Department of Electrical Engineering and Computer Science
4000 Central Florida Blvd
Orlando, FL32816-2993

Senior Design Report

Multi-Functional Hexahedron:
An Interactive LED Cube

Group 5 Team Members:

Roberto Amaya

Luis Ferrer

Eury Reynoso

Julio Romero

Table of Contents

1.0 Executive Summary

The main goal of this project was to redeem the University of Central Florida from previous failed attempts to make a LED cube. Another goal was to have fun for the last two semester of higher education by doing a versatile project. The Multi-Functional Hexahedron is an interactive electrical system displayed as a three dimensional array of light emitting diodes, which features three modes of operation with the overall purpose to entertain its user with a visually appealing light animation. The fundamental objective of the project is to display a set of pre-programmed three-dimensional images in rapid succession, as to appear in the form of an animation. Furthermore, the cube also performs two additional functions based on mode selection. On “movable” mode, it makes use of an accelerometer to simulate behavior similar to a liquid moving on a container. On “listening” mode, it relies on a VU meter to effectively listen to sound, and it displays light animations according to the sound input from the user.

The main inspiration and theme of the project is Mother Nature. Some of the animations display many elements of nature, such as rain falling down, a snake rattling around, and more animations. In the rain animation, the light emitting diodes lights up in such a sequence that looks as though raindrops are falling and disappearing into the floor. The snake animation uses four LEDs that find their way through the cube. Additionally, the “movable” mode simulates the flow and movement of water. This part is implemented by an accelerometer that turns LEDs on and off in a water-like fashion as movement is sensed. It represents the fragile stability of the water by reacting to any movement that a spectator applies to the cube. The “listening” mode reacts to ambient sounds. It is implemented by a volume unit meter that will sense the ambient sound to later be transferred and visualized as a spectacular light show.

To turn imagination into reality, four students from the University of Central Florida Senior Design class set out to form group five, with the main objective to overcome the entire obstacle. Roberto Amaya, Julio Romero, Luis Ferrer, and Eury Reynoso, the three first pursuing a Computer Engineering degree and the last pursuing an Electrical engineering degree, are the producers of this project. Within the boundaries of the two semester period, they were able to design, construct, test, and present the LED Cube. Throughout all phases of the process, the group obtained a great deal of experience pertaining to custom integrated circuit design, embedded programming and working with complex electrical gadgets and components. In the journey, they used many resources, such as previous class knowledge, experience from acquaintances and professors, as well as obtained much specialized knowledge that can be only acquired by doing a long project of such magnitude from start to finish.

The team first designed and built a cube with five hundred and twelve, three millimeter, blue diffused LEDs with a final dimension of eight square inches and eight by eight by eight LEDs. Due to time constraints, the team thought they would not be able to finish an RGB LED cube. Fortunately, the team was also able to design and present another cube with the same dimensions but uses five millimeter, four pin, clear Red-Green-Blue LEDs. The final design is enclosed in a cube built from a clear, shatter resistant, polycarbonate sheet that allows viewers, especially children, to lift up and manipulate the cube in a safe environment. The only two electrical components that are exposed to user reach are the power cord and a small switch used to select the different cube modes.

The main contents of the document include: research pertaining to previous designs, details about related components, detailed discussion of our design, final implementation and testing of the project. Also included is the administrative content, as well as relevant sources applied.

2.0 Project Description

2.1 Introduction

These mesmerizing structures built by formatting light emitting diodes (LED) into a cube, have become all the rage within the electronic hobbyist community. Cubes starting from 2x2x2 in dimension can become as large as 32x32x32 LEDs. They serve no functional purpose other than to be aesthetically pleasing and to wow friends and family. Majority of these LED cubes, leave the user standing at a distance admiring the three dimensional patterns. Our goal was to bring a new twist to the LED cube and make it an interactive user experience. Bringing different technologies together, the user is be able to pick it up and do more than just observe from a distance.

The plan was to construct an 8x8x8 cube of single colored LEDs. The cubes main function would be to simulate three dimensional animations by lighting up the LEDs in a pattern dictated by a microcontroller. Upon completion of the primary task, two more functions were added. The second function made the LED Cube's lights mimic liquid in a cube based on the input from an accelerometer. The third function had the LED Cube become a Volume Unit (VU) meter. Due to the success of the prototypes, more features were added. This gave way to construct the physical RGB cube. The Challenge was having the different technologies providing input to work in unison and give the proper visual output, as well as to make the cube itself sturdy enough to be picked up and handled by individuals.

This document discusses the development processes undertaken for the completion of this project. This includes research, architecture, software and construction. Technical background knowledge in digital and embedded systems is required to fully understand the ideas discussed in this document.

2.2 Motivation for the Project

Interest came about when hearing about the nearly failed attempt to create a 16x16x16 LED cube by a previous senior design group last summer. Also to top the non-impressive LED cube that was picked up and showcased this spring. Never having heard of such a contraption, the group searched the internet for further information. Having seen various LED cube projects of varying magnitudes, and the amazing capabilities of the grand 32x32x32 LED cubes. It was decided that this was a very captivating project if managed correctly. The LED cube seemed a little too simplistic at first glance, so we thought of different ways of make this project more interesting. The VU meter and color organ found on many audio electronic devices inspired us to apply this feature to the LED cube. This would make it an interesting center piece display at parties as the cube reacts to the sound of music. Another idea was based on a previous group that integrated an accelerometer to an LED cube. This allowed users to pick up the cube and shake it around having the lights mimic a liquid physics based on the outputs from the accelerometer. Another reason for selecting this project was that it would be programming intensive. This would allow for equal distribution of work since the group consists of three computer engineers and one electrical engineer. Being poor college students, a monetarily feasible project was of great desire. The Multi-Functional Hexahedron will require personal spending since sponsor was unobtainable at the time. But it is perceived that this project will cost significantly less than some $1,500 automated chess project to build. This will probably run about a few hundred dollars the most, the printed circuit boards be the most expensive components. Ultimately this seems like a reasonable, flexible, inexpensive and fun project to take on as we spend our final semesters in undergraduate study.

2.3 Goals and Objectives

The ultimate goal was to create an interactive RGB LED cube, encased in a clear acrylic glass container that would allow anyone to pick it up and rotate it around; as well as have the cube react to sound. A single pole triple throw switch would be located at the top or side of the casing which would allow the user to change the state of the cube’s behavior (display pre-programmed animations, a liquid physics simulator and a sound reactor). An extremely difficult part of this project will be in regards to the programming of the microcontroller. The programming language used will probably be C due to familiarity. With three computer engineers in the group, it should be manageable task to handle. Another important feature would be to have an input method for uploading animations.

This was achieved by dividing the project into sub-goals. The first was to perform extensive research on the subject matter. This included researching past projects and tutorials, as well as researching possible components. Following the research, a prototype of a 3x3x3 LED cube was constructed to test components and learn how to implement the ICs. The goal here was to obtain an understanding as to how the entire process works. This also provided a platform to achieve the next set of goals. The following objective is to gain complete understanding as to how the accelerometer, VU meter and LED sink drivers function. We can use the 3x3x3 cube to test these devices individually before application to the 8x8x8 cube. Once the final design with breadboards was functional, the printed circuit diagram was finalized and ordered. Once that was accomplished, construction of the final prototype commenced. After completion of the RGB cube, it was tested to make sure it met all the requirements set at the beginning of our class.

2.4 Project Requirements and Specifications

There are five major components in the system. The first being the LED cube which displays the output of the computed input from the sensing devices. The LEDs light up based on the calculated input data from the surrounding environment. The computations and processes are dealt with in the second major component being the embedded electronics, which includes the microcontroller, IC units and SPTT switch. The next three components deal with input into the entire system. This includes the accelerometer, the VU meter and the image code. Figure 2.1 below demonstrates how the components relate to each other as a whole in a simple block diagram.

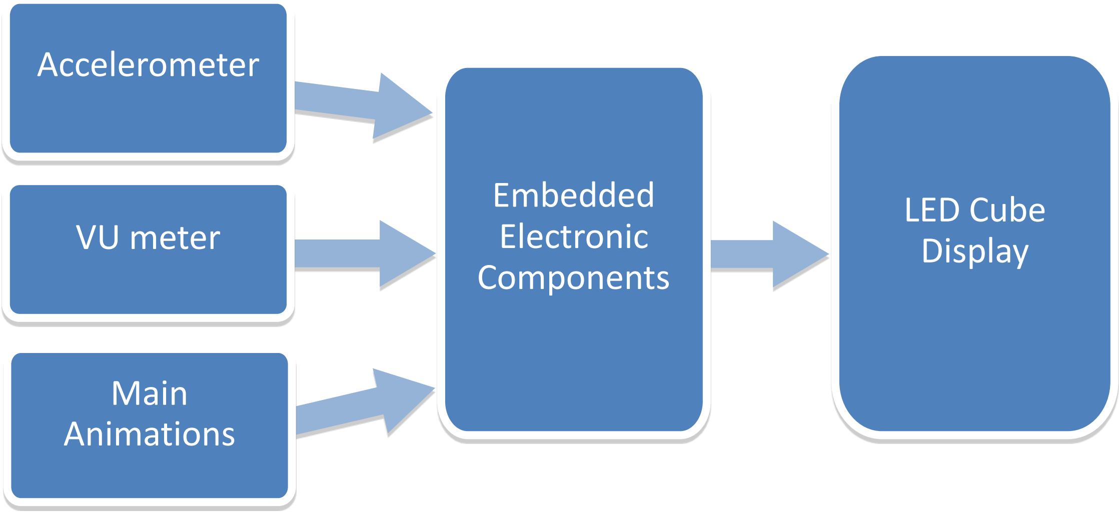


Figure 2.1: Block Diagram of the System

In the next four subsections we will be discussing the main specifications for the LED cube project.

2.4.1 Physical Specifications

The following specifications explain the physical dimensions the cube adheres to for this project:

- The LED resolution will be 8x8x8 = 512 minimum. Having five visible faces
- Blue light emitting diodes will be used
- Total mass should be approximately 10 lbs. or less
- The size will be approximately one cubic foot
- Electronics will be housed underneath the LED cube structure
- Casing will be made from transparent acrylic/Plexiglas
- FTDI cable connection should be accessible from the outside
- Three position switch accessible to the user
- PCB encasing should be made out of a light weight material

2.4.2 Environmental Specifications

The following specifications explain the environmental guidelines the cube adheres to for this project:

- System should work in a standard dry indoor environments
- Should withstand vibrations caused by sound waves at high volumes
- Function in possible humid/moist outside environments
- Withstand temperatures between 50˚ F – 100˚ F
- Be able to sense acceleration
- Be able to sense sound

2.4.3 Hardware Specifications

The following specifications explain the hardware requirements the cube adheres to for this project:

- Components must run off of 5V DC or less
- Have an on/off switch
- Method of changing state of the LED cube to one of three
- Method of communicating with PC
- Reliable microphone for registering sound
- Device to measure movement
- Bright LEDs

2.4.4 Software Specifications

The following specifications explain the software requirements the cube adheres to for this project:

- Program should not be more than the available memory on the microcontroller
- Have a minimum of three modes/states in which the LED cube will be in any given time
- Source code will be written in C programming language
- Be uploaded through a FTDI USB connection

3.0 Research Related to Project Definition

3.1 Existing Similar Projects and Products

It’s difficult to come across new ideas in today’s day and age, as they say “there is nothing new under the sun”. For example the LED cube has been repeatedly done with many different tutorials found online. In order to impress, the cubes are made larger and more colorful using more and more complex algorithms. Few people have gone up and beyond in attempting to do more with the LED cube than just make it a visual spectacle. There was a group that implemented an accelerometer and a different group that used a VU meter. What was accomplished is the combination of both projects into a super ultra-mega LED cube. This section discusses the mentioned projects’, along with some others that have contributed to our design. The idea is by learning from past attempts and failures the team is able to facilitate the creation of an original LED cube with multiple functions, something that has yet to be done that has not been posted online already.

3.1.1 8x8x8 Instructables LED Cube

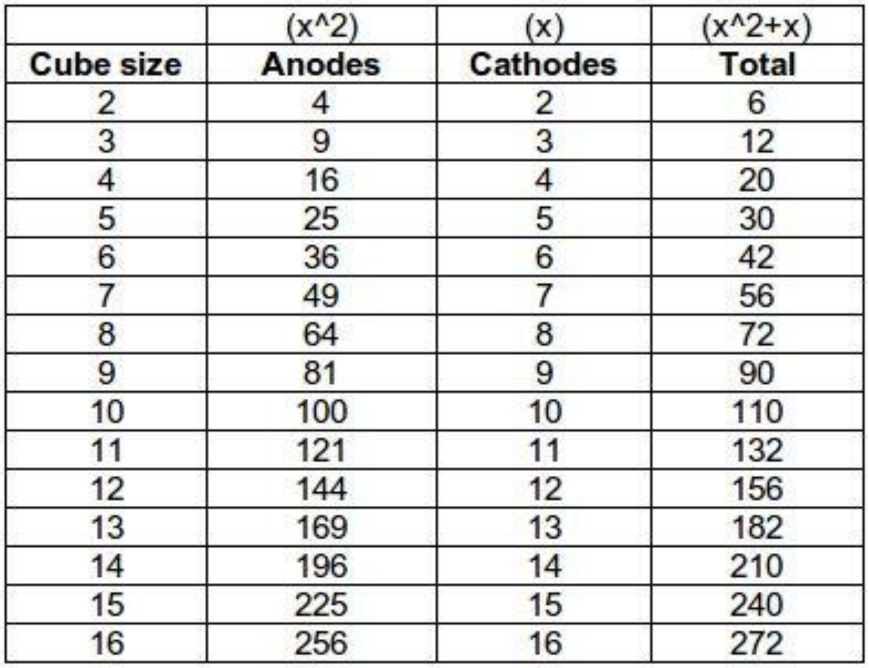
One of the first sites that we encountered in our research was instructables.com. The site provided a very thorough step-by-step instruction set to build an LED cube. From gathering of the materials needed, to theory of operation and detailed execution of each step, the guide was of great use to the group as we were able to gain broad insight on the subject. According to the project author, the build took about 4 days for the completion of the cube, and another 4 to 5 days were spent solely on the programming of the light animations.

The owner of this project suggests a few set of skills required in order to engage in this project, including a basic understanding of electronics, programming skills, soldering skills, and others. The owner also strongly did not recommended building an L.E.D. cube as a first electronics-intensive project. However, seeing as our group is composed of electrical/computer engineering senior students, this did not pose an issue at all, as we have had plenty of exposure to many electrical/software related projects.

In order to be acquainted with the inner workings of an LED cube, one must first understand the anatomy of such a device. A light emitting diode cube, as suggested by the instructables.com website, can be thought of as a device mainly composed of columns and layers. The positive cathode legs of every LED are soldered together forming a layer, and this is done for every layer. In the same manner, each of the anode legs in one column are soldered together. This setup allows for an efficient control of the lighting of LEDs. Each of the 64 columns is connected to the controller individually, and also each of the 8 layers have a connection to the controller board. The author of this project makes use of the transistors in the circuit to control the flow of current to one particular layer; to produce the 3D image, the transistor from the current layer must be turned off, the image for the next layer must be changed, and finally the transistor for the next layer must be turned on.

This cube is composed of 512 LEDs. Having a dedicated port to control each LED would mean needing a microcontroller with 512 I/O ports, as well as having 512 wires running through the cube. This would be very impractical and would diminish the visual value of the finished product. The way that the author addressed this issue is by taking advantage of a concept known as Persistence of Vision (POV). Persistence of Vision is the phenomenon of the human eye by which an afterimage is thought to persist by a fraction of a second on the retina. This means that if this concept was to be applied to the project, then if an LED is flashed really fast, the light will stay on the user's retina for a small amount of time after the LED turns off. By extension, if each layer of the cube is flashed individually one after the other really fast, it will create the illusion of a 3D image. This will allow the setup to only be composed of 72 IO ports (64 for the anodes and 8 for each layer) to control the cube, therefore solving the issue of having too many ports/wires for the LED cube, and making it more visually appealing.

As far as driving the LED cube, two sets of IO ports are going to be needed. The author discusses that one set will be used to source the anode columns, while the other to sink all the cathode layers. Following is a table provided by the site, table 3.1 in which the amount of anodes and cathodes relative to the cube size is shown:

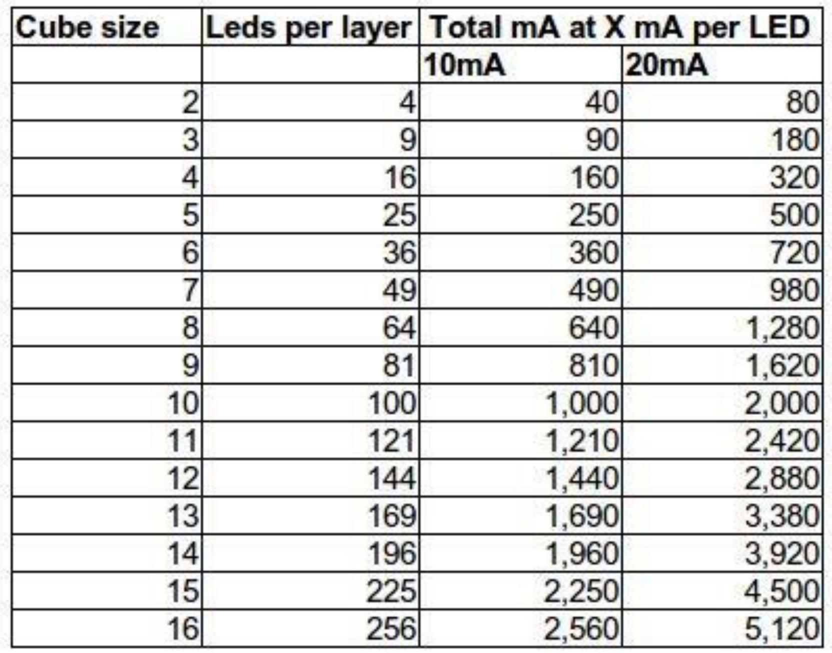


	(x^2)	(x)	(x^2+x)
Cube size	**Anodes**	**Cathodes**	**Total**
2	4	2	6
3	9	3	12
4	16	4	20
5	25	5	30
6	36	6	42
7	49	7	56
8	64	8	72
9	81	9	90
10	100	10	110
11	121	11	132
12	144	12	156
13	169	13	182
14	196	14	210
15	225	15	240
16	256	16	272

Table 3.1: Cube size table
Printed with Permission from Instructables.com [1]

As seen from the previous table, the number of IO ports will increase exponentially depending on what the size of the LED array will be. This is an important point to take in consideration, and greatly influenced our decision on working with an 8x8x8 cube. For a bigger cube array, the number of components would have increased drastically, and therefore would have required a much greater investment in both time and money. Furthermore, as our group has decided to increase the scope of our project features by adding an accelerometer to display water effects when tilting the cube, a bigger cube would have not allowed for the desired maneuverability and handling to properly demonstrate this effect.

Another important factor that the author suggests to take in consideration is the amount of current that will be flowing through each individual LED, as well as the other components of the circuit. If the scope of the project was to build a smaller cube (a 3x3x3 or 4x4x4 cube, for instance), this would not pose much of an issue, as each cathode layer could be connected directly to a microcontroller IO pin. However, since the scope of the project is a bigger, 8x8x8 cube, the current going through each IO pin will be too high, and there would be a risk of damaging any component. To get around this issue, the author is helpful enough to provide a table overviewing the power requirements for a LED layer of many different sizes. In particular, we are able to observe that for cube of size 8, a current draw of 640 mA will be required. Following is the table with power requirements:



Cube size	Leds per layer	Total mA at X mA per LED	
		10mA	20mA
2	4	40	80
3	9	90	180
4	16	160	320
5	25	250	500
6	36	360	720
7	49	490	980
8	64	640	1,280
9	81	810	1,620
10	100	1,000	2,000
11	121	1,210	2,420
12	144	1,440	2,880
13	169	1,690	3,380
14	196	1,960	3,920
15	225	2,250	4,500
16	256	2,560	5,120

Table3.2: Power Requirements for different cube sizes.
Printed with Permission from Instructables.com [1]

It is important to note that the above table, Table 3.2, shows the current draw with all the LEDs turned on in a particular cube.

Naturally, the next component to take in consideration is what type of microcontroller is to be used in the project. As stated before, it is not feasible to acquire a microcontroller that has at least 512 IO pins in order to control each LED individually. The way that the author approaches this issue is by using multiplexing to control the entire cube. The multiplexer that was used for this Instructables project was an 8-bit latch (74HC587), which will effectively multiplex 11 IO lines into the required 64 output lines for each column. This integrated circuit could be a potential candidate for the group's approach, because we are dealing with same size cubes. The 74HC587 is composed of the following pins:

- 8 inputs (D0 – D7)
- 8 outputs (Q0 – Q7)
- 1 "latch" pin (CP)
- 1 output enable pin (OE)

Following is Figure 3.1, a schematic depicting the arrangement of the latches in conjunction with a 74HC138 decoder:

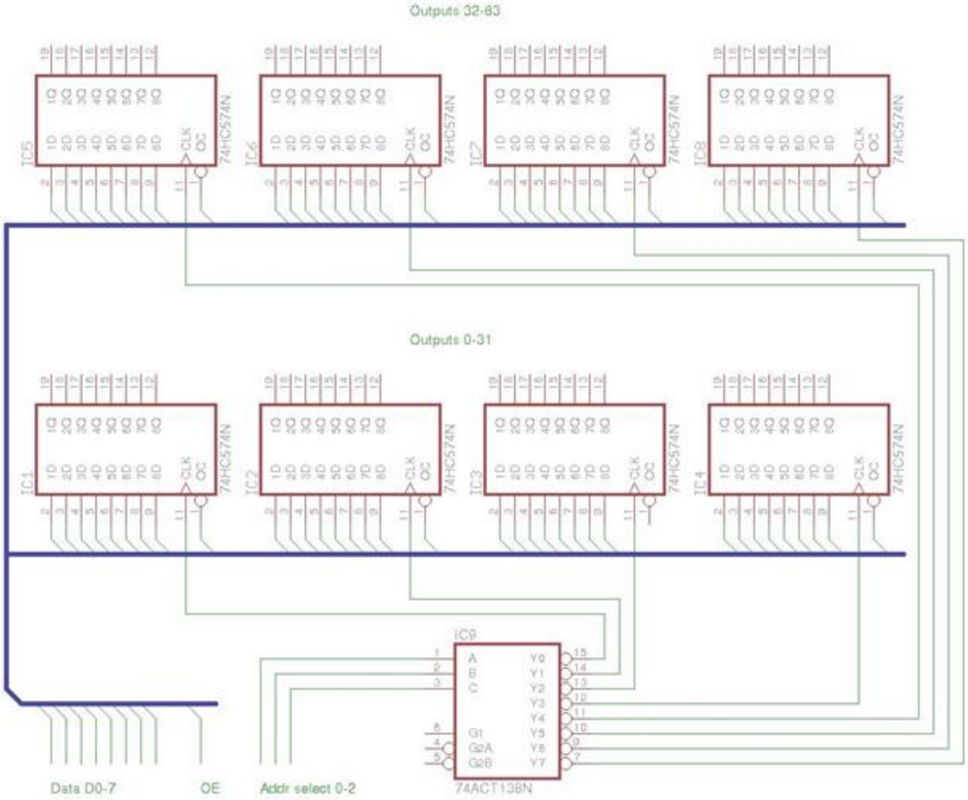


Figure 3.1 Schematic for 3-to-8 decoder and latches used to control LEDs Printed with Permission from Instructables.com [1]

In the previous figure, the anatomy of the circuit is laid out. The outputs of the 3-to-8 decoder connect to the clock of each latch. Each latch will serve as a kind of simple memory for the cube, holding 8 bits of information which are to be represented on the output pins. Also, the inputs for all of the latches are connected together using a bus. This way, the author utilizes the set-up in order to present an animation in the following set of steps:

1) Data is loaded onto the input lines
2) Latch is set to high, loading data to output lines.
3) Latch is set to low.
4) Load next state of the cube
5) Repeat until complete

The author goes on to describe the decisions behind the choice of other very important components that would comprise the LED cube, along with the factors that were taken in consideration to influence such decisions. From the importance of a suitable power supply to the size and shape of light emitting diodes to choose from, the author does a thorough analysis of each component. Making the right choice for each component is very important if one was to end up with a visually-pleasing, and most importantly, a working LED cube. More information of component research and decision is found within the Component Research section of this document.

