control the motor that moves the ring and also the sequences of light the LED Globe is going to display, not to mention it will monitor the power supply to the microcontrollers. The chosen microcontroller must be capable to run a fast clock speed, being low power consumption device, having enough appropriate analog and digital I/O, and have small size to be integrated on designed board. At this moment, our project have identified the need of having 2 voltage sensors and, 2 current sensors plus all the out needed to turn the lights on and off. So far our LED Globe will need two microcontrollers, but depending on the procedure chosen to maximize the output, more microcontrollers are going to be needed. If the procedure chosen is Charlieplex, we will need the microcontrollers with the maximum I/O pin capability; however, if shift register are used maximize the output, the space for coding will be what is going to determine the microcontroller used. Decision about if two unit or more are needed will depend on balanced between cost, performance; nevertheless, it is aimed that selected microcontroller unit meet the following model:

- Low cost on the unit and desirable on the development board as well
- Low power consumption
- A high level language to be programmed similar to C/C++
- Sufficient memory, +16K of flash memory
- Enough amount of analog I/O ports
- JTAG debugging
- Convenient software, libraries, IDE
- Processor speed exceeding our routines/tasks
- Practical to be integrated with external peripherals (Wireless, data logging)
- Good community support is not mandatory but desirable
- Good sleep mode when it is not in use

3.4.1 ATMEGA328P

The ATMEGA328P principal limitation is the flash memory, which is 32 KB. Most of the microcontrollers of this type have around the same capacity or even less. On the other hand, the language that is used to program this microcontroller is simpler in comparison to the assembly languages used in other microcontrollers. The power efficiency is also an advantage provided by the microcontroller, the feature of having 8 pins that can be used for PWM makes possible controlling multiple devices with different level of variant power. Finally the cost of the microcontroller is extremely low making possible to compensate the space limitation the microcontroller has by using many of them communicating between each other through an I^2C protocol.

The ATMEGA328P is a low power, high performance 8 bit microcontroller that has 14 digital input/ output pins, and 8 of this pins can be used for Pulse with Modulation (PMW); also has 6 analog pins and a 16MHz ceramic resonator.

The Microcontroller operating power is 5V, and the input voltage recommended is 7-12V never exceeding the 20V. With a total of 20 pins, its I/O pins provide a DC current of 40mA and its 3.3V pin provides a current of 50mA. It also has a Flash memory of 32 KB, an SRAM of 2KB, and EEPROM of 1 KB with a clock speed of 16MH.

3.4.2 MSP430G2553

The MSP430G2553 is also a low power, high performance but 16 Bit RSIC architecture microcontroller. Its architecture combines with five low power modes is designed to achieve extended battery life. It also has a digitally controlled oscillator (DCO) that allows wake-up from low power modes to active modes in less the one microsecond. Besides the above mentioned, the code composer used to programmed this microcontroller, allows the user to program it using two different languages, C and Assembly. Limitations on these microcontrollers are the 513b of SRAM space, which leaves not a lot of space for coding, and the fact that the microcontroller counts only with 20 pins including VCC and GND.

This is a low cost microcontroller; however, the evaluation debugging board can reach the $100 and IDE to load the code has to be purchased as well for about $400. Free open source alternative software is available but not really the best option. Furthermore, Texas Instruments proprietary IDE has a free version with full capability but bears a limitation of 16 KB of total code. Despites price to be invested on the software, this MCU is considered as a good alternative because its features and low power consumption.

The voltage range of this microcontroller is between 1.8V and 3.6V with an active mode of 0.230 Amps at 1MHz and standby mode of 0.0005 Amps. As it was mentioned before it counts with five different saving modes with a wake up time from standby mode of 0.0001 second. It also has 16-Bit RISC architecture with 62.5-ns instruction cycle time.

3.5 LEDs

There are many different LEDs with many different aspects and characteristics that we can choose from. To determine which ones that will be suited to our needs, we need to keep in mind that the LEDs are needed for POV so size and the way we mount the LEDs onto the apparatus are large considerations that we need to keep in mind.

The number of LEDs that we want to include in our LED Globe is ninety. The final number may be smaller due to the limitations of space on the ring. We want the LEDs to be RGB, which means that the colors red, green and blue can be used in our project. Combining the colors of red, green, and blue can make most of the colors that we need. This is done when red, green or blue light is added together in various ways to produce other colors.

To determine which LED that will suit our needs; we need to determine the characteristics that will fit what we want. The LEDs must be small in size and be easily mountable onto the LED ring structure. The LED’s will are small because we want to fit many them on the ring structure; easily mountable, so as to not have a hard time in trying to mount many LEDs onto the ring structure. The structure and a mock picture of the LED are shown below in Figure 3.5.

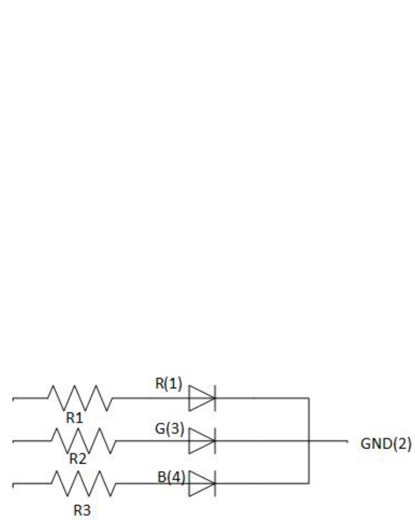


Figure 3.5 Common Cathode LED

There are two ways that we are exploring to control the LEDs. One involves shift registers. The other involves an inter-integrated circuit or IC^2. We talk about shift registers in a later section because that is the method that we want to use. IC^2 would involve connecting a series of microcontrollers together. One would act as a master while the others would be its slave. Basically, the set up would help us to control the LEDs better since we would have to program and send data only into the master and program the slaves to just react to the data that the master microcontroller would send to the slave microcontrollers. In theory, this would sound like it would work for us but this would increase the size of the housing unit for the microcontrollers and in turn would increase the cost and budget for the entire project. That is why we will be choosing the shift register method [3].

3.6 Power Supply

3.6.1 Design

The power supply needed for the LED Globe is a simple transformer- bridge rectifier circuit. The input voltage will be 120 Vrms and the transformer will step the voltage down to 24Vrms. After the rectification process and some filtering the supply voltage for the motor and the microcontroller will be about 32V RMS. The motor voltage will require 9V and so will the microcontroller.

To get the peak voltage, multiply the RMS voltage times the square root of two, so we have the input voltage maximum to be Vp=120*1.42=170.4V. Step down voltage maximum will be Vs.=24*1.42=34.1V and the supply voltage after filtering will be V=22.6*1.42 = 32.1V. Our power supply must be able to handle these peak voltages at any moment even though these peak voltages are not constant.

The maximum power that our power supply should deliver is a combination of the maximum power available for the motor and the maximum power available for the microcontroller. The maximum power available for the motor is 80 W, and the maximum power available for the microcontroller is about 9 W, so in total our power supply should be able to deliver about 90 W of power.

3.6.2 Advantages and Disadvantages

The main advantage of a simple transformer-bridge rectifier circuit is that it is simple to design and uses minimum components to make an effective power supply. The disadvantage is the size of the transformer, which takes up a lot of space, and can heat up.

3.6.3 Specifications

The power supply, rectifier, and smoothing capacitor specifications are shown in table 3.2, 3.3, and 3.4 below.

AC Voltage input	**Frequency**	**VA**
240Vrms to 120vrms	60HZ	100

Table 3.2 Transformer

Forward Voltage Drop	**If**	**Ifsm**	**Ir**	**Tj max**
1V	3A	200A	5µA	150deg c

Table 3.3 Four Rectifier Diodes 1N5406 DO-201AD

Capacitance	Peak voltage	Tolerance
52.08mf	**75V**	**1%**

Table 3.4 Smoothing Capacitor Cornell Dubilier CGS263U075V4C

3.7 Voltage Regulator

A voltage regulator is used to maintain a constant voltage level; this can be found after the bridge rectifier and smoothing capacitor circuit. This is necessary to help the motor maintain a constant speed. There are two things that the regulation is specified by: the load and the line or input. The load regulation is the change in output voltage for a given change in load current. Line regulation or input regulation is the degree to which output voltage changes when input voltage changes.

There are some other important factors that go into voltage regulation:

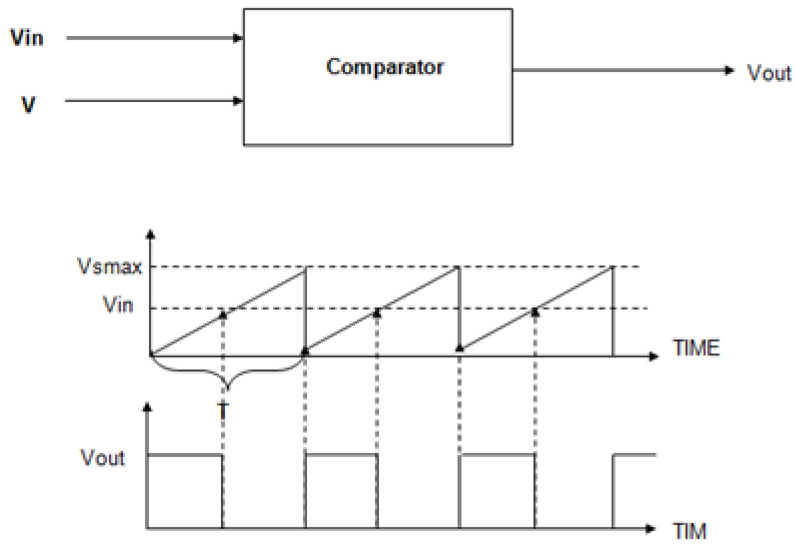
- Temperature coefficient of the output voltage
- Minimum difference between input voltage and output voltage so the regulator can still supply current
- Reaction of regulator when a change of load current or input voltage occurs
- Output noise and output dynamic impedance
- Absolute maximum ratings
- Quiescent current
- Initial accuracy
- Mirror-image insertion protection

There are some ways to make a simple voltage regulator other than just buying one. A voltage regulator can be made with just a resistor and a diode; they have to be in series with each other. You can make it this way only when precise voltage control and efficiency are not of major importance.

3.7.1 Motor Voltage Regulator

The motor will use a switching voltage regulator. The switching voltage regulator is use due to the fact that it is capable of reducing power loss and has a higher efficiency than the linear voltage regulator. The switching voltage regulator uses a pulse width modulator, which produces a varying periodic waveform with a desirable duty cycle. The pulse width modulator is nothing more than a comparator with a sawtooth waveform being fed into the comparator. The comparator has to voltage inputs: Vin, which is the input voltage and Vs, which is the input from the sawtooth waveform. If Vin is greater than Vs then the comparator has an output voltage; furthermore when the voltage of the sawtooth waveform increases higher than the input voltage, then the comparator output voltage goes to zero until the end of the sawtooth voltage duration. Then the voltage in Vs goes back down to zero to start a new cycle, and since the voltage now is less than the input voltage, the comparator output voltage is switched back on. The output voltage of the switching voltage regulator is a square wave signal with a duty cycle dependent on the frequency of the sawtooth waveform.

The Figure 3.6 helps illustrate.



Figures 3.6 Voltage Regulator

Since the output voltage is not constant, less power is being dissipated. Of course if the period of the square wave is too large, then the motor would just turn on and off. Therefore the frequency of the square wave should be a kilohertz or greater to make it seem like a continuous DC voltage.

3.7.1.1 Specifications for Motor SVR

Specifications about the motor SVR are shown in Table 3.5 on the next page.

LM25117	Switching Regulator	
Vin	Min=14v	Max=32v
Cin	10µf	
Cvcc	470nf	Vmax=10.0V
Rramp	118KΩ	
Cramp	820pf	Vmax=50V
Rt	20KΩ	
Css	15.0nf	Vmax=50.0V
Ccomp	1.56nf	Vmax=50V
Rcomp	39.2KΩ	
Rfb2	12.1KΩ	
Rfb1	1.18KΩ	
Cout	100µf	
Vout	9V	
Iout	10A	
L1	6.8µH	
D1	410mV	I=70mA
Rsns	.007Ω	
Cboot	100nf	Vmax=50.0V
M1	Vdsmax=40v	Idsmax=30A
M2	Vdsmax=40v	Idsmax=30A
Cres	470nf	Vmax=6.3V
Ruv1	6.34KΩ	
Ruv2	54.9KΩ	

Table 3.5 Motor SVR Specifications

3.7.1.2 Advantages and Disadvantages

The main advantage of the switching regulator for the motor is efficiency. The disadvantage of switching regulators is complexity. Several external passive components are required for the correct function of the switching regulator. In the case of high-current applications, external FET ICs are required as the IC-converter acts only as control logic for the external FET switch. Another disadvantage is output voltage ripple. This is generally handled with bypass capacitance near the voltage supply and at the load.

3.7.2 Microcontroller Voltage Regulator

3.7.2.1Design

The heart of the microcontroller linear voltage regulator is the LM7809 voltage regulator. This voltage regulator can accept a wide range of input voltages and outputs 9 V ± 3.7% the maximum output current is one amp. Two filter capacitors, one placed at the input and the output, will be used to filter out any noise or unwanted ripple.

3.7.2.2 Specifications

Specifications about the microcontroller voltage regulator are shown in able 3.6, 3.7, and 3.8 below.

Vin	**Vo**	**Tj**	**Io**	**Line Reg**	**LoadReg** **Io=5mA to1.5A**
5V to 18V	9 V ± .7%	+25C	5mA$\leq Io \leq 1A$	6mV+.18V	12mV +.180mV

Table 3.6 LM7809

	Value	**Peak Voltage**
C1	470µf electrolytic	30 V
C3	.1µf electrolytic	100V

Table 3.7 Filter Capacitors

Characteristics	**1N4934**	**Units**
Peak reverse voltage	100	**V**
RMS Reverse Voltage	70	**V**
Average rectified output current	1.0	**A**
Forward voltage drop	1.2	**V**
Peak reverse current	100	**µA**
Operating and storage temperature range	-65 to+150	**°c**

Table 3.8 Protection Diode 1N4934

3.7.2.3 Advantages and Disadvantages

The main advantage of the linear voltage regulator for the microcontroller is the output voltage regulation, its small size and its simplicity in design. The main disadvantage of the linear voltage regulator for the microcontroller is its power efficiency which can be quite low, about 60 to 40%. Besides its low efficiency, a regulated voltage is crucial for protecting the delicate internal components of the microcontroller.

3.8 Shift Registers

A shift register is a series of flip-flops that share the same clock and each flip-flop's output is the subsequent flip flop's input and so on. Basically, a chain of flip flops that result in a circuit that shifts by one position the bit array stored in it, shifting in the data present at its input and shifting out the last bit in the array, at each transition of the clock input.

There are many different types of shift register combinations that deal with inputs and outputs. There are bi-directional shift registers and circular shift registers. Shift registers can have both parallel and serial inputs and outputs. There are two configurations that can be done with a shift register: serial-in, parallel-out or parallel-in, serial-out.

There are many uses for shift registers. The main function of a shift register is to convert between serial and parallel interfaces. That being said, the use of the shift register in our project is to help with the function of the sequencing of our LEDs. To explain, we will send data into the microcontroller to send to the shift register. Using the clock and the data we sent, we can blink the LEDs according to the clock. Figures 3.7 and 3.8 show how the shift registers and the LEDs can be connected.

Each LED in our LED array needs to blink at the same time as all the other LEDs. That means that the colors need to be stored in order to be shown by the LEDs.

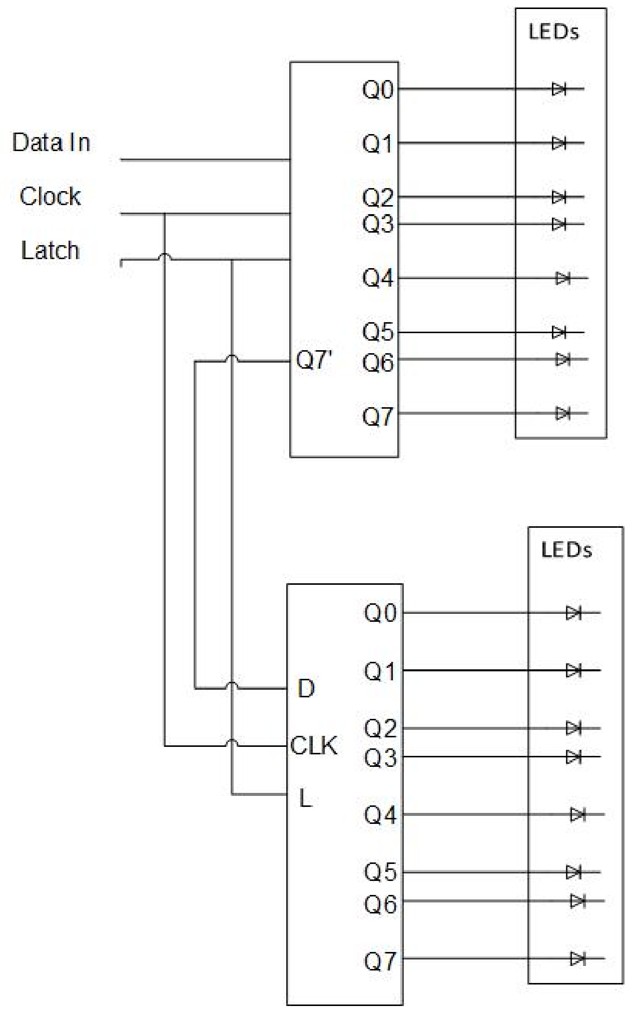


Figure 3.7 Two shift registers connected together connected to LEDs

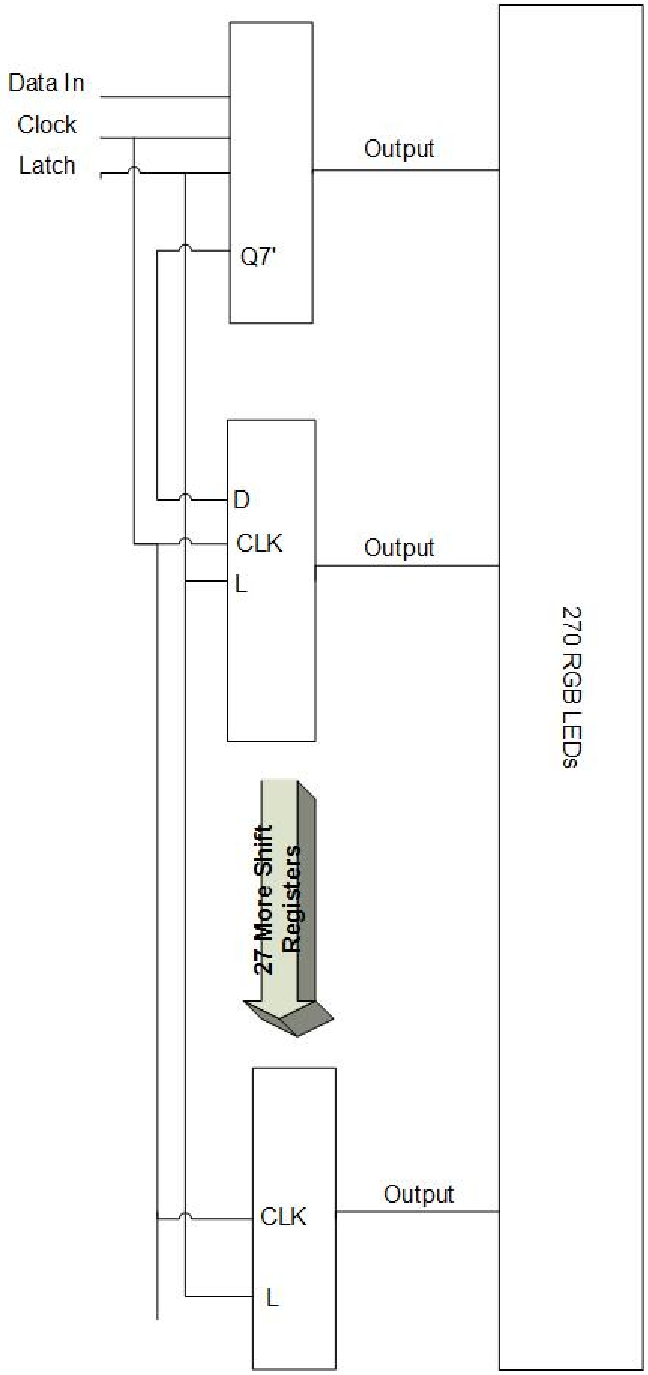


Figure 3.8 Example of How 270 LEDs will be connected with shift registers

3.8.1 Specifications

- 8-bit serial input
- 8-bit serial or parallel output
- Storage register with 3-state outputs
- Shift register with direct clear
- 100 MHz (typical) shift out frequency
- ESD protection:
- HBM JESD22-A114F exceeds 2000 V
- MM JESD22-A115-A exceeds 200 V
- Multiple package options
- Specified from -40 °C to +85 °C and from -40 °C to +125 °C

3.9 Programming Language

In order to make everything work in the way we want it to, there has to be some back-end design and coding that we must implement into this project. There are many programming languages to choose from and we should choose the best one to suit our needs. We need to build an application that has a user-friendly interface and that can allow simple serial communication. In order to make a user-friendly interface that can communicate between input and output we need to choose a programming language with an existing IDE that supports libraries that have able and supportive GUI development tools. We can consider C++, Java and assembly language as our candidates to choose from for our programming language. Below is Figure 3.8, which depicts the way we will try to develop our project using this lifestyle.

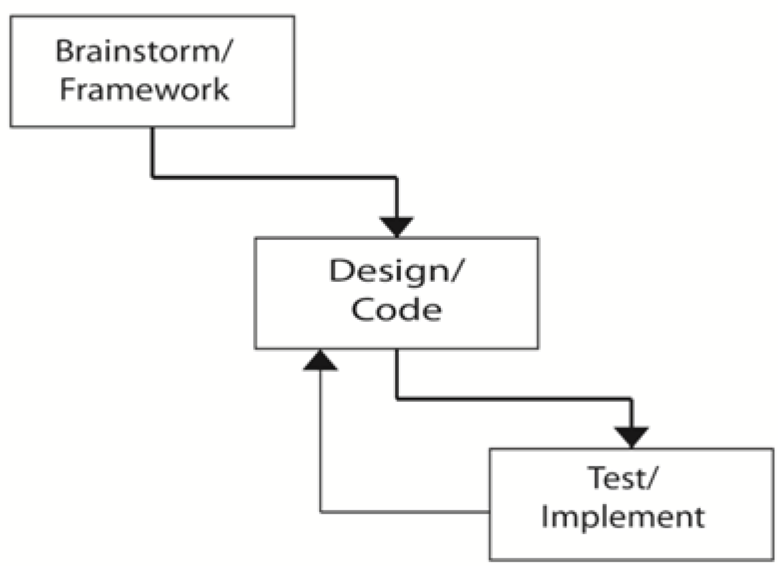


Figure 3.9 Software Design Lifecycle

The C++ programming language is the language that is most widely known at this moment. It is a language that is free-form, complied and general-purpose. C++ is compatible with C such is the language that a lot of us have been learning over the years here at the University of Central Florida. C++ supports C's syntax. C++ uses operators to perform arithmetic, logic, comparison and other operations. C++ supports memory management such as static or dynamic memory allocation. C++ is like the object-oriented version of C. The question is can we use C++ to build a user-friendly interface and program the LED globe to communicate between input and output? Through extensive research, we found that C++ is capable of building a user-friendly interface and also has access to GUI libraries for development. That being the case, there does not seem to be a visual GUI editor or IDE that can assist us in our endeavor even though C++ is such a powerful language.

The java programming language is an object-oriented programming language. It has a "write once, run anywhere" code, which means that the code can be ran on one platform and does not need to be recompiled to run on another platform. The Java programming language strives to be simple, robust, portable, high performance and dynamic throughout. Java's syntax is similar from C++. Java uses an automatic "garbage collector" to manage memory in the object lifestyle. Studies show that Java is slower than C++ in execution. Java is built exclusively as an object-oriented programming language. This means that all code is written inside a class and everything is considered an object with the exception of variables. Java can be used to develop mobile apps if needed. Java has a GUI visual editor as well as an extensive library in their IDE to help development. Java is one of our top candidates that can get the job done for us.

Assembly language is basically machine code or a low-level programming language. Our time with assembly language was not a very pleasant one but in development with the microcontrollers that we were using, it does give us the option of using assembly language instead of C programming if things go awry with the C++ programming. Assembly language can be used to program our microcontrollers but it does not seem there is a good way to build a user-friendly interface with assembly language.

3.10 Graphical User Interface

A graphical user interface is a possibility as far as our project goes. We can develop it for the PC and possibly for mobile devices if time allows. Development for the mobile device will allow us to send a text message to the LED globe and the LED globe should display it. This section will discuss how the graphical user interface will look as well as its functionality.

Identification of the requirements of the graphical user interface will help us build it more efficiently. Without identifying what makes a user interface effective, we constrict ourselves. We are going to be using the development cycle displayed below in Figure 3.10. Basically, we will first brainstorm what we want our graphical user interface to do. The brainstorming will provide a framework for our user interface. Next, we will design and code the user interface. Lastly, we will implement and test the user interface. Any trouble or troubleshooting will require us to go back into the design/code phase and back to testing and implementing our user interface until we are satisfied. The testing portion will be discussed in the subsequent section in the report.

One of the more common themes we want to use for our GUI is for the GUI to be user friendly and easily accessible to all ages, meaning easy to understand. The GUI should not require any technical knowledge about LEDs, centripetal force or any engineering or physics terms to operate. The GUI should not require training of any sort to operate. The GUI will be labeled accordingly with precise terminology that should accompany our project. The interface should look clean and easily maneuverable.