The next few sections of the author’s project are about the actual building of the cube, implementing the software to display the LED animations, and finally putting everything together to end up with the fully functional cube.

In essence, the construction of the actual cube really comes down to a lot of patience, and a steady hand. In order to make an aesthetically-pleasing LED cube, the author suggests that it is of utmost importance for it to be perfectly symmetrical; that is, each light emitting diode will have to be facing the same way, and both the cathode and anode legs of each led have to be roughly the same length so that the final product comes out as an almost-symmetric cube. This can be achieved by making use of a temporary soldering rig.

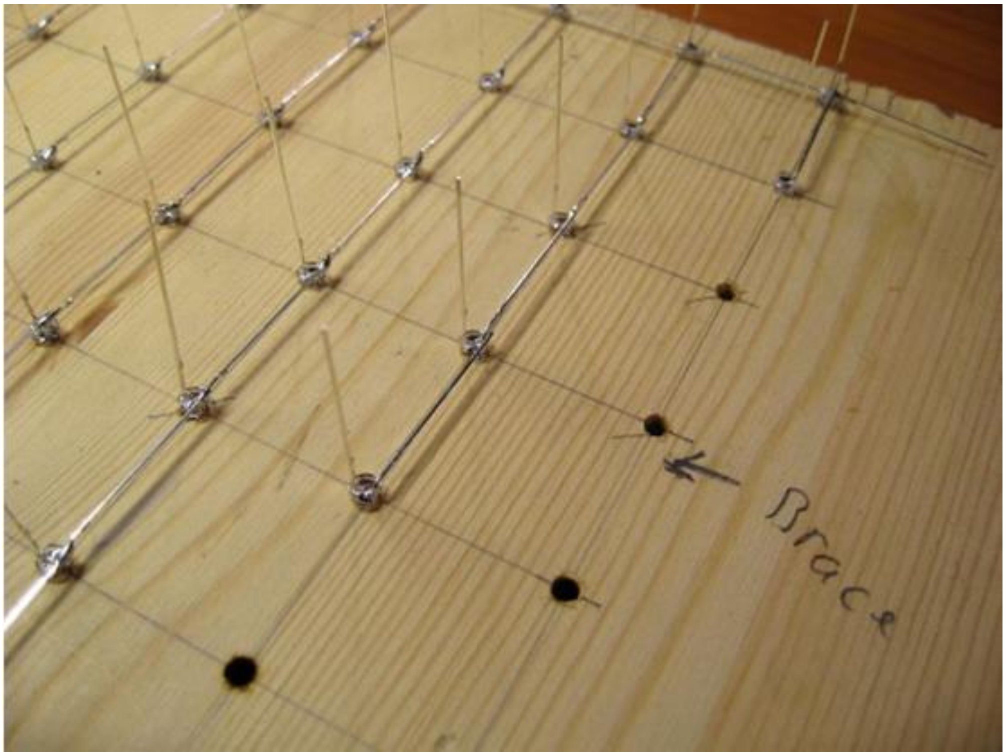


Figure 3.2: Using a piece of wood as a soldering jig for the LED cube. Printed with Permission from Instructables.com [1]

As seen in the previous figure, Figure 3.2, a piece of wood is a perfect candidate for a soldering rig. Holes can be drilled with ease, making sure that each hole is spaced according to the length of the anode leg. This simplifies the soldering job by a substantial amount, allowing the LEDs to be kept in place while everything is put together. Once all the soldering is complete, if the holes are drilled all the way through, the same wooden jig may be used as a stand for the final project.

Once the body of the cube was finished, the author attached the remaining circuit components, and then tested out the cube with some test code. The software for the cube was developed with an open-source compiler called avr-gcc, and the code itself was written in C programming language.

According to the author, it is very advisable to first start on a small scale model before actually engaging in the full project. The author worked in constructing a 4x4x4 dimensions cube in order to fully grasp the concept behind it, and get a feel for the actual project that was to come. Our group has come to the decision of following the same plan of work. Therefore, we will begin by constructing a 3x3x3 LED cube. This way it will be possible to get familiarized with the inner workings of the cube, as well as develop test cases to use in the full project, and also to implement accelerometer and UV functionality with this smaller prototype.

3.1.2 University of Portland Senior Design Project

This group applied an accelerometer to the traditional LED cube idea. In turn this has inspired us to make it a feature in our LED project. They used the free scale MMA7341L MEMS (micro-electro-mechanical systems) triple-axis accelerometer, which can register 3g to 11g of acceleration. Their main objective was to use it to light up the LEDs in a pattern that would mimic liquid in a cubical container. As the user would move the cube around, the LEDs would react to motion based on the input from the accelerometer.

The brain house of their cube which happens to be the same everywhere else on the internet was the Arduino Duemilanove Microcontroller development board. We plan to use a different microcontroller discussed later in this document. But it has provided some insight into qualities we looked in to for selecting the microcontroller. For example, the accelerometer produces output in analog which poses a problem for our digital circuit system. So an analog to digital converter will be necessary. The VU meter will probably also produce an analog output, so having an analog input on selected board will greatly facilitate the process, and reduce the number of components required. Also if other analog based devices are implemented, the conversion capabilities and pin count will be on hand. Another aspect of interest of the board is memory size. Much will be needed to house code that will allow multiple modes for the cube to be in. The animations code file will be the largest in size and require much memory. Being able to switch modes based on the input signals on one pin on the board controlled by a switch is a great idea. By directing the voltage to three different pins, three different modes can be created.

Another fascinating aspect of their project was the minimal number of parts used, compared to other home brew projects. The cube project in general involves much multiplexing and decoding. The bigger the cube, the more flip-flops and latches one needs. Further investigation revealed that the group had custom integrated circuit (ICs) chips produced by the MOSIS foundry. One type of MOSIS chip was the iterator which was used to iterate through the LED cube columns. The other being much more complicated acted as the sequential access memory and was called the SAM MOSIS. The design consisted of 1500 gates. This seems like a rather costly route to take for our team. This group on the other had had special funding therefore could afford to build custom chips.

The containment unit for their project inspired us to create a similar model. This team used the Arduino development board as mentioned before. This required the cube to have multiple connections at one time. The Multifunctional Hexahedron will have a reduced number of external connections, ideally one just for power. This project is one of the better ones found online but the use of LED drivers may have been a better option, which is what we considered. This served as a great jumping off point and we hope our project is an improvement to this project.

3.1.3 3x3x3 Pocket LED Cube

This particular LED Cube is the foundation for our LED Cube. We will be using this simple 3x3x3 pocket LED cube in order to better comprehend the workings of an LED Cube and how we can incorporate our own design as well as features. The pocket LED cube was built using ladyada's MiniPOV Version 3 kit, which includes 8 red LEDs, a preprogrammed microcontroller, a serial port connector and an AA battery case. The MiniPOV3 kit is originally meant to be a straight line of LEDS which can be programmed to follow any particular pattern the team uses. The team at Makezine, however, modified the layout of the LED's to match that of a cube.

The begin by soldering the LED's all together using a block of wood with the layout of the 9 LED's in each level. They drill the 9 holes and place the LEDs inside them upside down so that they can arrange the cathode and anode legs to the desired layout. They commence the soldering of the legs until they have one solid layer of LED's. They repeat the above steps three times until they have three layers of 9 LED's. From here they solder all of the layers together to have the finished LED cube, depicted in Figure 3.3 below.

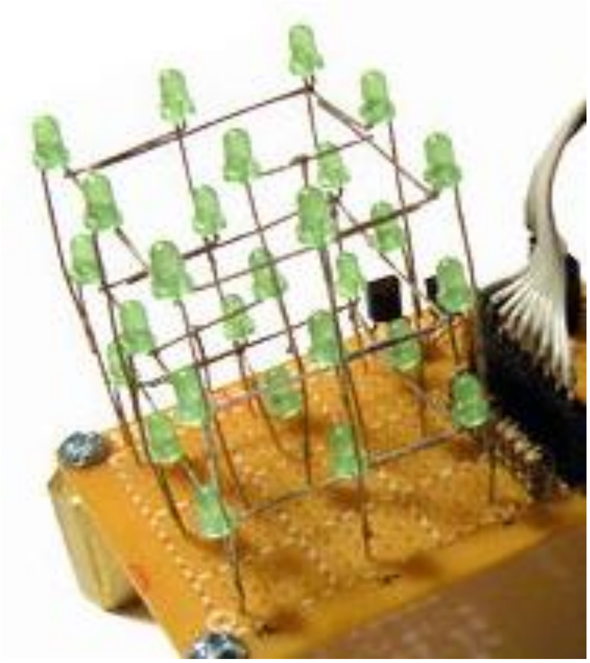


Figure 3.3: 3x3x3 LED cube.
Printed with Permission from Instructables.com [1]

Since the project requirements do not allow the usage of prebuilt kits, the kits included in this particular LED cube will not be used. Parts from this LED cube as well as parts from another LED cube project will be used to develop the first prototype of this project. The second 3x3x3 cube project that will be used as a guideline for our prototype does not use a prebuilt kit. It uses a low number of components and hooks up directly to an Arduino or in the case of this prototype, an MSP430 microcontroller. The list of parts required for this particular 3x3x3 LED cube are:

- 1 Perforated Board
- 3 NPN Transistors (2N2222, 2N3904, BC547, etc.)
- 12 Resistors (~220 ohms and ~10k ohms)
- 13 Headers (male or female)
- 27 LEDs
- Wire

Similar to the pocket LED cube, this LED cube uses the wood block jig in order to solder the three layers of 9 LEDs. Each set of nine LEDs will share a common connection amongst their cathodes (negative leads). Each of the nine LEDs on a level is connected to the corresponding LED on the other two levels through their anodes (positive leads). After the layers have been soldered, the three layers are soldered together. From here, the cube is placed on the perforated board with the appropriate spacing. Three stripped wires are connected to one leg of each of the center LEDs of each layer. The remaining part of the circuit is built using the following schematic, Figure 3.4:

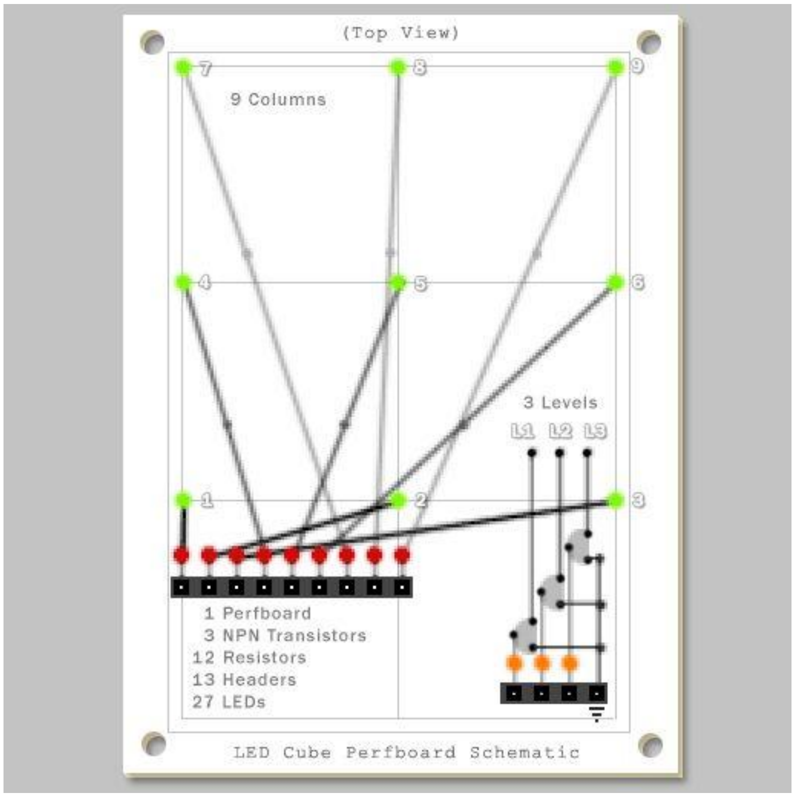


Figure 3.4: Schematic for Instructables 3x3x3 LED cube. Printed with Permission from Instructables.com [1]

As can be seen from the above figure, each of the nine columns will connect to a pin on the Arduino. This will be done through a current limiting resistor. Each of the three layers will be connected to ground through an NPN transistor when activated through the Arduino. The trick to this particular circuit is that only one layer will be allowed to be lit at a time. Therefore, the layers have to be lit up fast enough that it will appear to the human eye as if all three levels are lit at the same time. The actual circuit layout as well as the software for our 3x3x3 LED cube will be explained later.

3.1.4 3D RGB LED Cube (VU Meter)

During research, many volume unit meter projects using LED's were encountered. One of the designs was done as a homebrew assignment and presented on craigandheather.net. The design is a four by four preassembly RGB LED cube. Also, it was very specific about parts used and provided a complete schematic for the final circuit. It uses a condenser microphone and a LM324 op amp for the input. Then, the input is broken down into seven different frequencies. This part is done by a MSEGQ7, a seven band graphic equalizer that provides frequency band as output. The microprocessor used was an ARDUINO Uno with an ATmega328 chip. No special reason was given for the use of the ARDUINO, it just happens to be available at the moment. For the output, the design uses two TLC5940 LED drivers that provide 16 channels and 12 bit grayscale PWM control. Since the LED drivers are more expensive than the demultiplexer, the author uses a 74LS138 decoder/demultiplexer to reduce the amount of part need. Finally, eight TIP126 driver PNP transistors are connecting to the demultiplexer output and to the LED cube. The circuit is fed by a 5 VDC, 3 Ampere wall wart power supply.

What makes this design special is the use of a seven band digital controlled graphic equalizer which used a digital output to eliminate the necessity of an analog to digital converter. The total cost of the project is not discussed in the page, but many of the parts were used because they were available from previous projects. Also, the author used a pre-built LED cube with PCBs to eliminate the wire and have better visual appearance. If the group decides to use this option in our final design, it will increase the final price. but this would allow the LED cube to have a stronger structure which will increase the performance of the accelerometer.

The weakness of the design is seen in the way how LED's reacted to sound. The author did not use the ARDUINO library to help out with the software code and the demonstration turned out to appear like a random flashes of LEDs, without any pattern. Also, RGB LEDs are not used in full, in this project regular LEDs should work perfectly fine. The design can be greatly improved upon by changing the programming of the microprocessor. In comparison with our project, we have the technical background and human power to implement a better program.

3.1.5 How Not to Engineer

This individual went about the multiplexing of the LED cube in an uncommon manner compared to other projects around the web. The usages of shift registers were mostly implemented. In the best example found online, eight 74HC574 IC units were used to multiplex the Instructables' LED cube. The HowNotToEngineer (HNTE) cube used five STP16CPS05MTR LED sink Drivers, instead of eight. Even though this reduces the component count by three at a glance, when looking at the schematics of both projects as seen in Figure 3.5, it is evident that the use of LED drivers reduces the number of resistors significantly. This also includes a reduction in pin count and space required for the controller board, only using three Inputs to control all 5 drivers. Part of this is because the LED drivers receive serial signals which are further discussed in the components section of this document. This method has influenced the design, taking in the idea of "less is more." Less components means more space to organize the parts on the PCB. If the size of the PCB can be reduced, then money is saved. This individual used an Arduino development board like the group prior. This pattern has given us direction in researching the ATmega microcontroller series. Amore powerful chip may be required to accommodate the grand plans in progress. Another feature that was rather interesting is HNTE used a home built PCB. This engineer had access to special ink for printing customized PCBs. Though this could save money on PCBs and allow for much more flexibility in development, after some research into this method, it was decided not to follow homemade PCBs. It required technologies not currently accessible and time to learn the proper procedures to execute correctly. Costing mostly time, which the team did not have. HNTE then continued to create and RGB cube, building what was learned from the single color LED cube. Since the possibility of the RBG is on the horizon. The team did some research on this project as well which is discussed in the next section. Aside from that, implementing this design will create a learning curve in the programing of serial data communications, but its benefits outweigh the cost of time.

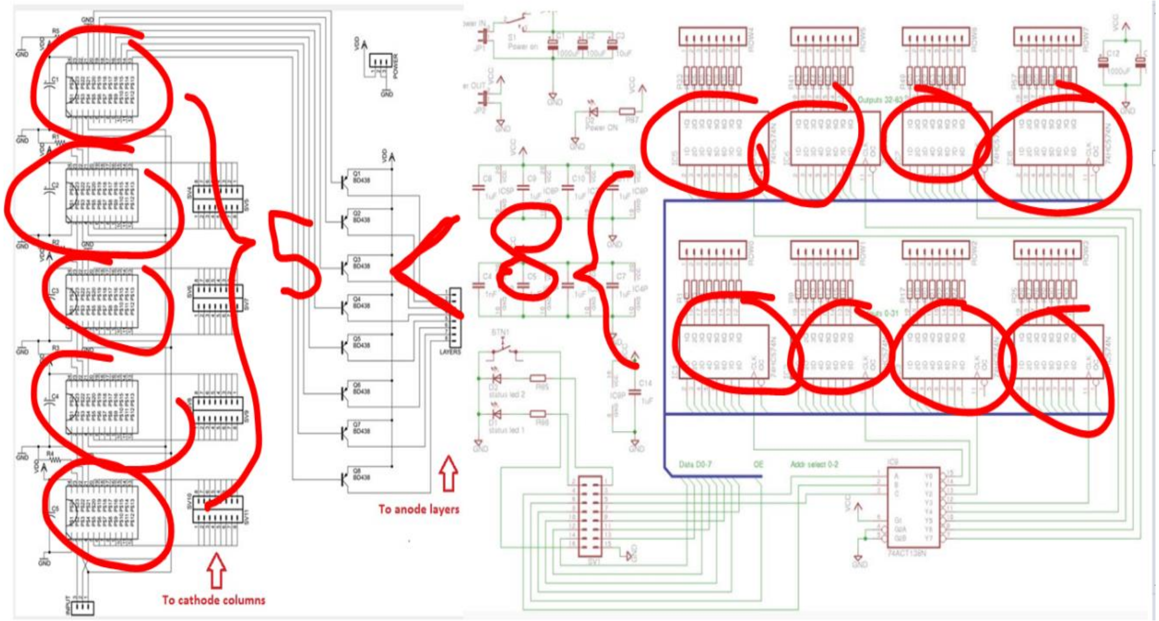


Figure 3.5 Schematic outline of the Instructables and HNTE controller board layout.

3.1.6 How Not To Engineer RGB 8 x 8 x 8 Cube

It was decided that research would be conducted on two LED cubes, the single color LED cube and the RGB LED cube. The team at How Not To Engineer also released their experience building the RGB LED cube and a guide on how to build it in addition to the single color LED cube. The following few paragraphs explain the differences between both cubes and how the team is planning to create these two separate cubes. In addition to these paragraphs, there will be an explanation on the project including list of components, code segments, etc.

One of the main differences between the RGB and the single color LED cube is obvious, the differences in color and animation. In terms of the RGB LED cube, the level of programming involving the animations is much more complicated than the single-color LED cube. The single-color LED cube only has two main settings, on or off. Within these settings, you can alter the intensity of the brightness through either the use of the PWM in the Texas Instrument LED driver TLC5940 (explained in section 4.1.3 Display System) or through the software created by the team while using the STP16CP LED driver. In regards to the RGB LED cube, there are three separate settings for each LED going through the primary colors. This requires not only to be in charge of more color settings (either on or off) but, also, the brightness of each of these colors. Which, like the single-color LED cube, will be either using the TLC5940 or the STP16CP LED drivers. In terms of the actual programming of the cubes, its only obvious that the animation for the RGB LED cube is much more extensive because of all these extra modes of operation.

Another main difference between the RGB and the single color LED cube is the number of LED drivers used to drive the LEDs. The single-color LED cube required only 5 LED drivers: One to control each of the eight layers of the cube, and four to control the 64 individual LEDs of each layer. In the RGB LED cube, there are 12 LED drivers: four drivers for each mode of the 64 LEDs in each layer. Therefore, there are four drivers for the red mode, four drivers for the yellow mode, and four drivers for the blue mode. The following figure, Figure 3.6, depicts the arrangement of the LED drivers for the RGB cube:

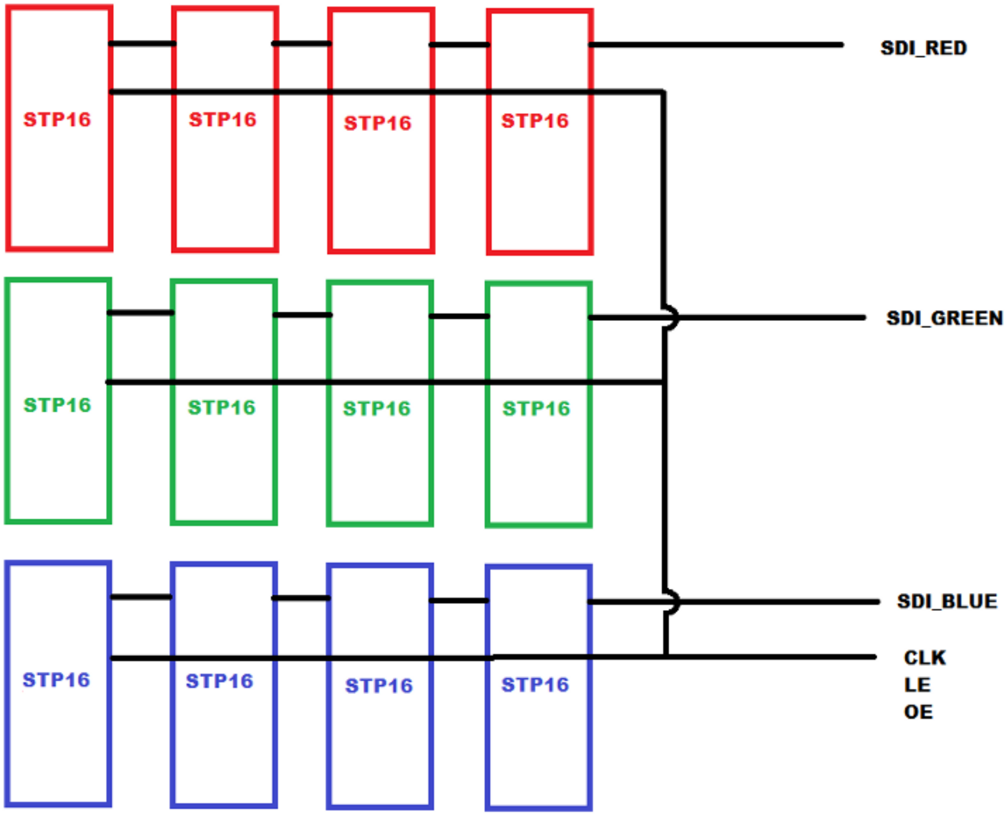


Figure 3.6: Layout of STP16CP LED Drivers for RGB LED Cube.

This particular LED cube contained a few more components than the single-color LED cube. The following is a list of the components needed to create this LED:

- 512 x 10mm, common anode RGB LED
- 12 x STP16 LED driver
- 8 x 2N5195 PNP transistors (next time I am using MOSFETS)
- 1 x ChipKIT UNO32
- 1 x 5V switched mode PSU
- 1 x 5v-3.3v DC-DC converter
- 12 x 10uF 16V ceramic capacitors (THIS SHOULD BE 0.1uF)

- 20 x 100R resistor (STP16 current set & transistor current limit) (USE 1K FOR THE CURRENT SET RESISTOR)
- 8 x 1.5K resistor (transistor pull up)
- 14 x IDC ribbon cable connector
- 1 x Roll 22AWG tinned wire
- 14 x pin headers (16x2)
- 16 way ribbon cable
- 12mm ply for the base and jig
- 20x12mm pine for the jig
- Screws, solder, bluetack, etc, etc.