We want our GUI to have options that the user can choose. One of the options would be to have a singular color globe. Another option would be to display the actual Earth globe with land and sea green and blue respectively. We also want to display the time/date and an option to display our names and our group number. The last thing we want to include is a text box to display any text we wish to enter and display on the LED globe. If time allows and if we have the necessary programming knowledge and research, we would like to implement a game of pong on our LED globe. So that could be the final option on our GUI and it would open up another window to accept inputs for the game of pong being played on the LED globe.

We want to include a visual representation of the GUI so we have Figure 3.10 below to show the options that we want to set out to have in our POV LED Globe.

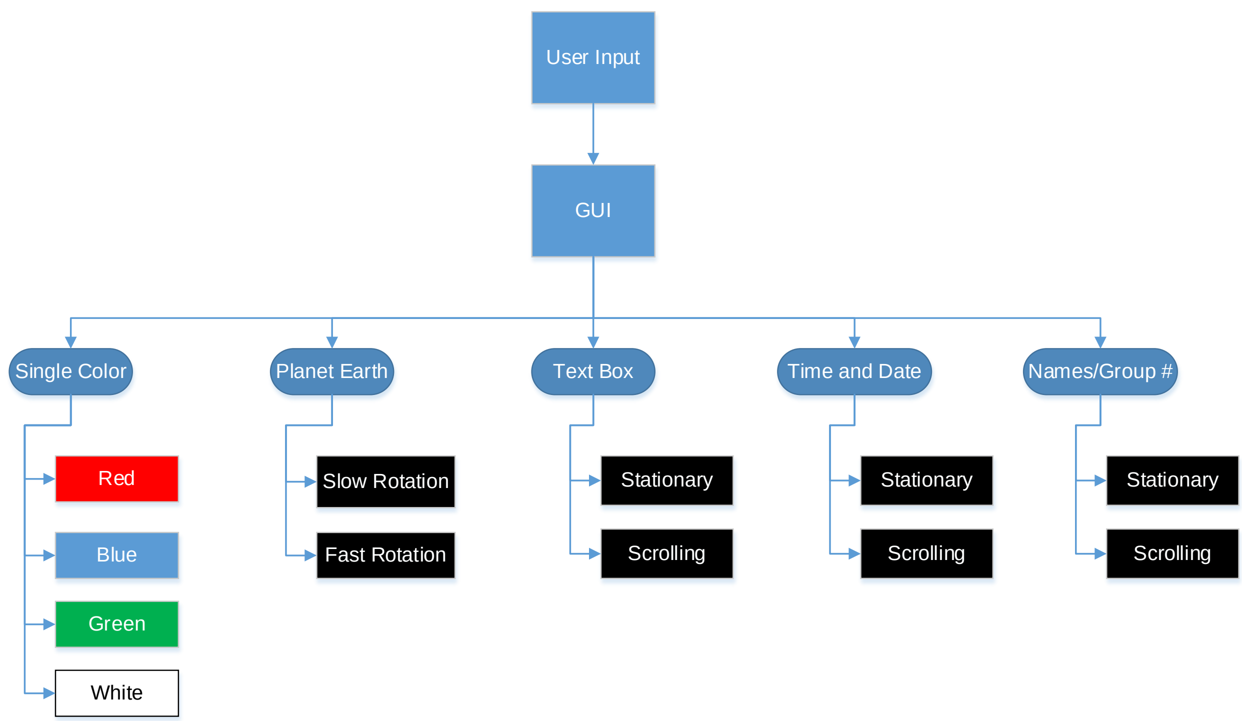


Figure 3.10 Visual Representation of GUI options

4 Project Hardware and Software Design Details

4.1 Initial Design Architectures

In the overall system, the initial architecture is been described along with the previous section of this documentation. The power supply will be in charge of regulating and converting the voltage coming from the outlet. Then, after the voltage has been adapted to the LED Ring requirements, it is going to be transmitted to the electric motor, which is going to engaged by a user interface and to the microcontrollers and shift, registers, and from here it will be finally transmitted to the 270 LED's.

Furthermore, along the first step of regulating the power and the last step of turning the LED's on, there are a lot of processes involved. Hardware and software described in this section will work together to be able to accomplish the projection of images, which is the main function of the LED globe. On the same way, the hardware will make possible the rotation of the LED ring, and software will be responsible for the timing between light sequences and configuration that will make displaying images possible.

4.2 Mechanical Design

4.2.1 LED Ring Structure

In order to display a POV globe, you must have a ring structure to hold the LEDs in place. Figure 4.1 shows the detailed design of the LED ring.

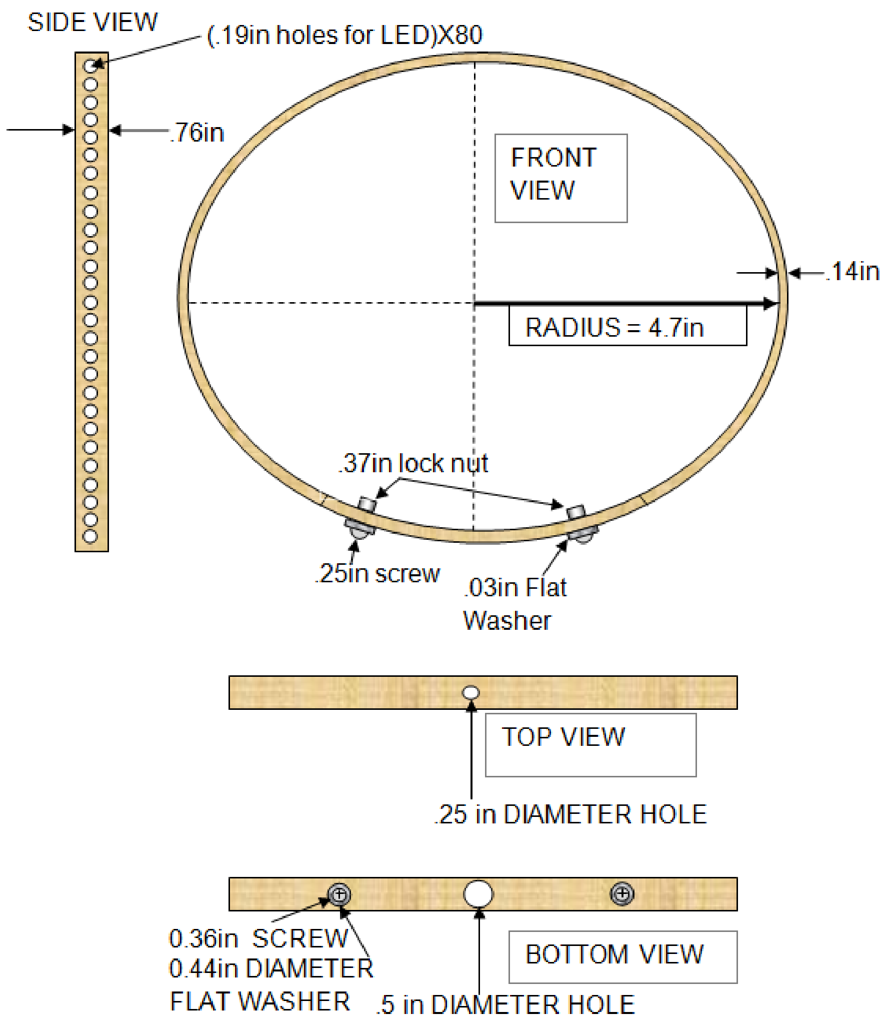


Figure 4.1 LED Ring Structure

4.2.2 Ring Support Shaft

The Ring support shaft is what will rotate the LED ring structure. The support shaft protrudes through the ring's center of mass; this helps with reducing rotational inertia which can help with the motor's efficiency. Two copper slip rings will be mounted in place with insulation underneath to prevent electrical shorting. Figure 4.2 shows the details of the ring support shaft.

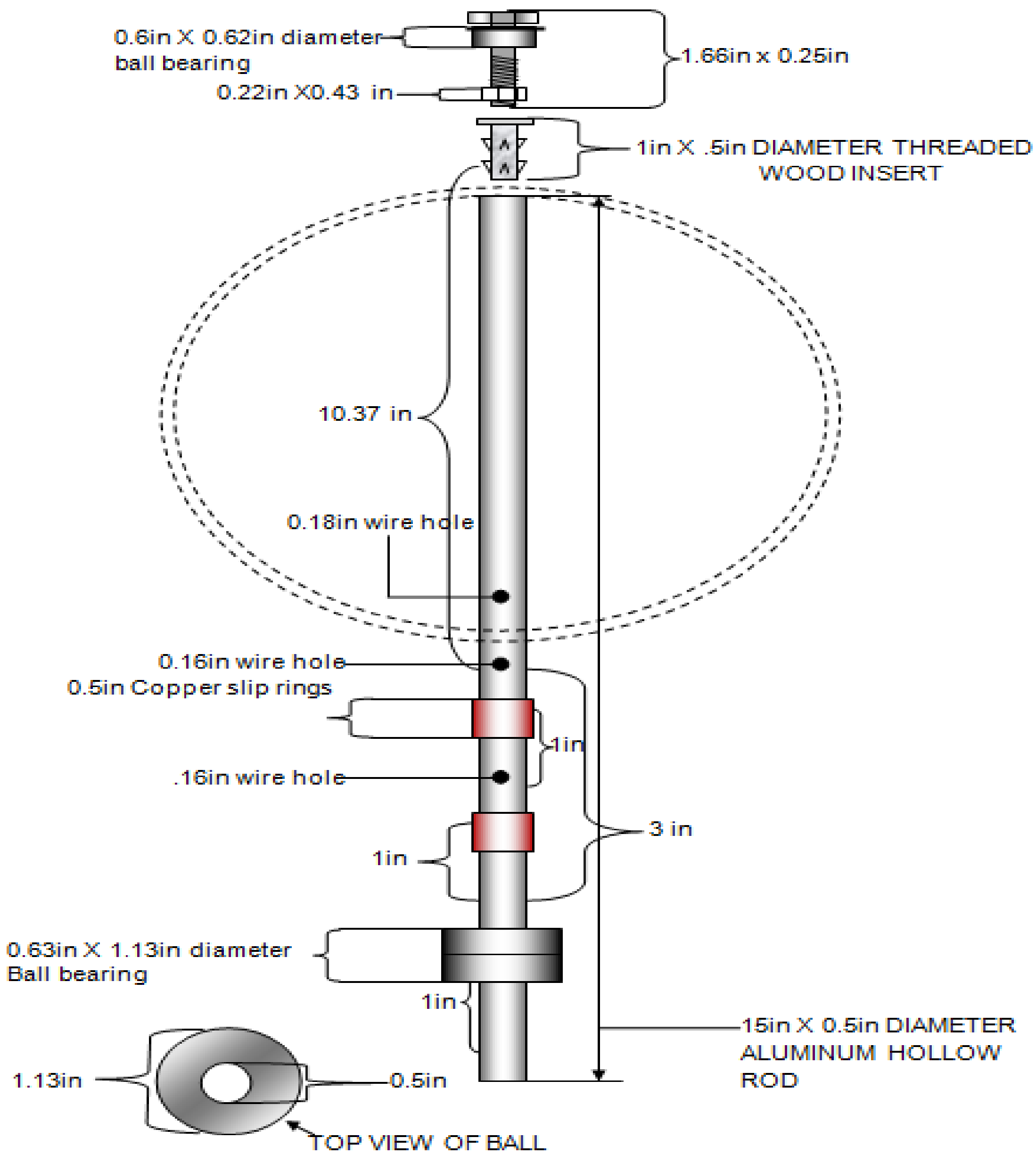
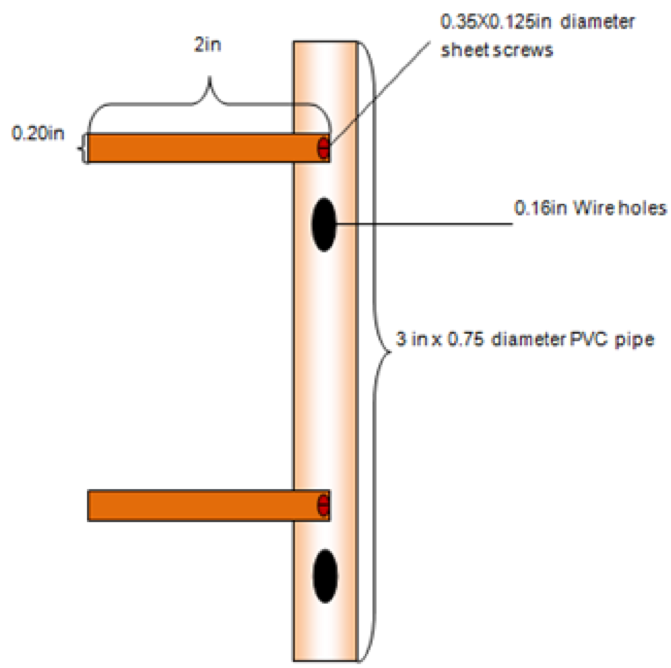


Figure 4.2 Ring Support Shaft

4.2.3 Slip Ring Voltage Terminals

The slip ring voltage terminal is how the microcontroller and the LEDs will receive power inside the ring structure while the ring support shaft rotates. It is made out of a 3inx0.125in PVC pipe with two copper 2inx0.2in copper strips. The top strip will be charged by a positive charge and a negative voltage will charge the bottom strip. The two strips will make contact with the copper slip rings on the ring support shaft. Figure 4.3 shows the design details.



. Figure 4.3 Slip Ring Voltage Terminals

4.2.4 Crucifix RING-Shaft Support

The crucifix ring shaft support is a piece of .6 inch diameter hollow aluminum tube with two perpendicular aluminum planes, to be fitted over the Ring support shaft and screwed onto the LED Ring. The purpose of the crucifix ring shaft support is to prevent the LED ring from spinning, due to angular momentum, after the motor stops. Figure 4.4 shows the design details.

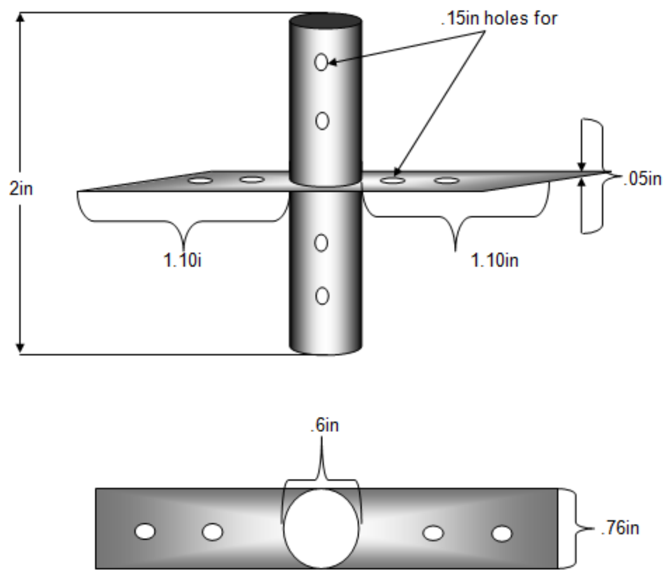


Figure 4.4 Crucifix RING-Shaft Support

4.2.5 Housing Unit Frame

The housing unit frame is designed to help with the plexiglass's structure and balance. It is made out of 12 6inx1inx0.08in aluminum bars, design shown in Figure 4.5.

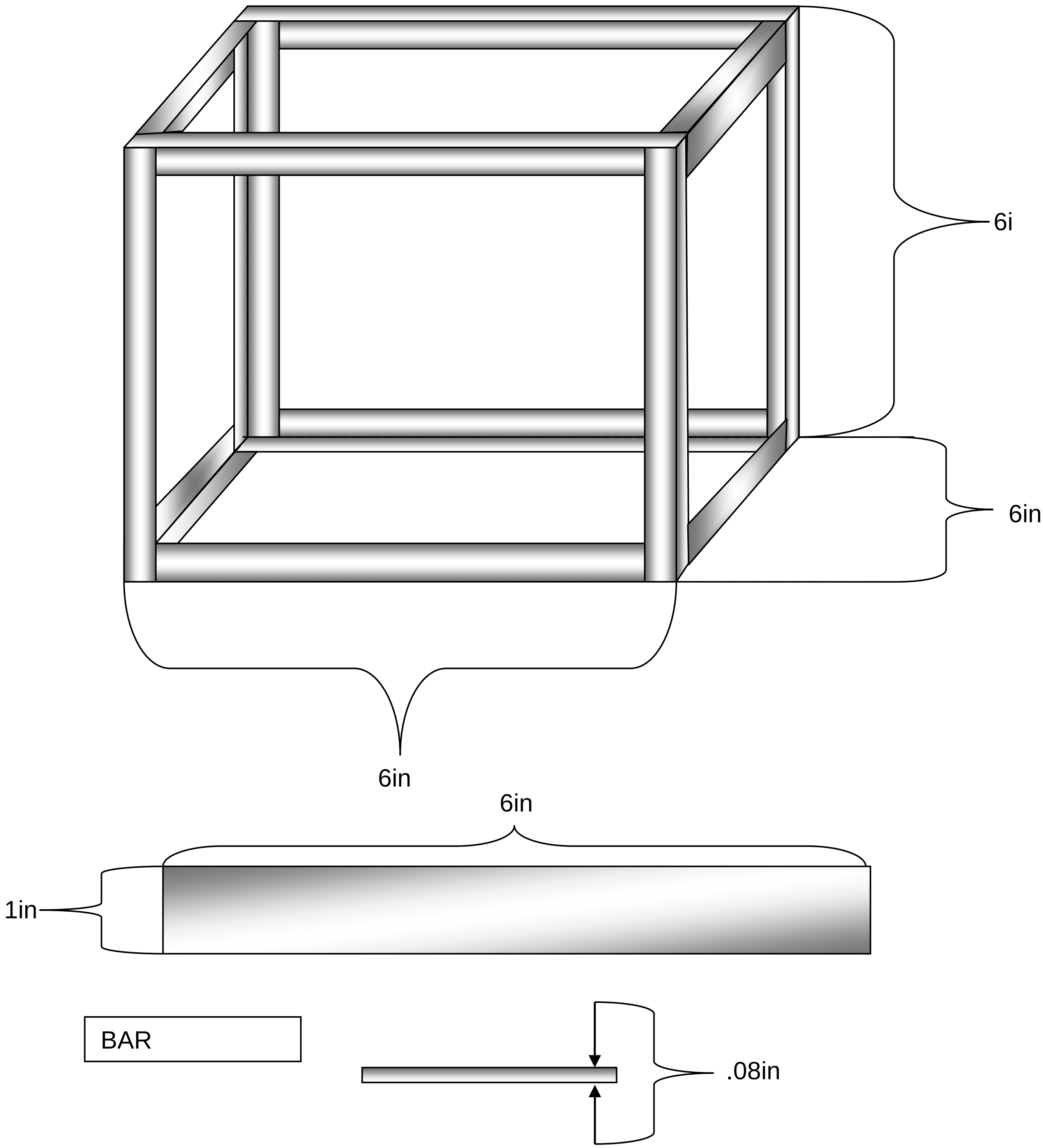


Figure 4.5 Housing Unit Frame

4.2.6 Plexiglass Encapsulation

The plexiglass encapsulation will house the motor, the power supply and it is the main structure for balance and support for the rest of the system. Figure 4.6, Figure 4.7 Figure 4.8 and Figure 4.9, show the top, back, bottom, inside, left and right view of the plexiglass encapsulation respectively.

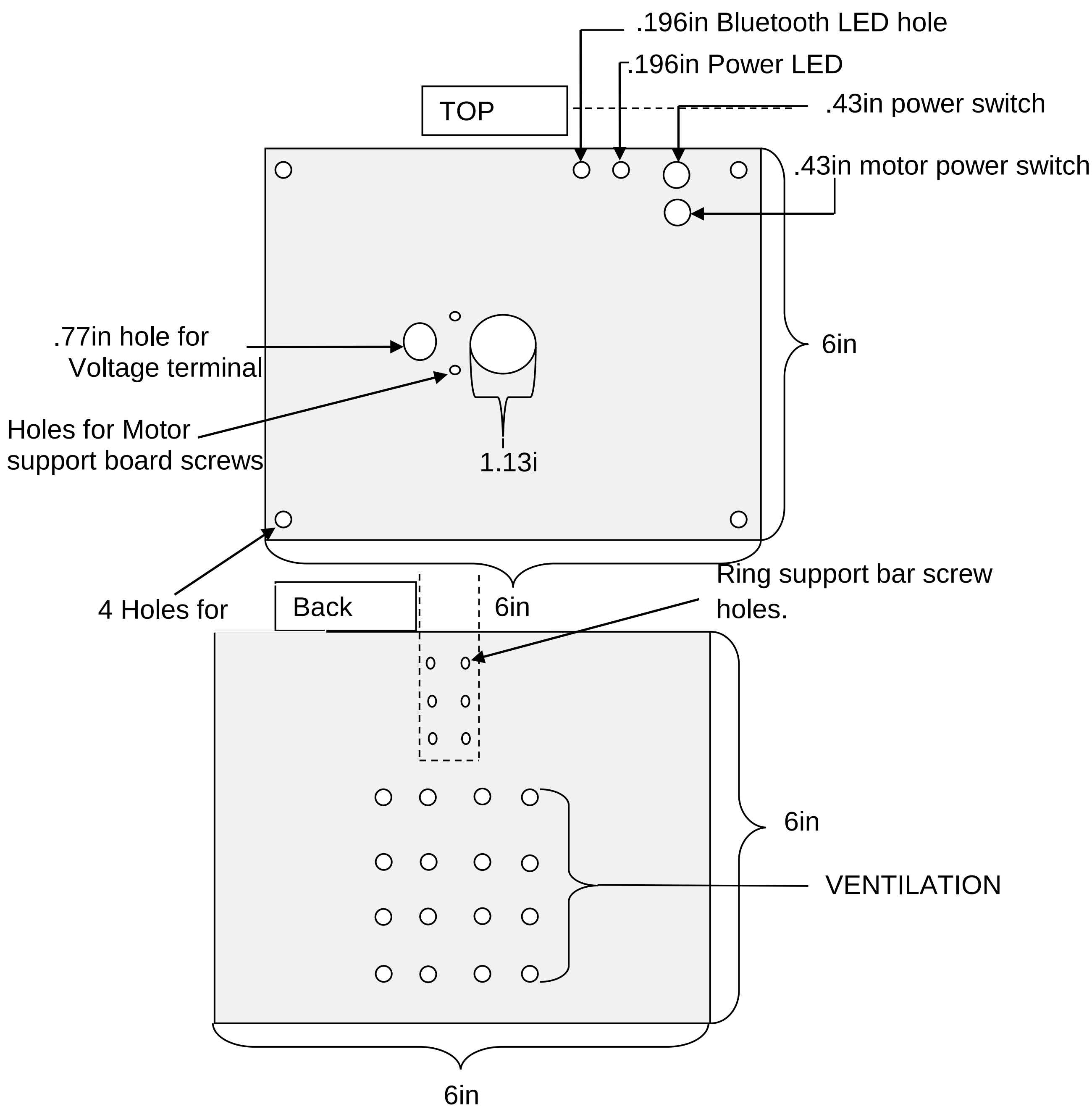


Figure 4.6 Plexiglass Encapsulation

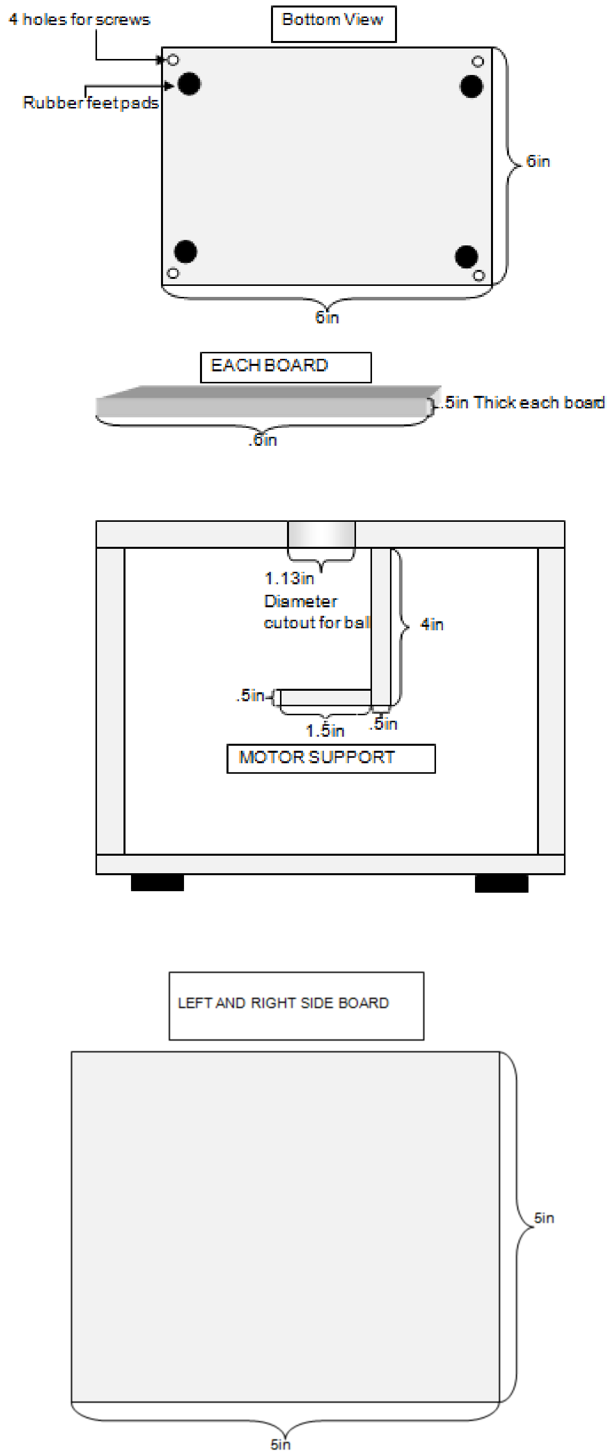


Figure 4.7 Plexiglass Encapsulation

4.2.7 PCB Mount

The PCB mount is the mount for the microcontroller, shift register and current-limiting resisters on the printed circuit board. The goal was to reduce any wind resistance and additional torque by mounting the PCB mount as close to the ring's center of axis as possible. An additional plate is mounted in the back and will be connected together with the front plate via four rods on each corner. Figure 4.8 shows the detailed design.