When it came to the assembling of this LED cube, the procedure was a bit different and definitely more complicated than that of the single-color LED cube. While the single-color LEDs had two connections, the cathode and the anode, the RBG LEDs have four connections, the common anode and the cathodes for each color (red, blue, and green). This means that when it comes to assembling, instead of soldering only 128 different connections per layer as it is for the single-color LED cube, there will be 256 different connections to solder per layer. This increases the likelihood for errors as well as the time it takes to build the entire cube. The How Not To Engineer group also mentions that for this particular cube, a set of wires for each cathode color column was used instead of the legs of the actual LEDs, which calls for more component purchases.

In terms of the microcontroller board as well as the layout of the components, this cube is a bit different from the single-color LED cube. Firstly, instead of joining the 12 LED driver chips in a long cascade, they were separated by color. Four LED drivers (controlling the 64 LEDs in each layer) were connected together and three groups of these four LED driver chips were cascaded together in shorter cascades. These were connected to the same amount of transistors as the previous cube and in the same arrangement. The reason for joining the different colors into shorter cascades is that this way red, green, and blue data at the same time which reduces the time to load the data into the LED drivers.

Another main difference between the RGB LED cube and the single-color LED cube is that the RGB cube was designed to be a self-sustaining cube without the direct connection to a computer or an external PSU. The team at How Not To Engineer purchased an internal PSU that would provide a decent current to drive the LEDs. Because the Chipkit used by the team provides 3.3V IOs, they decided to use PNP transistors to drive each of the layers. Figure 3.7 below depicts the PSU used by the HNTE project.

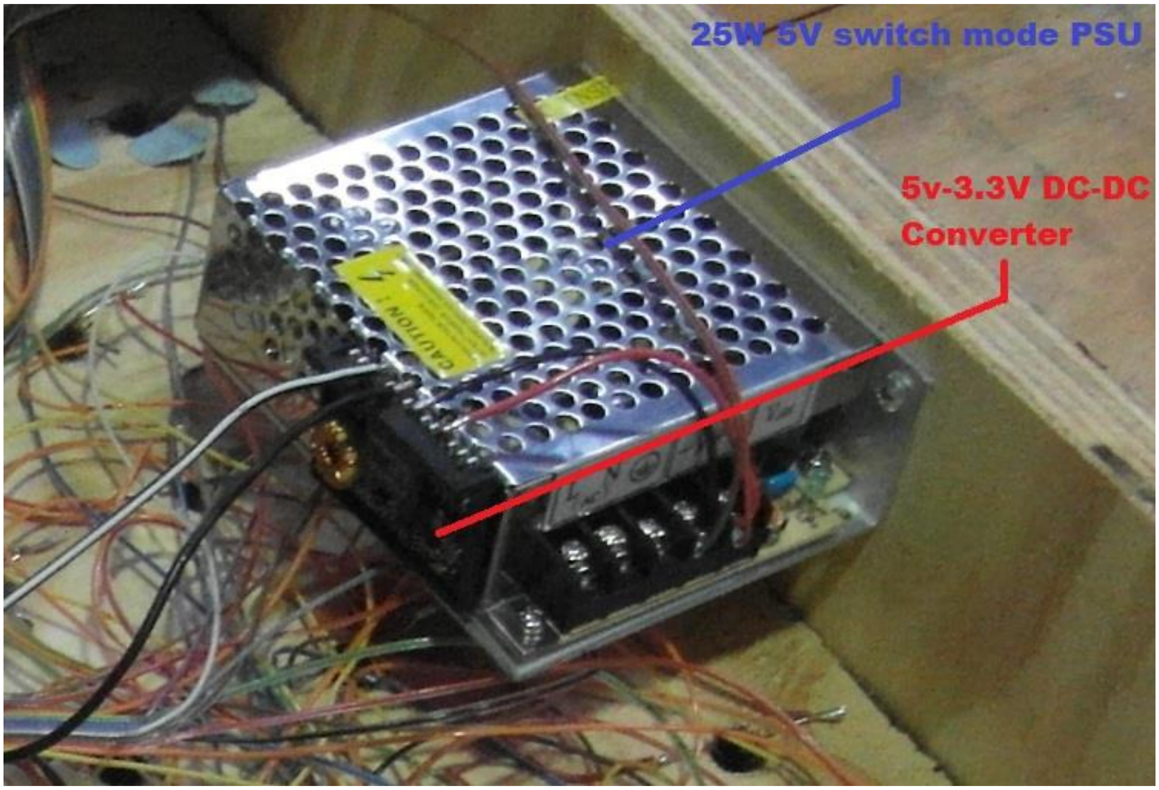


Figure 3.7 : Power Supply for HNTE RGB LED Cube.

In conclusion, the RGB LED cube is considerably more complicated to successfully build than the single-color LED cube. The RGB cube requires more components (LED drivers) to be able to drive it, more connections to each LED which increases build time and possible error making, more calculations to be able to display the different colors of the LED's, and finally a more powerful power supply to withstand the current needed to drive this cube. Overall, though challenging, it has been decided that the team will attempt to build this cube alongside the single-color LED cube.

3.2 Relevant Technologies/Component Research

After doing a sufficient amount of research, enough to have gained understanding on the workings of an LED cube. It was time to begin to gather the components required to build the cube. It was discovered that many variations and types of the components sought were on the market and further research was required to select the parts that best fit the project's specifications. Components selected were quickly ordered so that they may arrive in a timely fashion before the Summer semester commenced. This section discusses the components and technologies considered and chosen for the design.

3.2.1 Embedded Processor

The embedded processor is the most important part of this project since it will be acting as the brains of the LED cube. There is a large market for microprocessors on the internet and in turn will only discuss a few of the options considered for this project. There are a few requirements in consideration that came about through the research of previous projects discussed in Section 3.1 Existing Similar Projects and Products. Flash memory, pin count, operating frequency, analog to digital conversion capabilities and price are the main focus of the embedded processor consideration.

Texas Instruments

Texas Instruments is a well-known provider of signal processing technology. Due to familiarity with the MSP430 thanks to the Embedded Systems class at UCF, Texas Instruments was highly favored amongst the team. There were two model options from Texas Instruments, the MSP430 and Stellaris. The C2000 and Hercules series are too overpowered with non-need extra features, this included copious amounts of memory and a 32 bit ARM processor. Sure it would be great to have, but it is not necessary. The Stellaris LM4F120C4QR nearly fell into the same category with the C2000 and Hercules processors. But due to its price and lack of availability, other options were considered. The MSP430F5525 model was nearly selected for this project. It had all the features sought with an excellent amount of Flash memory and GPIO count. Including a 12 bit ADC and the low power usage was a plus. But it was not selected since a slightly more powerful processor at a lower price was found. Though an MSP430 will not be used for the final prototype, an MSP430 development board will be used to run a 3x3x3 LED Cube for the practice run.

Atmel

Most projects researched shared the common factor of the microprocessor used; the ATmega168 seemed to have been the user’s choice. Wanting to improve upon this design, a similar but slightly more powerful controller was sought. The AVR UC3, AVR XMEGA and megaAVR were the three model types being decided on. The UC3 has a 32 bit processor which is not a necessity. The XMEGA and megaAVR were close in comparison. The XMEGA had slightly more flash memory and a higher operating frequency. Also the megaAVR has an 8-bit processor, while the XMEGA has the capability to switch between 8 and 16 bit processing.

Other Producers

Zilog, Renesas, and Cypress are three semiconductor manufacturing corporations that arose during microcontroller research. Even though their products at a glance could meet project specifications, their lack of renown deterred continual research into what they had to offer. And since these are smaller companies, products are not readily available. One would have to wait some time to receive orders. Another issue is compared to Atmel and Texas Instruments, there lacks resources on the internet in regards to working with said devices. This could ultimately pose an issue in the long run.

ATxmega64A4U

The microcontroller in second in consideration for this project was the ATxmega64A4U. This is a high-performance controller with low-power usage, reducing heat production. It served general purposes with Xtreme performance 8-bit processing, also containing a full-speed USB device interface. This section will discuss the features and their use within this project.

Parameters

- 8/16-bit AVR CPU
- Flash Memory: 64 Kbytes
- SRAM 4 Kbytes
- Pin Count: 44
- Max Operating Frequency 32 MHz
- ADC Channels 12
- Operating Voltage 1.6 to 3.6
- Temp -40˚C – 85˚C

Most projects reviewed used between 16 - 32 Kbytes of memory. Since the controller will be housing the source code for three different functions the selected microcontroller has 64 Kbytes, double the amount. It is hypothesized that a large amount of script will be stored compared to past projects. Therefore this amount should give us some space for wiggle room. It is unclear if this will be sufficient memory until trials are performed. also if time permits and more features are added, this will mean that the memory will have to increase.

The ATxmega has analog to digital conversion capabilities (ADC) which will be of great help. The accelerometer and VU meter we plan to work with produce analog signals. The objective is to work with digital signals to communicate with the LED drivers. This will save us time and money, since we no longer have to come about with other means of converting the analog signals. The ATxmega has 12 ADC channels. The This was chosen over 4 or 8 channels since multiple analog devices will be connected and it is unsure how many ports with will take up. A greater number was chosen as a precaution. Also the sampling rate for the ATxmega's ADC is 2000 ksps which is 133 times more than that of the plain ATmega. This will reduce the error rate, allowing us to collect clear samples.

The amount of general purpose input/output pins (GPIO) was of concern. The chosen controller has 44 pins, which should be more than enough. Based on research, about 32 pins should work fine. Once again a larger amount was selected out of precaution. Our design plan as of now is to implement serial data input into the LED sink drivers. this will reduce the amount of pins used overall, but just in case something goes awry. The option to multiplex multiple pin outputs is still available.

Two options of communication between computers are presented with the ATxmega. The universal asynchronous receiver/transmitter (USART) and 2.0 universal serial bus interface (USB). One of these methods will be used to upload program code to the controller. Also the USART and serial peripheral interface (SPI) can both be used to send bits serially to the LED drivers.

3.2.2 Accelerometer

Accelerometers are very useful little gadgets that provide many different applications. In essence, they are used to sense both static and dynamic acceleration. The LED Cube that the group has built will allow for a “movable” mode, where the actual cube is handled and tilted by the user, making the cube display a conveying water-like effect. Such effect could not be achieved in a more efficient way than with the use of an accelerometer.

Similar to other components employed in the project, there are a couple of factors to discuss when considering which type of accelerometer to employ. First and foremost, a decision has to be made between an analog or digital accelerometer, which has been determined by the nature of the project’s hardware. Also, other specifications have been taken in consideration, such as number of axes, maximum swing, and sensitivity.

The aim of the project, as stated in the project objectives, was to have functionality for the LED cube to display a water-like effect. This implies that the cube will be subject to 3 dimensional movements, as well as tilting the cube along the x, y, and z axes. There exists a possibility for the group to implement three-dimensional functionality, which means that the cube will be subject to tilt on all 3 axes and upside down rotation. The group took in consideration this functionality, and it will be implemented in the final design. The group also discussed the possibility for gyroscope functionality for even more realistic and conveying movement effects. However, due to time constraints, gyroscope functionality was not taken in consideration for the final design.

The following types of accelerometers were researched by the group in order to consider one that will be most suitable for the design of this particular mode of the cube.

DE-ACCM Buffered ±5g Accelerometer

During the research phase for this component, one of the first accelerometers that were found was the DE-ACCM 2-axis accelerometer, manufactured by Dimension Engineering. Figure 3.8 below displays the size of the accelerometer in comparison to a nickel.

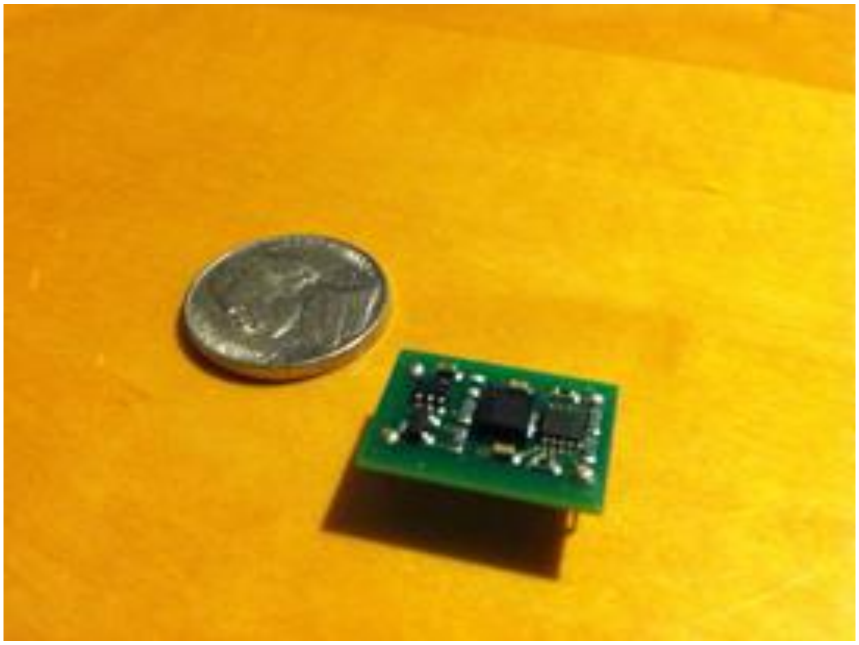


Figure 3.8: DE-ACCM Buffered ±5g Accelerometer

This analog accelerometer solution offers integrated op-amp buffers for direct connection to a microcontroller's analog inputs. The main features of the DE-ACCM include:

- Dual axis ±5g sense range.
- Up to 312mV/g sensitivity.
- Operating voltage 3 to 5V.
- Draws under 2mA.

These specifications, the sensitivity and operating voltage in particular, made the DE-ACCM a suitable candidate for our project. Additionally, this product includes protection features, such as reverse voltage protection, output short protection, and improper insertion protection, so the device would have been convenient in case there was a fault in any of these areas. It is important to note that the group was already in possession of this kind of piece, therefore reducing the cost of acquiring one.

The group's main concern with the use of the DE-ACCM accelerometer is the fact that it is only a two-axis piece, therefore there was uncertainty on whether the desired water-tilting effect would be feasible. Based on research, many sources stated that two-axes accelerometers would allow for motion and tilt applications, but the group decided to develop a small test-case to prove this functionality.

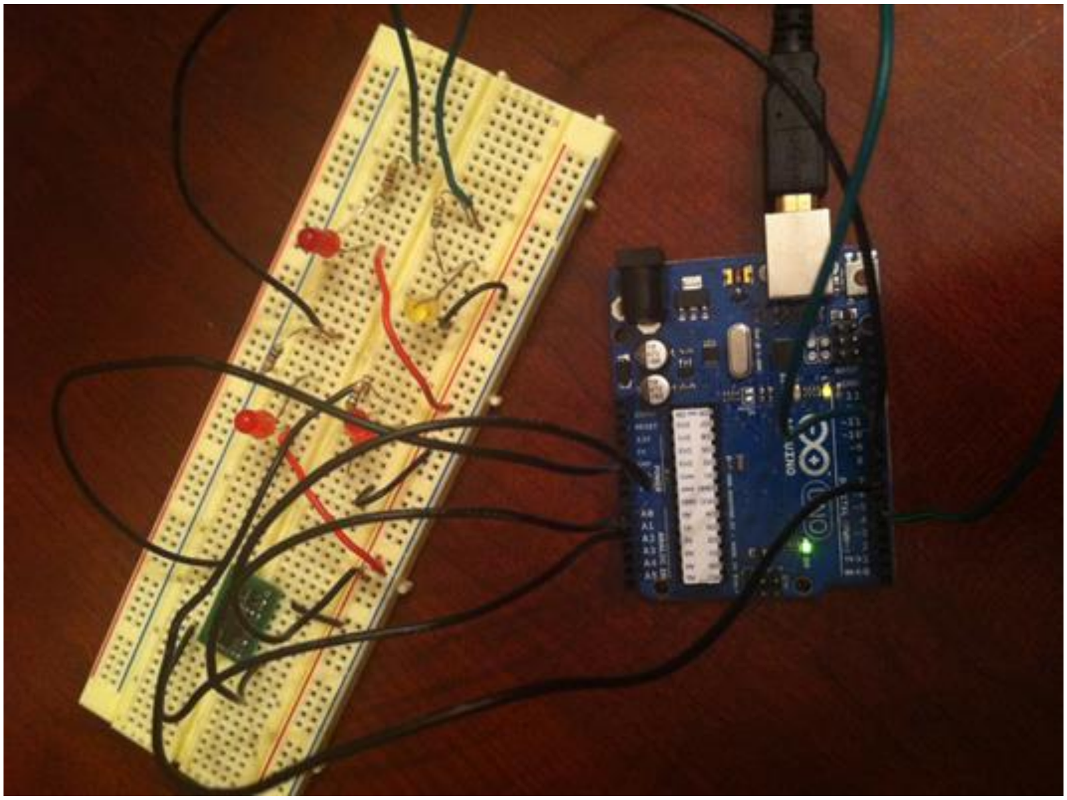


Figure 3.9: DE-ACCM Tilt test. Arduino board and Breadboard set-up

As illustrated in the above figure, Figure 3.9 a test case for the DE-ACCM was easily made with the use of some LEDs and an Arduino development board. In this particular setup, each of the LEDs is connected in parallel to a resistor and then connected to the digital pins. Also, the X and Y inputs of the accelerometer are connected to the analog pins of the board. The expected result of the test case was to make the corresponding LED light up based on the direction in which the accelerometer was tilted, and the test cases confirmed this behavior. Connecting the LEDs to digital pins, and the X and Y inputs of the accelerometer to analog pins, the expected result of the test case was to make the corresponding LED light up based on the direction in which the accelerometer was tilted. Therefore, it was successfully proven that a 2-axis accelerometer would indeed work for the movable feature of the LED cube.

However, upon further inspection of the previous test case, the group was able to determine that a three-axis accelerometer would be needed for this kind of application. This is due to the fact that, with a two-axis accelerometer, the only way to display water behavior effects was to hard code the accelerometer mode. With the implementation of a three-axis accelerometer, the group would be able to mirror the x,y, and z axes of the cube to the x,y, and z axes of the accelerometer, and thus easily implement this behavior.

DE-ACCM3D Buffered ±3g Tri-axis Accelerometer

While researching about the previous two-axis accelerometer, the group stumbled upon an extension of the DE-ACCM. The DE-ACCM3D is a complete three dimensional ±3g analog accelerometer solution. Similar to the DE-ACCM accelerometer, it features integrated op amp buffers for direct connection to the microcontroller’s analog output. In addition to the applications that the DE-ACCM offers, the DE-ACCM3D also offers device positioning as well as vehicle acceleration logging.

Additionally, the DE-ACCM3D has been designed to fit the DIP-16 form factor, which means that this device is suitable for any applications involving perf-boards, breadboards, and insertion into standard chip sockets.

The main factor that separates this accelerometer from the rest is the fact that it features three axes of movement. This allows for an improvement in accuracy, as the additional axis will provide the group with additional information regarding the movement of the cube. The group was able to take this information and apply it to the project in order to enhance the animation of the moving liquid effect. For instance, if the cube was to be positioned upside down, the animation would be displayed accordingly, thanks to the additional axis provided by this accelerometer. However, this particular accelerometer was a little above the group’s price range, and the decision was made to go for an alternate triple axis accelerometer.

ADIS16003 Dual-Axis ±1.7g Accelerometer

Another possible solution that was of the group’s interest during research and would complement the project was the ADIS16003 dual-axis accelerometer by Analog Devices.

The ADIS16003 is a low cost, low power accelerometer, featuring dual-axis capability with an integrated serial peripheral interface, as well as an integrated temperature sensor.

Similar to the DE-ACCM accelerometer, the ADIS16003 is capable of measuring dynamic acceleration as well as static acceleration, which would prove useful in the detection of vibration and gravity, for instance. However, this particular accelerometer only measures acceleration with a full scale range of ±1.7g as a minimum, and when compared to the DE-ACCM, it was rather a tad limited in this aspect.

Another important factor to note is the fact that the ADIS16003 features a digital interface, as opposed to the simple and analog output that other accelerometers provide. While digital accelerometers are less susceptible to noise than their analog counterparts, this choice would be most difficult to get integrated with the microcontroller as it features an integrated serial peripheral interface, and would therefore require a higher learning curve to overcome due to the fact that the group has never had any previous experience when it comes to dealing with accelerometers.

Some other features of this accelerometer include highly integrated, minimal external components, an externally selectable bandwidth, and an externally controlled self-test. The group decided that these additional features will not provide a radical improvement to the project overall.

In addition, given the fact that a DE-ACCM accelerometer is already available to the group, acquiring this digital accelerometer would imply a higher cost addition to include in the project expenses, and this is undesirable.

MMA7341L Triple Axis Accelerometer

The MMA7341L is a complete three dimensional ±3g analog accelerometer solution. Similar to the DE-ACCM accelerometer, it features integrated op amp buffers for direct connection to the microcontroller’s analog output. The MMA7341L also offers 0g-detect which detects linear free-fall.

Additionally, the MMA7361 has been designed to fit the DIP-10 form factor, which means that this device will be suitable for any applications involving perforated boards, breadboards, and insertion into standard chip sockets. The main factor that separates this accelerometer from the rest is the fact that it features three axes of movement. This allows for an improvement in accuracy, as the additional axis provides the group with additional information regarding the movement of the cube. For instance, if the cube was to be positioned upside down, the animation will be displayed accordingly, thanks to the additional axis provided by this accelerometer. Additionally, this accelerometer was less expensive than the DE-ACCM family of accelerometers, which proved a potential candidate to the final design.

This was the group’s final decision for the accelerometer component. The group ruled out the possibility of a two axis accelerometer due to the fact that the liquid behavior effects would not have been achieved with such a component, the only way being a hard-coding method. The MMA7341L allowed the group to successfully utilize each coordinate axis and perform the corresponding logic with the three axis of the physical cube.

3.2.3 VU Meter

Another requirement for the final design was a set of animations based on music interaction. The first step to consider is how the interaction is going to be. The objective is that the LED will mimic the equalizer display in a radio; in addition with different pattern to have a full use of all LED’s available. More details on the visual display will be address in the software section. The next step was to research the different components that were needed for the interaction. There were two options to act as the input for the Volume Unit (VU) Meter. We could use an audio jack and any music player as input or use ambient sound. Fewer electronic components were needed to implement the audio jack and the input would have been very reliable but, we would have needed an outside device to play the music. Therefore we decided to use ambient sound as our input to increase the maneuverability required in the project specifications.

To implement the audio jack, the music would be played by a digital music player such as an Ipod. Then, we would use a stereo audio plug cable to connect to a regular 1/8" stereo audio jack mount in the circuit board. The stereo input would need two separate circuits to break down the different frequencies or use coupling capacitors to integrate the stereo signal to mono input. The design did not require high sound fidelity to operate; therefore, the mono input would have been implemented for simplification purposes. After the signal was broken down, it was sent to the microprocessor. In the audio jack option, fewer electronic components were needed to implement the VU meter circuit and the input would be very reliable but, we would have needed an outside device to play the music and audio cord attach to the project. It would decrease one of the main specifications, which is the maneuverability of the project. The other option was to implement the ambient sound. A microphone attached to the circuit board would pick up the sound wave and send the signal to an operational amplifier. Once more, the signal would go to an integrated circuit to be broken down into different frequencies and finally be sent to the microprocessor. With this option, the project would react to noise in the surrounding area and any kind of music played, with the down side being that an audio player and speakers are needed to play music. Therefore, we decided to use ambient sound as the input to our vu meter to increase the maneuverability requirements in the projects specifications.

To keep to the same sequence as the input signal, the first component to research was the microphone. In general, any speaker can be used in reverse to create a microphone but for simplicity, an actual microphone was implemented. The specification of the design was used to select the ideal microphone. The design did not require special effects for the VU meter or high fidelity in the sound signal since the audio will not be reproduced. Therefore, any microphone can be used. The only parameter was that it had to be suitable to be attached to a PCB. Also, it needed to be easy to implement as an RC circuit, and the frequency response needed to be close to the integrated circuit that broke down the frequency. To narrow down the research, octopart.com was used because some of the parts were in stock. The following table, Table 3.3, describes the characteristic of the different microphones considered in the project. At the end, the ABM-707-RC was chosen because it was not too small in comparison to the TOM-1545P-R and the ABM-703-RC and had a higher frequency response than the AOM-6545P-R.