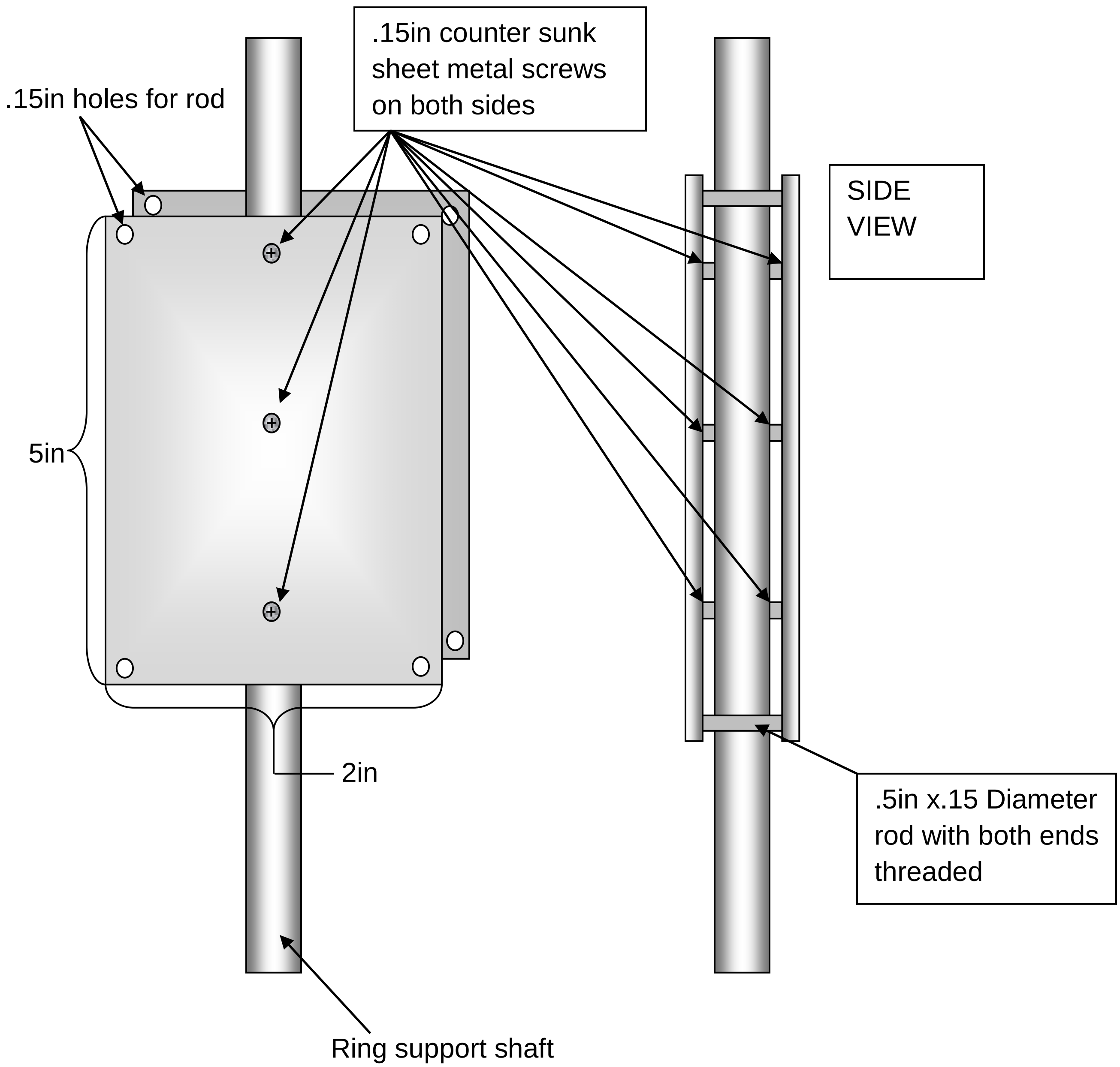


Figure 4.8 PCB Mount

4.2.8 Support Bar

The support bar will help support the LED ring shaft and will also help minimize any wobbling when the LED ring is spinning. Figure 4.9 shows the design details.

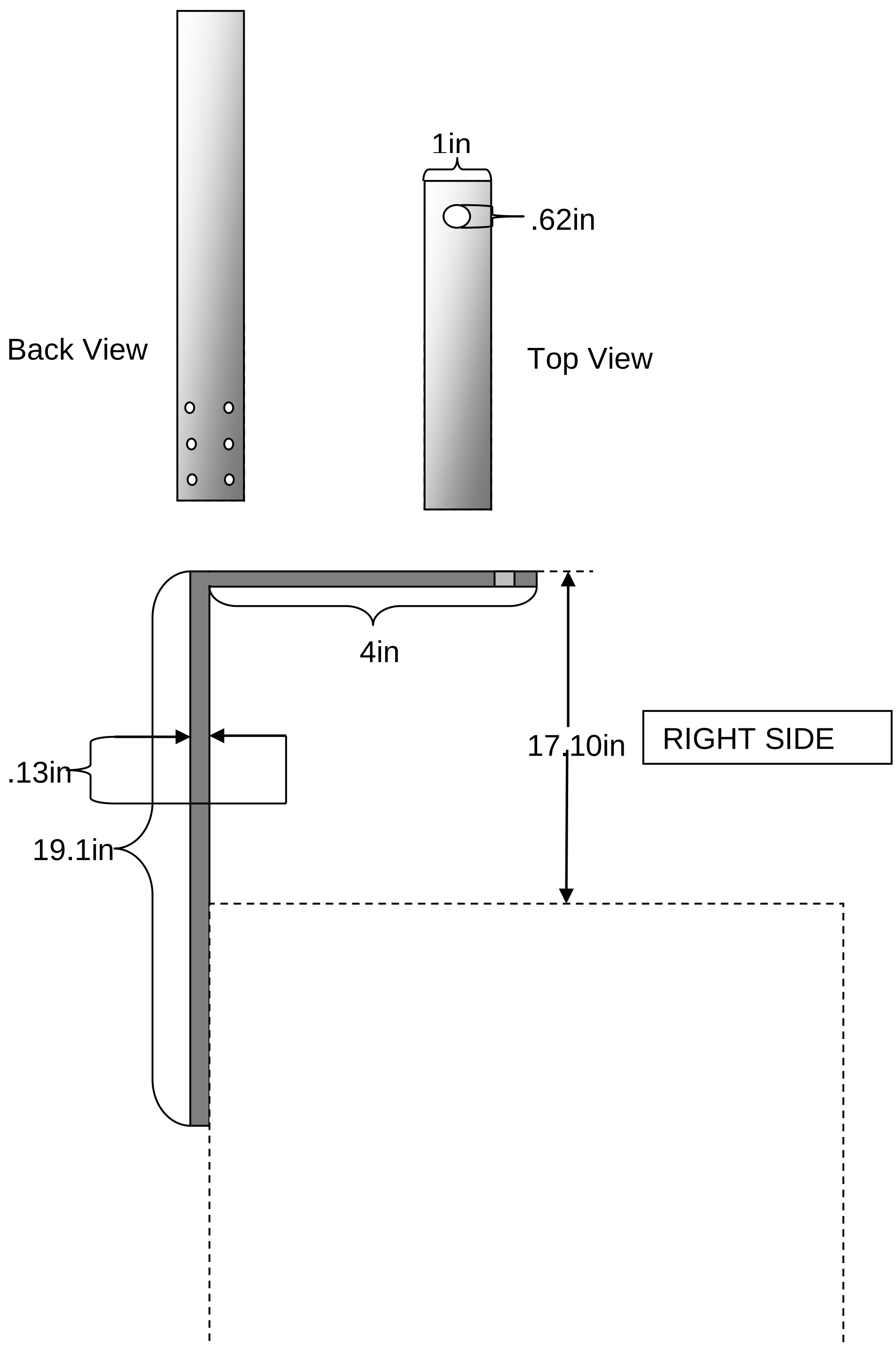


Figure 4.9 Support Bar

4.3 Power Supply Design

Figure 4.10 below shows a flowchart of our Power supply design.

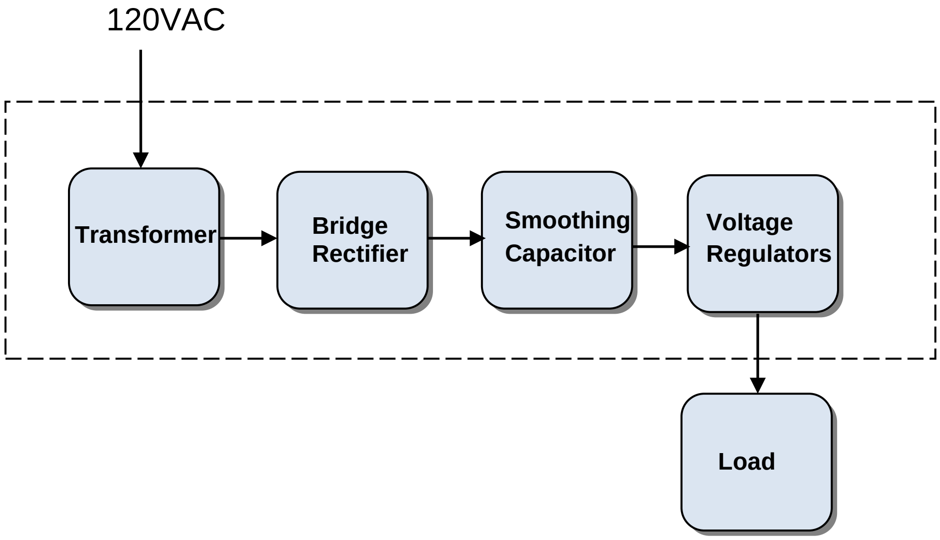


Figure 4.10 Power Supply Design Diagram

4.3.1 Transformer

The transformer we are going to be using is the TR100VA001US. This transformer is a 120 to 24 VAC, 100 VA, 50 or 60 Hz transformer with a circuit breaker included for over current protection and operates at temperature of -30 to 140°F. Below is the TR100VA001US in schematic circuit form depicted in Figure 4.11.

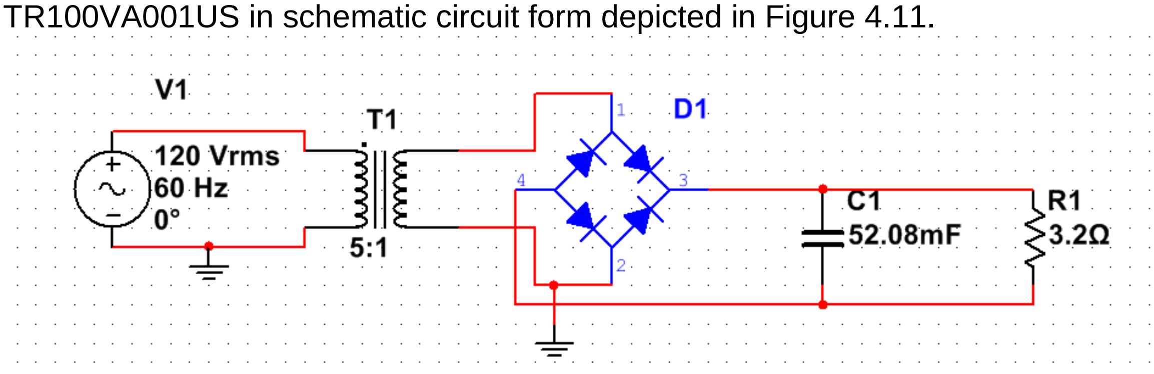


Figure 4.11 Transformer Circuit

4.3.2 Bridge rectifier

The bridge rectifier will be composed of four NTE5812HC diodes. This diode has a maximum recurrent peak reverse voltage of 100 V, maximum RMS voltage of 70 V, maximum DC blocking voltage of 100 V and of ten amps and can operate at temperatures of -55° to 125°C. More Specifications are depicted in Table 4.1.

Maximum recurrent peak reverse voltage	Maximum RMS voltage	Maximum DC blocking voltage	Forward average current	Operating Junction Temperature Range Celsius	Forward Voltage Drop
100V	70V	100V	10A	-55 to 125	1V

Table 4.1 Bridge Rectifier Specifications

Each diode is capable of handling the power we need:
$P_{Diode} = Vf * Imax = 1 * 8A = 8watts.$ See Figure 4.12 for the Bridge Rectifier circuit.

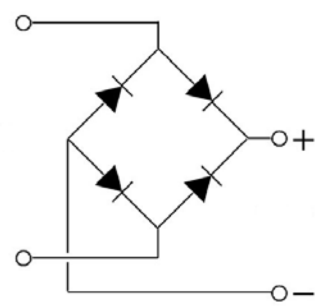


Figure 4.12 Bridge Rectifier

4.3.3 Smoothing Capacitor

In order to calculate the correct smoothing capacitor, we must determine the peak voltage before and after the bridge rectifier. The voltage on the secondary winding of the transformer is: Vs= 24Vrms. Its maximum value will be 24$*\sqrt{2}$ =33.9V or Vs(max) = 34V. In order to calculate the maximum output voltage after the bridge rectifier, we must take the maximum input voltage before the bridge rectifier and subtract this by the forward voltage drop, 'Vf', of the two conducting diodes, therefore we have: Vo(max) = Vs(max) $- 2 * \mathrm{Vf} = 34 - 2 = 32\mathrm{V(max)}$ where Vf=1V for each diode. In order to determine the kind of capacitor value we need, we can use the following equation: $\mathrm{C} = \frac{\mathrm{VO(max)}}{2\mathrm{f}*\mathrm{R}*\mathrm{Vr}}$ where Vr is the ripple voltage, R is the load resistance and f is the frequency in Hz.

The output of the bridge rectifier will have an output ripple voltage of twice the input frequency due to the diodes inverting the negative voltage to the positive side. So the input frequency as seen by the capacitor will be 120Hz. The output voltage and current can be used to solve for R. Since the peak output voltage is 32V and a total of 10 amps will be drawn out, then R= $\frac{32}{10} = 3.2\Omega$ and let's have 5% ripple, so Vr is .005*32V = 1.6V and f = 60Hz. So we have C= $\frac{32}{2f*3.2*1.6}$ = 52.08mf. The capacitor will be two Cornell Dubilier® CGS263U075V4C Aluminum Electrolytic Capacitors - Screw Terminal, 75volts, 26mF in parallel.

4.3.4 Motor Voltage Regulator

The motor voltage regulator will be based on the LM25117. The LM25117 is a high voltage synchronous buck controller for step-down regulator applications. The input voltage has a wide range from 4.5V to 42V. The pulse-width modulation circuit embedded inside the IC, is programmable from 50kHz to 750kHz. The LM25117 can drive external high-side and low-side NMOS power switches with adaptive dead-time control. The LM25117 also has thermal shutdown capability in case of overheating which can destroy this delicate IC.
The following Table 4.2 lists all of our known values and design parameters.

Output voltage	Vout =9V
Full load current	Iout = 9A
Minimum Input Voltage	Vin(min) = 14V
Maximum input voltage	Vin(max) = 32V
Switching Frequency	fsw = 230KHz
Diode Emulation	Yes
External VCC Supply	No
Cycle by Cycle Sense Voltage Threshold	Vcs(TH) = .120V
Minimum HO On-time	$t_{ON(MIN)}$ = 100ns
Ramp Capacitor	Cramp = 820pf
Current sensor amplifier gain	A_s = 10
UVLO hysteresis voltage	V_{HYS}=1.1V
Regulator startup voltage	$V_{IN(STARTUP)} = 12.07V$
Soft-start time	t_{ss}=1.2ms
Restart time	t_{res}=58.75ms

Table 4.2 Motor Voltage Regulator Specifications

First we must determine the timing resistor for the switching frequency by using the following equation:

$$R_T = \frac{5.2X10^{\wedge}9}{fsw} - 948\Omega = 21.7K\Omega$$

Next we determine the output-inductor L_o:

$$L_o = \frac{Vout}{\text{Iout*fsw}} * \left(1 - \frac{Vout}{Vin(max)}\right) H = \frac{9V}{10A *.413*230\text{kHz}} * \left(1 - \frac{9}{32}\right) = 6.8\mu H$$

Now we need to determine the inductor peak-to-peak ripple current with the following equation:

$Ipp(max) = \frac{Vout}{Lo*fsw} * (1 - \frac{Vout}{Vin(max)})$ A = $\frac{9v}{6.8\mu H*230KHz} * (1 - \frac{9v}{32})$ = 4.14A

$Ipp(min) = \frac{Vout}{Lo*fsw} * (1 - \frac{Vout}{Vin(min)})$ A = $\frac{9v}{6.8\mu H*230KHz} * (1 - \frac{9v}{14})$ = 2.1A

So Δ Ipp =2.04A

Next we determine the current sense resistor R_s with the following equation:

$R_s = \frac{V_{cs(TH)}}{Iout(max)' + \frac{Vout*K}{fsw*Lo} - Ipp/2}\ \Omega \ = \frac{.12}{10*1.5 + \frac{9*.555}{230*10^3*6.8*10^{-6}} - 2.1/2} = 7m\Omega$, K is the damping factor and $Iout(max)$' is the max output current with a 150% higher value to account for tolerances and ripple current.

Now we have to determine the maximum power dissipation of Rs:

$\mathrm{P_{Rs}} = \left(1 - \frac{\mathrm{vout}}{\mathrm{Vin(max)}}\right) * \mathrm{Iout}^2 * \mathrm{Rs} \ = \left(1 - \frac{9}{32}\right) * 10^2 * .007$ =.5W

Now we calculate the worst case peak inductor current under the output short condition:

$\mathrm{I_{LIM\_PK}} = \frac{\mathrm{Vcs(TH)}}{\mathrm{Rs}} + \frac{\mathrm{Vin(max)*t_{on(min)}}}{\mathrm{Lo}} = \frac{.12}{.007} + \frac{32*100*10\hat{}(-9)}{6.8*10\hat{}(-6)}$ = 17.61A

Now we determine the ramp resistor:

$\mathrm{R_{ramp}} = \frac{\mathrm{Lo}}{\mathrm{K*Cramp*Rs*As}} = \frac{6.8\mu\mathrm{H}}{1*820\mathrm{pf}*.007*10}$= 118.46KΩ

Now we determine the UVLO $R_{UV1}, R_{UV2}\ and\ C_{FT}$ which will determine the desired startup voltage and the hysteresis by setting up a voltage divider with the resistors and the capacitor provides the filtering:

$R_{UV2} = \frac{V_{HYS}}{20\mu A}\Omega$ and $R_{UV1} = \frac{1.25V*R_{UV2}}{V_{IN(STARTUP)-1.25V}}\Omega$,∴ $R_{UV2} = \frac{1.1V}{20\mu A} = 55K\Omega$,

$R_{UV1} = \frac{1.25V * 55{,}000}{12.07 - 1.25} = 6.34K\Omega$

Next we find capacitor C_{VCC}. According to the data sheet for the LM25117, C_{VCC} should be no smaller than .47µf. So we chose .47µf.

Now we find the output capacitor C_o. According to the data sheet for the LM25117, C_o can be about 100µf.

Next we must find the input capacitor C_{in}.We want to have a good capacitor that will filter out the ripple voltage at the Vin pin while supplying most of the switch current during the on-time. Two 10µf ceramic capacitor will be sufficient for this purpose.

Now let's find the soft-start capacitor C_{ss} by using the following equation:

$$C_{ss} = \frac{t_{ss} * 10\mu A}{.8V} = 15nf$$

A similar equation is used to find the restart capacitor C_{res}.

$$C_{res} = \frac{t_{res} * 10\mu A}{1.25V} = 470nf$$

Now let's calculate the output voltage divider resistors R_{FB2} and R_{FB1} with the following equation: $\frac{R_{FB2}}{R_{FB1}} = (\frac{Vout}{.8V} - 1) \ = \ (\frac{9}{.8} - 1)$ = 10.25. Let's set $R_{FB1} = 1.18K\Omega$ then $R_{FB2} =$ $12.1K\Omega$.

Next we need to find the loop compensation components C_{comp} and R_{comp}. These components can be found using the following equations:

$$R_{comp} = 2\pi * R_s * A_s * C_o * R_{FB2} * f_{cross}, \text{ where } f_{cross} = \frac{fsw}{10}$$

$$R_{comp} = 2\pi * .007 * 10 * 100 * 10^{-6} * 12.1 * 10^3 * 23 * 10^3 = 12.24\text{K}\Omega$$

$$C_{comp} = \frac{R_{load} * C_{out}}{R_{comp}} = \frac{\frac{9}{10} * 100 * 10^{-6}}{12.24 * 10^3} = 7.353\text{nf}$$

Figure 4.13 shows the schematic diagram of the Motor Voltage Regulator:

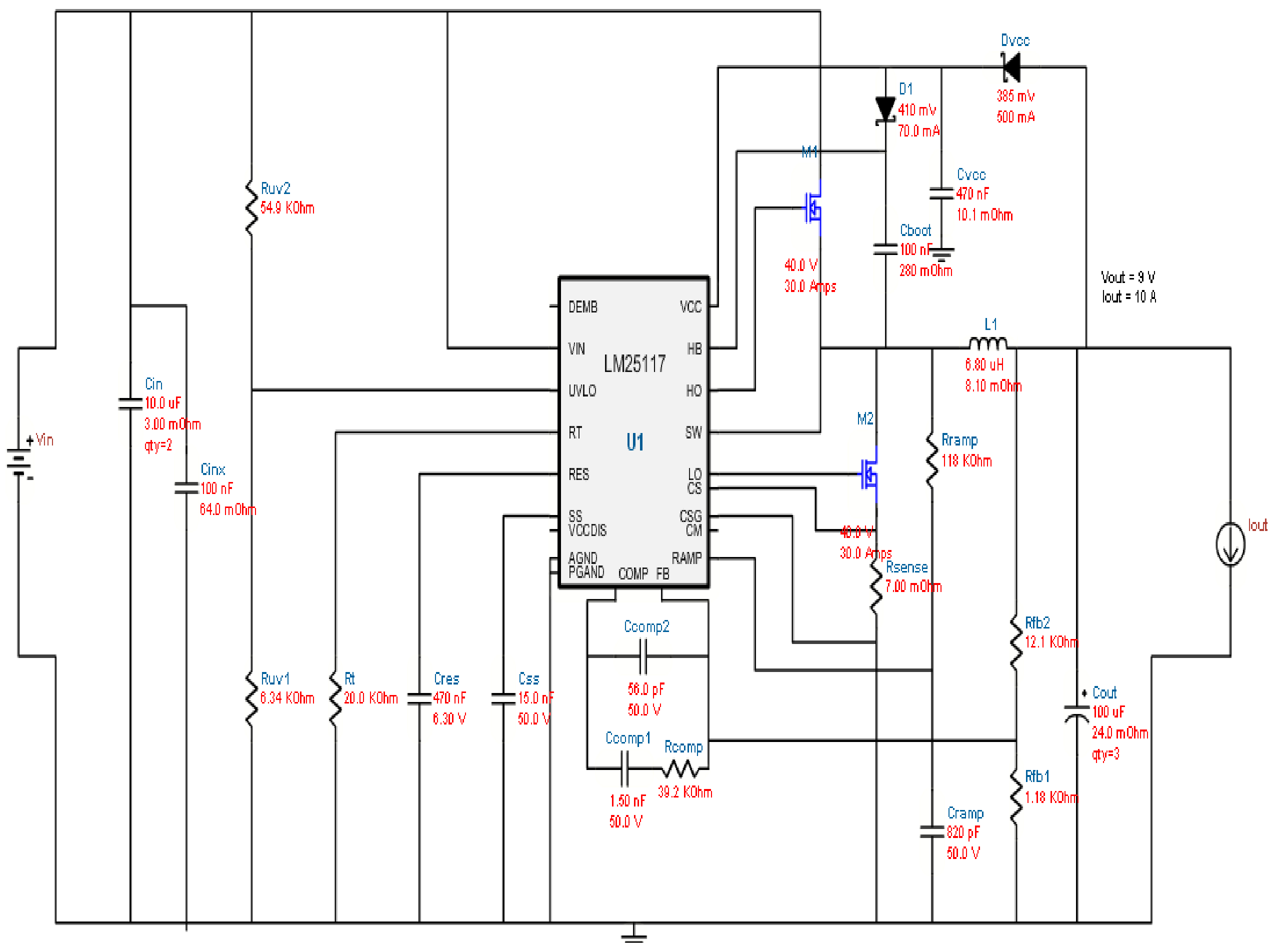


Figure 4.13 schematic diagram of the Motor Voltage Regulator

4.3.5 Microcontroller Voltage Regulator

The microcontroller voltage regulator is based off of the LM7809 voltage regulator. C2 is the bypass capacitor, which is needed to stabilize the regulator to prevent oscillation and improve transient response. C3 is a coupling capacitor to help suppress any noise coming from the power supply. D1 is a 1N4934 fast recovery diode with a current rating of one amp. This diode protects the regulator for the following reason: The input smoothing capacitor has a very large value, which will cause a temporary short-circuit when it completely discharges. After the power supply is turned off and there is a higher potential at the output of the voltage regulator versus the input, say due to C2 > C3, then the energy stored in the output is greater than the energy stored in the input which can destroy the voltage regulator. Therefore adding a diode across the input and output terminals of the voltage regulator will allow the current to flow through the diode, bypassing the regulator and therefore protecting the delicate components inside the regulator. See Figure 4.14 for the circuit with the LM7809CT in it.

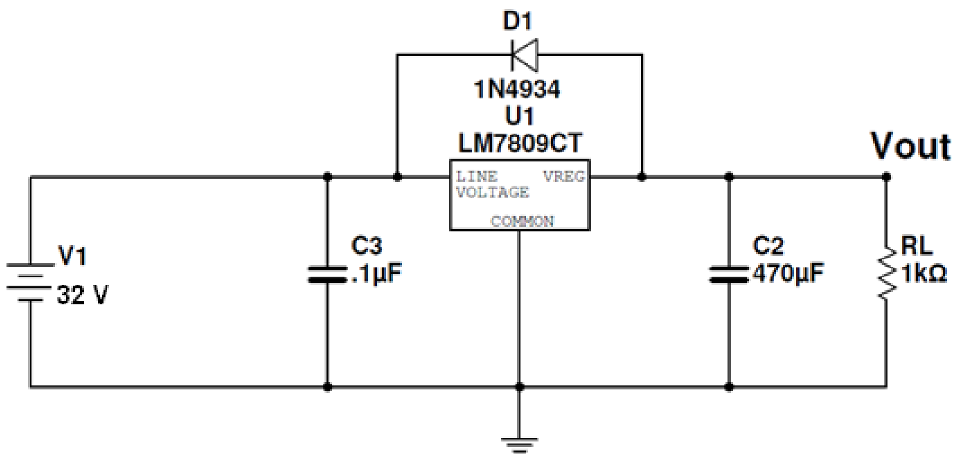


Figure 4.14 Microcontroller Voltage regulator schematic

4.4 PCB Design

One of the most important issues that come into constructing our LED Globe will be the printed circuit board design. Without this important part, our project will never come into fruition. One of the reasons why printed circuit boards are necessary in the design and implementation of an engineering project is because of the ease of back-end design and programming aspect of engineering. A printed circuit board will streamline our needs to have a microcontroller that will suit our needs. In other words, it is a customizable and programmable circuit that can fulfill what we are looking to accomplish. We can have resistors, capacitors, wires, and other electronic circuit parts soldered by surface mount technology directly to the PCB for us. Without surface mounting technology, the process would take an extreme amount of pain-staking time and energy because we would have to manually connect every single piece of the circuit by soldering each individual piece to the other. Printed circuit boards are assembled step by step using computer software applications that make it easy to make the schematic and then the PCB design and will even order the parts for you and solder it together.

For our project, we will be relying on the CadSoft Eagle PCB design software to assemble our printed circuit boards. This software can be download at the website [4]. Using this software, we can make a schematic, immediately translate the schematic into a board design, configure the board design to our needs and then order the PCB directly with the software. Just with those features alone, the CadSoft Eagle PCB design software is perfectly suited for our needs. The procedure we used to design a new printed circuit board using CadSoft's Eagle PCB design software for our LED Globe is as follows:

1. Create a new project to store all of our project files in.
2. Look up libraries with the parts we want to use in our schematic. A library contains an extensive list of parts for users to choose from so there is room for endless customization to make your own unique PCB. Remember to add the libraries you found into the library directory in order to use it.
3. Create your schematic, name it any name you want.
4. Start adding your parts to the schematic, making to add names and values to your parts.
5. Connect your parts with the connector tool and make sure you have a power source and a ground.
6. Once your schematic is finished, there is a tool that generates a board design for you from your existing schematic. Simply click that tool and you will have a board design loaded with all you parts that you connected in your schematic.
7. Simply edit your board design by placing the parts in the positions of your choosing. Define the size of the board and your net class. Net classes help with auto routing your wiring of the board.
8. The Eagle software has an auto-router tool at its disposal. You can edit how it will define the routes by choosing which layers you want the tool to reference. Connect your circuit via auto-router tool.
9. Make sure to define and connect your ground ports and vias. Vias are electrical connections to ground. Place vias in various locations.
10. Make any other changes (reroutes, ground ports, debug, moving the silk screens) before finalizing for production of our PCB.
11. Do a design rule check. Design rule check is basically the Eagle software's way of checking through and making sure the board is manufacture-able.
12. Now we are ready to get a quote or generate Gerber files. These are certain tools to help with the manufacturing and ordering of the Printed Circuit Board.

There is a standard that is referred to when considering the design of printed circuit boards. It is called the IPC-2221A document and the IPC stands for the Institute for Interconnecting and Packaging Electronic Circuits [5]. They are the people who control most aspects of printed circuit board design, testing and manufacturing. The document described above is titled "Generic Standard on Printed Design." The document entails parameters that we must follow. Concerns about hole spacing, hole size, pad sizes, spacing, trace width and most importantly board size are just some of the characteristics covered in the IPC-2221A. Standard board spacing for routing could go up to 2 inches; some space for the actual routing and additional spacing on the border for processing. Next is the track width. The width of the track affects current resistance and inductance somewhat proportionately. A larger track width will have low direct current resistance and a small inductance and a smaller track width will have opposite of the larger track width. The minimum limit of the track width depends on the manufacturer because the manufacturer determines the track resolution that is in the manufacturer's reach of producing. The size of the board also has small resistance that must be accounted for.

The pads are the portion of the PCB that used in conjunction for the surface mounting electrical components. Pads come in all sizes, shapes and dimensions. Pads have something called a pad to hole ratio. What this means is there is a general rule of thumb when it comes to come holes in your pad. Usually, the pad on the PCB should be 1.8 times the diameter of the hole. Another thing to take notice and keep in mind is the trace width and spacing. The trace width depends on the maximum temperature increase of the current and the impedance tolerance. Trace spacing tells the designer how to layout the trace width and the spacing between holes. There are many things to consider when designing a printed circuit board but luckily the software, CadSoft Eagle, we are using actually streamlines the process considerably.

4.4.1 Design Equations for Printed Circuit Boards

There are few things that we must calculate before we look into designing our printed circuit board. Conductor Resistance, Conductor Capacitance and the Characteristic Impedance are some of the important calculations that must be done before we design our circuit board.

The first characteristic we will talk about is Conductor Capacitance. It will tell how much electrical energy stored for a given potential. Finding the capacitance is easy one you have the thickness of the conductor, width and distance between conductors.

$$C = Q/V$$

$$C = \frac{2\pi L}{2ln(2h/b)}$$

d = distance between conductors (inches)

L = conductor width (inches)

b = substrate dielectric constant

h = thickness of the conductor (inches)

The second characteristic we will talk about is Conductor Resistance. Conductor Resistance is affect by the thickness of a wire, wire length, wire temperature and the conductivity of the base material being used. The thickness of the wire is basically the cross sectional area of the substance being fabricated. Area is material length times material width. In order to determine conductor resistance, only conductor width is taken into account.

The third characteristic we will talk about is characteristic impedance. The characteristic impedance of an electric component is the ratio of amplitudes of voltage and current waves moving along an infinitely long line. Characteristic impedance is basically resistance due to having the same unit measurements (ohms). For a printed circuit board, this is the formula used below to calculate the characteristic impedance of an infinitely long line.

Zo = $\sqrt{\frac{Lo}{Co}}$

Co = capacitance (F)

Lo = inductance (H)

Zo = impedance (Ω)

The microstrip impedance is the last characteristic that we will talk about. A microstrip is an electric medium that can be made using a printed circuit board. A microstrip is made up 4 components: conductor, upper dielectric, dielectric substrate and the ground plane. Illustrated below is the method to calculate the impedance of a microstrip line.

$$Z_{\text{microstrip}} = \frac{Z_0}{2\pi\sqrt{2(1+\varepsilon_r)}}\ln\left(1+\frac{4h}{w_{\text{eff}}}\left(\frac{14+\frac{8}{\varepsilon_r}}{11}\frac{4h}{w_{\text{eff}}}+\sqrt{\left(\frac{14+\frac{8}{\varepsilon_r}}{11}\frac{4h}{w_{\text{eff}}}\right)^2+\pi^2\frac{1+\frac{1}{\varepsilon_r}}{2}}\right)\right)$$

h = dielectric thickness

W = microstrip width

Er = substrates dielectric constant

Here is the PCB design for the microcontroller voltage regulator. We used element14pcb.com to make this particular design. Please see Figure 4.14.1

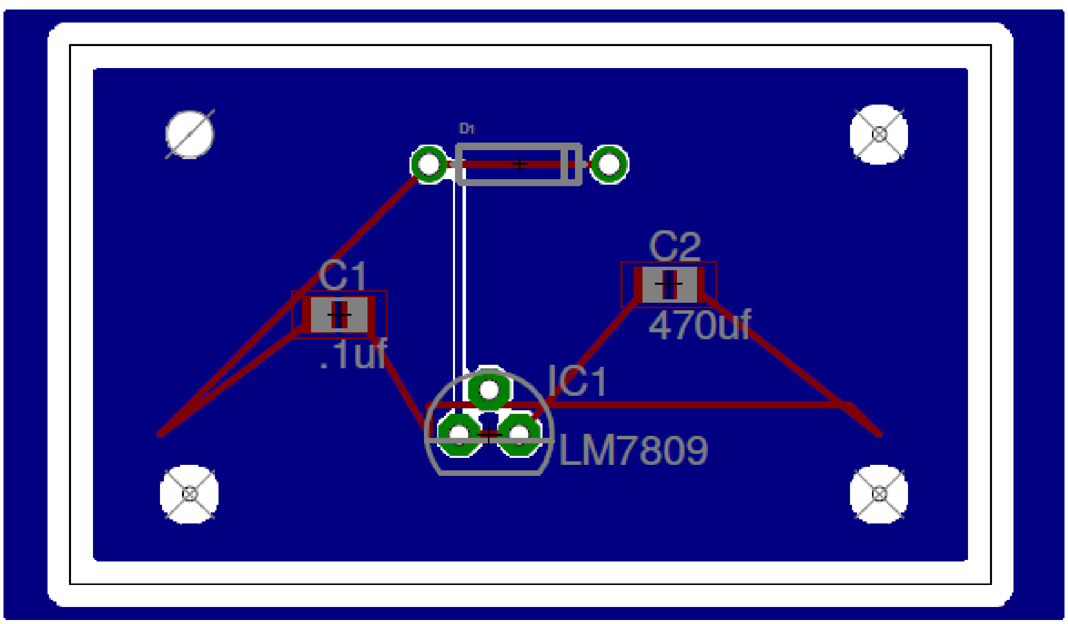


Figure 4.14.1 PCB Layout for Microcontroller Voltage Regulator

Here is the PCB design for the motor voltage regulator. We used element14pcb.com to design this part. Please see Figure 4.14.2

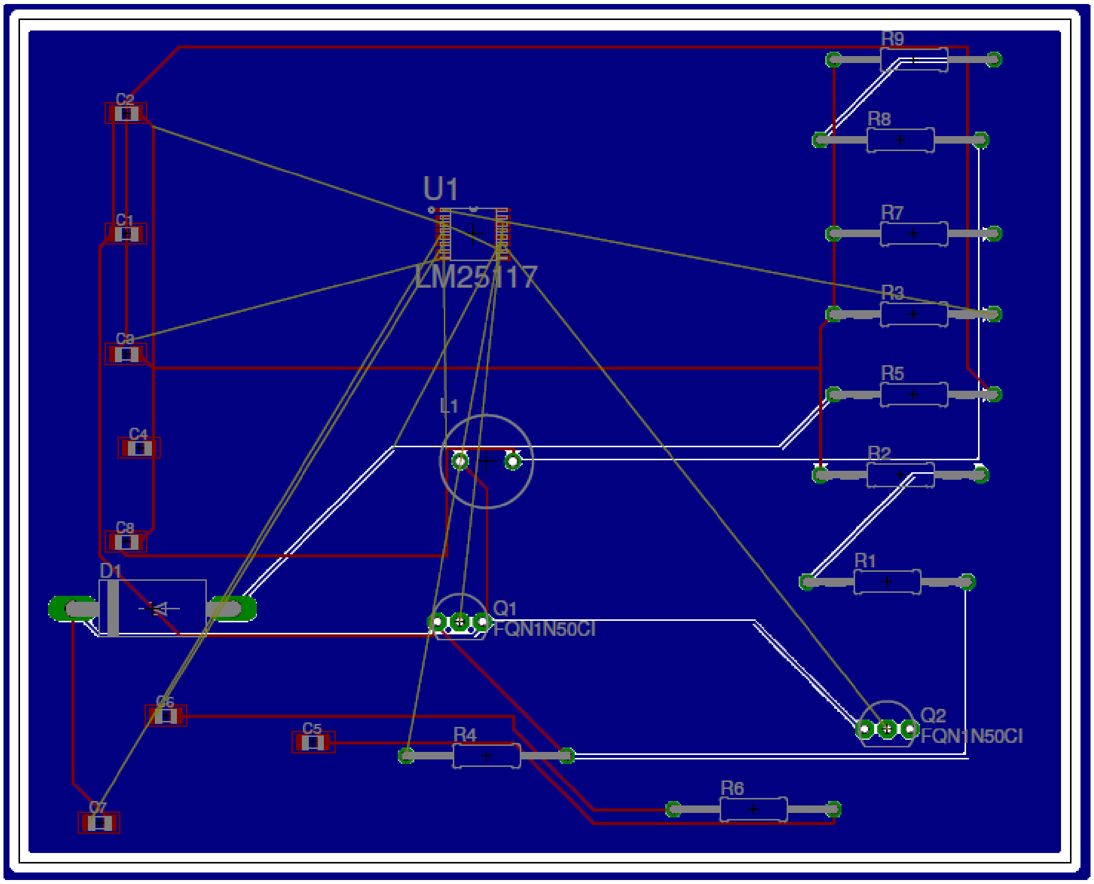


Figure 4.14.2 PCB Layout for motor voltage regulator

Here is the PCB design for the main power supply. We used element14pcb.com to design this part. Please see Figure 4.14.3

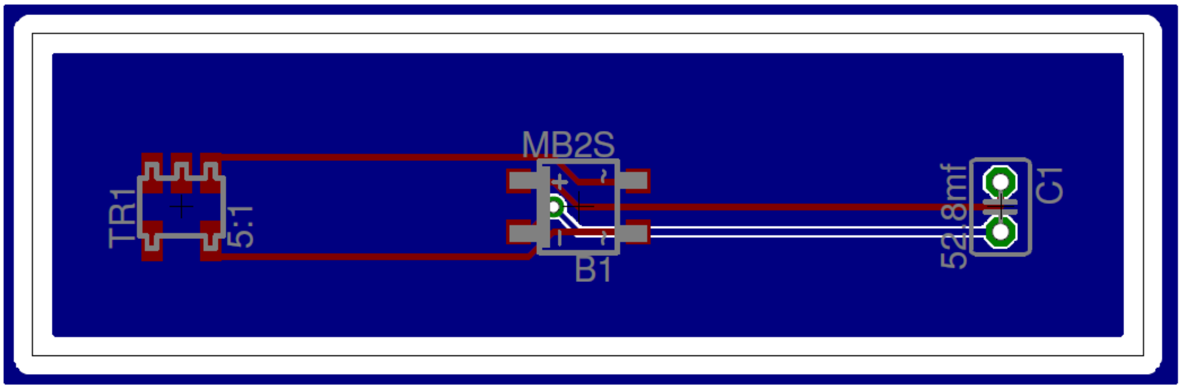


Figure 4.14.3 PCB Layout for main power supply

4.5 System Design

4.5.1 Microcontroller Unit

The Atmel ATMEGA328P-PU microcontroller would be used to control the light sequence of the LED Globe; this microcontroller will be pre-loaded with the preference of Arduino Uno boot loader. The Atmel ATMEGA328P-PU counts all the adequate hardware and software capabilities to meet all the design goals for this project, providing enough pins to handle all the outputs, in other words, the 90 LED's the LED Globe has. Moreover, the Atmel ATMEGA328P-PU has serial communication capabilities that make possible controlling the microcontroller via Bluetooth. Finally, another advantage of the Atmel ATMEGA328P-PU is to have 6 digital pins that provides pulse with modulation (PWM) and this capability can help us to control the motor speed if is needed. A list of the specifications for the Atmel ATMEGA328P-PU is given below:

- 16-bit microcontroller
- 16 MHZ clock speed
- 32 KB flash memory
- 2 KB SRAM
- Two pins UART
- 14 digital I/O pins
- 6 analog inputs pins
- 6 PWM pins
- SPI Interface
- I2C bus communication
- Input power 7 – 12 V DC
- Output power 5 V DC

Two of the six analog pins would be utilized to detect voltage coming to the motor and LED'S. In the event that multiples microcontrollers are used, the pins 13 (A4 SDA) and 14(A5 SCL) are going to be used. By using these two ports, we are able to set one of the microcontrollers as the master and the rest of the microcontroller as the slaves. SDA is wire Serial Interface Data and SCK is the wire Serial Interface Clock, both of these pins are set to one (digitally) to enable the 2- wire Serial Interface. The protocol used to implement this communication is calledI^2C, and this protocol allows the master microcontroller to interconnect up to 128 different devices using only two-bi- directional bus lines, SDA and SCL.

To connect this microcontroller, a common line for the SDA and another common line for SCL port have to be set and these two lines will be connected to two independent resistors and then to the VCC terminal. If any extra device needs to interact with microcontroller, I^2C compatible parts would be highly recommendable over analog devices. In case that only analog device can be further implemented due to certain limitations, then it is deemed necessary to include analog-to-digital converter in order to be in harmony with I^2C bus.

Finally, the master device will initiate and terminate the transmissions and also generate the SCL clock; the master will address the slave microcontrollers. This connection is depicted in Figure 4.15.

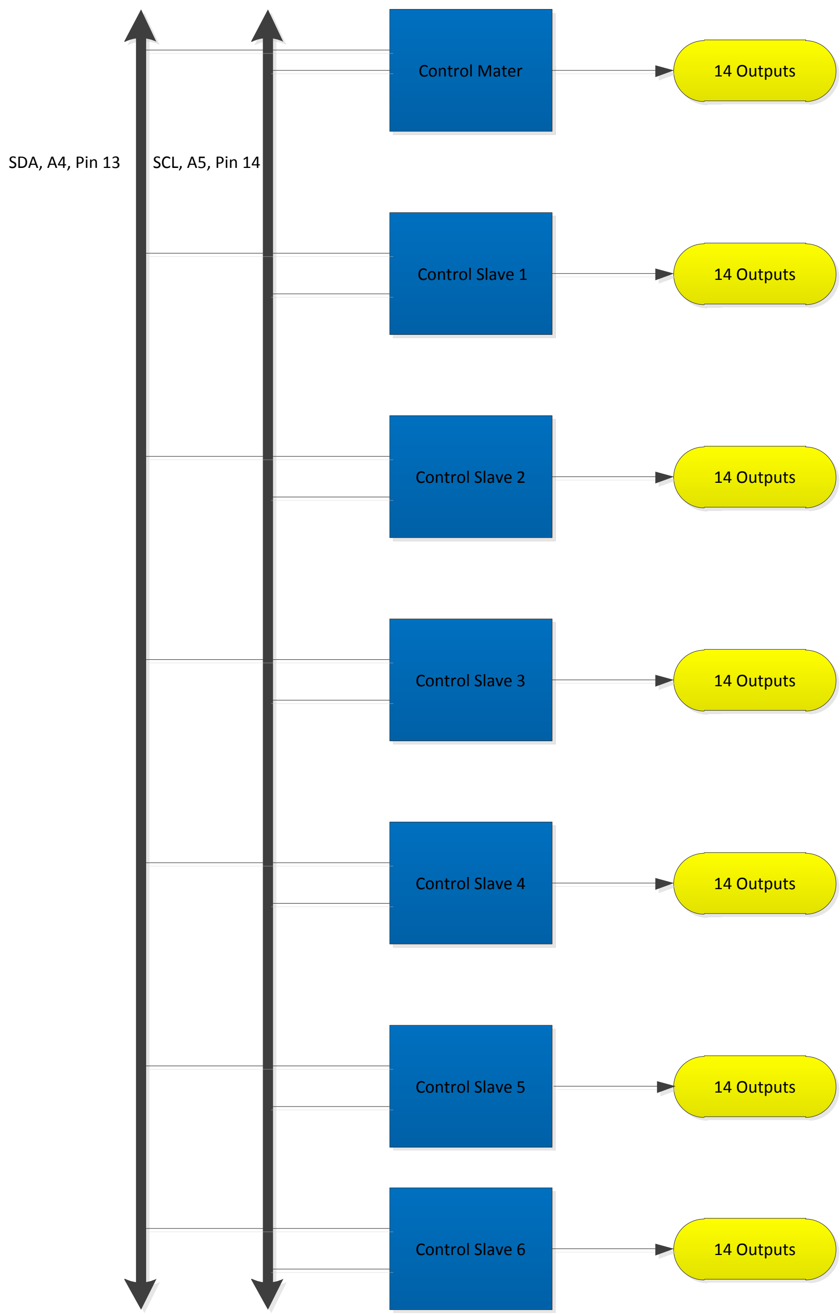


Figure 4.15 I^2C Microcontroller connection

Now, that the communication between the microcontrollers has been set, we are able to use the 14 output pins that each of the microcontrollers provide, so for our LED Globe, we are going to drive 90 LED's (90 outputs), which means, we will use seven microcontrollers to drive the 90 outputs on the LED Globe.

As was previously stated, the LED Globe would be controlled by seven microcontrollers, one master and six slaves, and in the master-slave communication for I^2C only one microcontroller will be the master, and in our case the other six will be slaves. One of the advantages of this method is that the space in which the code is going to be written is going to be increased by seven, but on the other hand, the synchronization of the microcontroller can be challenging. In the I^2C protocol, the master is in charge to send the first command to any of the slaves and after this occurs, they all able to communicate each other using the I^2C protocol. Furthermore, to execute this protocol each of the microcontrollers will have a number assigned, so it can be called from the other microcontrollers when in need. In addition, it is worth to mention that the SDA and the SCL lines will be connected each to a resistor and then to Vcc; this connection is not shown is Figure 4.15.

4.5.1.1 Microcontroller Unit with Charlieplex Configuration

By simply using the outputs of the microcontrollers, it is been discussed previously seven microcontroller will be needed; however, there is a process that allow us to maximize the out of each microcontroller, this process is called Charlieplex.

To use the Charlieplex method, we use the tri-state (Hi-Z) logic capability our microcontroller has. Each of the pins that are used to drive the LED's have three different states: high, low and Hi-z which classify them as a positive, negative (GND), and high impedance respectively. In other words, in addition to the 0 and 1 logic state, the microcontroller also has a high impedance state. This high impedance state is achieved because each of the pins is enable with one pull resistor, when this resistor is disable and the pin is set as an input the high impedance is achieved. Is worth to mention that the I^2C protocol which as was mention before is a bidirectional communication protocol uses pull up resistors on the two communication line, SDA and SCL, so when a communication between two of our devices the outputs of the rest of the devices are set to high impedance, and by doing this we remove their influence on the circuit.

Nevertheless, the tri-state capability of out pins and the Charlieplex method allows reducing the 90 pins needed to drive our LED's to 10, and consequently reducing the number of microcontrollers to one, leaving four pins unused.

For instance, when three pins of the microcontroller are used, two of them can be set up as outputs, logic 1 and 0; in other words, high and low, and the pin that is left is set to Hi-Z, so far we can have two different outputs because if we called the two output pins, PINA and PINB we have an output if PINA is set to 1 and PINB is set to 0, and we will have another output when we reverse the values, if PINA is set to 0 and PINB is set to 1. Now, remember we still have another pin which was set to Hi-Z; let that pin to be PINC. If we change the state of PINA from 1 to Hi-Z, and the state of PINC from Hi-Z to 1, we will immediately will obtain another output between PINB and PINC because remember that PIN B was set to 0. Furthermore, if at this point the state of PINB and PINC are interchanged will obtain another output for a total of four outputs. Finally, if the state of PINB is changed from 1 to Hi-Z and the state of PINA from Hi-Z to 1, again we will immediately obtain an output between PINA and PINC; by interchanging PINA and

PINC state, another output will be obtain, for a total of six outputs using three pins . As we increase the number pins the output in a relation calculated by the following formula:

$$Number\ of\ Pins = \frac{1 + \sqrt{1 + 4 * Outputs}}{2}$$

The numeric relation between the pins and outputs using the formula above can be seen in Table 4.3

Number of Pins	**Outputs**
2.302776	3
3	6
3.701562	10
5	20
6	30
6.844289	40
7.588723	50
8.262087	60
8.881527	70
9.458236	80
10	90

Table 4.3 Pin vs. Output

Then, if we know that between two pins we can have two outputs at a single time as it can be seen in diagram in Figure 4.16 which shows the connection of the 90 outputs using only ten pins; it is valid to mention that the reason why we have two outputs is because an LED (Figure 4.16) is a light emitting diode with a positive terminal (anode) and a negative terminal (cathode) and by placing two diodes with its polarity inverted with respect to each other, the current will flow in different direction when we change the state of the pins from 0 to 1.