	TOM-1545P-R	ABM-707-RC	ABM-703-RC	AOM-6545P-R
Directivity	Omni	Omni	Omni	Omni
Frequency Response Min	20 Hz	50 Hz	100 Hz	20 Hz
Frequency Response Max	19 KHz	16 KHz	16 KHz	1 KHz
Standard operating voltage	2.0 V	1.5 V	2 V	4.5 V
Max operating voltage		10 V	10 V	
Sensitivity	-45 dB	-41 dB	-40 dB	-45 Db
Drain Current Max	0.5 mA	0.5 mA	0.5 mA	0.5 mA
Impedance	2.2 KΩ	2.2 KΩ	2.2 KΩ	2.2 KΩ
External Diameter	4 mm	9.7 mm	6 mm	9.4 mm
Type		Electret condense	Electret condense	
Unit Price (USD) Newark	$1.90	$1.13	$1.09	$1.29

Table 3.3: Microphones

The next device needed to be researched was the microphone preamplifier. Almost any operational amplifier could be used to get the desired gain for the circuit. The main parameter for the selection of the operational amplifier was the power supply. The design uses a USB port as a power supply which delivers 5 volts and low current, as well as other modes of power supply. For the selection of the right op amp, the selection had to be low on power consumption with single voltage supply being less than 5 V. For better performance with the frequency comparator, it had to be able to handle frequencies between 63 Hz to 16 KHz or -20 to +3db. The next section discusses the reasons for the parameters. The design will use bias decoupling capacitor for the cut-off frequency to reduce the amount of operational amplifier. The gain will be done in two stages. Finally, the previous parameter was included in the search in optopart.com. Many options were search, but in table 3.4 are the most important parts that were taking into consideration. Then, the group decided to request some free example from Texas Instrument. The AD8552ARZ and TS912ID were discarded because their manufacture does not offer the free example program. The TLC27M2 is a military issue and it need to go to a background check to order or request a free sample. Therefore, the LM324NE3 is the operational amplifier that will be implemented.

LM324NE3	Is a low power quadruple operational amplifier from Texas Instruments with a bandwidth of 1.3 MHz and large voltage gain of 100 dB. But, the special about this part is the voltage range of +3V to 30V and a typical output sink current of 20 mA. This op-amp will work perfectly in our low voltage low current project
TLC27M2	Is dual operational amplifier from Texas Instruments, it operate with a single power supply from 3 to 16 volts, Iq per channel max of 0.28 mA, Typical gain bandwidth 0.525 MHz, minimal common-mode rejection ratio of 65 dB, and come in PDIP, SO,SOIC,TSSOP package.
AD8552ARZ	Is a dual, rail-to–rail input and output swing operational amplifier from Analog Devices, it operate with a single power supply from 2.7 to 5 volts, supply max current of 0.7 mA, a bandwidth of 1.5 MHz, common-mode rejection ratio of 130 dB, and come in SOIC package. This op-amp reduces the amount of part need in the project because does not require any external capacitor.
TS912ID	Is a dual, rail-to-rail operational amplifier from STMicroelectronics. Operate with a single power supply from 2.7 to 16 volts, supply max current of 0.4 mA, a bandwidth of 800 KHz, common-mode rejection ratio of 70 dB, and come in SOIC package.

Table 3.4: Op-Amps

The next component to research is the basic and most important part of the volume unit meter. The frequency divider will break down the frequency and send it to the microprocessor. The input signal can come from an audio jack or an operational amplifier into a frequency comparator. It can be built by implementing multiples comparators using operational amplifier or simply getting an integrated circuit with the function already built in. To implement the comparator, a set of minimum eight operation amplifier are necessary to control each layer of the cube. In addition with multiple resistances are require to set the different comparison points. With this option, an exactly point of reference can be assign and can be as many comparator as need. The negative side s of this option is the need for more resistor and operation amplifier. Also, the circuit will not be very accurate; therefore, this option will not be considered. The other option is to implement and integrate circuit that breaks down the frequency. One of the most famous ways to do a VU meter is by using a Dot/bar display driver. The most popular method is the use of integrate circuit UV meters such as the LM3914, LM3915, and LM3916 from Texas Instrument. Another popular method was the use of a band graphic equalizer such as the MSEQ7. No decision has been made yet about which to use. Both option will be research and discuss, also two prototypes will be made before making the final decision.

Dot/Bar Display Driver

This integrated circuit uses multiple operational amplifier implemented as frequency comparator. Each of the comparators gives an analog output. NTE Electronics and Texas Instruments are the two major company companies that produce this type of integrated circuit. Both company produce very similar products, but this project focuses on the Texas Instruments because of the convenience of being able to request free sample. The LM39xx provides 10 drive output. It can be used to drive LEDs, LCDs, or Vacuum Fluorescents and incandescent bulbs. The display can be change from Bar or Dot by providing or not voltage to pin 9.Have a wide operating power supply and low output current. The display reference can be adjusted but the 10 step voltage divider cannot. It can also, easily be used in cascade to increase resolution and range. Table 3.5 provides the different characteristic for the three display driver produce by Texas Instruments. From the table we can see that the main different among LM39xx family is the display. LM3914 is a linear display, while the LM3915 is a log display with 3db increments and LM3916 display -20 to +3db. The inconvenience of choosing the LM3916 is the analog output. The project needs a digital output; therefore, to implement this type of driver an analog to digital converter, such as a multiplexer, needed to be add to the project. The other option was to use a band graphic equalizer such as the MSEQ7discuss in the next section.

	LM3916	**LM3915**	**LM3914**
Minimum input voltage	3 V	3 V	1.2 V
Maximum input voltage	35 V	25 V	12 V
Output Voltage	1.28 V	1.28 V	1.34 V
Maximum output current	35mA	30mA	30mA
Number of output	10	10	10

Expandable to display	70 dB	90 dB	100 steps
Display Type	Audio VU	Logarithmic 3dB/step	Linear
Package	DIP	DIP	DIP

Table 3.5: T.I. Dot/Bar Display Driver

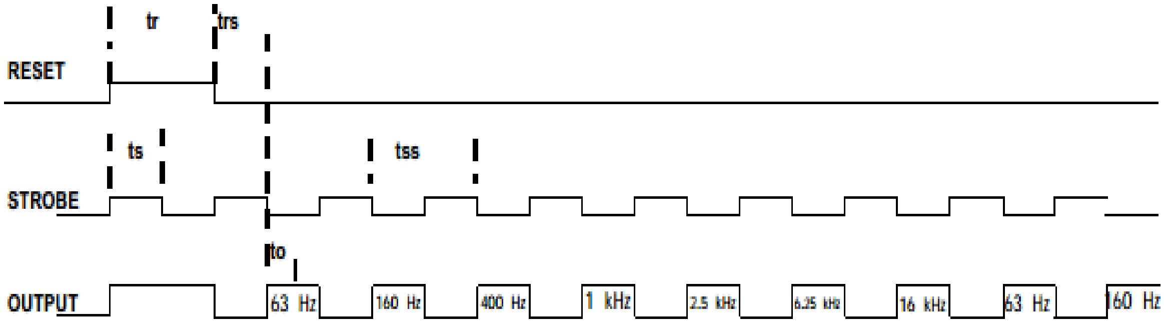
MSGEQ7

The MSEQ7 is a seven band graphic equalizer produce by Mixed Signal Integration. This integrate circuit take the input sound signal and divides the audio into frequency of 63Hz, 160Hz, 400Hz, 1KHz, 2.5KHz, 6.25KHz, and 16KHz. These integrate circuit is easy to implement because only require four coupling and decoupling capacitor and two resistances. Also, the most important features are already provide in the data sheet as follow: Low power consumption

- Only two external components
- On Chip Ground Reference
- Switched - Capacitor Filters
- 3.3 or 5 volt Operation
- 20 dB of Gain Typical
- On Chip Oscillator
- Output Multiplexer
- Variable Decay Time
- 8 Pin Package 20

The low power consumption makes it ideal for the design because we are going to use a very low power supply. Also, the voltage operation is ideal because it can be implementing with the other components. The major problem of this integrated circuit is that only provide seven band frequency and the cube have eight layers. The issue can be resolve by always lighting up the bottom layer or simply let the cube react only from layer one to layer seven. The final decision will be discuss in the software section. Another problem to take into consideration is the errors that may occur while obtaining data from any analog device. But the Figure 3.10provides a time diagram and help to understanding of how the MSEQ7 works. As show in the figure, first we have to send a signal to reset; then a signal for the strobe. Finally, after the strobe signal goes low, the output signal can be read. The strobe and reset input can be provided by the microprocessor, and the output goes directly back to the microprocessor. Therefore, only three connections are requiring between the chip and the microprocessor. The other positive aspect of Figure 3.10 is that it provides the specific delay period among signal. Knowing those delays will drastically reduce the chance to go into error code. In the market there are more options for different band graphic equalizer, but the price will dramatically be increase in our project. The unit price goes for around $1.89 dollar and varies from sellers. This is a very affordable price. The final decision whether to use an MSDEQ7 or the LM3916 has not been made yet. Therefore, to have a better understanding of how to implement and to help in the final decision of which one to use, a prototype for both chips will be made. The following list is a reference for all the pins and it functions

1. VDD/A: positive power supply
2. VSSA: negative power supply
3. OUT: multiplexed DC output
4. STROBE: channel selection
5. IN: input
6. GND: ground
7. RESET
8. CKIN: clock



tr - Reset Pulse Width		100 nS min	
trs - Reset to Strobe	Delay	72 uS min	
ts - Strobe Pulse Width		18 uS min	
tss - Strobe to Strobe	Delay	72 uS min	
to - Output Settling Time		36 uS min	(with Cload = 22 pF and Rload = 1 Mohm)

Figure 3.10: Strobe Timing Diagram for MSGEQ
Permission pending from MSI [3]

3.2.4 Power Supply

Another electrical component that is of the utmost importance for the success of this project is the power supply. During the research stage of the project, three options for the power supply were considered: a rechargeable battery, a wall wart power supply, and a USB port. Within the confines of this section, many factors that led to the selection of a suitable power supply will be discussed. The design specification of our project states that the cube will be required to be portable, and also be able to resist abrupt movements. In addition, for the cube to be operational, the use of sensible electronic parts from different manufacturers and voltage ranges are required.

The light emitting diodes that were chosen for the project each require 3.4V and 20 mA. There are 512 LEDs that make up the whole cube. Therefore, if one was to turn on all LEDs at the same time, amperage of 10.24 Amp will be required, keeping in mind that this is only from the LEDs, and not counting the remaining electrical components of the circuit. In essence, the circuit could go from 20 mA to more than 11 Amp in a matter of milliseconds, and thus the microprocessor will be subject to damage. This is where the principle of persistence of vision comes into play, as only a given layer of the cube will be turned on at any given time.

The microprocessor, ATmega1284 has a general operating power supply voltage from 1.6 V to 3.6 V. Absolute maximum rating:

- Current into a VCC pin (I_{VCC}) 200 mA
- Current out of a ground pin (I_{GND}): 200 mA
- I/O pin sink/source current (I_{PIN}): 25 mA

The accelerometer uses the DE-ACCM5G Buffered. It has an operating voltage range from 3 to 5 V and maximum current of 2 mA. The graphic equalizer has an operating voltage range from 3.3 to 5 V and maximum current of 1 mA. The final selection of all the parts has not yet been done; but, the design will require less than 5 Volts and less than 1 Amp. Now, a power source can be chosen taking in consideration of the previous parameter

The first option is to use three AA batteries. One possibility is to use a rechargeable nickel–metal hydride battery (NiMH) with 1.5 V and 2500 mAh or a 9 V and 1200 mAh lithium batteries. Both option can be in the PCB and will make the final design independent. This option is affordable and very easy to implement. The voltage will have not ripple and an inexpensive linear regulator can be used. The negative part of this option is the need of multiple batteries as backed up. In the 9V cases the regulator will have to sink around half of the voltage and the power efficiency will decrease. If AA batteries are used, then it will have to be recharged or replaced after 2 hours of use.

The second option is to use wall wart power supply. In this case we will have to use a step down transformer, a rectifier, and filter. This option provides a higher power but, the output will have a ripple voltage. This option is of convenience, because the voltage regulator can be outside and far away from the microprocessor and not add extra heat to the circuit.

The third option is to use a computer USB port as the power supply. The USB 2.0 specifications provide a reliable and stable power supply. The down sides of this option are the sensibility and lower deliver power. USB have a closed voltage range from 4.4 to 5.25 V and a maximum high-power current to 500 mA and the power that can be drawn to a maximum of 2.25 Watt. Therefore combination of wall and USB option was chosen to power up our project. It will have USB connector mount to the PCB. It will give us the option to supply power from a PC or wall wart with a USB port.

3.2.5 Voltage Regulator

This project have sensitive electronic component, and any of the power supply implemented will required a voltage regulator to keep stable the voltage operation point. The classic two options are linear regulator or switching regulator. To select the over the best option we are going to follow guide line from a published paper by national semiconductor which advice to evaluate the following requirements

- Maximum Load Current
- Type of Input Voltage Source
- Output Voltage Precision
- Quiescent Current
- Special Features

The best choice for a specific application can be determined by evaluating each component separately. Table 3.6 has the parameters to be consider for the selection. The total max current is 393 mA. The common voltage point for all the different part is 3.3 to 3.6. The select voltage operation will be 3.4V because we don't want get too closed to the max operation point of the microprocessor. Also, the others parts don't have to work up to the max capacity; in addition, 3.4 volts is in the middle of the LED's operation point. The USB supply voltage is 5V. The regulator will have to sink 1.6 volts which is the 32% of the input. The type of input voltage will vary from a from a PC or wall wart. For the precision requirement, the voltage regulator can goes down to 3.3 volts or 9.7% and up to 3.6 volts or 10.59%. The quiescent current is not an issue in the design because not battery will be used. The final project does not require special feature such as low power shut down or reverse input protection because there is only one way to connect a USB port and always will be connected.

	Part number	**Voltage range**	**Max load current**
Microprocessor	ATxmega128D4	1..8 to 5.5 V	200 mA
LED driver	TLC5940NT	3 to 5.5 V	120 mA
Accelerometer	DE-ACCM5G	3 to 5 V	2 mA
Op-Amp	LM324N	3 to 30 V	50 mA
VU meter	MSEQ7	3.3 to 5 V	1 mA
LED's	3mm	3.2 to 3.4	20 mA

Table 3.6: Voltage and current operations

A linear regular is the best option because is the more economically affordable. Also, the design can have proximally ten percent tolerance and linear regulator is within five percent tolerance. The linear regulator implemented is the classic 7805 because was already available from the previous electronics II laboratory. The regulator will be adjust to supply an output of 3.4 volts by selecting the ground to output leg resistance of 1kilo ohm and ground leg to ground of 2.4 kilo ohms.

3.2.6 LED Sink Drivers

STP16CP

The STP16CP LED sink driver is a chip that for the purposes of this project can receive a serial signal and sink the cathode of 16 separate LED's to ground. The reason why this chip could have been useful for this project was that it allowed running multiple chips in cascade to control up to a certain number of LED's as well as several layers, depending on how the circuit was arranged. In terms of the researched 8x8x8 LED cube from How Not To Engineer, they used the chips in cascade to control 64 LED's simultaneously as well as 8 layers. Using the datasheet provided for the chip as well as some circuit arranging, they derived the following schematic (Figure 3.11) which guided the team in the development of a prototype as well as the finished product.

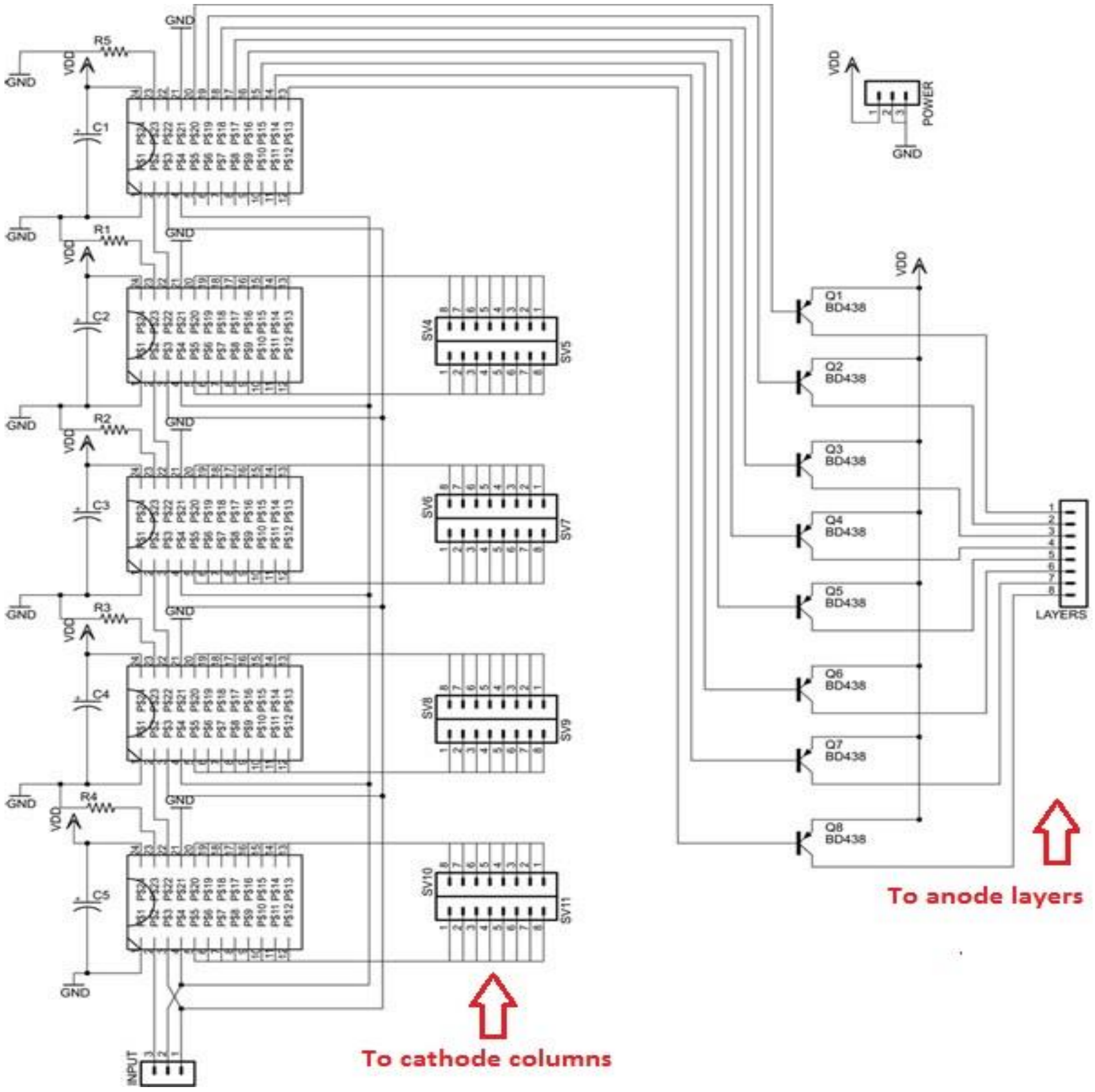


Figure 3.11: Schematic illustrating a cascade of five STP16CP LED Drivers

TLC5940

The TLC5940 has 16 channels, which contain an individually adjustable 4096-step gray scale PWM brightness control as well as a 64-step constant current sink. This is also called Dot Correction, which adjusts variations in the brightness between LED channels as well as other LED drivers. These functions are both controlled and accessible via the serial interphase, which will be discussed in the SPI section of this document. Connecting one resistor sets the maximum amount of current for all of the 16 channels.

Based on calculations of other teams who have developed an eight by eight by eight LED cube, the needed supply voltage ranged from 3 volts, if just using the programmed animation, to 5 volts if running the cube while simultaneous using the USB to serial converter. The range of supplying voltage allowed by this LED driver was anywhere between 3 to a maximum of 5.5 volts, which means this LED driver fell within the needed parameters to operate this LED cube.

Fortunately, the Arduino boards support the libraries of the TLC5940, which facilitated the coding and operating process of the program. Through the use of the TLC5940 library we were able to facilitate the process of programming the Grayscale PWM brightness operation, which allowed us to gradually control the brightness of the LEDS giving us a smooth eye-catching visual of the LED lights. The following are a few of the functions which allowed the team to quickly program the LED driver through the Arduino Board:

- Tlc.init(initialValue (0-4095)) – Used for setting up the timers before using any other TLC functions. InitialValue defaults to zero which means all 16 channels are off.
- tlc_initServos (uint8_t initAngle) – Initializes the TLC for usage.
- Tlc.set(uint8_t channel (0-(NUM_TLCS * 16 – 1))), int value (0-4095)) – Used for defining the grayscale data for channel which relies of Tlc.update())
- Tlc.setAll(int value(0-4095)) – Sets all 16 channels to a certain value which also relies of the Tlc.udpate()) function.
- Uint16_t Tlc.get(uint8_t channel) – Returns the grayscale data for channel to be compared by the TLC5940 against grayscale counter value.
- Tlc.update() – Sends the changes from the basic functions of clear, set, or setAll.

LP3943 RGB/White/Blue / 16-LED Fun Light Driver

Another LED driver researched, the LP3943 RGB/White/Blue LED driver, allowed for both driving 16 RGB or single-color LED drivers. The main difference between this LED driver and the TLC5940 was the PWM control. Unlike the TLC5940's ability to control the brightness of the LED using its 4096 step PWM brightness control, the LP3943 provided multiple programming states—on, off, input and dimming at a specified rate).

The LP3943 is another integrated device that allowed for independent driving of 16 LED's. It provided the ability to dim each individual LED though the use of SMBUS/I^{2}C applications where required to reduce bus traffic. In comparison to other LED drivers, in order to dim LED's the use of a serial shift register, like the 74LS594/5, provided large amounts of traffic/congestion on the serial connections. The LP3943 only needed the arrangement of the duty cycle and frequency for each of the output pins.

3.2.7 Latches and De-multiplexers

In regards to controlling the 8x8x8 LED cube, it was important to stress the fact that the whole cube required 72 IO lines to operate: 64 for each column, and 8 for each layer. This was an issue because there are no microcontrollers with a DIP package that have that many IO lines available for use. While researching previous projects, there were a couple of suggestions regarding the solution to this issue, one of them being the use of latches or de-multiplexers.

It was in the best interest of the group to research these components, and weight the advantages and disadvantages that they provide when compared to the alternative of using LED drivers.

74HC574 8-bit Latch

The 74HC574 is an octal D-type flip-flops by NPX Semiconductors. It features separate D-type inputs for each flip-flop and 3-state outputs for bus-oriented applications. This particular type of latch is used in the instructables.com project that was included in the research portion of this document, and thus information regarding the inputs of this chip can be found at this section. Additional features and benefits of the 74HC574 include:

- ESD protection
- Specified from −40 °C to +85 °C and from −40 °C to +125 °C

If latches are to be implemented in the final design, they will prove to be very useful, as they can serve as a kind of simple memory. The latch is able to hold 8 bits of information, and these 8-bits are represented on the output pins. In order to remember and load the on/off states of 64 LEDs, 8 latches will be required.

However, the LED drivers are a preferable solution as opposed to the 74HC574 8-bit latches. If the group decided to use these latches in the final implementation, an array of 8 latches will be required. However, using the LED drivers such as the STP16CP, only 5 of these drivers will be required, therefore reducing the cost of materials and having an efficient and more reliable set-up with fewer components.