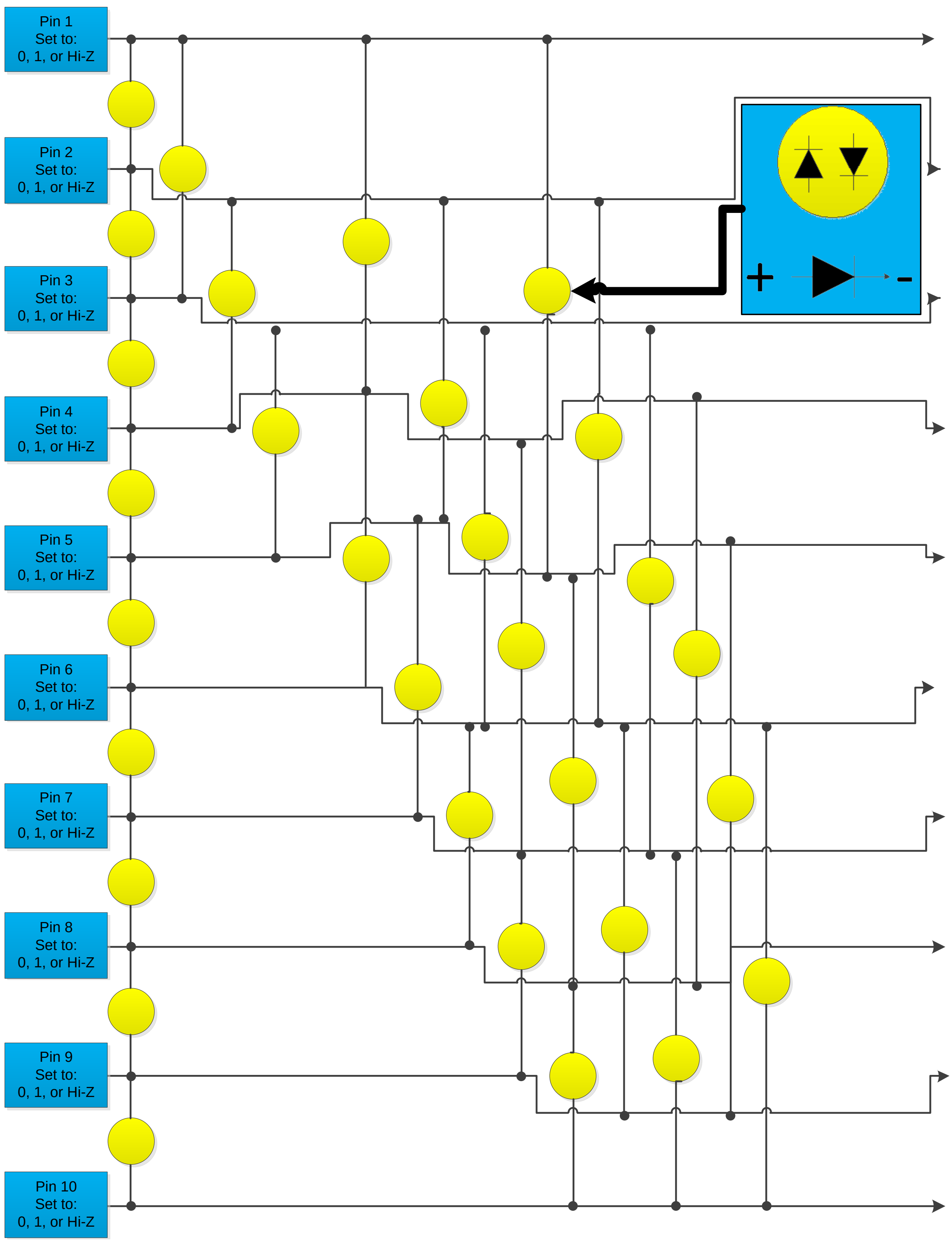


Figure 4.16.1 Charlieplex 90 Outputs part 1

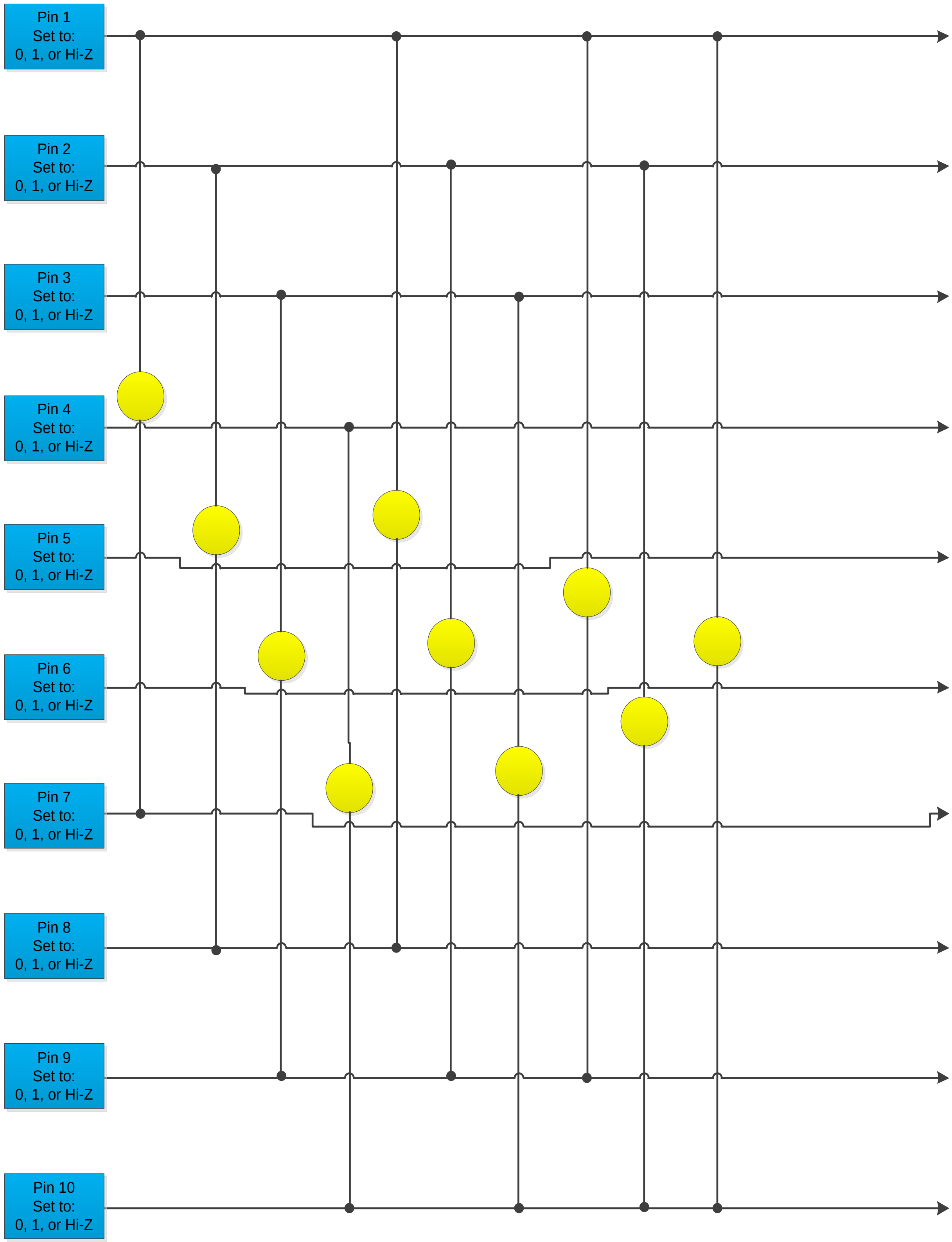


Figure 4.16.2 Charlieplex 90 Outputs part 2

As it can be seen, in the second version of the Charlieplex diagram, pins 2, 3, 4, 5, 6, 7, 9, 10 are connected to different diodes through its cathodes, and on the other end of those diodes (the anode), the diodes are commonly connected to pin (Pin 1). This process is repeated until we use all the pins as common pins as is depicted in Figure 4.17

Figure 4.17 Charlieplex 90 Outputs version II

So far, we have discussed the pin assignation for regular diodes (one anode and one cathode of one color LED); however, the diodes we will use in the LED Globe are common anodes RGB LED's (red, blue, and green), and consequently this diodes have a common anode and three cathodes, one for each color; therefore, each diode needs two more pins than a regular diode to execute the three different states (RGB).

Now that, the configuration for regular LED's have been exposed, we have a better understanding in how to configure the RGB's LED's. For each RGB LED's, we have three output a red light, a green light, and blue light, so each output is multiply times three; therefore, the 90 output we need for the LED globe are going to increase to 270. By using the formula for Charlieplex previously exposed we can calculate the pins we need for this many outputs:

$$Number\ of\ Pins = \frac{1 + \sqrt{1 + 4 * Outputs}}{2}$$

$$Number\ of\ Pins = \frac{1 + \sqrt{1 + 4 * 270}}{2} = 16.94 \approx 17$$

As it can be seen, by Charleiplexing the outputs, we reduced the usage of pins to 17, and each of the microcontrollers provide 14 pins, so we need only two microcontrollers.

Furthermore, the connection of the RGB diodes using the Charlieplex method keeps the same principle, but increases in complexity because of the fact that for each three common anodes we need three different cathodes this condition can be better observed in Figure 4.18.1 and 4.18.2, which is an scaled version of the entire connection in which instead of three pins, 17 are going to be used.

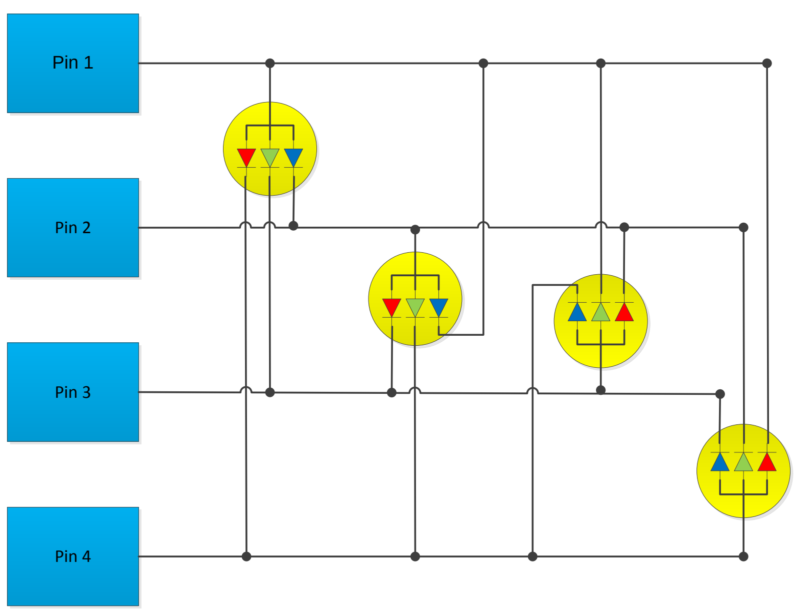


Figure 4.18.1 Charlieplex RGB Outputs part 2

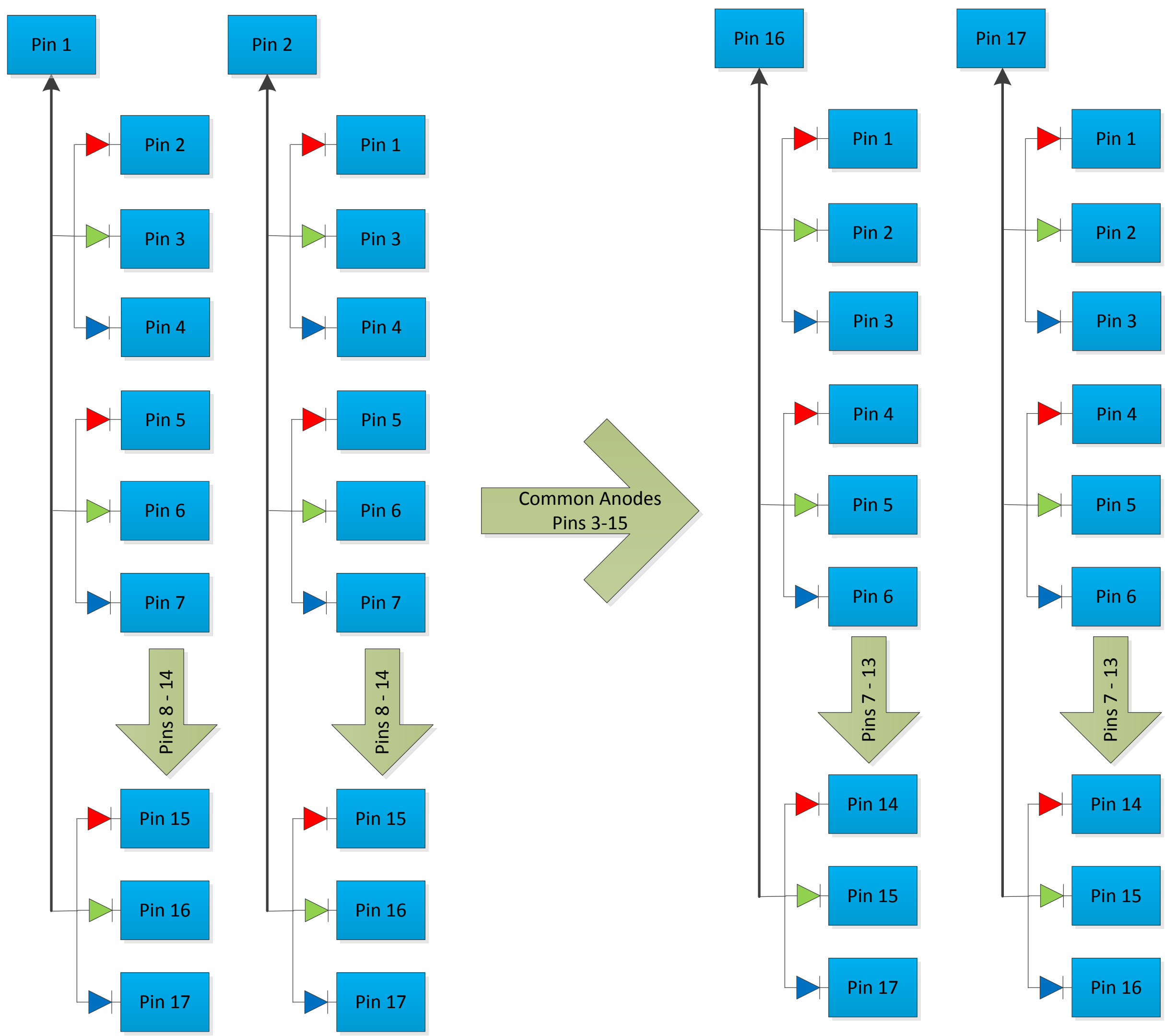


Figure 4.18.2 Charlieplex RGB Outputs version II

4.5.1.2 Microcontroller Unit with Shift Registers Configuration

On the other hand, controlling shift registers through the microcontroller, gives second option to control the outputs. As it was exposed before, each shift registers have the capability to provide eight parallel outputs at the same time, and they can be connected together, so we can have as many outputs as needed. Only three from the microcontroller will be used, one for the data, one for the clock, and one for the latch.

Since the shift register 74HC595 are 8-bit serial-in, the data have to be sent to the register through the data pin, this data is sent digitally; in other words, the data is sent using binary numbers, and this number equivalent to the outputs needed. On the same token, as it has been exposed previously, our LED globe has 270 outputs; therefore, we need 270 binary numbers, and the numbers are coming from the microcontroller imbedded in it through an algorithm programmed previously. The data has logic values (0's and 1's) and it is going to come out from the microcontroller as an array of 270 digits.

This process is executed through synchronous serial communication, and it is required that clocks in the transmitting and receiving devices are synchronized. In contrast I^2C uses asynchronous serial communication which relies in the sender and the receiver to be set independently to an agreed upon specified data rate.

Using shift register, we will only use three pins of our microcontroller one pin for data (DS), one pin for the clock (SH_CP), and another pin as our latch pin (ST_CP). The 270 digits representing each of the output will come out from the microcontroller through the pin designated to be the DS pin (Serial Data Input), and each of this outputs or digits are going to be detected or taken as outputs at the rising edge of the clock. In other words, every time our pin designated to be the SH_CP (Shift Register Clock Pin) pin goes from LOW to HIGH (0 to 1), one digit for the output is going to be valid; however, after the data is taken (270 digits) using the rising edge, the data have to be latched and to do this we have to designate the latch pin ST_CP (Storage Register Clock Pin); after the 270 digits for data are valid or taken the latch pin has to go from LOW to HIGH (0 to 1), and by applying this procedure the data flow out of the microcontroller and through the shift register until we complete the 270 digit as it can be seen in Figure 4.19. This figure is an scaled version for only 13 outputs; therefore, as it is depicted in the Figure 4.19 there are 13 rising edges in the clock while the real scale for our LED Globe the clock pin will 270 rising edges. Also, notice the data pin is LOW (0) for the first three rising edges, meaning the first three outputs are LOW, so those LED's will be off; for the next two rising edges the data pin is HIGH, meaning those outputs are HIGH, so those LED's are on. Finally, for the last cycle of the clock pin when all the outputs have been set, the latch pin goes from LOW to HIGH, meaning this data is taken and is storage in the shift register to be sent out to the outputs (LED's) in parallel.

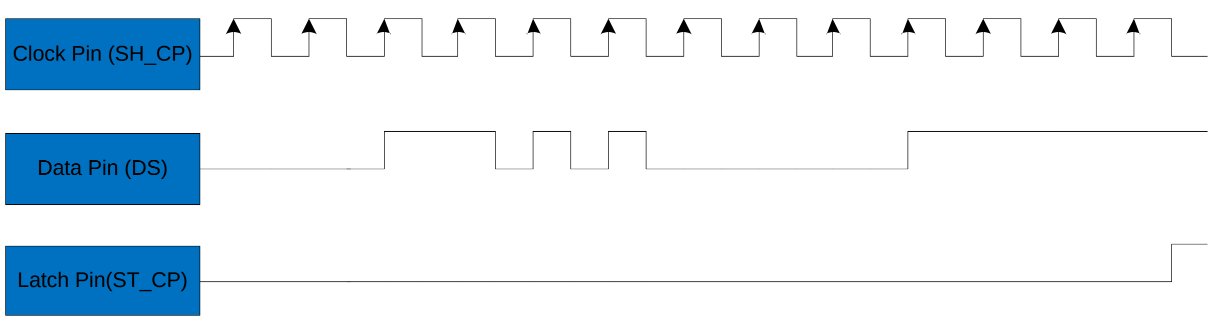


Figure 4.19 Pin Usages and Timing Diagram for Shift Register Option

Even though this method is using more hardware, the usage of the space to program the microcontroller is maximized significantly. The maximization of the space for programming is exposed previously and even though the complexity of the circuit layout decreases when we use this method, since more hardware is used (34 shift registers, 34 capacitors, and 270 resistors), the size and the connections of the circuit layout increases significantly as is depicted Figure 4.20.

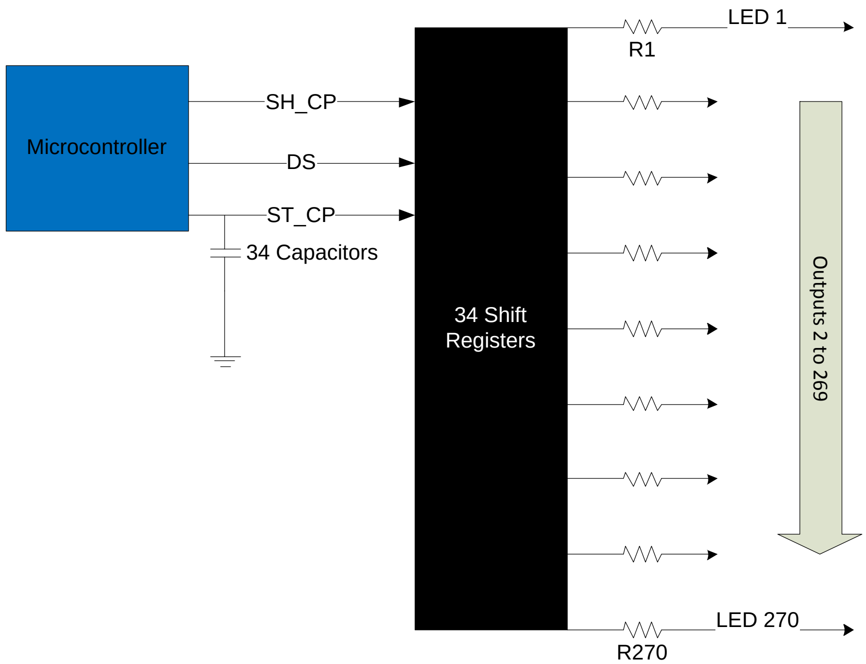


Figure 4.20 Hardware Shift Registers Implementation

Pursuing this further, the microcontroller will receive commands through serial communication using a Bluetooth device to execute sequences previously programmed. In the same token, our microcontroller (ATMEGA328P-PU) has a number of facilities to communicate with computers and other devices. The ATMEGA328P-PU provides UART TTL (Universal Asynchronous Receiver/Transmitter Transmitter-Transmitter Logic) controller for serial communications and it is available on digital pin 0 (RX) and pin 1 (TX) of our microcontroller.

Following this further, as it was explained before, among the four pins on the Bluetooth module, there are two pins, TXD and RXD. The RXD on the Bluetooth module will be connected to TX (pin 1) on our microcontroller; the TXH on the Bluetooth module will be connected to RX (pin 0) on our microcontroller. In the same way, a software serial library has been developed to allow the communication between the devices; this library is accessed when the microcontroller is programmed.

4.5.2 Algorithm Implementation

4.5.2.1 Algorithm Implementation for Charlieplex

The microcontroller will execute specific algorithms which are going to perform the LED Globe functions and the Bluetooth function, and also a simple function such as checking the status of the voltage coming to the LED ring and turn on an single LED if voltage is been supply bye the source. As it is been exposed previously, two different hardware's logic implementations have been analyzed to execute the functions of the LED Globe (I^2C and shift registers), and one or a mixed of those two is going to be used. Previously, we also discussed the I^2C and the shift registers configurations in detailed. Now, for the I^2C configuration, the algorithm uses the Wire Library which allows establishing communication between the microcontrollers as it was depicted previously in Figure 4.15.

On the same token, the master microcontroller is the one that possess the algorithm to be executed at first, and a part of this algorithm is designated to activate the execution of the other algorithms programmed in the rest of the microcontrollers (slaves). After the first interaction, when the master microcontroller communicates with any of the slaves, the communication does not have a specific order, in other words, any of the microcontrollers can be activated to perform functions through the algorithm by any of the other microcontrollers as it is depicted in Figure 4.21

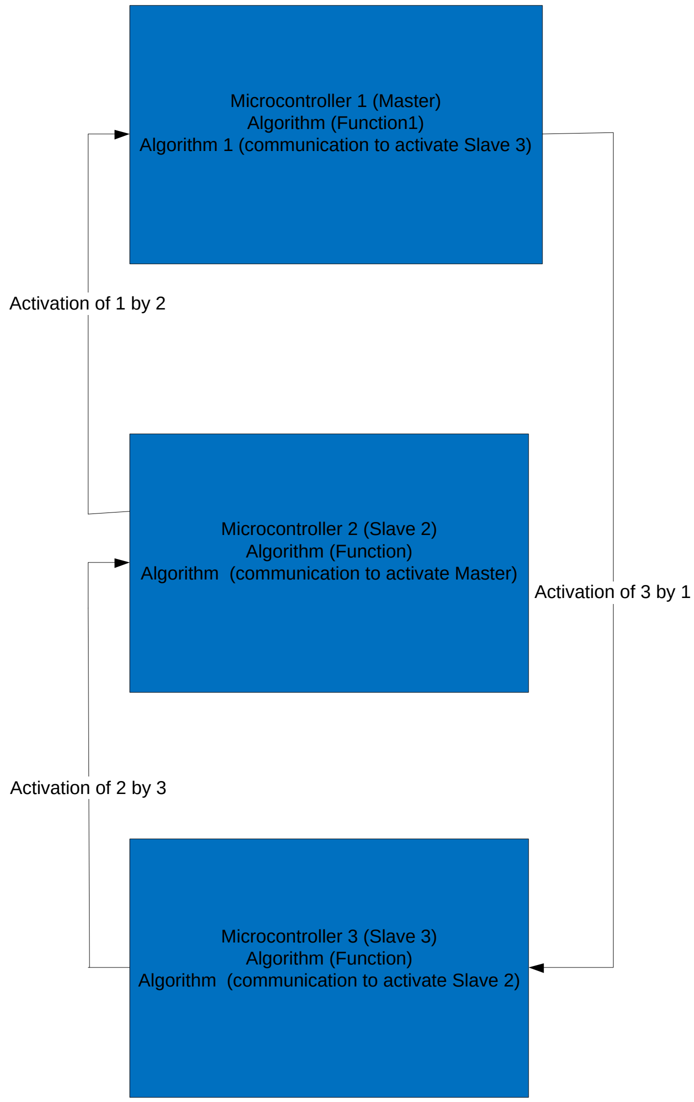


Figure 4.21 Algorithm Implementation for I^2C

4.5.2.2 Algorithm Implementation for Shift Register

Using shift registers, the algorithm configuration changes. The algorithm will not be as extend as it was for the previous configuration; the algorithm consists of three parts: output array, clock, and latch.

The algorithm for the output array will have 270 digital values (0 or 1); the second part of the algorithm, the clock, will take each of these digital values while toggling the clock pin, so it goes from LOW to HIGH (0 to 1). Finally, when the clock algorithm has toggled the clock pin 270 times, the third part of the algorithm, the latch, is going to make the latch pin to go from LOW to HIGH (0 to 1), and once this process is executed, the LED's will perform a sequence of turning and of course more of sequences are going to be needed to display images on the LED Globe. Consequently, there will be a variety of arrays with the same 270 digital values, but ordered differently, and these three parts of the algorithm are going to be enclosed in a loop (Figure 4.22) until an entire sequence of images is display on the LED Globe.

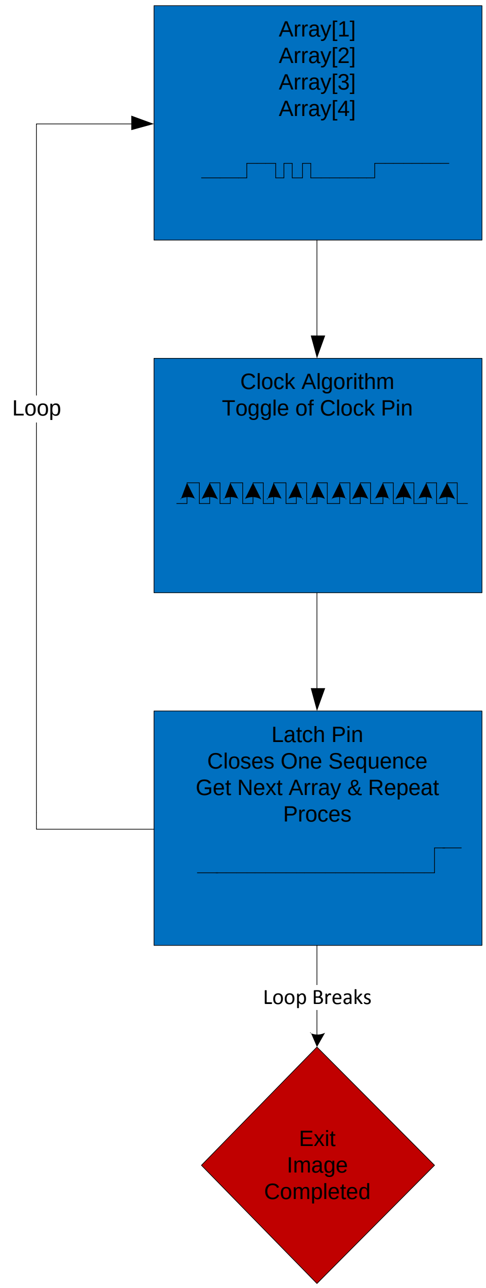


Figure 4.22 Shift Register Configuration Algorithms

4.5.2.3 Algorithm Implementation for Bluetooth

For the Bluetooth algorithm, we used the same serial library we used previously. The algorithm designed to operate this device will establish communication between the JY-MCU Bluetooth module and the device we chose to transmit the commands to the Bluetooth module.

The ATMEGA328P serial library allows the microcontroller to read character through serial read; the algorithm used to receive commands via Bluetooth will be based in conditional statements, so after the wireless connection is established, the microcontroller will receive a character from the transmitter wirelessly, and using a conditional statement depending the type of character received, a light sequence will be displayed. Lastly, using wireless communications, we will activate the execution of specific algorithms that allows us to have complete control only of the LEDs.

4.5.3 Light Emitting Diode

The diodes are two terminals electric components that have low resistance to current flow in only one direction. In addition, the light-emitting diode (LED) is a semiconductor diode, which emits light through an effect called electroluminescence. Like a regular diode the LED has a positive and a negative terminal called another a cathode respectively.

Pursuing this further, for this project, we have chosen to use the common anode RGB light emitting diodes, meaning that these light emitting diodes are able to emit light in three different colors, red, green and blue and as its name describes it, this diode has a common anode and three diodes, one for each color (Figure 4.23). A list of the most relevant specifications for RGB common anode light emitting diode is given bellow:

- Forward current 20 mA
- Forward Voltage 2.0 V to 3.4 V
- Peak forward current 30 mA
- Operating temperature $-40℃\ to + 85℃$
- Luminous intensity
  - Red 800 mcd max (millicandela)
  - Green 4000 mcd max (millicandela)
  - Blue 900 mcd max (millicandela)

In final analysis, since the desired amperage of the light emitting diode is 20 mA, a calculated resistor of 150 Ohms (Figure 4.24) is going to be used for each light emitting diode taking in to account an input voltage of 5 volts normally provided for the microcontroller or shift register.

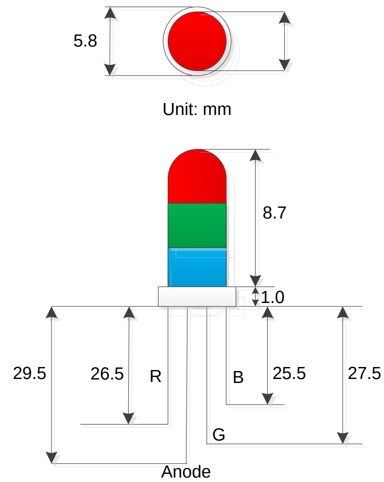


Figure 4.23 RGB light emitting diode

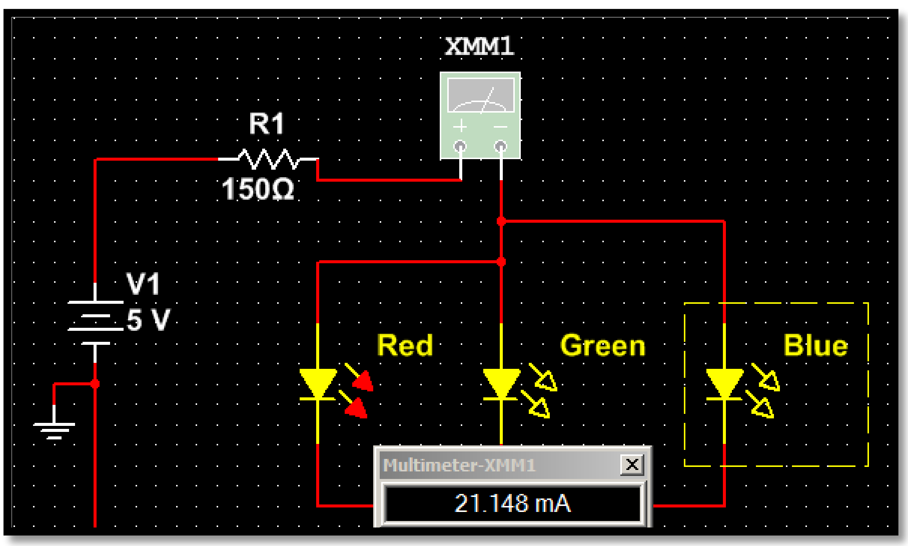


Figure 4.24 RGB light emitting diode amperage and resistor

4.5.4 Shift Registers

The 74HC595 is a Complementary Metal-Oxide Semiconductor (CMOS); this shift register has 16 pins and a serial input (DS) and a serial standard output (Q7S) for cascading, which allows the device to be connected to other shift registers (74HC595). The 74HC595 consists of an 8-bit shift-register and 8-bit D-type latch with three-state parallel outputs (LOW, HIGH, and Hi-Z).

The configuration of this device was explained previously when we exposed the microcontroller configuration, and as it was exposed then, the output of the microcontroller are: SH_CP, DS, and ST_CP which represent the clock, data, and latch. Furthermore, these outputs are connected directly to inputs on the 74HC595 shift register which have the same name and have the pin numbers 11 (SH_CP), 14 (DS), and 12 (ST_CP). In addition to these three pins we are going to use pin 9 (Q7" or Q7S), serial data output for cascading inputs, meaning that after receiving eight inputs through the DS pin, the first shift register will transfer the inputs to a second register through the Q7S pin, and after the second shift register receive eight inputs, the second register will transfer inputs to the third one through the Q7S pin of the second register and so on. The pins 8 and 16 are used for ground (GND) and input voltage (VCC = +5 Volts) respectively; the rest of the pins have default values pre-set conveniently, so they are not shown in Figure 4.27.

On the same token, for the 270 outputs, which are 270 LED's (90 red, 90 Blue, and 90 green), we are going to use 34 shift registers, which will drive the three different colors of our LED's. In other words, 11 and ¼ register are going to be used for each color. Lastly, since we want the LED to be protected from being overload, we are going to add a 150 ohm resistor as shown in Figure 4.25.

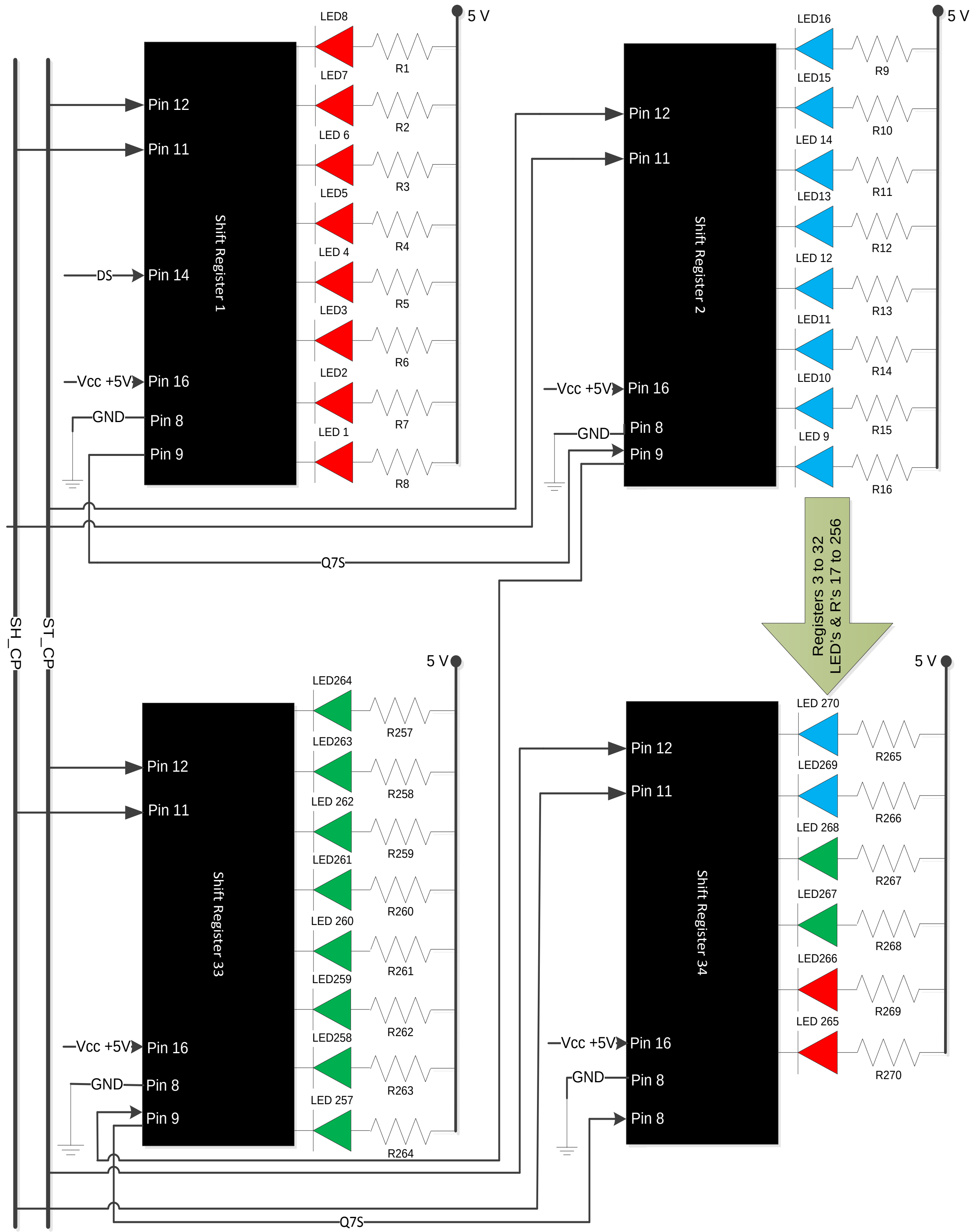


Figure 4.25 Shift Registers Configuration

4.5.5 Bluetooth Module

After power is detected in the LED Globe system, and the device (LED Globe) is ready to display images, the LED Globe will be able to receive commands via Bluetooth from another device enable to send commands through Bluetooth, in our case a cellphone. Pursuing this further, the Bluetooth chosen to perform such a task is the JY-MCU Bluetooth module; the JY-MCU easy installation and accessible ports make the installation straightforward. Also, the device uses the same serial communication exposed on section 4.5.2 and it can be configured using the same procedure we used for I^2C protocol (Maser-Slave) depicted in Figure 4.17 BACK. The functionality of transmitting data wirelessly is significant because allows us to update and upload new programs without the necessity of having USB interface added to our PCB design; consequently, this improve our design and makes it neater. Also, it is worth to mention now that when configuring this Bluetooth module, the communication ports have to be carefully chosen in the sense that the computer used has to be configured through the device manager, so the bits per second, data bits, parity, stop bits, and flow control match the specifications of the Bluetooth module which are: 9600, 8, none, 1, and none, respectively. These specifications are also valid for all the serial commutations used in this project.

A list of the most relevant specifications for the 74HC595 is given bellow:

- Input voltage 3.6 – 6.0 volts
- PCB size 35 x 15 millimeters
- Depth 2 millimeters
- Transmission distance 10 meters (maximum)
- Weight 3.6 grams
- 4 ports: VCC, GND, TX, RX,

As it was mentioned previously, the ATMEGA328P UART serial communication pins are pin 0 (RX) and pin 2 (TX), and these two ports are going to be connected to TX and RX of the Bluetooth module respectively as is depicted in Figure 4.26.

Of course, the transmitter device needs Bluetooth capabilities, and the device chosen to perform such a task, is a Samsung HTC mobile device (cellphone). Using one of the device applications (Bluetooth Chatting) that allows sending via Bluetooth, we will send the commands to the microcontroller. Before this, the transmitter and receiver need to paired, and the Samsung HTC will locate the Bluetooth devices available around, including our Bluetooth module, which will appear as a device with the name of Linvor; it will also require a password which is 1234, and after the connection is established a flashing light on our Bluetooth will stop and stay solid.

Lastly, the commands that are sent to the Bluetooth are going to be single characters, which are going to be sent individually; each of them will start a sequence of images previously programmed in the microcontroller. The programming part for this procedure was exposed in section 4.5.2.3; the sequences of images are described with more detail in section 2.2.8 where the output of the system was explained.

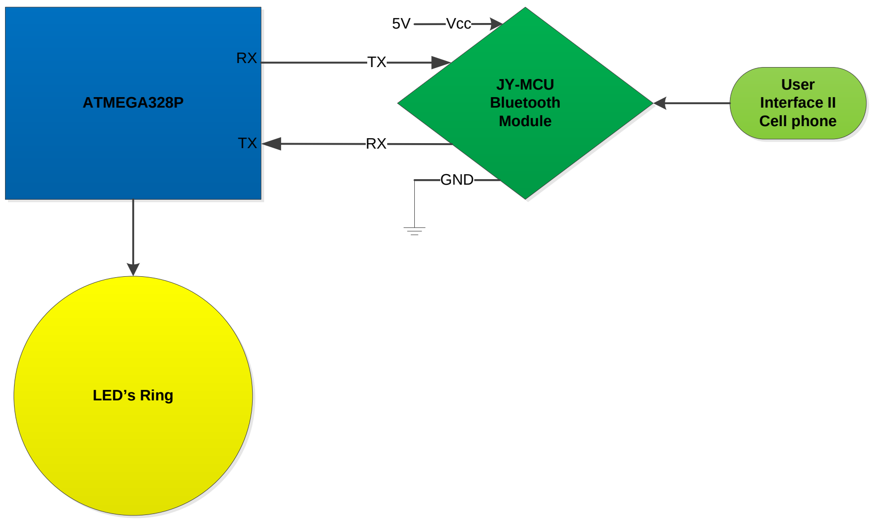


Figure 4.26 Bluetooth Module

4.5.6 USB to TTL Module Adapter

Nevertheless, when uploading programs to the ATMEGA328P via Bluetooth, communications issues are commonly encounter; therefore, as an alternative a PL2303HX USB to TTL Convertor Module Adapter will be used. Even though this alternative way of uploading sketches represents a solution to possible communication problems between the microcontroller and the Bluetooth, it creates a conflict because the PL2303HX USB to TTL Convertor Module Adapter as the Bluetooth uses digital pin 0 (RX) and pin 1 (TX) of the microcontroller; the ATMEGA328P boot loader allows to execute this procedure.

In this situation, both JY-MCU Bluetooth Module and PL2303HX USB to TTL Convertor Module Adapter are connected to digital pin 0 (RX) and pin 1 (TX) and even though both of these devices are separately programmed, the fact they have digital pin 0 (RX) and pin 1 (TX) as common pin, can create a communication problem also.

At this level, if after testing, the communication problem exists, a switch controlling the voltage feeding the Bluetooth module can be placed, so the voltage can be interrupted when the USB to TTL convertor is been used depicted in Figure 4.27. Lastly, it is worth mentioning that each microcontroller will have a PL2303HX USB to TTL Convertor Module Adapter and a JY-MCU Bluetooth Module to upload sketches, but when we are controlling the LED's, only one them is going to be used.

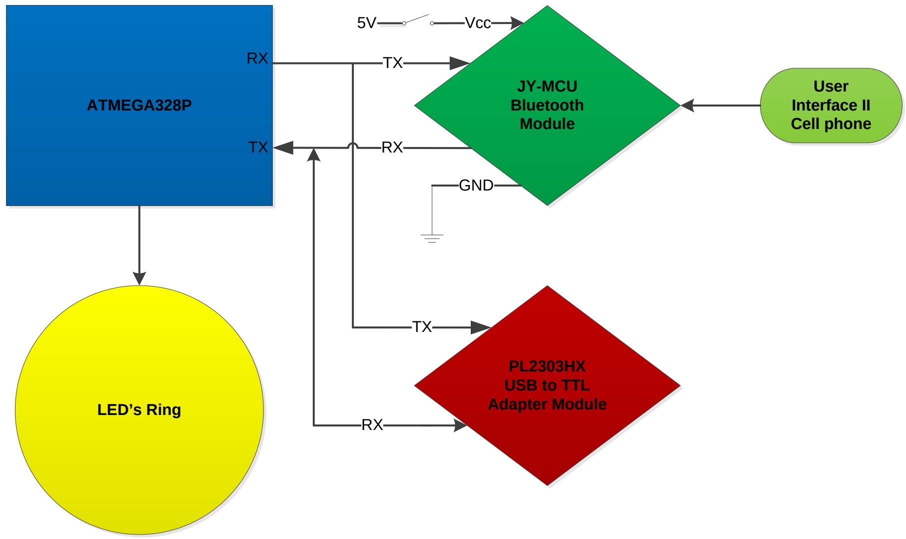


Figure 4.27 Bluetooth Module with PL2303HX USB to TTL Convertor Module

5 Design Summary of Hardware and Software

5.1 Hardware Summary

To summarize the hardware design of the LED globe system, let us first describe the power supply. An AC Voltage of 120rms will be stepped down by the 5:1 transformer and rectified by the bridge rectifier circuit. The RMS AC voltage will be 24V and its peak voltage will be 32V before the rectifier. After the rectifier circuit, the RMS voltage will be 22 volts and its peak will be 30V. This is due to the forward voltage drops of the two rectifier diodes, which are 1 V each. Next, the smoothing capacitor will try to smooth out the ripple voltage out of the bridge rectifier to a near constant peak voltage of 30 volts.

Next, we describe the voltage regulators, which are part of the power supply. The maximum output power available to the voltage regulators will be 352 Watts. The motor voltage regulator will be in parallel to the microcontroller voltage regulator, which is both feeding off of the smoothing capacitor output. The motor and the microcontroller voltage regulator will have a maximum input voltage of 42V and a minimum input voltage of 14V. The output voltage of the motor voltage regulator will have a voltage of 9V with a max current output of 10A. The maximum power dissipated due to the motor will be 90 Watts. The motor will spin the support shaft at about 3169 RPMs. The output voltage of the microcontroller voltage regulator will have a voltage of 9V with a maximum current output of 1 A which means it will run at about 9 watts max. This output voltage will be available at the slip ring voltage terminals to power up the microcontroller, bluetooth and give power to all the shift registers and the LED's.

The Plexiglass encapsulation will house the power supply and the motor. The plexiglass will be perforated on the backside for ventilation of the motor and power supply. The support bar will be mounted on the back and will help stabilize the ring support shaft. The slip ring voltage terminals will supply the slip rings on the support shaft with a constant 9 V. The support shaft is responsible for supporting the microcontroller PCB, and the LED ring structure; the support shaft will also rotate the LED ring using the motor. The LED ring structure will rotate while it holds all 90 LED's in place. The ring has to be very lightweight, which is why it is made out of balsawood. Please see Figure 5.1 and Figure 5.2 for more details about our hardware design.

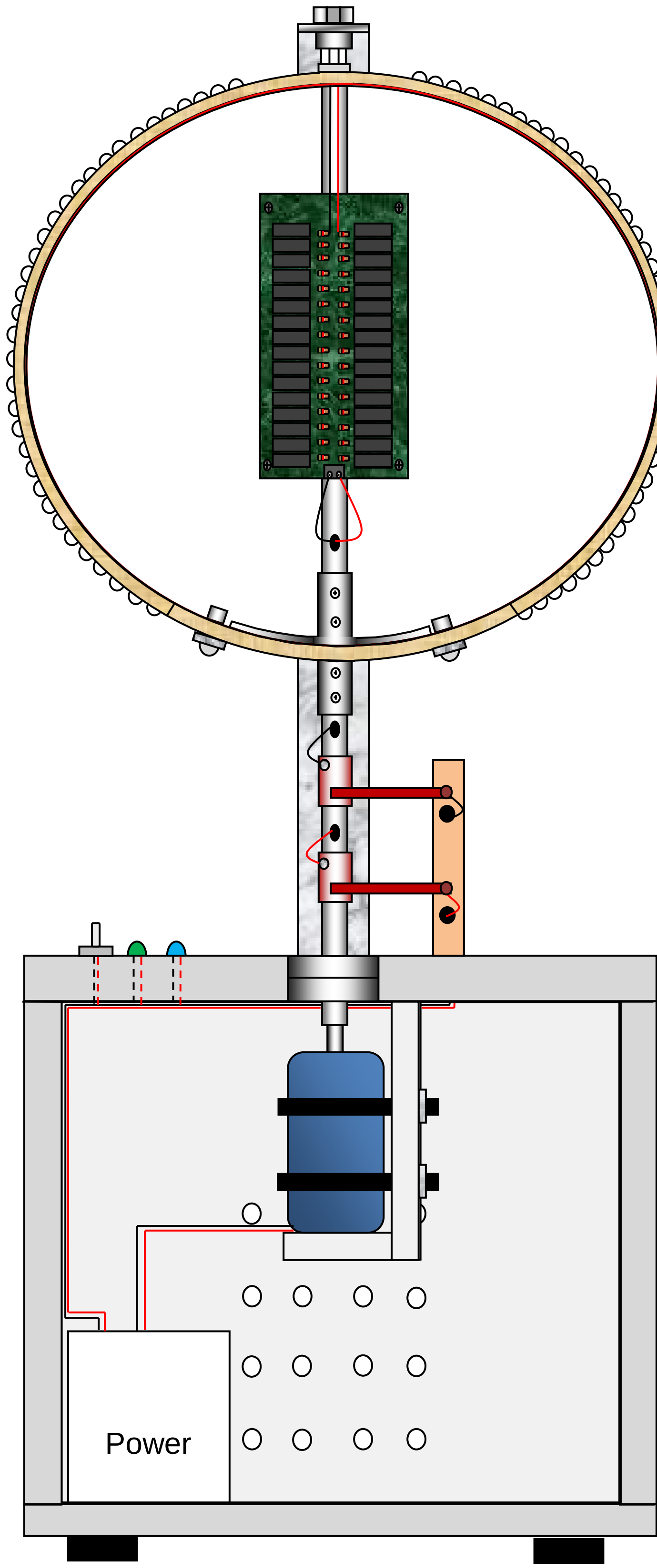


Figure 5.1 LED Globe Front View

Right side view

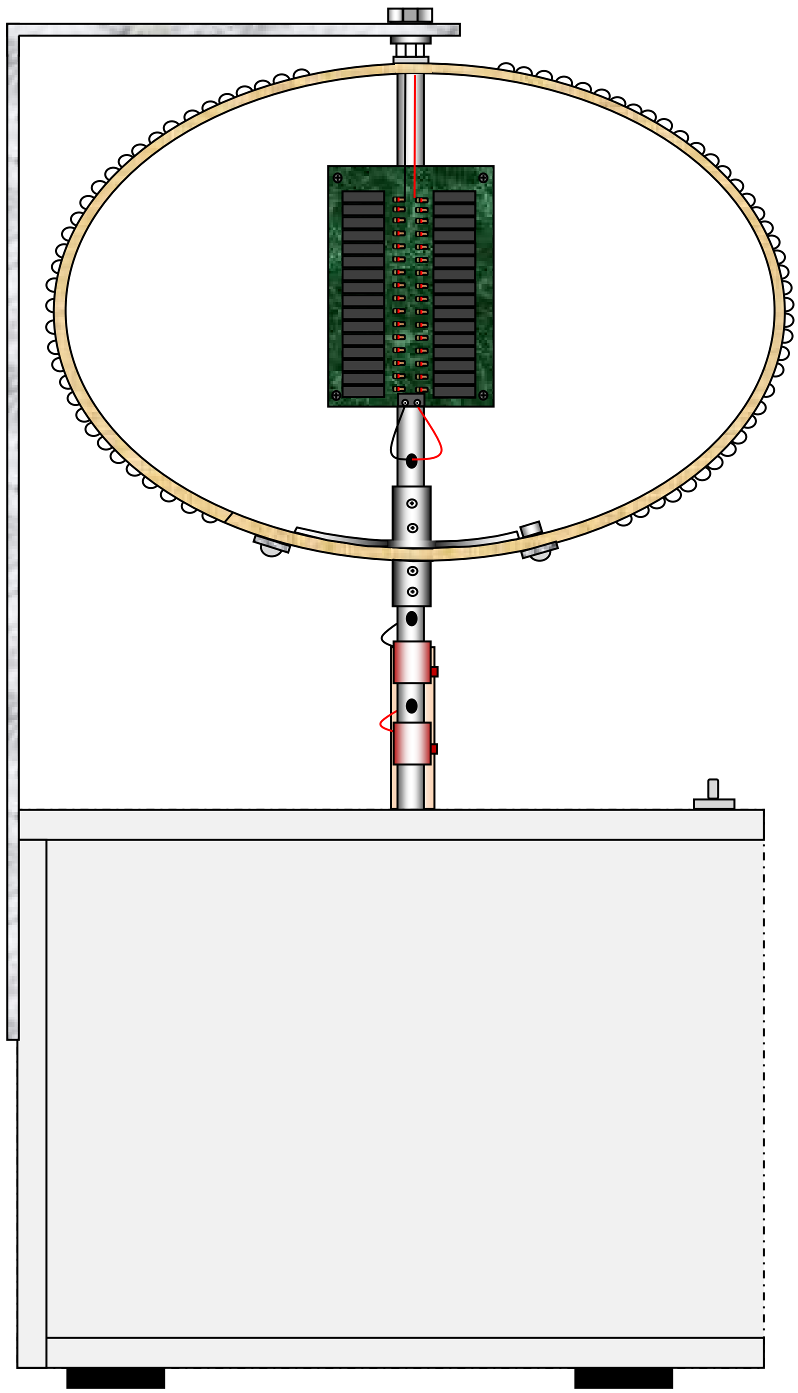


Figure 5.2 LED Globe Right-Side View

5.2 Software Summary

The main control of the LED Globe is the microcontroller, which will execute the following functions:

- Check power status for the LED's
- Receive data from two different sources
- Provide data to 34 shift register
- Establish communication with others microcontrollers

As was exposed previously, these few functions are the brain of the LED which allows it to perform its principal function of displaying images, and even though, the other portions of this design are important, nothing is more important than the software. The user interfaces that the microcontroller uses, (USB to TTL module and Bluetooth module) enhance the accessibility to the microcontroller, allowing to update data and consequently to be able to upload new sketches and display new images. This improvement on the design reduces the space limitation the microcontroller has. Also, the discussed method explains the pros and cons of the software design for each of the cases, and basically establish an inversely proportional relationship between amount of hardware and algorithm complexity, meaning that applying methods that uses less hardware such as Charlieplex, we increase the algorithm complexity and also the length. On the other hand, when we implement shift registers, we increase the amount of hardware, but we decrease the complexity of the algorithm and length. Finally, Figure 5.3 depicts the overall microcontroller and a functionality diagram

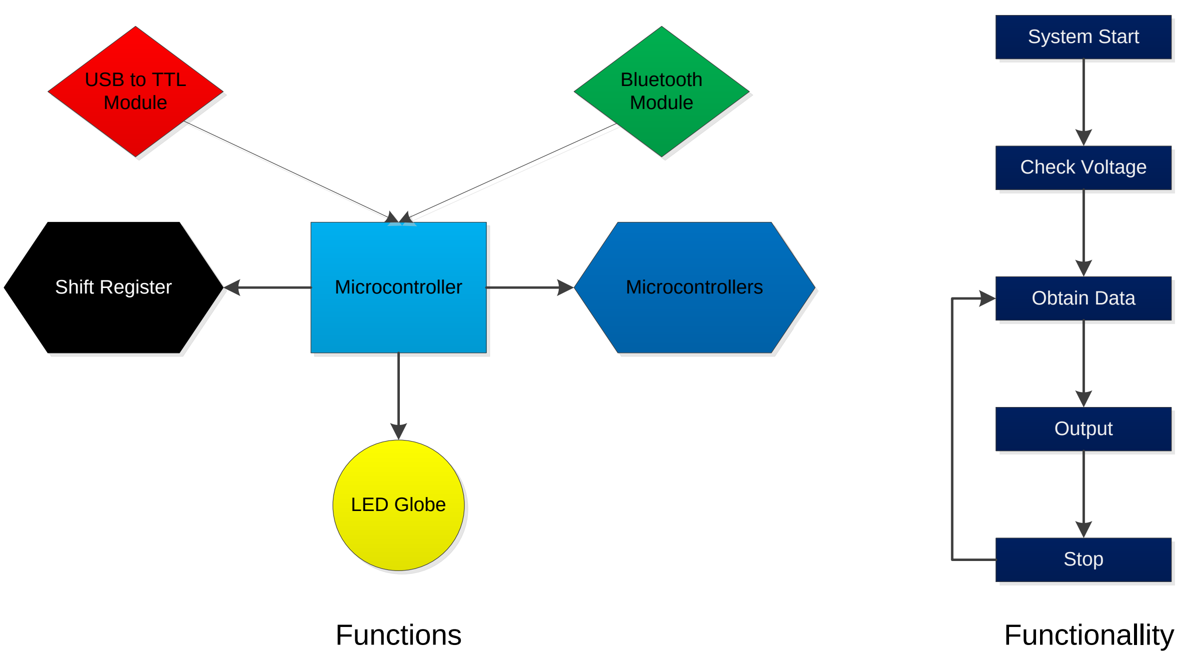


Figure 5.3 Microcontroller Functions and Functionality

6 Project prototype construction plan

6.1 Parts Acquisition

Many of these parts will be ordered through CadSoft Eagle software because of the ease of finding the parts while assembling and building the printed circuit board simultaneously.

Control Circuit
Microcontroller – Atmel ATMEGA258P-PU
There are various that provide the ATMEGA258P-PU used in our project because it is a very commonly used microcontroller. That being the case, we are going to order our microcontroller when complete our printed circuit board design and order it together with the other parts of the printed board design. The cost varies from vendor to vendor but through the printed circuit board design software, the cost will be around $1.80 at the very minimum.

Bluetooth module – JY-MCU bluetooth module
Our Bluetooth module will be bought through Sparkfun Electronics. The Bluetooth module is our communication device and thus a very important part of our project. The JY-MCU Bluetooth module will cost $8.20 and we will only need one of these parts.

USB port module – PL2303HX USB to TTL converter module adapter
The PL2303HX USB to TTL converter module adapter will be purchased through Ebay because of the price. It is set at $2.40 and usually has free shipping. If Ebay does not work for us, we will order the part from DealExtreme, which puts the PL2303HX USB to TTL converter module adapter at about $2.70 and also free shipping. We only need one of these parts for our programming prototype phase.

Shift Register – from sparkfun, 74HC595
There are a lot of shift registers to choose from but we chose the 74HC595 because it was simple and cheap. We will need at least forty of them to connect to all the LEDs that we have and have some leftover in case of faults or accidents. The vendor we will use to purchase these parts will probably be the printed circuit board software company. The shift registers are extremely cheap at $0.10 per part pushing the total cost of the 74HC595 to be $4.00.

Structure Assembly
Metal/Wood materials
We will be getting various materials to assemble our structure. Materials such as screws, pipes, washers, rings, plates and the like will be bought from Home Depot. A more detailed look into our materials can be found in the Administrator Content in the Budget section.

LEDs - Ultra-Bright 5mm 4 pin RGB Diffused Common Anode LED
We will be purchasing our LEDs from EBay because of the cheap price point of the LEDs. We only need ninety LEDs but this comes in a pack of one hundred LEDs. The cost comes out to $12.00 which makes each LED cost $0.12 each. Also, free shipping is a plus.

Power System
Voltage regulators - lm25711, lm7809
We are going to purchase both the LM25711 high voltage synchronous buck controller and the LM7809 voltage regulator from DigiKey but it is manufactured by Texas Instruments. One of the voltage regulators will regulate the motor and the other will regulate the microcontroller. The LM25711 high voltage synchronous buck controller costs $5.70 while the LM7809 voltage regulator will cost $6.40. We will only need one of each part and there is likely a shipping fee.

Transformer – TR100VA001US
The TR100VA001US transformer will be purchased at EnergyControl.com at the price point of $56.00. There are other transformers at cheaper prices but the one from EnergyControl.com has the specifications that we need to fit with our project. We will only need one of these and there is likely a shipping fee added.

Smoothing capacitor - Cornell Dubilier CGS263U075V4C Aluminum Electrolytic capacitors
These capacitors will be ordered from Mouser Electronics. These capacitors cost $80.29 each and we need to have two of these parts because of the power supply that we are using which comes to a total of $160.58. Alternatives did not compare to the ones that Mouser Electronics has so it was a fairly easy choice.

Bridge rectifier – NTE5812HC diodes
We are ordering our NTE5812HC diodes for our bridge rectifier from Newark and element14. The price of these NTE5812HC diodes are $1.77 each. We need four diodes to create a rectifier circuit so the total amount we have to pay is $7.08. There is no shipping fee.

Electric Motor – HobbyWing XERUN 3.5T 3650-9100KV
The HobbyWing XERUN 3.5T 3650-9100KV electric motor will be order from HobbyPartz.com and it is a RC car motor that can be sufficient in our needs of spinning our ring structure really fast to produce POV. This electric motor is small and compact and will fit in the plexiglass encapsulation. The cost of this electric motor is $76.00 and it has a shipping fee. We will only need one motor for only project.

Home Depot
Home Depot will provide the little things such as screws, washers, metal materials and other things that we will need and is detailed in the administrator section under budget.

Home Depot is inexpensive and has a nice hobbyist environment where the help is plentiful and the goods are top-notch.

6.2 Assembly

We will be building two printed circuit boards and since the price of the printed circuit boards is dependent on the size, the design will have to be efficiently and strategically placed in order to save money. We must choose the smallest parts that meet the needed criteria of the project as well as look at cutting costs wherever is necessary. The prototype will be smaller than the actual main printed circuit board that we will be using. As such, there will be less parts to surface mount and also a smaller schematic which results in a smaller board.

Our main printed circuit board that will control all the LEDs will have thirty-four shift registers surface mounted to the board in addition to the microcontroller and various other electric components. We will need to find the smallest shift registers in order to conserve space. Most of our components will be purchased and surface mounted by CadSoft's Eagle software Direct Link function. They have decent prices for parts. If we are not satisfied with Eagle then we will turn to 4PCB.com to do our surface mounting and ordering of our parts. Below is Table 6.1, which details the minimum requirements and restrictions in the making of printed circuit boards.

Min. qty. 4 Boards	White Legend (1 or 2 sides)
Lead Time 5 Days	1 Part Number Per Order (extra 50 charge for multiple parts or step & repeat)
2-Layersm FR-4, 0.062'', 1 oz. cu. Plate	Max. size 60 sq. inches
Lead FREE Solder Finish	No slots (or overlapping drill hits)
Min. 0.006'' line/space	No Internal routing (cutouts)
Min. 0.015'' hole size	No scoring, tab rout or drilled hole board separation
All Holes Plated	Routed to Overall Dimensions
Green LPI Mask	Max. 35 drilled holes per sq. inch
Credit Card Order Only	

Table 6.1 Printed Circuit Board Specifications

7 Project prototype testing

In order to make sure our theoretical design works and is in sync with our physical design, we have to set some procedures up to see if our components are working properly. Testing procedures should proceed as follows:

- Make sure components worked as ordered
- Make sure parts are not faulty
- Make sure programming executes
- Calibrate components according to specifications

In the following sections, we will discuss the procedures we will run in order to test each component to deem it worthy for presentation and to help us calibrate it to make the components work effectively. We will list the prototype we are testing as well as the method of testing and the desired results we would like to see from our testing procedures. A short paragraph might follow for our desired modifications to existing prototypes if testing was deemed unsuccessful.

7.1 Power supply testing

Testing for the power supply will be done at the UCF senior design laboratory. To test the power supply, a multimeter will be required to check for voltage, current and resistance, a function generator to create a sinusoidal signal and an oscilloscope will be required to see and analyze all signals.

7.1.2 Transformer Testing

Testing the transformer is a very simple process. First, the primary side of the transformer will be attached to a function generator set to a 60 Hz sine wave with 12 V peak amplitude to simulate the 60 Hz signal coming from an outlet. Since the transformer has a turn's ratio of 5 to 1, the expected voltage across the secondary leads should be 2.4 V. An oscilloscope will be attached to the primary side of the transformer and the secondary side to compare input and output amplitudes and to see if there is no change in phase. This low voltage preliminary test is necessary in order to verify that the transformer is in working condition before introducing it to a higher voltage source. If the transformer passes this test, then the transformer can be introduced to a 120 V outlet; furthermore the secondary side of the transformer can be tested with an oscilloscope to verify if a 24 V RMS 60 Hz sine wave is present. If so, then this indicates that the transformer is in functioning condition.

7.1.3 Bridge Rectifier Testing

The secondary leads of the transformer will now be connected to the bridge rectifier circuit in low voltage testing mode. An oscilloscope will be attached to the secondary leads of the transformer and the output of the bridge rectifier circuit. The incoming signal should be a 2.4 V peak 60 Hz sine wave. After the rectification process, the signal should look more like a DC signal but with peak amplitude of about .4 V. This is due to the forward voltage drop of the two diodes as Figure 7.1 shows:

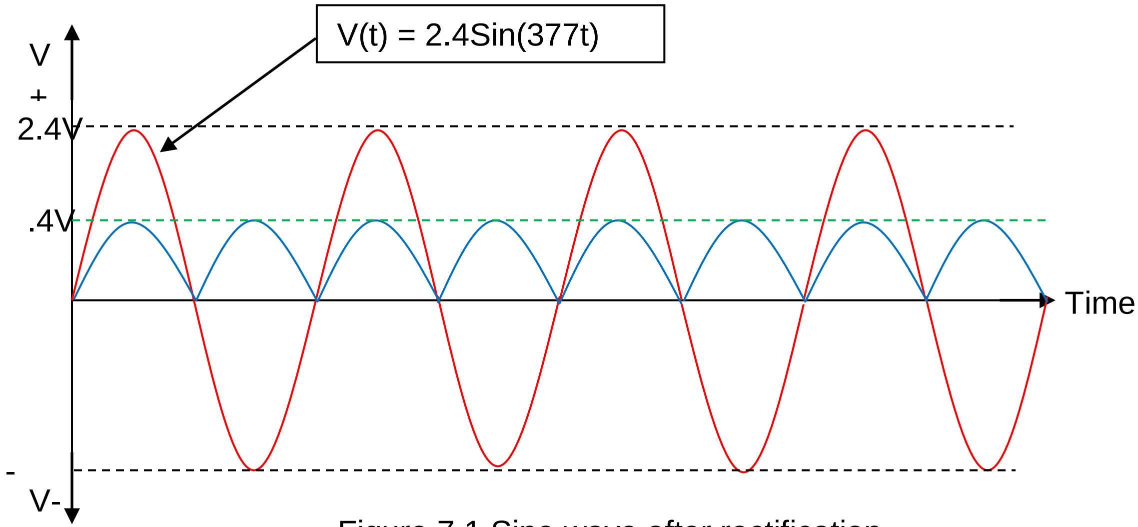


Figure 7.1 Sine wave after rectification

If the bridge rectifier passes the low voltage test, then the bridge rectifier is functioning properly. If the bridge rectifier's functioning properly, then we can connect it to the secondary output of the transformer and test the output signal on the bridge rectifier to see if the signal is rectified. If so, then the signal should look like the low voltage signal but with amplitude of $24x\sqrt{2}V-2Vf = 33.94Vpeak -2 = 31.94$ V peak as Figure 7.2 shows:

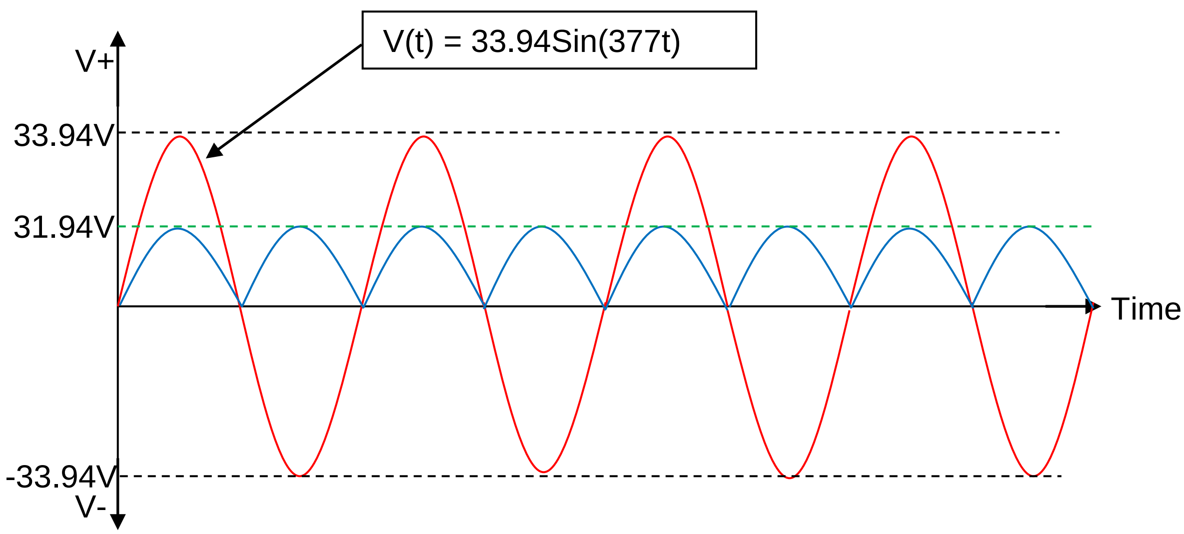


Figure 7.2 Sine wave after rectification at full voltage

7.1.4 Smoothing Capacitor Testing

The job of the smoothing capacitor is to smooth out the ripple voltage after the bridge rectifier circuit. Because the diodes are forward bias, the capacitor will charge up to the peak voltage, and will discharge as soon as the voltage swings lower. The voltage across the capacitor is modeled by the following equation:

$Vc(t) = (Vmax)e^{-t'/RC}$, where t' is the time after the output has reach its peak value, and RC is the time constant. The smallest output voltage of the capacitor while it is discharging is modeled by the following equation: $VL(t) = (Vmax)e^{-\frac{T'}{RC}}$
Where T' is the discharge time of the capacitor. The ripple voltage will then be equal to Vr= Vmax – VL = Vmax(1- $e^{-\frac{T'}{RC}}$) .We would like the ripple voltage to be as small as possible. To have a good smoothing capacitor we want the RC time constant to be as large as possible. R is $\frac{\text{Vmax}}{\text{Io}}$ = $\frac{31.94}{10}$ = 3.19 ≅ 3.2Ω and C is 52.08mf. If we estimate T' to be the period of the signal T which is
$\frac{1}{2*f} = \frac{1}{2*60Hz}$ = 8.33ms and approximate e^x, where x = $\frac{-T\prime}{RC}$,to its equivalent Taylor series and keep only the linear terms: $1 + \frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots \cong 1 - \frac{T\prime}{RC}$ then the ripple voltage will be
$31.94\left(1 - e^{\frac{.0833}{.1665}}\right) = 31.94\left(1 - \left(1 - \frac{.0833}{.1665}\right)\right) = 1.6V,$ which is 5% of the maximum voltage. See Figure 7.3 for a visual of Vmax over time:

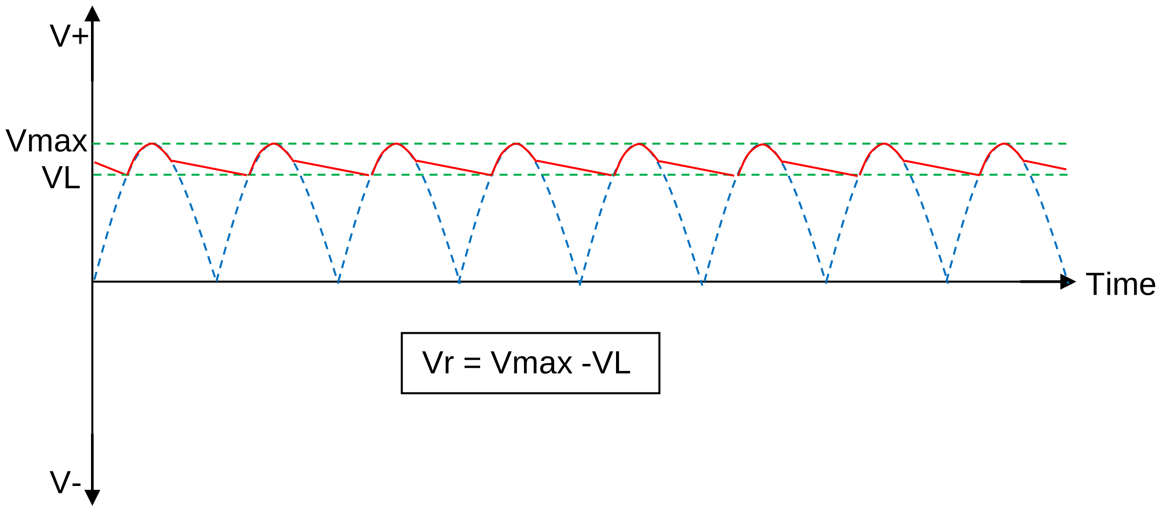


Figure 7.3 Ripple Voltage after Smoothing Capacitor Filter.

The smoothing capacitor can be tested with an oscilloscope across the two capacitors. The ripple voltage can then be determined on the oscilloscope to see if in fact we have a ripple voltage approximating to 1.6 V. Please see Figure 7.3 for further detail concerning the ripple voltage.

7.1.5 Motor Voltage Regulator Testing

The Motor voltage regulator can be tested by first determining if the voltage divider resisters, Ruv2 and Ruv1, will produce a voltage greater than 1.25 V for the UVLO pin. This can be done with a voltage meter. If the input to UVLO, which stands for under-voltage lockout programming, is less then .4 V, then the regulator will go into shutdown mode with all of its functions disabled. If UVLO is greater than 0.4 V but less than 1.25 V, then the regulator is in standby mode with the VCC regulator operational but the SS pin will be grounded and no switching will be found at the H0 and L0 outputs. If the UVLO voltage is above 1.25 V, then the SS pin is allowed to ramp and then the pulse width modulation gate drive signals are delivered to the H0 and LO pins. Next is to test the FB pin. The FB pin is the feedback inverting input of the internal error amplifier. The feedback resistors, Rfb2 and Rfb1, determine the voltage applied to the FB pin and requires a regulation threshold of 0.8 V which can be determined with a voltage meter. Next we need to test the RAMP and the AGND pin to see if a square wave pulse-modulated signal is present. This can be tested with an oscilloscope. The RAMP and the AGND pin set the pulse width modulation ramp slope, which is determined by properly selecting the external resistor and capacitor R ramp and C ramp. These two components emulate the AC component of the inductor L1 with a slope proportional to the input supply voltage. Next we determine if the following pins are within their absolute maximum ratings. Below is the configuration of the motor voltage regulator tests depicted in Table 7.1.

NAME of pin	PIN #	Voltage limits
VIN to AGND	20 to 6	-0.3 to 45V
SW to AGND	17 to 6	-3.0 to 45V
HB to SW	19 to 17	-0.3 to 15V
VCC to AGND	16 to 6	-0.3 to 15V
HO to SW	18 to 17	-0.3 to HB+0.3V
LO to AGND	15 to 6	-0.3 to VCC+0.3V
FB, DEMB, RES, VCCDIS, UVLO to AGND	8,2,3,7,1 to 6	-0.3 to 15V
CM, COMP to AGND	10,9, to 6	-0.3 to 7V
SS, RAMP, RT to AGND	4,11,5, to 6	-0.3 to 7V
CS, CSG, PGND, to AGND	12,13,14, to 6	-0.3 to 0.3V

Table 7.1 Motor Voltage Regulator Voltage Tests

7.1.6 Microcontroller Voltage Regulator Testing

The voltage regulator testing involves three tests: Ripple filtering, Vout vs. Vin and Efficiency. To test for ripple filtering, we test the input of the LM7809 regulator with an oscilloscope to see if the voltage ripple is at a minimum, if not, then we must troubleshoot the smoothing capacitor first. Second, we measure Vout, with a voltage meter, over a wide range of input voltages within the recommended voltage range according to the LM7809 data sheet, to see if the output voltage changes, if the output voltage changes, then the LM7809 voltage regulator is faulty. Finally to test the efficiency, we set up an ammeter on both the input and output of the regulator.