74HC138 3-to-8 Line Decoder/De-multiplexer

For an LED cube using the previously mentioned latch set-up, a problem arises where 8 IO lines are needed to control the CP line for each of the 8 latches. The solution to this problem is the use of a 3-to-8 Line decoder/de-multiplexer, such as the 74HC138 manufactured by NPX Semiconductors. Some of the features and benefits of the 74HC138 include:

- De-multiplexing capability
- Ideal for memory chip select encoding

- Active LOW mutually exclusive outputs

This particular integrated circuit has 3 input lines and 8 output lines, which is very convenient for an LED cube of size 8. The input lines can be used to control which of the 8 output lines will be low at any time, while the rest are high. Each of the 8 outputs will correspond to each of the 8 latches.

Because this component is used in conjunction with the latches, it will not be part of the group’s final implementation, as the group will opt for the LED driver set-up. However, these components provided great insight into the alternative setups that can be applied to achieve the project’s objectives.

3.2.8 RS232 Communication

MAX232

One important concept for which the team almost overlooked was the communication between the microcontroller and any given personal computer. The Instructables site provided guidance in establishing a connection between a pc using an RS232 cable. The MAX232 IC makes a connection with the universal synchronous and asynchronous serial receiver and transmitter pins found on the ATmega microcontroller. Looking into the MAX232, it was found out that it requires 5 volts power supply to function properly. This falls out of line with our low powered project.

MAX3232

Texas Instruments provided another option. The MAX3232 requires 3.0V to 5.0V of supply voltage. This will require less components, time and energy to assimilate into the design. Specs for the MAX3232 can be seen below:

Parameters

- RS-232 Bus-Pin ESD Protection Exceeds ±15 kV
- Exceeds Requirements of TIA/EIA-232-F and ITU v.28 Standards
- Two Drivers and Two Receivers
- Accepts 5-V Logic Input With 3.3-V Supply
- Alternative High-Speed Pin-Compatible Device (1 Mbit/s)
- SNx5C3232
- Operating Voltage   3-V to 5.5-V
- Operates   250 KB/s
- Low Supply Current   300 µA Typical
- External Capacitors   4 × 0.1 µF

The next decision would have to be to construct the RS232 connection or purchase a pre-fabricated board with a DB9 male connecter. Building it would also require the construction of a modified RS232 connecter. It may be more expensive, but it would save us time. It is currently undecided which path to pursue.

3.2.9 Light Emitting Diode

Light Emitting Diodes (LEDs) are a type of semiconductors that emit light when current runs through it. Used often by devices to indicate running systems. Current must be set on a forward-biased in order for the electron to react and release energy in the form of photons. LEDs come in varying shapes and sizes, this section will discuss selected component for this project along with other options. Specs for the LEDs used in this project are down below.

3 mm diameter		
Blue Round Diffused		
Forward Voltage	min = 3.2 V	avg = 3.4 V
Wavelength	min = 460 nm	avg = 465 nm
Luminous Intensity	min = 4000 mcd	avg = 5000 mcd
Reverse Current	Max = 10 µA	
Forward Current	20 mA	
View Angle	120˚ – 140˚	
Life Span	50,000 hours	
Operating Temperature	-20˚C – 55˚C	

The diffused LEDs were chosen over the clear round LEDs. The clear casing emits light in an upward direction, lighting up the LEDs above, causing unwanted reflections, taking away from the spectacle. With the diffused casing, light is somewhat evenly distributed to all visible sides of the cube. Also in regards to brightness, the chosen LED is able to produce a desired brightness at 1/8 the duty cycle that the layer is turned on due to the multiplexing involved.

Though larger LEDs do provide a brighter light, brightness is not the objective. Also Larger LEDs take up space and add weight (though miniscule, every gram counts). The chosen 3 mm size allows us to create an almost transparent cube to create the three dimensional light illusions. With larger size bulbs, the risk of insufficient spacing in between each LED could cause a series of bright flashes grouped together obtruding the three dimensional pattern. Perhaps ideal with much larger cube sized projects, smaller sized LEDs are better suited for the 8x8x8.

Round LEDs were chosen since they are much easier to work with in the soldering portion of this project. Though the square head LEDs would better fit the cube theme, they are more expensive which will pose financial issues.

The single color LED compared to the RGB was chosen out of simplicity. The tri-LED would involve more I/O connections and parts, in turn meaning more money. Though different colors would produce a more visually pleasing experience; the objective is to produce three dimensional images, not multi-colored ones. This would also take away time from the more important functions being the VU meter and accelerometer. If time permits perhaps an RGB prototype can be produced.

Final thought on the chosen LED is in regard to the leg lengths. The legs will be the skeleton of the cube holding it up. Further support may be added since the goal is for the cube to be manhandled.

3.2.10 RGB Light Emitting Diode

RGB LEDs are light emitting diodes, just like the single-color LEDs, with the capability of showing an array of colors by mixing the 4 primary states of this LED. The colors red, green, blue, and white are the main modes of this LED. These types of LEDs come in various sizes but for the purposes of this project, the team will use 5mm diffused RGB LED's. The following image, Figure 3.11, depicts the structure of the RGB LED which will be described in the paragraph following.

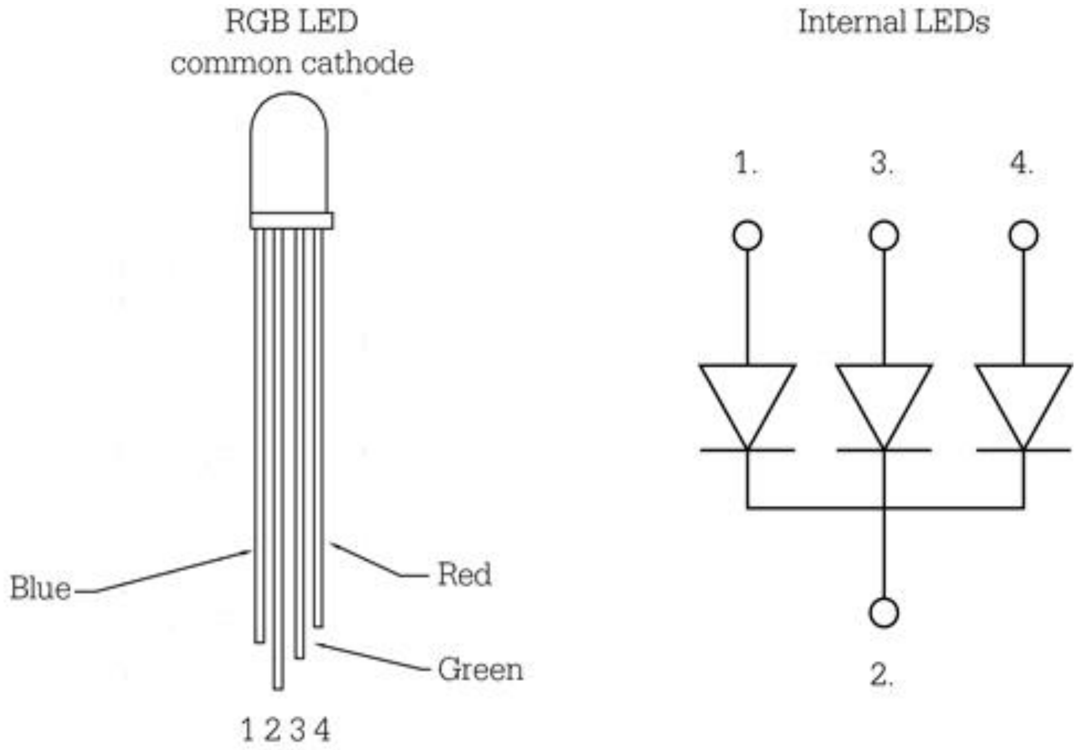


Figure 3.11: Diagram of RGB LED
Printed with permission from Bare Conductive [4]

The RGB LEDs differ physically from the single-color LEDs. The once used for this project have 4 legs. Three of these legs are anodes controlling each color (Red, Green, Blue) and the fourth leg (the longest of all) is the cathode. The way this LED works is a bit different than the single-color LED. Feeding a current through the selected color anode leg controls the color lights. The difference is in the variation of colors. If current is supplied to more than one color anode at a time, the LED will light both colors and make a mix of colors. This allows for diffusion of color to happen, which is something that cannot be done with single-color LEDS. Having all anodes receive the same amount of current simultaneously will result in a white light, which is not common, and more than likely will not be used in this project. The reason being that this will draw up too much current in the cube and there will be a possibility of overheating the controller or the LEDs.

Using the RGB LEDs in this project will give the team an advantage over using the single-color LEDs. Esthetically speaking, the capability of producing images and patterns on the cube in multiple changing colors will have a greater impact on the visualization over the single-color LEDs. Another advantage of using RGB LEDs over the single color is the opportunities to create visually intriguing patterns will be much greater. Some patterns can only be visualized if there is a differentiation in the color patterns.

3.2.11 Switches

The type of switch required is of some importance to the overall project and therefore noteworthy in this design document. The current plan of having three modes/states in the cube, a method to switch between these states is required. The idea of using a touch screen interface to manipulate states was taken into consideration. This was dropped due to the increased cost and was set aside. A simple selector switch with three positions (single pole triple through) and four pins was selected. The mode selection will be based on the pin number receiving current on the micro-controller. The moving the switch to one of three positions will direct current to said pin, changing the state of the machine. The Amico 3 Position SP3T 1P3T PCB Mount Slide Switch Right Angle 4 Pin switch was selected due to its simplicity and cheap cost. Down side of the part being cheap, is the shipping time takes several weeks. Info for the switch can be seen below:\

Parameters

• Pin Width:	0.5mm/ 1/50"
• Knob Height:	5mm/ 1/5"
• Rating:	DC 50V, 0.5A
• Contacts Resistance:	30M?
• Insulation Resistance:	100M?
• Minimum Load :	DC 500V
• Dielectric Strengths:	AC 500V / 1 minute
• Life Test:	10000 cycles/ minute

3.2.12 Control Boards

Based on research conducted on the How Not to Engineer LED cubes, the team decided to research which control board manufacturer would be most beneficial for our team. Since the team has two cubes to build, the eight by eight by eight RGB LED cube as well as the eight by eight by eight single-color LED cube, there were definitely a few ways to arrange the purchasing of personalized control boards to best suit our time frame as well as our financial needs and constraints.

The two companies the team decided to look at were BatchPCB, stationed in the US, and RushPCB, stationed in the UK. BatchPCB offered great low cost prototypes. The drawback to this company was their four-week turn-around time. Since the second class of senior design is only about 11 to 12 weeks (shorter than the average semester of 14 to 16 weeks), this is of great concern. The team would have to have a finalized design that has been revised to be in working condition by the first week of Senior Design 2 in order to have 6 to 8 weeks to work on the building and testing. On the other hand, RushPCB has a great turn-around time of just one week with the drawback that it is slightly more expensive. Even if paying a little extra to have the boards delivered on time, ordering from the company with the fastest turn-around time would be more advantageous to the team. Considering both these great options, the team has decided to use the manufacturing company which gives more build and test time and that would be the RushPCB Company.

The team will order the control boards for both the eight by eight by eight RGB LED cube and the eight by eight by eight single-color LED cube at different times. Since, the team would like to finish the single-color LED cube first, the control board with the personalized design of that cube will be order first before the beginning of Senior Design 2. As for the RGB LED cube, since the team will commence working on this cube after the successful completion of the single-color LED cube, the control board will be order a few weeks after the ordering of the single-color LED cube control board.

3.3 Architecture and Related Diagrams

The block diagram seen in Figure 3.12 in the specifications section, almost demonstrates how the entire system can be divided into three major components or sub-systems; sense, process, and actualize. Figure 3.12 illustrates the components in each subsystem and how they connect. Each of the three major sub-systems in turn is made up of many parts Now that the components have been selected, this section will further discuss the selected instruments and explain their interrelationships with one another. Both the hardware and software aspects will be covered.

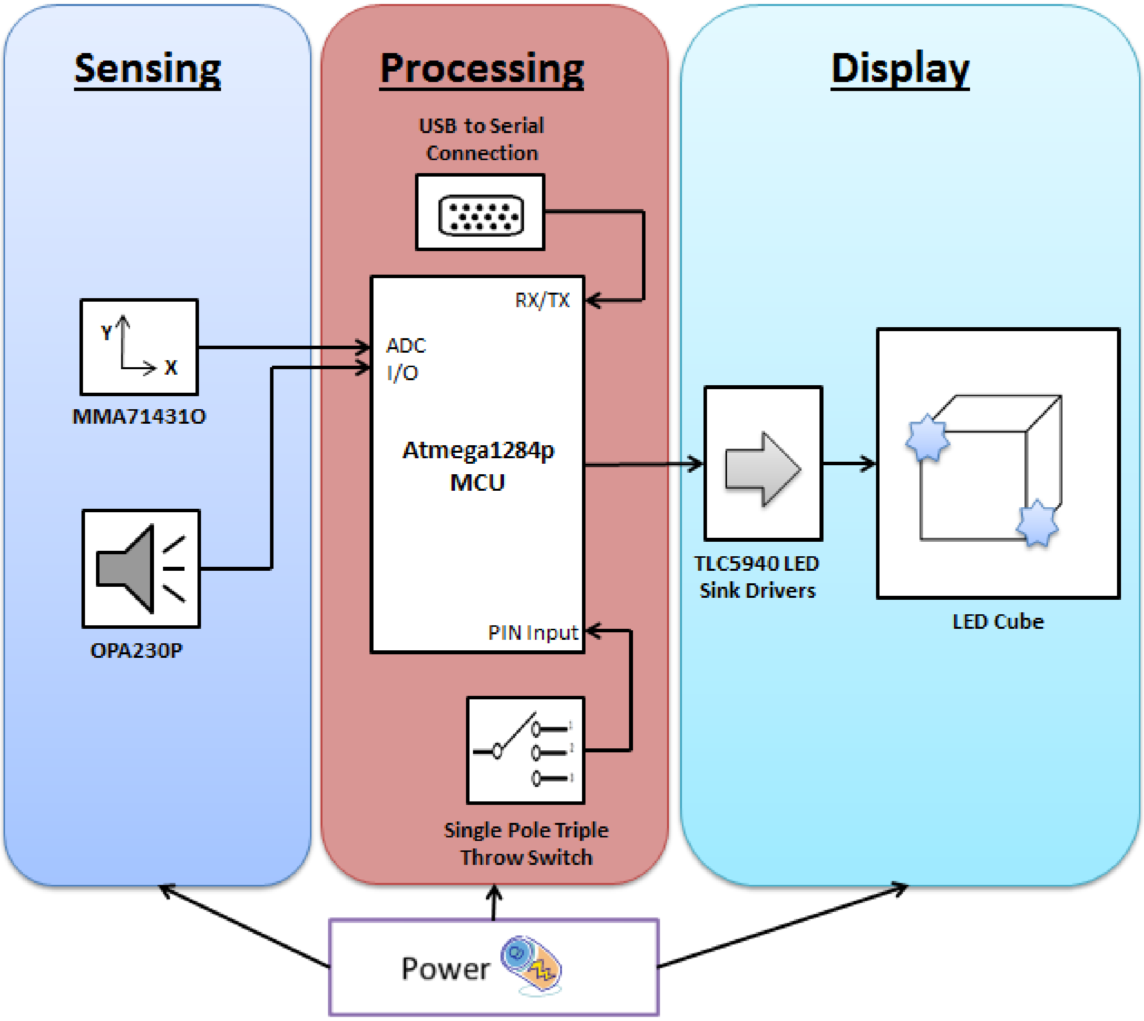


Figure 3.12: Subsystem block diagram

3.3.1 Hardware Architecture

Display Subsystem

The simplest subsystem would have to be the display. The display produces the visual spectacle that is calculated by the processing subsystem. The components that make up the Display architecture is the LED and LED driver, 513 LEDs and 5 LED drivers to be exact.

The cube shape will be constructed by eight layers of LED sheets, each sheet being eight LEDs in width and length. The completed product will be encased in acrylic for protection from the users. The Cube is held together through the LEDs themselves by using the anode and cathode legs as support through soldering them together. The soldering will be very labor intensive since there will be over 600 soldering points. In a layer the cathode leg is bent for 64 LEDs and used to create an 8x8 net in which every bulb is connected. The layer will then have 64 anode legs which will be used to connect the layers by stacking them onto each other. This will create 64 columns of bulbs connected by the cathode leg. During this process, it is important to keep enough space between the bulbs to in order to maintain sight of the LEDs through the cube. Too far apart or close together, take way from the three dimensional visual. Also replacing a faulty bulb could once the cube is constructed is of great difficulty, that is why testing each individual LED before assimilation is vital. Eight individual wires will be used to provide current to each layer. With the 64 cathode columns and 8 anode layers, the cube will be treated as a two dimensional array in the code, when activating a bulb it would simply be calling Cube [layer] [column].

The 72 connections go to the LED sink drivers. Each driver has 16 outputs therefore only requiring five of these components. Four of which will control the cathode columns and one will juggle the layers. It is still unsure as to whether the drivers will be placed on their own PCB board or share one with the processing subsystem. Other components sharing space with the drivers are transistors, capacitors and resistors. No special arrangement is required; as long as it appears organized that should suffice. These components will be housed in a container, out of sight, beneath the LED cube. The only external inputs required is power and data which will be passed in by the processing subsystem.

Processing Subsystem

This subsystem takes in the input from the sensing subsystem and provides data signals to the display subsystem. This is where all the information is processed using the ATmega1284. The architecture of this system is made up of the ATmega1284 microcontroller, MAX232 IC unit and the Amico 3 position SP3T switch components. The goal is to place all these components on the same PCB. The PCB will be stored in a plastic container with a protruding switch, RS-232 connection socket and a power input.

The switch is important since this will dictate what will be displayed. The sensing subsystem will constantly be feeding the microcontroller information so long as the power is on. What gets ignored and computed is based on which one of three pins on the microcontroller is receiving voltage at the time specified. Toggling between modes should be rather simple, since majority of users are familiar with a standard switch, a labels will be used to indicate the current mode on the box.

Communication between the computer and microcontroller will be done so through the MAX232 IC unit. An RS-232 cable will be used to connect to the MAX 232, which will convert the RX, TX, CTS and RTS signals for the ATmega1284. This is how source code will be uploaded to the microcontroller. The ATmega1284 will be programmed in the C language. As noted, practically every component will be connected to the microcontroller, therefore making it important that a sufficient number of pins are available. This also includes the components in the sensing subsystem.

Sensing Subsystem

This subsystem takes in the data based on its surroundings and produces an output which is sent to the processing subsystem. It responds to sound and motion based on the user's action. The system is comprised of two components the accelerometer and vu meter.

The DE-ACCM two-axis accelerometer will be used to record movement of the cube. It produces two analog outputs for the X and Y axis, which will be converted to digital signals by the Atmega1284 microcontroller. A three axis accelerometer was considered for measuring in the Z axis direction. But do to its price and lack of value added to the project, it was placed on the backburner. The size of said device is very small, almost equivalent to a five cent nickel. This is optimal since it will require less space on the PCB.

The VU (volume unit) meter is comprised of a microphone and an op amp. The microphone will be attached to the front of the casing holding the PCB board, possibly next to the mode switch. Sound will be analyzed by the op amp on the PCB board and produce analog output. This will be converted to digital signals in the microcontroller. There is possibility of having an IC op amp unit that produces digital signals allowing for flexibility

3.3.2 Software Architecture

The software is, like everything else, a major component; containing the instructions required to actually run the system and make it do something. The source code will be written in either C or C++, depending on the comfort level of the team. This will be written using the Atmel Studio 6 development platform, which is free of charge. Using the many software libraries and tutorials should facilitate the learning curve required to program the ATmega1284 embedded device. As long as the cube is receiving power, it will be in an infinite loop constantly checking the state or mode it is in. basic coding techniques, such as the use of if statements, for and while loops can be used to accomplish this. Figure 3.13 demonstrates a very high level overview of how the software will run.

There are three modes/states, VU meter, Pattern Display and Accelerometer. Each of which will have its own functions/methods, running the code for each process, of which will be called from the Main function. There will also be an output method which deals with sending the signals to the LED drivers. This send method will be dealing with the interrupt routine that will be required to multiplex the LEDs. It is uncertain how many more functions will be required to perform other task related to each state. One thing that they will all share is the manipulation of the LED 2D array, dictating which one is turned on or off.

In Pattern Display mode, there will be several animations that will run for a given amount of time, time being predetermined. Upon completion it will go to the next animation and have it run. The animations will be randomized, showing one that has not been displayed. Once all patterns have been seen, the animations restart.

For the VU meter and Accelerometer mode, these methods will be processing the signals received and converting it into data that the LED cube can represent. They will continue to process data so long as they mode is called from the main function. The output is then sent to the output method for visual display.

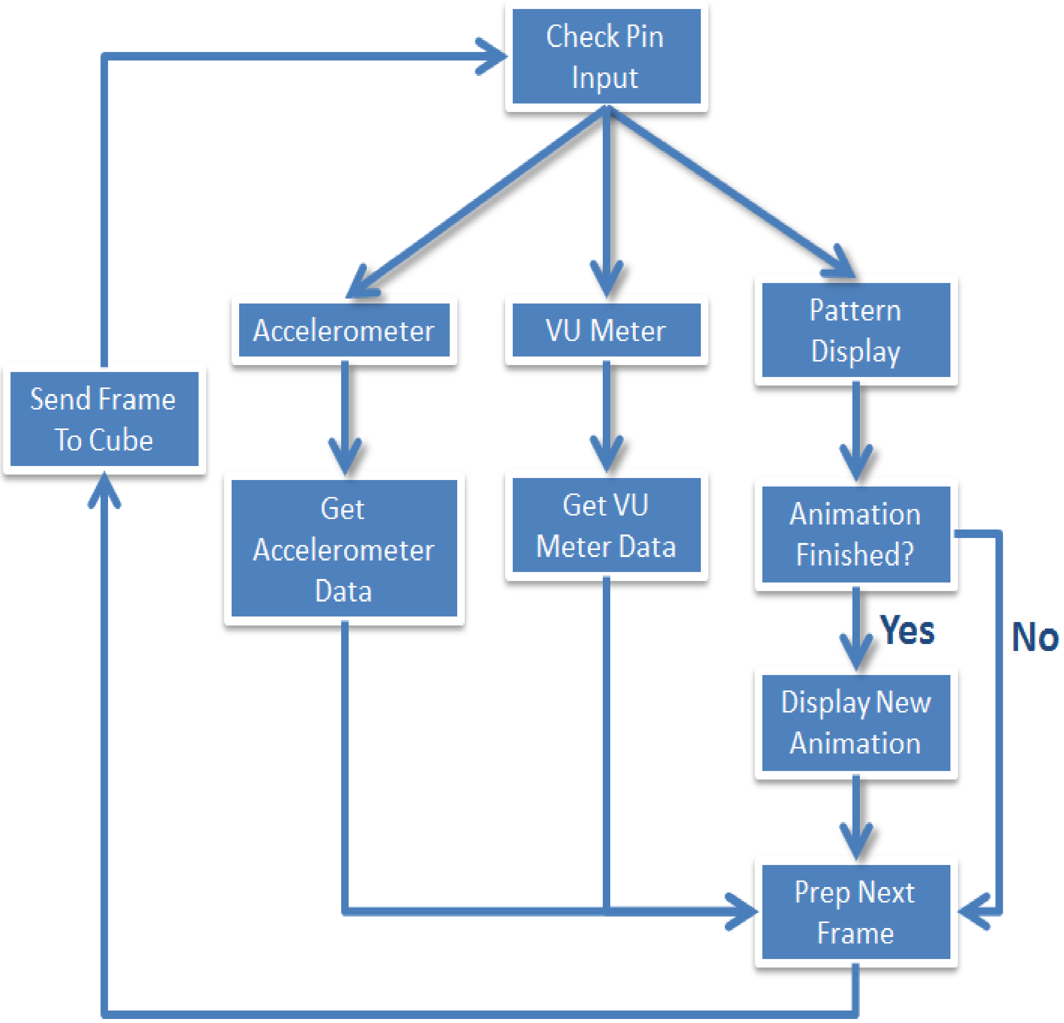


Figure 3.13: Software Flow chart

4.0 Project Hardware & Software Design

This section will discuss the projects function in greater detail. This includes selected parts and methodologies.