We then measure the current going in the regulator and the current going out through the load “microcontroller circuit”. Then we calculate the input power by multiplying the input current by the input voltage and the output power by multiplying the output current by the output voltage. The efficiency is then the ratio of the output power "P_{out}" to "P_{in}"

$$\text{Efficiency} = \frac{P_{out}}{P_{in}} * 100\%$$

We would like the efficiency to be as close to 100% as possible which will indicate less power loss.

7.2 Motor Testing

DC brushless motors are not really difficult to test, and can quickly be checked out for any problems. In order to test the motor, a drill with a chuck, an oscilloscope and an AC volt meter are required. Brushless motors are three phase motors, or perhaps a synchronous three phase motor with a variable speed electronic drive called the ESC unit. You can look at a three-phase voltage wave-shape signal from the brushless motors with a four channel Oscilloscope, and three resistors (star) wired in to provide a neutral connection for the scope.

The first step is to disconnect the motor (ISOLATE ALL THREE WIRES FROM EACH OTHER!) from the ESC, chuck the shaft into the drill, and slowly wind it up while holding the body of the motor with the three lead wires. The rotation should be as smooth as possible, with no rattling or noise; there should as little drag as possible. Run the motor with the drill at full speed (1000 RPM or higher if possible) for a minute or two. As the motor is spinning, feel the motor with your hand; nothing inside the motor should get warm or hot.

If the motor does get hot, then this indicates that you got a shorted winding or two or a magnet might have come loose dragging against a stator inside the motor, which indicates a real problem. If the “bell” of the motor wobbles while the motor is running, then the magnets have a potential of hitting the stator.

The second step is to isolate the three motor wires from each other and label them as #1, #2, #3. Next we set the voltmeter to AC range, and connect it to terminals #1 and

#2, wind up the drill to full speed and write down the AC voltage. We repeat the same step to terminals #2 and #3 and again on terminals #3 and #1. All three voltage readings should be almost identical, provided the drill is running at the same speed on each test. Next we connect an oscilloscope to the three leads with a 1 KΩ resistor in series on each lead to act as a limiting current resistor and see if the generated signal is 120° out of phase, with the same amplitude and no distortion. Amplitude variations indicate turn imbalances between poles which can reveal weak magnets in the rotor and/or spacing (magnet to stator clearance) variations or a (badly) out of round rotor, assuming turns are known to be correct.

The third step is to test for drag. First we short out all three wires together directly at the ends of the motor windings. Then we slowly wind up the variable speed drill while holding the motor's body.

The drag felt should be very even with little variation or chugging. Fourth step is to check for any short-circuit between the windings of the stator by setting the multimeter to continuity testing. Fifth and final step is to check the no load current of the motor and see if it passes the manufacturers specifications.

7.3 LED Testing

To test the LEDs, we need to see if the LEDs can output the colors we need. Our microcontroller will need to write to each LED in the prototype and consequently control red, blue and green and various other RGB combinations. Below is table 7.2 on which color we would like the LEDs to output.

Test Color	0XFFFF	0X0000	0XF800	0X07E0	0X001F

Table 7.2 LED Color Scheme

We will test these colors above in various sequences and also singularly. So in order to test the LEDs, we need to build a small-scaled prototype with enough capability to test those colors. Another test for the LED array can involve varying the flashing speed and determining if the LED controllers can indeed be updated if we need the rate of flashing to be faster or slower.

The design for the LED array prototype will be similar to the how the full scale LED array will work. We will need to create the design to implement the test procedures while trying to emulate what our full scale LED design will do. For the prototype, we can connect the shift registers to the LEDs and the microcontrollers to the shift registers, just smaller scale instead of all the LEDs on the ring structure. Once the small-scaled prototype is made, we can test it.

7.4 Structure testing

The main structure is the LED ring, ring support shaft, metal housing frame, support bar and plexiglass encapsulation all assembled in its place. What we will need to look for is to see if there is any oscillation or excessive vibration of the structure, especially the LED ring and the ring support shaft, due to the motor running. Any vibration can add stress on the motor, the microcontroller circuitry inside the LED ring including the LED themselves and excessive vibration can distort the POV image display. Any excessive vibration is the result of an unleveled structure. To test this, we use a construction water level to level out the metal frame and plexiglass housing by placing the structures on a near-perfect leveled plane. If the structures are leveled, then the next step is to test and see if the motor is perfectly aligned vertically. This can be done with a construction water level resting on the shaft of the motor.

7.5 Microcontrollers and PCB Testing

The plan to test the microcontrollers and the PCB together will be based on a checklist. We will test certain scenarios and allow for constant debugging. We will check the performance of the algorithms as well as the outputs of the LEDs. The test plan will be based on steps, what the expected result should be and other notes. A similar test plan can be implemented for the PCB in order to assure the correctness of the functionality of the device. Below is Table 7.3 depicting the steps and the expected results of our test.

Step	Procedure	Expected Result	Other Notes
1	Development board correctly functioning with necessary libraries	Proper functioning and programming code uploading	Debug often.
2	Check power source	Algorithm take correct decision based on value captured	Arbitrary check
3	Check motor speed	Algorithm takes and constantly checks value or default one is assumed	If speed is too low, LEDs will not flash.
4	Check GUI	Algorithm takes in value and reroutes to display code	
5	System Output	Algorithm saves entered value to be displayed	

6	Display	Algorithm takes saved values and outputs to LEDs	
7	Repeat Steps 3-5	Algorithm can take in different values and then respond accordingly	
8	Repeat step 6 using step 5 result	Algorithm will determine new display using new values	

Table 7.3 Microcontroller & PCB test plan

7.6 Bluetooth and USB to TTL Adapter Testing

As it was previously discussed, it is common to encounter communication problems when wireless devices are used; therefore, testing this module is imperative because the user interface is based on this function. The JY-MCU Bluetooth module is the least expensive Bluetooth module on the market, but the downside is that documentation about this device is very limited. Using an oscilloscope, we are able to determine accurate specification for this device, so we can counterbalance the documentations issues. In addition, the device firmware has to be updated to establish reliable communication with the microcontroller. Consequently, communication tests have to be performed to evaluate any possible communication issues; characters have to be sent contemplating any possible scenario to avoid complications after the PCB is manufactured.

On the same token, PL2303HX USB to TTL Convertor Module is not expensive, but there is no documentation about this device. Similarly, the same precautions and testing procedures that are applied to any Bluetooth module, will have to be applied to this device specifically because of the type of configuration we have chosen for it (Figure 4.26).

7.7 Software Testing

The most important function of the software is to display images, and this task is executed through the algorithm written, and the sketches uploaded to the microcontroller. On the same token, the software is going to enable the microcontroller interaction with the Bluetooth module, USB to TTL adapter, shift registers, microcontrollers, and the LED's; therefore, testing through each of this interactions have to made. For the modules, testing procedures have been discussed; however, for the interactions with the shift registers and the other microcontrollers, communication testing is necessary, and we accomplish this, again, using the oscilloscope to evaluate the I^2C protocol, and also to check values of the data pin, clock pin, and latch pin when we are using the shift registers. After the communication and the data transmission tests are performed, the most important test takes place, and during this test the 270 outputs (LED's) will be evaluated while they are mounted on the ring, but they are not rotating. This test will proof the effectiveness and synchronization of software, which are most important aspect for the displaying procedure. Finally, the last test will be executed when the ring is mounted and rotating and at this time all the algorithms will have to be uploaded to the microcontroller, and by displaying the light sequences, we will test and solve any timing and synchronization issues we can possible encounter.

7.8 GUI testing

In order to test the Graphical User Interface, we will use the requirements specified in the research section. What was described in the research section for the GUI was that the GUI had to be clear, user-friendly and easily accessible. We also listed options that we would include in the GUI that the user can control. So we will test the communication between the GUI and our prototype as well as the user controllable input. Testing the GUI will require a prototype that is connected from PC to the prototype to control the GUI and test option control. The prototype must include the LED array. The procedure to test the GUI is described in Table 7.4 below.

Step	Procedure	Expected Result	Other Notes
1	Graphical User Interface displayed on screen, ready for user input.	Proper functioning and programming code uploading and ready to use.	
2	Check connection to prototype	Algorithm successfully checks for prototype.	Arbitrary check
3	Check single color run (option on GUI)	Prototype flashes RGBWBl on the LED array	
4	Check Stationary Time and Date (option on GUI)	Prototype outputs current time and date	
5	Check Scrolling Time and Date (option on GUI)	Prototype outputs current time and date and scrolls that information	
6	Check Stationary Name/Group# (option on GUI)	Prototype outputs names of group and number	
7	Check Scrolling Name/Group# (option on GUI)	Prototype outputs names of group and number and scrolls that information	
8	Test Text Box	Algorithm will take text that was	Numerous checks needed

		inputted in GUI and output it onto LED array	

Table 7.4 Graphical User Interface Test Procedure

8 Administrative Content

In order to have a successful project, we must manage our finances, deadlines and planning efficiently. Having a strong foundation in these areas whilst going through the steps of the project will speed the process up. Our goal is to lay out an administrative plan to help guide us through the various stages of this senior design project.

8.1 Budget

It is important to set a budget for our project. To understand the cost to fund such an endeavor is important so we know if our project is possible and feasible. Setting up a budget will help make our vision realistic and put a perspective on the range of our project. Below is Table 8.1 that displays a list of tentative parts that will be used as well as their prices and how many of each parts we will be purchasing and using.

Part list	**Cost per Part**	**# of Parts**	**Total Cost**
Fiberglass/Aluminum/Wood materials			
Flat Washers	$0.14	3	$0.42
Screws	$0.20	4	$0.80
Lock Nuts	$0.15	2	$0.30
Ball Bearings	$1.00	2	$2.00
Threaded Wood Inserts	$0.50	1	$0.50
Metal Pipes	$2.00	4	$8.00
Slip Rings	$0.20	2	$0.40
Copper Brushes	$1.00	2	$2.00
PVC Pipes	$1.00	1	$1.00
Metal sheets	$2.00	3	$6.00
Steel Frames	$1.00	12	$12.00
Rubber Foots	$1.00	4	$4.00
Bar	$5.22	1	$5.22
Switch for LEDs	$1.00	1	$1.00
Balsa Wood	$1.50	1	$1.50
Power supply			
Transformer	$56.00	1	$56.00
Motor Voltage Regulator	$5.70	1	$5.70
Microcontroller Voltage regulator			
LM7809	$0.71	1	$0.71
470 µf Capacitor	$0.63	1	$0.63
.1 µf Capacitor	$0.09	1	$0.09

1n4934 diode	$2.00	1	$2.00
Bridge Rectifier	$7.08	1	$7.08
Smoothing Capacitor	$80.00	2	$160.00
Wire			
Magnetic Wires	$9.95	1	$9.95
Resistors			
100 Ω resistors	$0.12	100	$12.00
Microcontrollers			
Atmel ATMEGA258P-PU	$3.50	1	$3.50
LEDs			
4 pin RGB Diffused Common Anode LED	$0.12	100	$12.00
Bluetooth Module			
JY-MCU bluetooth module	$8.20	1	$8.20
Electric Motor			
HobbyWing XERUN	$76.00	1	$76.00
Shift Register			
74HC595	$1.50	34	$51.00
USB Port			
PL2303HX USB to TTL converter module adapter	$2.40	1	$2.40
		Total	$452.40

Table 8.1 Budget table

8.2 Finance

We need to determine where are finances are going to come from. Is it going to come from our own pocket? Do we need to fundraise or seeking backing by potential sponsors? We are interested in seeking sponsorships to support our project.

A potential sponsorship that has come from the relationship between one of our group members and one of his family members that works for Siemens. The potential for this partnership at this time is yet unknown but we would like to pursue a working sponsorship with Siemens because they found interest in the POV LED Globe.

The only other option if no other sponsorships come to our aid, which is likely, then we would have to budget our project properly and then split the cost three ways equally between the three group members. Although, this might put a strain on the parts that we need for the project, nonetheless the functionality of the project will not suffer. When it comes time to decide ownership of the project, there are some options to consider. One potential option would be to donate the project to the University of Central Florida's Department of Electrical Engineering and Computer Science so further study can be conducted on it. Another potential option is to have one of the group members keep it if interested in furthering the project's potential as a business product. Likely, our group will vote on the future of this POV LED globe when the time comes.

8.3 Work Distribution

The work distribution was about choice, skillset, as well as interests. The three sections we decided to split the whole project up into were the mechanical aspect, the electronic aspect and the programming aspect. We are all majoring in electrical engineering and thus all have experience dealing with the electrical side of things. With that being said, we each chose our parts that we wanted to work on with this project and the distribution is shown below in Figure 8.1.

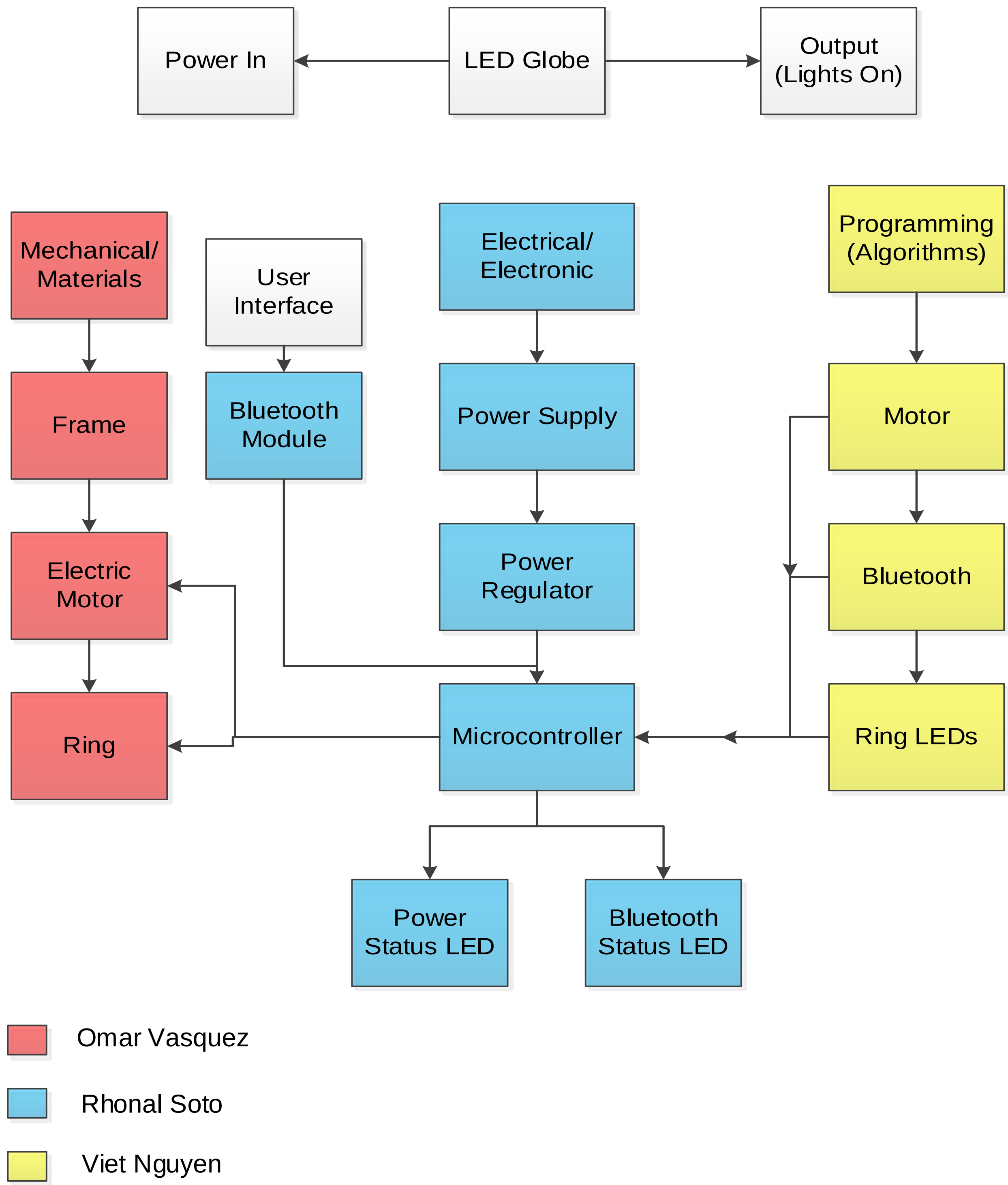
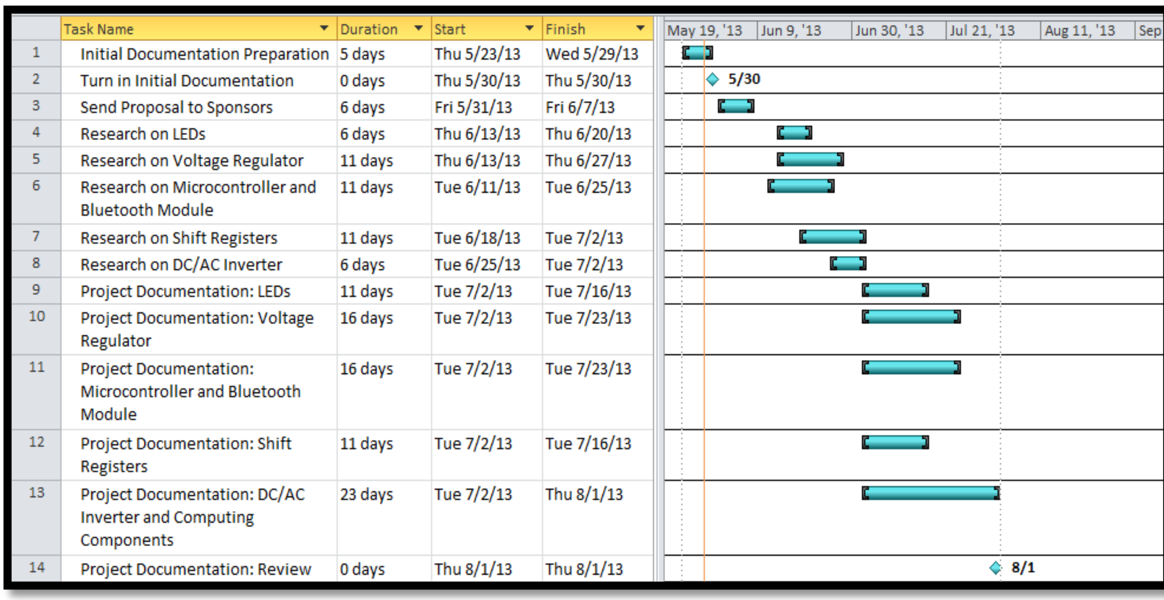


Figure 8.1 Block Diagram of Work Distribution and Legend

8.4 Schedule and Milestones Discussion

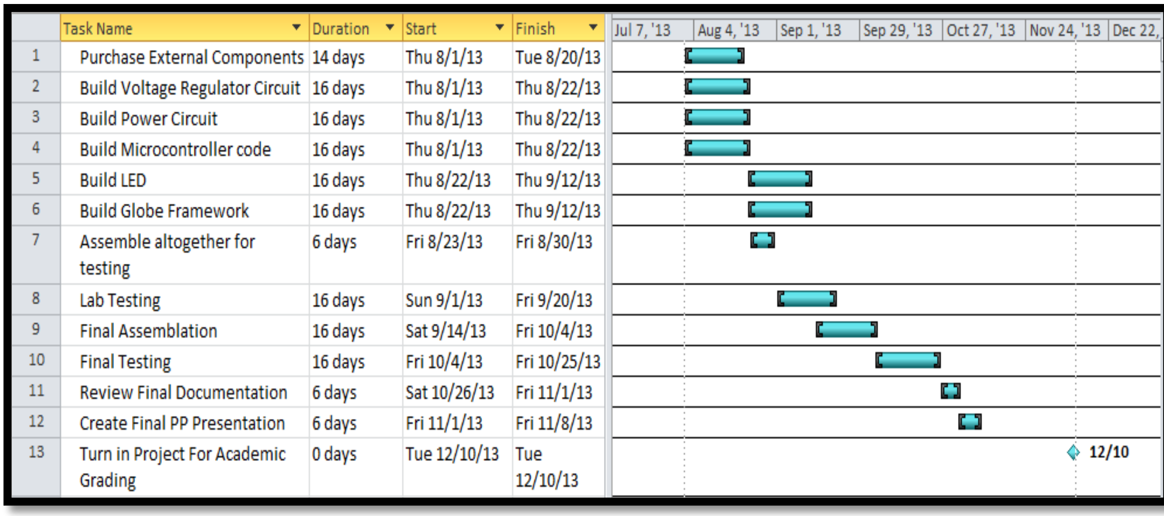
This project is encompassed around two semesters of work. In the first semester, which spans from early May to the beginning of August, we focus on the formation of the idea, the design of our concepts and the overall research and documentation of the whole project. Figure 8.2 below shows the various deadlines we set upon ourselves to keep up with the final Senior Design I report. The first half of the timeline includes research on all topics involving our project. The latter half of the timeline includes the documentation aspect as well as reviews. The research will create a strong foundation when we get into the hands-on portion of our project.



	Task Name	Duration	Start	Finish
1	Initial Documentation Preparation	5 days	Thu 5/23/13	Wed 5/29/13
2	Turn in Initial Documentation	0 days	Thu 5/30/13	Thu 5/30/13
3	Send Proposal to Sponsors	6 days	Fri 5/31/13	Fri 6/7/13
4	Research on LEDs	6 days	Thu 6/13/13	Thu 6/20/13
5	Research on Voltage Regulator	11 days	Thu 6/13/13	Thu 6/27/13
6	Research on Microcontroller and Bluetooth Module	11 days	Tue 6/11/13	Tue 6/25/13
7	Research on Shift Registers	11 days	Tue 6/18/13	Tue 7/2/13
8	Research on DC/AC Inverter	6 days	Tue 6/25/13	Tue 7/2/13
9	Project Documentation: LEDs	11 days	Tue 7/2/13	Tue 7/16/13
10	Project Documentation: Voltage Regulator	16 days	Tue 7/2/13	Tue 7/23/13
11	Project Documentation: Microcontroller and Bluetooth Module	16 days	Tue 7/2/13	Tue 7/23/13
12	Project Documentation: Shift Registers	11 days	Tue 7/2/13	Tue 7/16/13
13	Project Documentation: DC/AC Inverter and Computing Components	23 days	Tue 7/2/13	Thu 8/1/13
14	Project Documentation: Review	0 days	Thu 8/1/13	Thu 8/1/13

Figure 8.2 Senior Design I Milestones

After turning in the Senior Design I documentation, we will go into Semester 2 of the Senior Design course. We have planned out that the break in between the summer semester and fall semester is the perfect time to purchase and acquire and the parts needed for the project. The allotted time should be enough for ordering beforehand as well as waiting for shipping. Once we acquired all of the parts needed, we can start building and assembling our various parts together. In Figure 8.3 below, we state the various deadlines we set for ourselves in acquiring and assembling the parts we ordered. The design stage should run smoothly in order to give enough time for prototyping and testing the various functions and constraints of our project.



	Task Name	Duration	Start	Finish
1	Purchase External Components	14 days	Thu 8/1/13	Tue 8/20/13
2	Build Voltage Regulator Circuit	16 days	Thu 8/1/13	Thu 8/22/13
3	Build Power Circuit	16 days	Thu 8/1/13	Thu 8/22/13
4	Build Microcontroller code	16 days	Thu 8/1/13	Thu 8/22/13
5	Build LED	16 days	Thu 8/22/13	Thu 9/12/13
6	Build Globe Framework	16 days	Thu 8/22/13	Thu 9/12/13
7	Assemble altogether for testing	6 days	Fri 8/23/13	Fri 8/30/13
8	Lab Testing	16 days	Sun 9/1/13	Fri 9/20/13
9	Final Assemblation	16 days	Sat 9/14/13	Fri 10/4/13
10	Final Testing	16 days	Fri 10/4/13	Fri 10/25/13
11	Review Final Documentation	6 days	Sat 10/26/13	Fri 11/1/13
12	Create Final PP Presentation	6 days	Fri 11/1/13	Fri 11/8/13
13	Turn in Project For Academic Grading	0 days	Tue 12/10/13	Tue 12/10/13

Figure 8.3 Senior Design II Milestones

The planning that is put into this project will determine if it is a success or a failure. Adhering to this schedule and reaching certain milestones will be a big determinant towards our success.

Some of the major milestones that will define our senior design project as stated below in no particular order:

- Senior Design I Documentation Due
- Assembly of Prototype
- Project Design finalization
- Testing
- Senior Design II Documentation and Final Project Presentation

Some of the minor milestones that are less important to the project but will show growth in our group and individual basis and these must be completed before moving on to the major milestones are stated below in no particular order:

- Research
- Preliminary Design Review
- Prototype Fabrication and Testing
- Making of Foundation
- Making of Ring structure with LEDs mounted
- Documentation Design Review

8.5 Final Client Price

If it is decided that the LED globe can be a profitable venture by an outside source then we will need to determine a pricing point. To calculate the final client price, we will be using the cost-plus pricing strategy. According to Godfrey, the strategy "determine[s] the expense associated with producing a product and add an additional amount to that number to generate profit." Also, "Cost-plus pricing is a straight-forward and effective strategy because it ensures that all costs are covered before profits are calculated."[6]

Using the cost-plus pricing strategy, we need to calculate the cost of our system which includes the assembly and the cost of materials. We add a profit margin which is a percentage of the system costs and then determine how much installation will cost to get our total final price.

Final price = System costs + margin profit + installation costs
Final price = $450.90 + 0.2($450.90) + 0.1($450.90)
Final price = $450.90 + $90.18 + $45.09
Final price = $586.17

The final price of $586.17 will a suitable price giving us just about $150 in profit from selling and installing our POV LED Globe into the client's place of choice.