4.1 Hardware Design Details

The following subsections will describe each of the subsystem's hardware design details. The subsystems discussed are the processing, the sensing, and the display subsystem. The section concludes with a discussion of the power supply.

4.1.1 Processing Subsystem

ATmega1284

As established in a prior section, the ATmega1284 had been the selected microcontroller for the Multi-Functional Hexahedron. This piece of hardware was going to be the center of the entire system. It mainly contains 64 Kbytes, which after much research and deliberation, could have not containe sufficient memory to run the extensive programs desired. An upgrade to 128 Kbytes was considered, still following the same model. Upgrading to the ATmega1284 for its DIP pin package was ideal. Both microcontrollers were ordered from ATMEL for testing. The rest of this section will discuss the design with the ATmega1284 microcontroller in consideration.

The ATmega1284 has forty pins, 32 of which are I/O pins. To see the schematic layout of the connections see Figure 5.0. This is the current pin distribution:

- **4** ADC pins for accelerometer and VU meter (total)
- **3** GPIO pins for the VU meter
- **2** (RX/TX) pins for the RS232 connections
- **3** GPIO pins for the SPTT switch
- **3** SPI pins for LED drivers

Single Pole Triple Throw Switch

The SPTT switch is easily implemented into the design. This switch has four pins, one takes in a voltage, while the other three throw the voltage into different destinations depending on the switch position. Three GPIO pins are required from the microcontroller, pins twelve (PC2), eleven (PC1) and ten (PC0) are used. The switch used and schematic connections can be viewed in Figure 4.2.

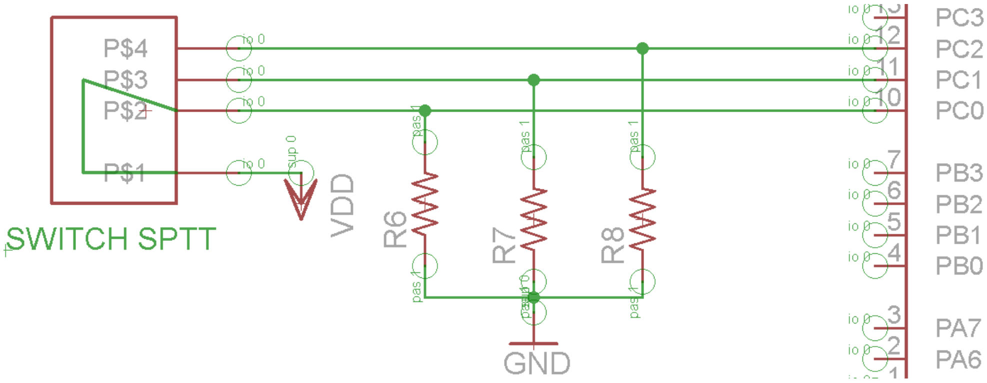
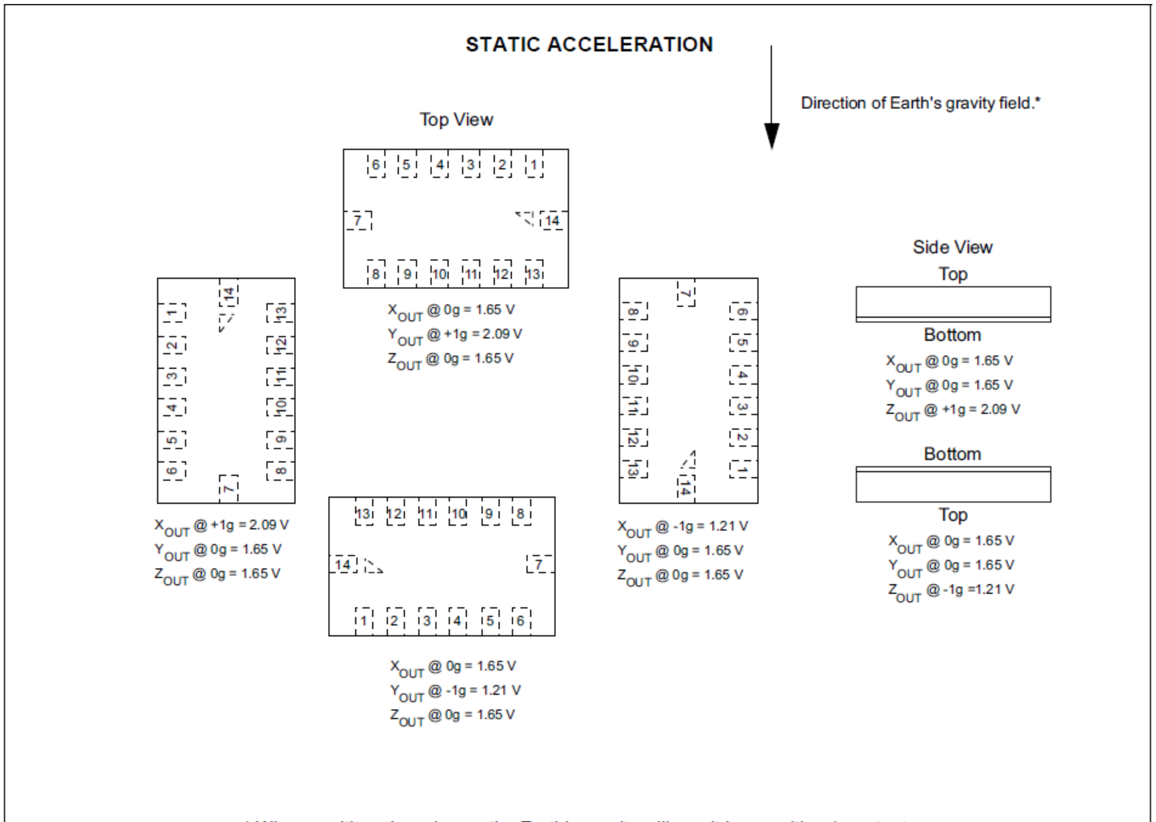


Figure 4.2: Custom Switch Component Design

4.1.2 Sensing Sub-Systems

MMA7341L Triple Axis Accelerometer

In order for the VU meter and accelerometer to communicate with the microcontroller, they must be connected to the proper pin. PortsA and PortsB which consist of a total of twelve pins, pin numbers one through seven and forty through forty-three. These pins can be used for analog to digital conversions. The MMA7341L buffered ±5g accelerometer functions between three to five volts. Wanting to maintain a low voltage usage, three volts will be used to run this IC. Therefore providing an output around 1.5V (give or take .2V) depending on position relative to the earth's surface, further examples can be seen in Figure 4.4. The X, Y, and Z output pins on the accelerometer will be connected to pins A1, A2, and A3 on the ATmega1284p.



* When positioned as shown, the Earth's gravity will result in a positive 1g output.

Figure 4.4: XYZ output relative to the Earth's surface.
Printed with permission from Dimension Engineering [5]

VU Meter prototype

During the research portion, two integrated circuits were selected to be built as a prototype and then run a series of testing to ensure the circuit is worked properly and is providing the correct output signal or voltage. The three test cases are analyzed; finally, one of the two will be selected to be implemented in the final design.

LM3916

This prototype is show in Figure 4.5 with the input provide by and iPhone playing a YouTube video. The prototype was easy to build because only require three resistances. Two connected as voltage divider to select a reference point and the other to have a LED voltage lower than voltage supply as require in the data sheet. A linear voltage regulator was included in the circuit because was on hand and only used for the test with the iPhone to ensure not damage from a feedback voltage, but is not require. For this prototype we follow the schematic already provide in the data sheet, but in this cases not coupling capacitor are need between the anode and ground because the distance between the chip and the LED are less than six inches.

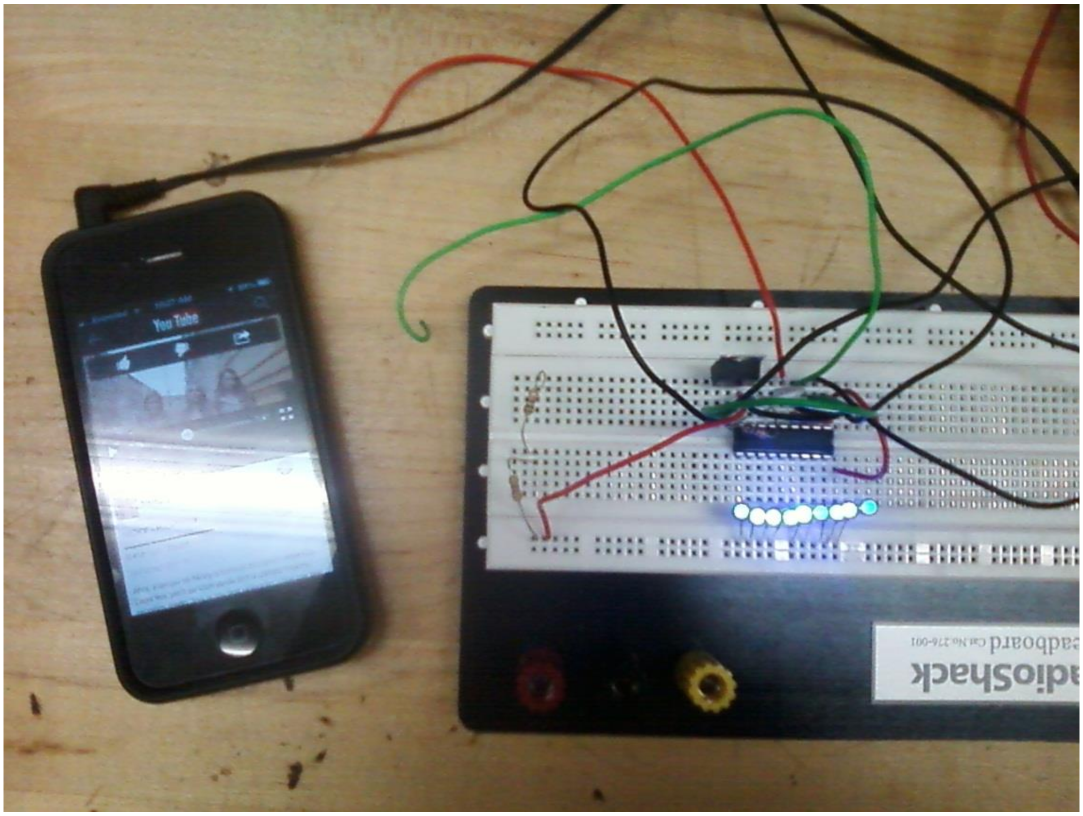


Figure 4.5: LM3916 Prototype

MSGEQ7

The prototype is show in Figure 4.6,the schematic for the MSGEQ7 is the same as the one provide in the data sheet for typically use with the only different that two 100 pF capacitor connected on parallel from the clock oscillator to ground was use instead of the 33 pF because was the only available at the point on time. As show in the picture an Arduino UNO was use in the prototype. The LED output uses the digital pin six to twelve with a 440 ohms series resistance. The strobe is attached to digital pin two and the reset is attached to digital pin three. The output from MSGEQ7is read from multiplexer using analog input 0. The code implemented was taken from the Arduino open library. The codes basically take the input and put it into an array of seven bit binary number. Then, is sent to the select output pins.

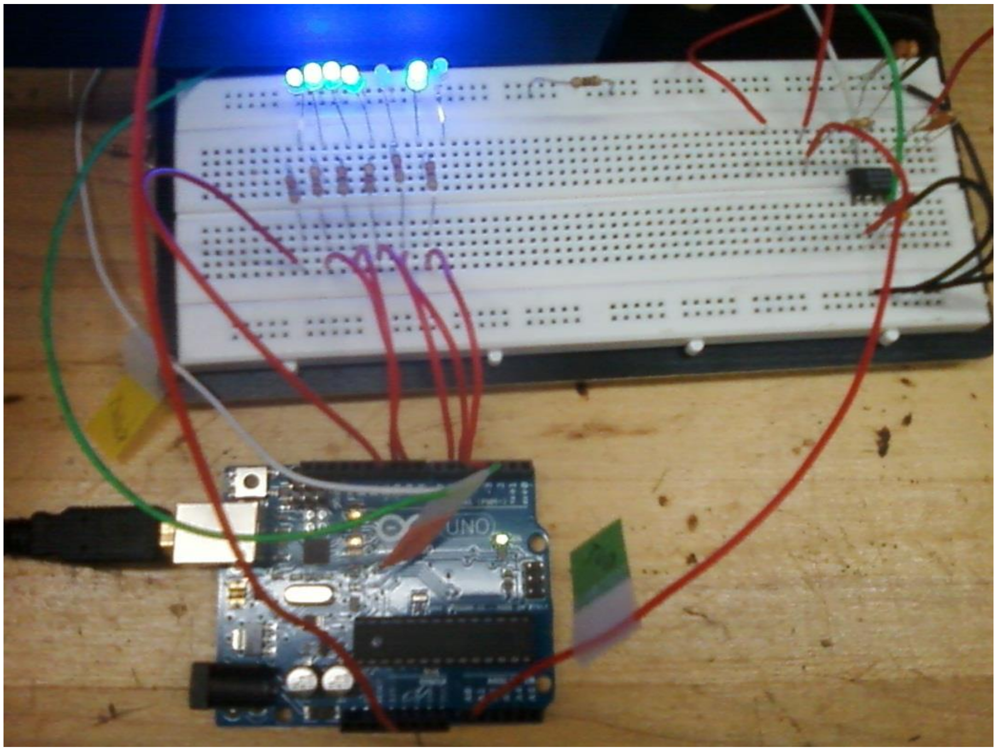


Figure 4.6: MSGEQ7 with Arduino UNO Prototype

Test # 1

Setup:	Connect two wire two the input signal
Description:	Test for noise and to see if the circuit is working
Initial Condition:	Not touching the input wire
Steps to run test case	Wrap the input wire with the hands and move it around to produce noise signal
Expected Behavior:	Not LED's should light up in the beginning and then, some will light up but not all
Result LM3916	All the LED light up, this proof to be very susceptible to noise
Result MSGEQ7	Only four out of the seven light up.

Test # 2

Setup:	Connect the input signal to a frequency generator
Description:	Test for noise and to see if the circuit is working
Initial Condition:	Set the frequency generator to it minimum value of Hz
Steps to run test case:	Gradually increase the frequency until all the LED light up
Expected Behavior:	The number of on LED should increase as the frequency increase
Result LM3916	The results were as expected and all the LED’s light up
Result MSGEQ7	The results were as expected and all the LED’s light up

Test # 3

Setup:	Connect two wire two the input signal
Description:	Test for noise and to see if the circuit is working
Initial Condition:	Not touching the input wire
Steps to run test case:	Increase the volume of the music player until all the LED light up
Expected Behavior:	The number of on LED’s should increase as the volume increase
Result LM3916	The LED’s continually when up and down in a liner patter
Result MSGEQ7	The LED’s continually when up and down in a liner patter but the output connect to pin seven have error.

After analyzing the results, the final decision is to send the signal directly from the op-amp to the microcontroller in the final design because not multiplexer are need it, also is less susceptible to noise signal. In the final design the noise can be also control by the software coding.

MSGEQ7 VU Meter

The MSGEQ7 model takes audio and divides the spectrum into seven bands. The bands are multiplexed then released. The MSGEQ7 requires between 2.7 to 5.5 volts functioning. The output produced is in a digital format, not needing to use the analog to digital converter on the microcontroller as we originally thought.

There are three pins required to attach the MSGEQ7 to the ATmega1284. The STROBE, OUT and RESET pins on the VU device are hooked up to the microcontroller at pins 41 (PA1), 40 (PA0) and 42(PA2). The schematic layout of the connected sensing subsystems can be viewed in Figure 5.0. Further discussion of what the source code will do with the data from the MSGEQ7 and the DE-ACCM5G accelerometer can be found in the software design section of this paper.

Resistance R13 and R12 are use as voltage divider and to make the microphone work around one point seven volts. Also, R12 works together with the microphone to form and RC circuit to get a higher quality response. Resistances R14 and R17 are used for the inverter amplifier to have an ideal gain of fifteen. R18 works with the potentiometer to produce an ideal gain from one to five. Capacitor C5 is used to filter out any noise. Below is the schematic layout of the VU meter.

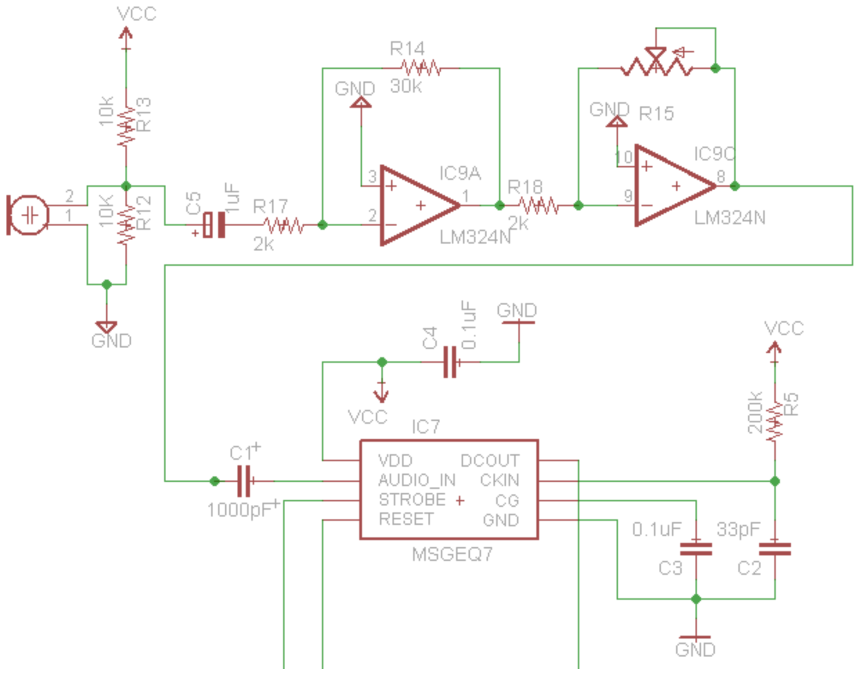


Figure 4.7: VU Meter schematic layout

During the senior design lab, the schematic circuit changes to more simply schematic to reduce the amount of part and ensure a reliable working circuit. Figure 4.7B is the VU meter schematic used in the final project.

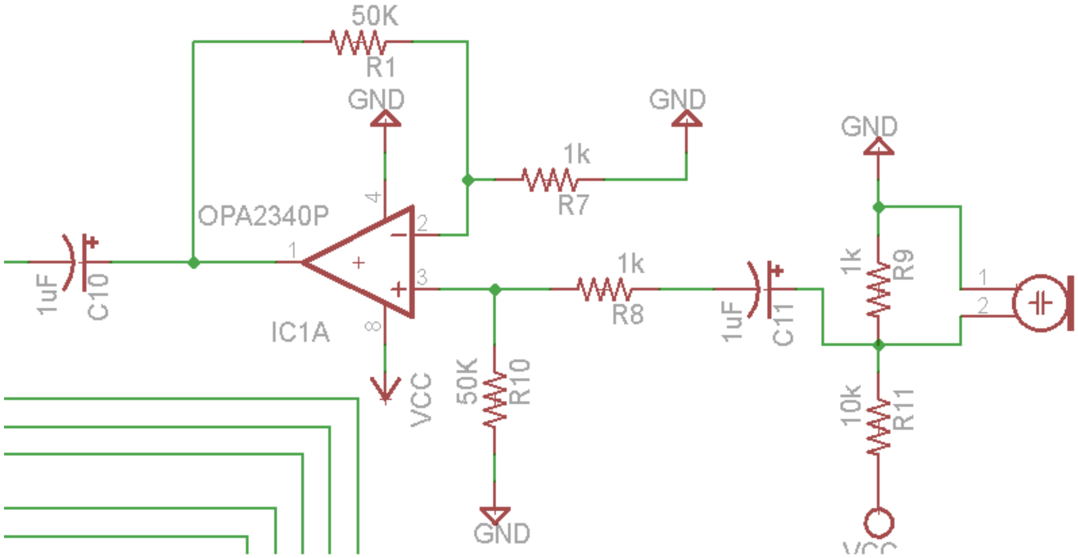


Figure 4.7B: update VU Meter schematic layout

4.1.3 Display Subsystem

TLC5940 LED Sink Driver

The TLC driver was mainly chosen due to its free of cost method of obtaining. The down side is the driver is difficult to work with. The driver has twenty-eight pins of which twenty-four will be used. Pins one through fifteen will be used to connect to the LED cathode columns and the rest will be connected in between each other and the microcontroller. The Drivers will be in a daisy chained layout as seen in Figure 4.8, displaying the three of the drivers used. The Serial Peripheral Interface protocol will be used and issued through pin 16 (PC6). Pin 7 (PC7) will be used or the clock and the SS (slave select) is connected to pin 14 (PC4). For daisy chaining, pin 12 (SOUT) will be connected to the pin 26 (SIN) of the next driver. Since only a single color LED will be used, the rest of the pins will be unused since they are used to control color and brightness. If RGB LEDs were to be used, 12 TLC5940 sink drivers would be required taking up much space and energy.

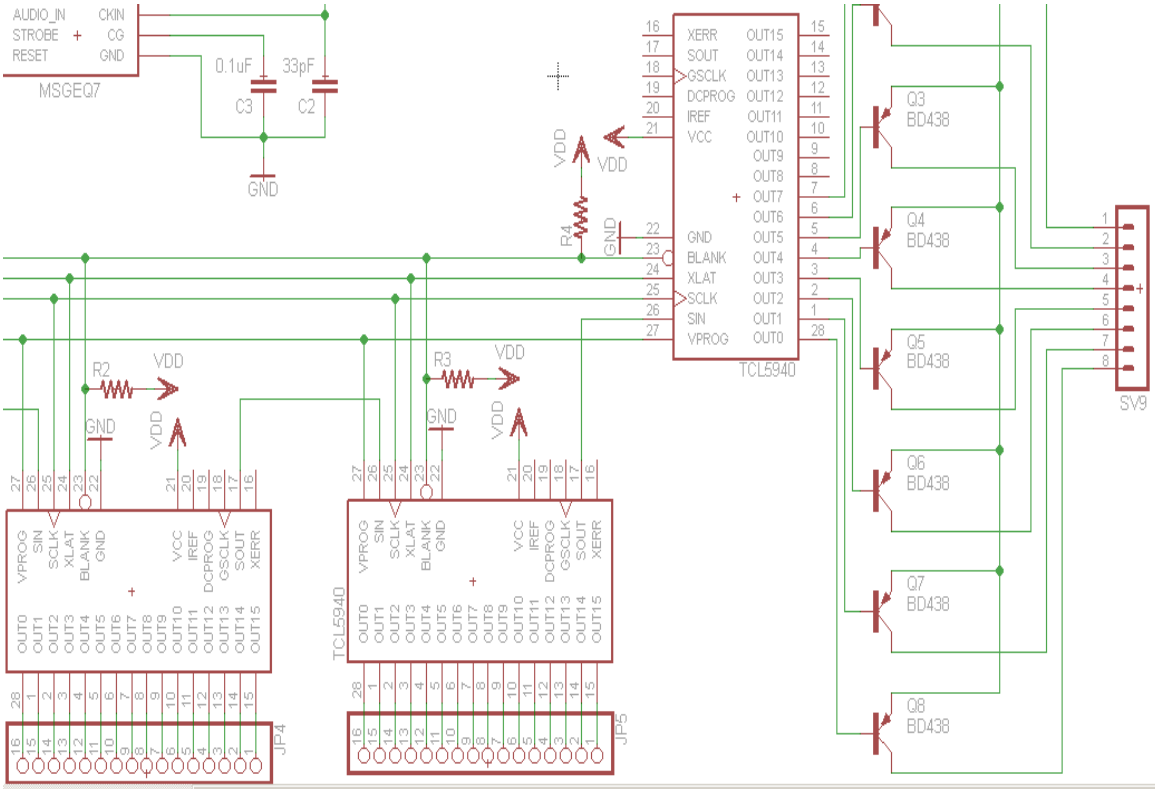


Figure 4.8: LED drivers formation

Grayscale PWM Operation

Since the team decided on using the TLC5940 instead of the more expensive STP16CP, the PWM brightness control was used from within the driver instead of through the software. The PWM brightness control works by oscillating the power received by the LED at a rate at which the eye cannot see. Unfortunately since a microcontroller is being used, only the digital output pins will be used which only have the setting of high or low. The LED driver including the PWM control allows for the oscillation of the power received by the LED to be at a rate of 30 MHz in 4096 steps that can be individually controlled.

The grayscale PWM included in the TLC5940 helps us control the brightness by a series of complex timings and relationships between the different signals. It begins with the falling edge of blank. The first GSCLK pulse after blank goes low increases the gray scale counter by one and switches on all OUTn with gray scale value of zero. Each following rising edge of GSCLK increases the grayscale counter by one. The TLC5940 compares the grayscale value of each output OUTn with the grayscale counter value. All OUTn with grayscale values equal to the counter values are turned off. A BLANK = H signal after 4096 GSCLK pulses resets the grayscale counter to zero and completes the grayscale PWM cycle. When the counter reaches a count of FFFh, the counter stops counting and all outputs turn off.

Serial Peripheral Interfacing

The Arduino board being used for the mockup cube contained 4 pins which support Serial Peripheral Interfacing, which allowed quick communication between the Arduino board and other local devices, which in the case of this project will be either the 6 TLC5940 LED drivers used in the single-color LED cube or the 12 TLC5940 LED drivers used in the RGB LED cube. With the SPI connection, there was always one master and depending on how many extra devices, one or multiple slaves. In the case of this project, there were 11 slaves The slave to master relationship will be controlled by the MISO, MOSI, SCK, and SS lines found in all of the TLC5940 LED drivers. These lines can be found at the following pins on the Arduino board:

- MOSI – Pin 11
- MISO – Pin 12
- SCK – Pin 13
- SS - 10

The Master In Slave Out allows for communication of data from the slave to the master, the Master Out Slave In allows the master to communicate data to all the peripherals attached, the SCK provides clock pulses which synchronize all data transmitted by the master, and finally the Slave Select, the most important pin in regards to our project, allows the master to select which peripherals to enable or disable. The advantage and necessity of using this interfacing is that it allowed the master to control all different LED drivers for all functions needed by just simply setting the slave select to high or low. This means that the LED drivers can be either executed simultaneously or separated. In some cases, both the LED drivers that control a certain layer as well as the LED drivers that control a certain color will be needed to be on simultaneously. In conclusion, the use of the SPI was helpful in controlling the LED drivers at the rates needed to perform the animations.

LED Cube

The LEDs cube is essentially a two-dimensional array that is arranged to appear like a three-dimensional hexahedron. The manner of construction is discussed in section 6.0 of this document. 72 pin connections were required to connect the 64 cathodes and 8 anodes input/output connections that were created. These connections were established between the TLC5940 LED sink drivers. The output pins 1 - 15 and 28 were connected using male pins from the PCB created. A series of ribbon pin connectors will be used to do this accordingly

4.1.4Power Supply

The voltage input is from regular power socket. The follow resistance and capacitor selection are use in reference of picture 4.9. two hundred twenty micro-Farad is use as bypass capacitor to ensuring only direct current is supply to the circuit. The power supply uses the linear regulator LM317N and is connect to make it adjustable. The resistance R11 is chosen to be one kilo ohm and R9 is two kilo ohms as a result for the calculation to select five volts as the circuit operation voltage supply. Also, one kilo was connected to the LED voltage indicator reduce the brightness.

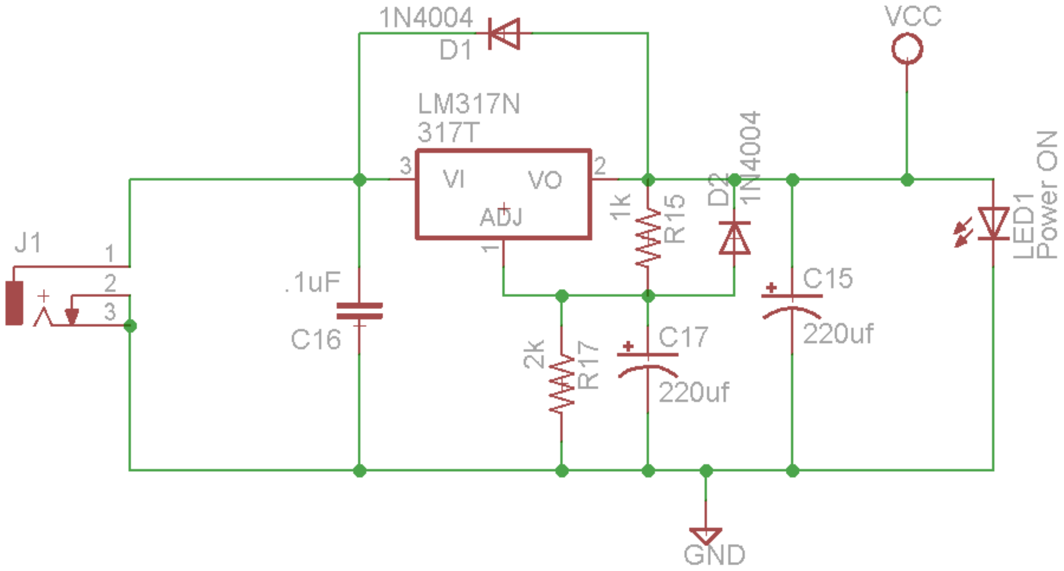


Figure 4.9 : Voltage Regulator Eagle Design

4.2 Printed Circuit Board

The printed circuit board (PCB) was the most expensive component and was taken into consideration for a while before ordering to reduce cost. For the design of the PCB, CadSoft EAGLE PCB Design software was used. This product tends to be widely used and very popular with hobbyists. It provides a free version which had everything required for our project’s needs. Since its widely used, many resources are readily available for learning how to use the software in designing schematics, custom parts and PCBS. The EAGLE website provides in depth tutorials, workshops, libraries and other resources to aid one in their PCB portions of their projects. The figures in this document regarding schematics have all been done through EAGLE.

The size of the PCB was 7.7 in. x 7.7 in. Ultimately, fitting all the components on one board saved us money. The goal was definitely achieved since many of the components were rather small and others were not going to be directly attached to the board. This was large enough to fit inside the container that held the LED cube above.

The number of layers is rather limited. The free version of the software only allowed two layers to be implemented, the bottom and top layers. Therefore only two layers were implemented. Trace thickness will be the default value for the

PCB software. It will be altered accordingly. Overall the main issue was to reduce cost, since this piece of hardware was going to run us the most money.

4.3 Software Design

There were four basic classes to manage during the coding section of this project. The four classes were Main, Animations, Accelerometer and VU Meter. The code was written in C++ like structure (Arduino IDE).

The Main class initiated the cube and its variables instantiating the public cube[] array and temp_cube[] array being the main variables for manipulation. This function will have an infinite loop, constantly checking the switch input interrupt, dictating the current mode. The main class also handled the output method for sending signals to the LED sink drivers. The clearing of the cube array was constantly taking place as the cube was changed from state to state. A series of IF statements or switch cases were used to alternate between the different modes calling the required classes to implement the given action. Also the main class took charge in directing though the signaling of buffers and interrupts.

The Animations class was the largest out of the four. This was due to storing all the varying animation functions. Each given animation has its own function and specified run time. The more animations, the longer the code in turn the more memory got used up. The instantiation function set all the ports and enable switches. Functions to delay time before sending byte data will be implemented

The Accelerometer class handled the data input and calculations in regard to said device. There were functions that calibrated and setup the required ports. A function to take measurements and link results to the cube was implemented as well. Gathered and modified data was sent to the Main class for further manipulation and then sent to the out function for visual manifestation.

The VU Meter class was set up similar to the Accelerometer class. But obviously perform different tasks. There were functions that calibrated and setup the required ports. A function was required to read the spectrum values and modify for cube use. A function to calculate the spectrum into visual form was called upon and then sent out to the Main class for true visualization. For testing and simplification purposes, the VU meter output display was set to an outer face of the cube, in which a function was created for.

The method for implementing this code will be done in phases discussed in the Testing Section. Each Class will be coded in such a way to work with the Main class and then tested and debug. Once each class is believed to properly work, it will be touched up to be integrated with the rest of the classes onto the microcontroller. It is believed that this project can turn out to be coding intensive. This works out since the team is composed of three computer engineers. This will allow the electrical engineer to focus more on the hardware aspect of this project. In regards to the LED cube, If time permits, more complicated animations can be coded. Also the possibility of make a game with the accelerometer is on the table based on time availability.

4.4 Housing Unit

The housing unit was holding the PCB and LED cube. It was divided into two parts. The box unit was made out of acrylic plexiglas. Aluminum was also under consideration since it is an easy material to manipulate. But since aluminum can conduct electricity, have electric components inside could cause severe problems. The bottom of the box had rubber stubs to stand on and prevent it from slipping. The SPTT switch was mounted in the front, as well as the power input and RS232 connecters. The hollowed out inside will hold the PCB and wires which will be bolted down to prevent it from moving around and banging on the sides, possibly loosening wires. The LED cube itself will be mounted on top of the box passing the wires through the 64 holes to connect to the PCB. Plastic rods may be used to provide further support for the cub structure since its main support will be the anode and cathode legs for which it was built upon. Adding more support will be wise since the cube will be handled, often by possibly untrained individuals who wish to be amazed and experience its life changing awesomeness. A cube acrylic casing was placed over the LED cube, enclosing it for protections. The acrylic casing was removable and secure with four screws. The acrylic casing was slightly bigger than the LED cube itself with about an inch between the LED and the wall of the. The color of the print circuit that hold the LED cube was painted black for special affect. Everything previously mentioned was applied for the RGB LED cube construction. Since the RGB LEDs are larger (5mm), the container will also increase in size. This is also due to the fact that a larger or more PCB boards will be required to run the RGB cube. Figure 4.11 is a mock up representation of the finished product.

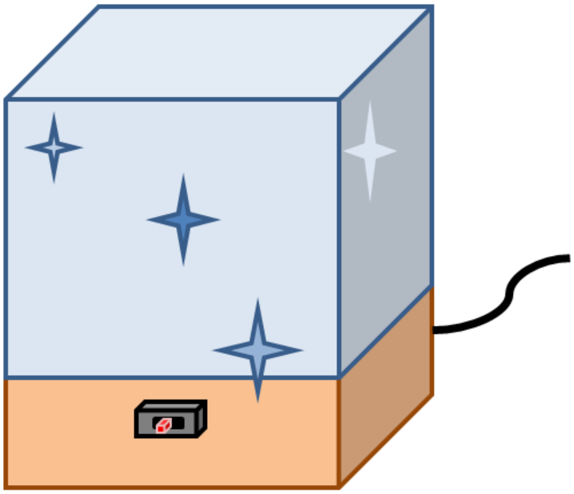


Figure 4.11: Representation of the housing unit for the LED Cube

4.4.1 Casing Cut Process

The casing is final part of the project and will capture the attention of the spectator. Therefore, it has to look the best, most like and art work. The casing look as the easier task of the entire project, but when the group decided to build a casing for the prototype, a few lections where learn. We use the try and error process. First attend was to cut the Plexiglas by using a ruler and a knife. The cutting edge have to pass multiple time next to the ruler until a track is form; then bent the material to separate into parts. This method turns to be no as expected because in one of the tutorial video it show to be very easy task and a perfect line as result. In our case, the edges were very sharp and crooked and took too much effort to cut a single piece. The second method was to cut the Plexiglas with a hacksaw. This method was easier but any of the group members has a very steady hand to cut a perfect line. Therefore, the pieces were crooked and also it create small crack on the sides. The small crack is unacceptable because the main structure can fail if is expose to abrupt movement as describe in the design parameter. The third method was to cut the material using a table saw. This option is maybe too powerful and dangerous but at the end get better results. In the beginning the Plexiglas was shattering and the final cut looks worse than the previous method. The Polycarbonate sheet use have a thickness of 0.093 inches, the cutting disk of the table saw was lower to cut approximately 0.07 inches. And the final cut is done by using a knife and bending the material. With this method an extra edge is add to the final structure; also, adding extra support and more space for the glue to hold the final pieces together. Figure 4.12 shows the final product and the red circle shows the extra cutting edge.

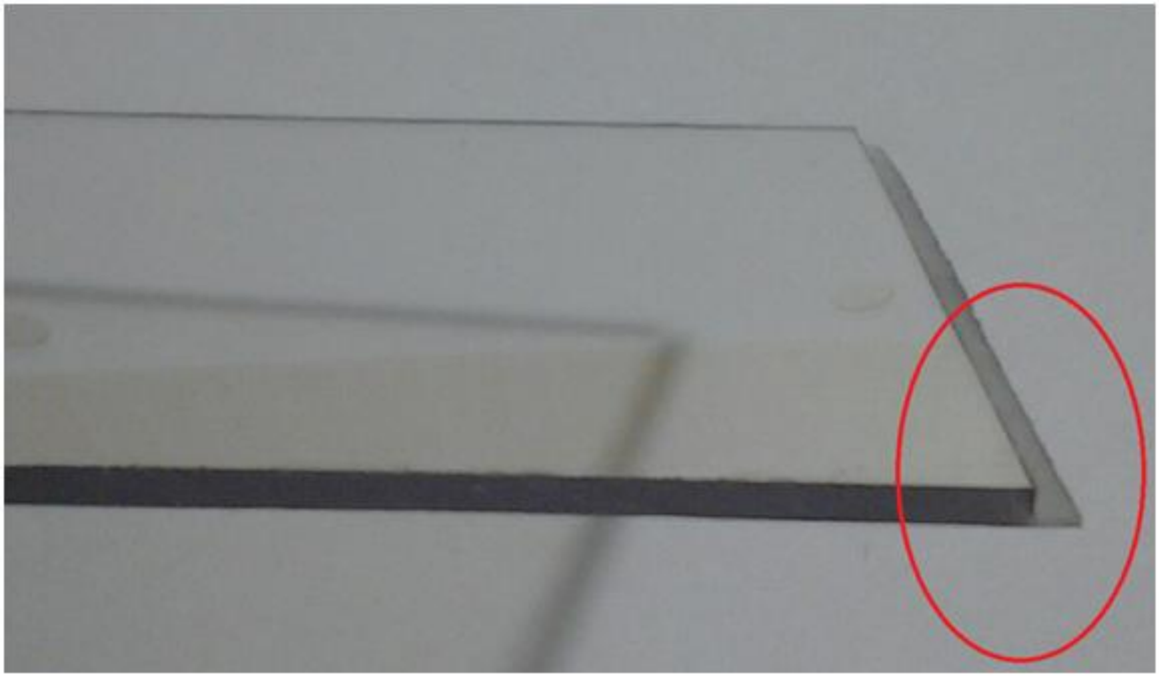


Figure 4.12: Final Plexiglas Cut

5.0 Design Summary of Hardware &Software

The team had already decided on what was intended to be built as well as on the components that were used to build both sets of LED cubes (RGB and single-color). In order to have a better understanding of how LED cubes function, the team decided to build a small scale LED cube that had some similar animations to both the eight by eight by eight single color LED cube as well as the eight by eight by eight RGB LED cube. This phase of the project was already done and documented. The next step was to develop the more complicated, full prototype of the eight by eight by eight single-color LED cube. Before the actual construction of these LED cubes, the team decided to simulate the behavior of the software as well as the hardware in real life situations using CADsoft Eagle PCB Designer, Multisim, etc.

5.1 High Level Design

The next few subsections will focus on the high level design of the cube. The sections will mainly focus on the prototyped single-color LED cube.

5.1.1 8x8x8 Single-Color LED Cube

For the eight by eight by eight single-color LED cube, the general structures Is not very different from the RGB cube. The cube had three different modes. The first mode is the main animation mode. The main animation mode cycles through various pre-determined animations that do not rely on any components other what has already been pre-programmed. These animations will have various physical displays. The cube light up in a diffused manner; it light up layer-by-layer, etc. The exact different types of animations was be determined when the actual cube is built and tested.

The second mode was the vu mode. The vu mode of the animations was determined by the input source connect to the vu meter. This animation will work similar to most equalizer animations in music players on computers. The vu meter read the different types of signals received from the input and communicate them to the microcontroller, which then drive the LEDs to light up in different patters based on the signals. The actual animations that was displayed vary depending on what type of signal is coming in, as well as what animation cycle is being displayed. The model allowed for the use of either a microphone or a music device with a regular 8mm headphone jack. The team was shooting for more than one style of animation but will at then end one working by the finalization of the cube.

The third mode of the cube is the accelerometer mode. This mode of animations will rely on the accelerometer chip of the teams choosing. The goal behind this mode of animations is to be able to pick up the physical LED cube and rotate it to see that the animations follow the movements. One of the main animations the team implemented on this mode is the water feature. The user is be able to pick up the cube and there will be some LED's on resembling that of a cube filled with water. When the cube is tilted in any direction, the LEDs are able to follow the movements smoothly throughout the cube. The following diagram depicts the main blocks of the design in general not exact):

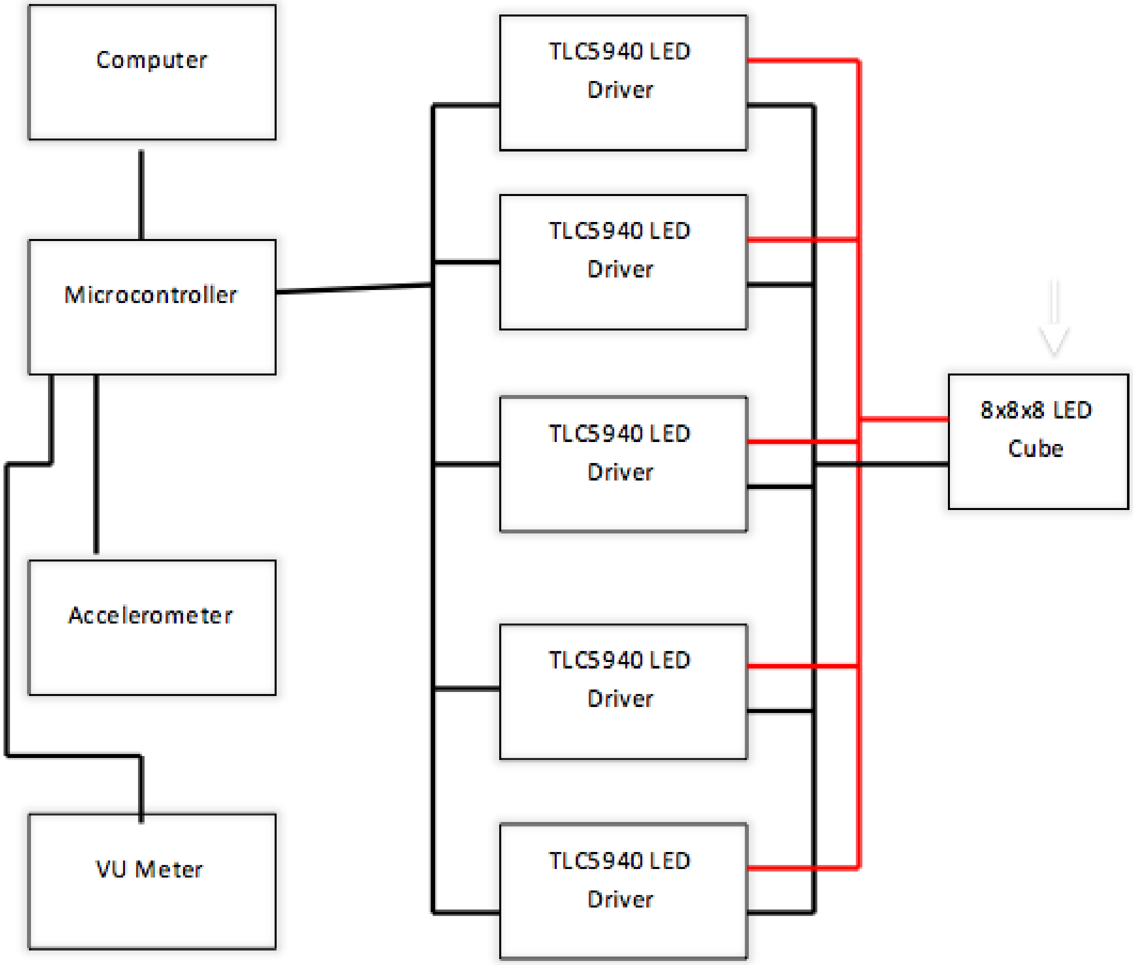


Figure 5.1: Block diagram for general LED cube design

5.1.2 Parts list

The following table lists the various components and their respective quantities needed to build the single color LED cube.

8x8x8 Single Color LED cube

Part	Quantity
Microcontroller ATmega1284	1
3mm 2-pin Round top Diffused LED	513
2N3904 Transistor	8
TLC5940 LED Driver	5
Ribbon Cable	1
Dual Row Header	5
MAX3232 RS232 to TTL Converter Module	1
3 Position Switch	1
Circuit Boards	1
DE-ACCM5G Accelerometer	1
MSGEQ7 Equalizer	1
Electret Microphone	1
LM324 Operational Amplifier	1
Capacitors	10
USB Class “A” Connector	1
Potentiometer	1
Resistances	16
7805 Voltage Regulator	1

The following table lists the various components and their respective quantities needed to build the RGB LED cube.

8x8x8 RGB LED Cube

Part	Quantity
Microcontroller ATmega1284	1
5mm 4-pin Clear RGB LED	513
2N3904 Transistor	8
TLC5940 LED Driver	13
Ribbon Cable	1
Dual Row Header	5
MAX3232 RS232 to TTL Converter Module	1
3 Position Switch	1
Circuit Boards	1
DE-ACCM5G Accelerometer	1
MSGEQ7 Equalizer	1
Electret Microphone	1
LM324 Operational Amplifier	1
Capacitors	10
USB Class “A” Connector	1
Potentiometer	1
Resistances	16
7805 Voltage Regulator	1

Figure 5.1 in the next page displays the schematic layout of the LED cube.

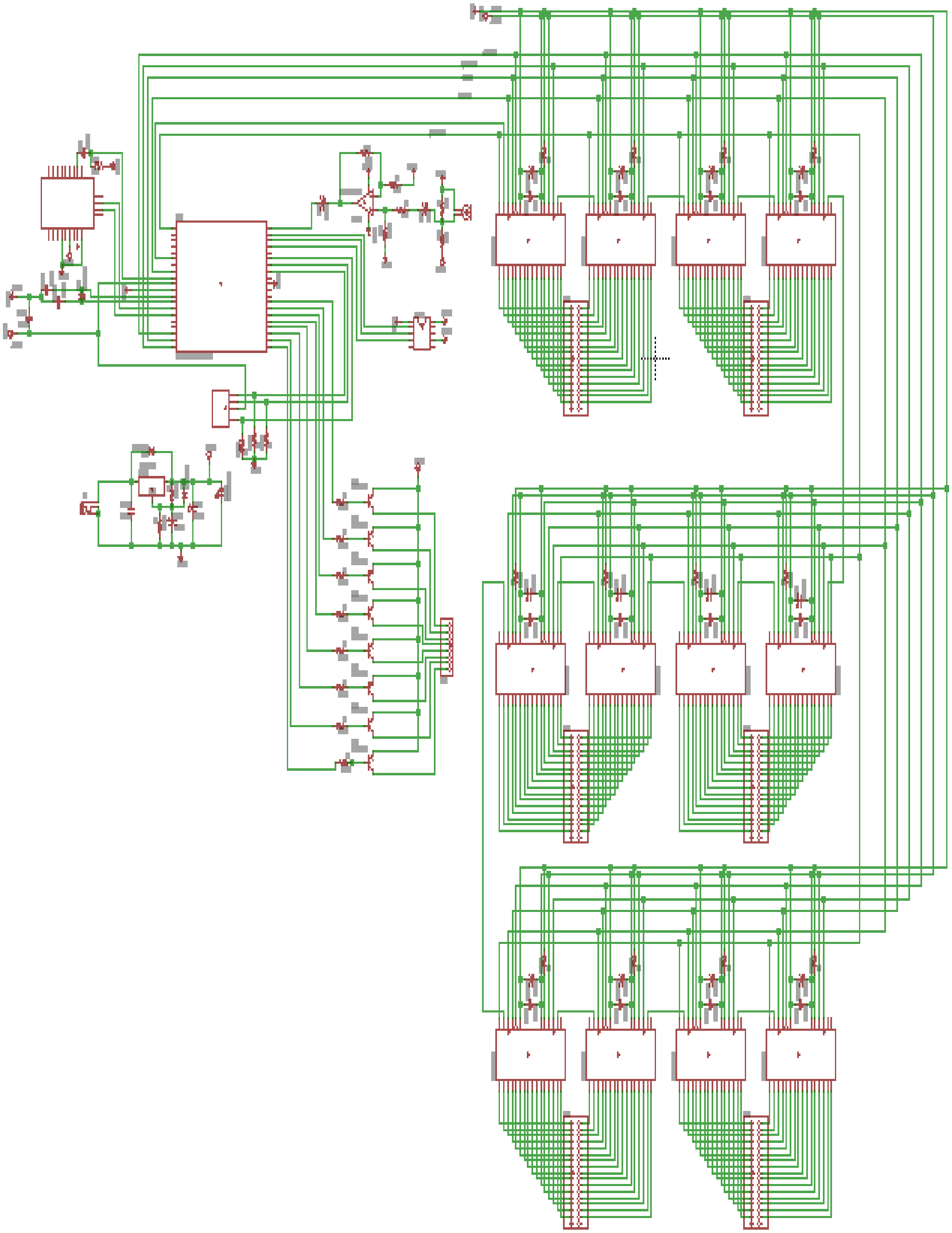


Figure 5.1: Schematic Layout of the LED Cube

5.2 Software

The programming of the design was done in C, using the Arduino IDE. The program has the main routine and three subroutines. The main routine takes the reading provided by a switch and decide which subroutine will be addressed. The first subroutine is the animation mode. It has a set of different animations that runs for a predetermined amount of time. The animation subroutine increased in complexity and was expanded as time allowed during the Senior Design II semester. The second subroutine is the accelerometer mode. It takes a first reading from the accelerometer and sets it as the original position. Then, it reads the accelerometer again and determines if motion has occurred and displays any changes. The third subroutine is the VU meter mode. It takes the reading from the first analog pin, filters the noise and then displays the appropriate animation. A better description for each of these three subroutines is shown in the following class diagrams. Each subroutine checks if there is a change in the input switch. Also, the LED brightness is done by the LED driver and is not considered in the programming, only the highest brightness settings will be employed for the purpose of these animations.

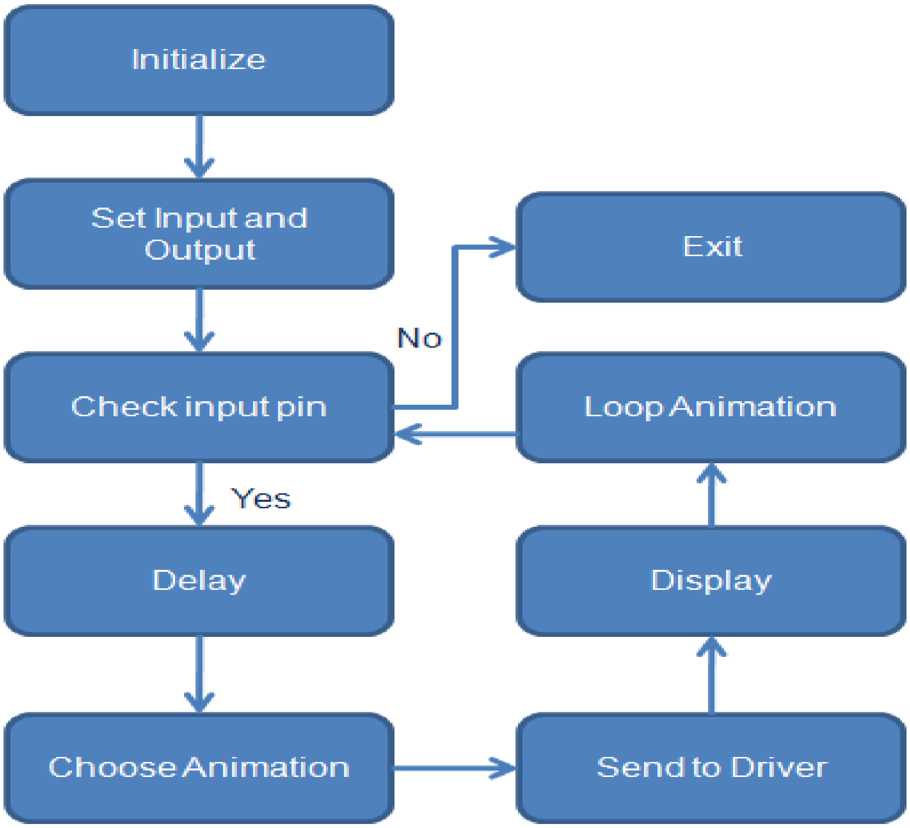


Figure 5.3: Animation Class Diagram

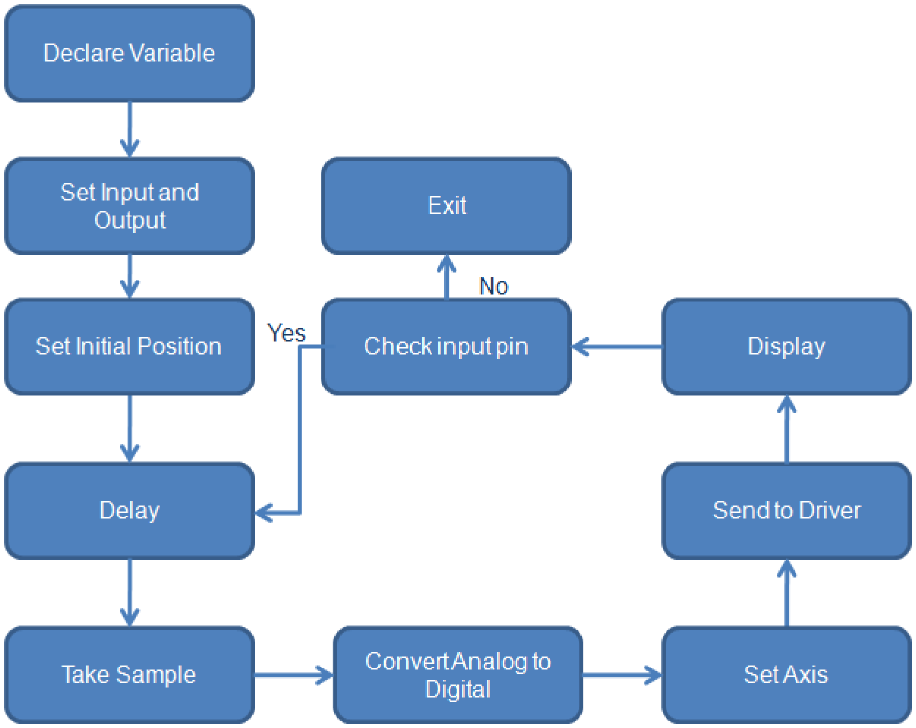


Figure 5.4: Accelerometer Class Diagram

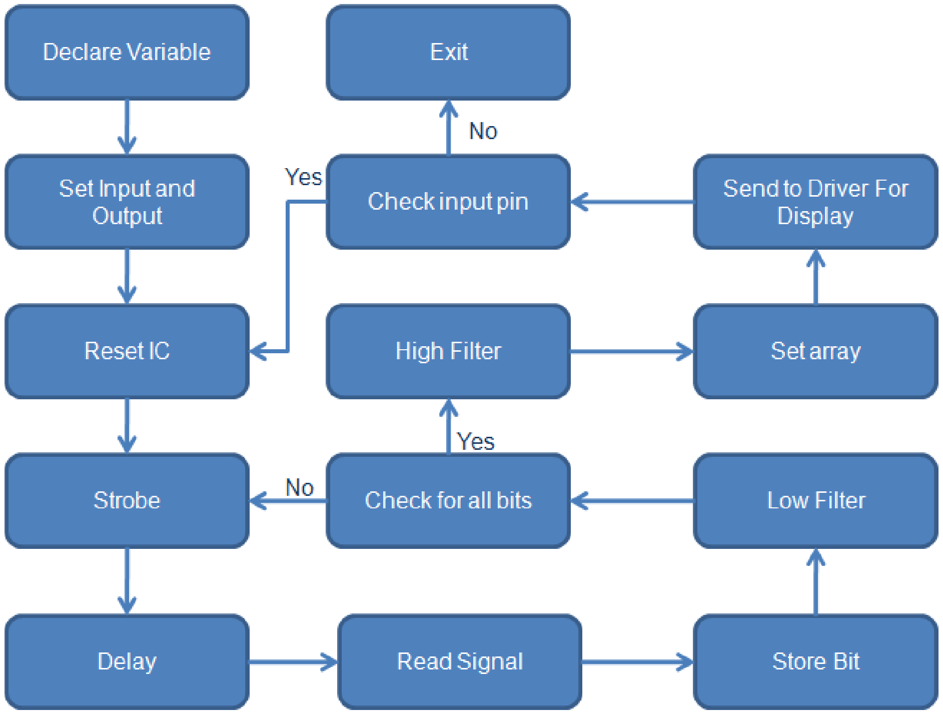


Figure 5.5: VU meter Class Diagram

5.3 Design Issues

Due to the fact that the team decided to build two different light emitting diode cubes, the group has without a doubt come across various design issues, which have resulted in redesign phases as the project progressed.

Single color LED Cube

Maintainability

The team decided to encase the LED cube. The team chose a set of LEDs, which are low in energy consumption, and this means they should withstand longer than other high-energy consumption LEDs. As for the software, the controller board was designed so that by connecting it to the computer, the programs could be modified or replaced altogether, making the LED cube a more modular project. Once the software is tested and loaded on to the controller, the software does not require maintenance.

Testability

To get a better understanding of what to expect when making the designs and schematics created a reality, the team decided to build a mockup LED cube of dimension three by three by three. Fortunately, there are many guides online to reference yourself on. The team decided to follow a few of these guides to build the three by three by three cube using an Arduino microcontroller. The software used in this mockup cube is all open sources, which allowed the team to save time that could be used for potential errors in building the physical cube. Once this cube was built, the team had a better understanding of how LED cubes work. In terms of the actual eight by eight eight, after getting the experience of how to create an LED cube from creating the mockup, it would be easier for the team to recognize any errors or mistakes on the physical cube, its connections, as well as its software. For the animations used on the 8x8x8cube, the team could use any of the open source software to test the proper functionality of the cube.

Performance

The cubes performance was observed during the testing phase of the project. It was imperative that all LED layers as well as all individual LEDs light up correctly and in a timely manner. In addition, the LED’s also have a stable brightness throughout. After the completion of the cube, the Team had solid animations running on the cube with all the LEDs capable of precise control of brightness diffusion.

Portability

Portability played a large part in the successful creation of this project. The accelerometer mode required that the cube be picked up in order to properly display the animations of the mode. Due to this, the dimensions of the cube as well as the weight of the entire system were reasonably small enough for any user to safely pick it up. The cube itself was covered in an acrylic casing in order for the user to be able to hold the cube comfortably and safely to properly display this mode of the animations. The inner workings of the cube (the controller board as well as any other external component besides the cube) were also properly secured underneath the acrylic casing in order to not obstruct the view of the LEDs.

Safety

The eight by eight by eight single-color LED cube was protected by an acrylic casing, which was built by the team. This casing served a dual purpose of not only protecting the case and the user from any direct contact to the LEDs but also it went hand in hand with the portability aspect of the project. Having an acrylic casing on the LEDs allows users to pick up the cube without directly coming in contact with the LEDs and possibly shorting any of the LEDs or layers of LEDs. This way, the users are able to test out the accelerometer mode of the cube, which allows the cube to act like it is full with water and portray animations of water moving around in it.

RGB LED Cube

The RGB LED Cube was built and presented as the final design. The cube followed the same characteristics as the single-color diffused cube. The main difference was the use of more LED sink divers in order to control more LEDs within a package, and the design required more complex software programming. The animations were also subject to heavy change from the single-color diffused cube, since a wide range of colors were considered. Following are the different characteristics of the design.

Maintainability

The lifetime of a RGB LED was more than twenty five thousand hours. The duration of this project was around three months, which is not even close to said lifetime. Therefore, the team did not expect for any of the LEDs to burn up. After all the code was uploaded to memory, there was no maintenance required. After the design is complete, it will stay for years without any fail in the system or any special attention.

Testability

The different components were tested separately using the Arduino UNO board, and a series of test cases were performed to ensure a proper working device. First, the accelerometer was tested with four LED's in each cardinal direction, where the corresponding LED would light up depending on accelerometer orientation. Then, the volume unit meter was tested with different input signals and seven LEDs connect to the digital ports of the Arduino. After finishing the first half of the class, a three by three by three RGBLED cube prototype was done. In the final design, the software was gradually added until a final product was achieved.

Performance

The circuit had to run at a fast rate to ensure the persistence of vision principle was carried out. The LED drivers provided the same brightness for every color and every RGB LED. The final design changed modes based on input from a switch mount on the circuit board, and was accessible from the outside after the casing is mounted. Within each mode, the display was also changed accordingly and was done so depending on by the predetermined time in the programming.

Portability

The final design is eight inches long, eight inches wide, and ten inches tall, and achieved a total weight of less than five pounds. The dimensions stated are the maximum dimensions possible. However, even the maximum dimensions made the design very portable and may be manipulated by any person.

Safety

The final casing of the design was made out of impact-resistant and shatter-resistant clear polycarbonate sheet. The circuit board will be secured by screws attached to the casing. The LEDs were securely attached to the circuit board and also to the casing. Overall, the final product is impact resistant and safe to use.

Animations

As the functional specifications describe, the single-color LED cube as well as the RGB LED cube design include three different modes. The switch on animations is done by sending a pulse voltage to one of the microprocessor pins. The specifications for the RGB cube modes are as follows:

Main Mode

The Main mode includes different animations. For instance, one of the animations turns on multiple LED on the top layer and the LEDs on the lower layers light up acting as raindrops falling. In this mode, only blue LEDs are used to enhance the illusion of water drops. The other animations are used the different colors in a variety of different ways. Additionally, letter patterns are displayed on the front faces and will move back to front.

VU meter

The VU meter reacts to ambient sound, and displays according animations in three different patterns. In the pattern employed, only one layer lights up all the time, and the LEDs in the remaining layers light up and down as the sound wave increases or decreases; essentially, they follow the patterns shown on the dot diagram.

Accelerometer

As far as the accelerometer implementation goes, the objective is to simulate the behavior of a liquid in a container. Therefore, only blue lights will be displayed in order to effectively mimic the illusion of water movement. When no input is read form the accelerometer, the cube will be in rest position, meaning that only the four bottom LED layers will be lit up with no animation whatsoever. However, once the accelerometer senses movement in any direction, the LEDs will react accordingly in a way such that the lit up LEDs represent water in a container.

Cube Construction

The RGB LED cube was built in a different fashion as the single-color diffused LED cube. However, the team followed the same bending method discussed in section 6.3. First, eight LEDs were put in a line in such a way that the three anodes are aligned with the next three anodes; then, the eight LEDs were soldered together, forming a column, and the process was repeated until a total of sixty four columns were achieved. Afterwards, the resulting eight columns were placed in a two-dimensional array of eight by eight dimensions, and the cathodes of each LED were soldered to form an eight by eight LED layer. The process was repeated until eight LED layers were achieved. Finally, the completed cube structure was placed on a wooden surface, where the remaining circuits were housed. More information about the cube construction can be found in sections 6.3 and 6.4 of this document. The printed circuit board was done using EAGLE PCB design software, and printed by DigiKey Corporation. All the integrated circuits for the design are in PDIP package size, all the resistances were half a watt. The team selected all components in PDIP size for the project, given that this package size provides a much convenient soldering process; therefore, all soldering will be performed by the team.

6.0 Project Prototype Construction &Coding

With all the project software and hardware details laid out in the two previous sections of this document, the next important aspect to take into consideration is how the group will handle the implementation of software and hardware regarding a project prototype.

As it is the case with many development cycle methodologies, it is of utmost importance to expose the project in question to some type of testing environment by making use of one or more preliminary designs. As with all other steps of the development process, this step in particular requires a great deal of trial and error and, most importantly, patience, as it directly reflects on the quality and functionality of the final product. It is in this phase where the group will be able to construct the body and circuits of the cube and work on the implementation of the software to make it work, all the while subjecting it to many tests under different environments. This way, it will be possible to extract and understand any faults or misbehaviors of the prototype, and consequently find a solution to such issues. The final results of all tests employed on the prototypes will allow the group to create a final product that is fully functional and free of any technical issues.

Within the confines of this section, many aspects of the prototype construction phase are discussed. The prototypes taken in consideration are the following:

- Purchasable Light Emitting Diode Cube Kit
- 3x3x3 LED Cube Prototype
- 8x8x8 LED Cube Prototype
- 8x8x8 RGB LED Cube Prototype

The group plans to make use of several cube prototypes. With the implementation of each prototype, the group will hopefully be able to see the architecture first hand and understand the benefits and flaws of each method and concept behind it. In essence, the knowledge from the previous prototype can be effectively applied to the next, therefore allowing for maximum efficacy in our final design.

It is within the tentative schedule of the group to have the first project prototype to be conducted during the spring semester of senior design. The next iteration of prototyping will begin sometime between the end of spring term and the start of summer term. As time allows, the group has also discussed the possibility of implementing an additional prototype design, which will hopefully feature additional features and functionality that will add to the quality of the final product.

The next subsections will explain in detail the reasoning behind the choice for each prototype, as well as studied decisions debating whether to utilize said prototype to the group's benefit.

6.1 LED Cube Kits

One of the first proposed prototypes that came to the group's knowledge during the research phase of our project was a LED Cube Kit. This option proves to be very practical, as acquiring a kit for prototyping would be much more convenient, as opposed to getting each particular component separately. Also, another advantage to this solution would be the timely acquisition of materials, due to the fact that all components will be shipped as a kit, all at once. However, there are some limitations involved in the selection of this option, such as the restriction of using the components provided by each kit.

The many advantages and disadvantages of this solution are described in more detail in this subsection, as well as the decision of the group whether to utilize such solution to the group's benefit.

6.1.1 8x8x8 LED Cube Kit by Jameco Electronics

While brainstorming and thinking of many different ideas, the group took into consideration the possibility of obtaining a LED cube kit by Jameco electronics. This Kit includes all materials and components needed to set-up and construct a light emitting diode cube of dimensions 8x8x8. As seen from the picture below, Figure 6.1, the lights of the cube are blue and the dimensions are shown. A green cube kit was also available for purchase at the site.

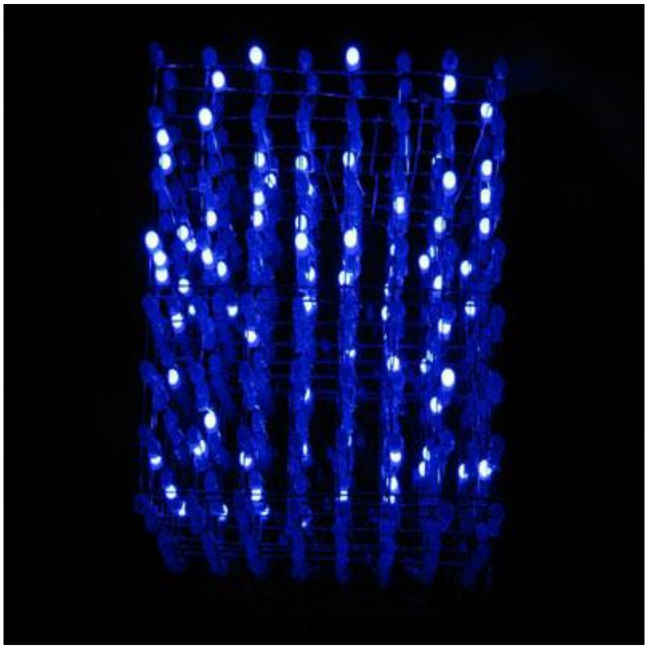


Figure 6.1: 8x8x8 LED Cube Kit by Jameco Electronics. Printed with Permission from Jameco Electronics [6]

The cube itself utilizes code based off the Instructables.com website, which is included in the Research section of this document. Also, this particular kit employs an Arduino UNO development board in order to power the cube and act as a programmer. One interesting fact that was found in this kit was that the Arduino Uno development board was already included, which was not the case for most other kits.

If the group was to purchase a LED cube Kit to employ in our first prototype, a few factors had to be considered. First was the cost of the kit. Since this was only to be a prototype, the group did not want to spend more than 50$ on it. However, this particular cube kit was in the price range of $139.95, not including shipping.

Also, the group had to take in consideration the types of components that were included in the kit. Many of the available kits that were found online did not include the light emitting diodes, making it a necessity to acquire these from another source. This was not the case, however, with the kit provided by Jameco electronics. The downside to including a kit would be the limitations that it would imply to the group, because the group is constrained by the components that are included in the kit. If the group decides to opt for other components or a different implementation of the cube, this would defeat the purpose of obtaining a kit in the first place.

The final factor to consider would be the size of the cube. This particular kit would be a perfect choice, due to the fact that these same dimensions will be utilized the final design of the project, therefore the group could effectively reuse the body of the cube and move it over to the final circuit design that would be plugged in to the printed circuit board. This is a definite advantage to choosing this kit.

6.1.2 3x3x3 LED Cube Arduino Shield by Maker Shed

Another kit that captured the group's attention was the 3x3x3 LED Cube Arduino Shield by Maker Shed. Like the Jameco 8x8x8 LED Cube Kit, this particular kit provides the user with all necessary components in order to construct a LED cube of smaller dimensions. One of the reasons for this particular kit to be very convenient is the fact that it includes a printed circuit board that acts as a shield; therefore, it can be easily mounted on top of an Arduino development board, as can be seen by figure 6.2 below. However, it is important to note that the development board must be purchased separately.

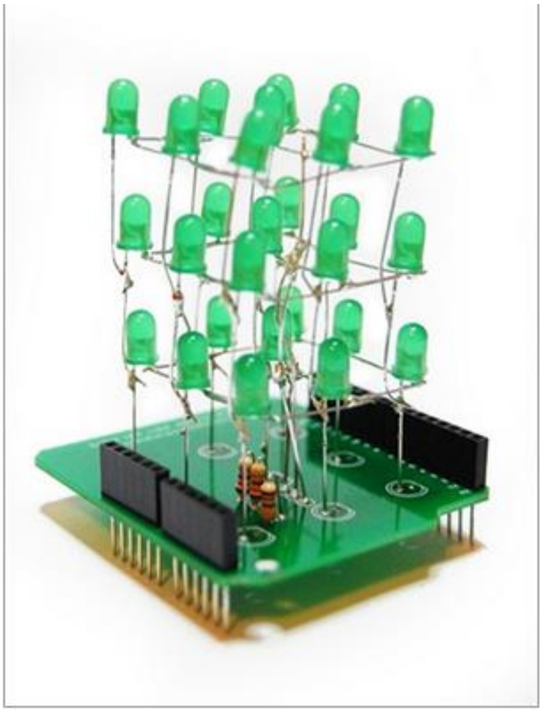


Figure 6.2: 3x3x3 LED Cube Arduino Shield by Maker Shed. Printed with Permission from Maker Shed [7]

The contents of this kit include:

- 1 – PCB
- 27 – LED's
- 2 – 6 Pins for Arduino
- 2 – 8 Pins for Arduino
- 3 – Resistors
- 1 – Foam Guide
- 1 – Small Wire
- 1 – Medium Wire
- 1 – Large Wire

It is important to note that some of these components were already available to the group, which may influence in the decision to opt for this kit solution.

This cube kit, available at a very convenient price of $15.00 not including shipping, seemed like a very strong option for the group's first prototype. However, upon further inspection, the group took into consideration the fact that this cube would not be mounted in a stand-alone surface like the Jameco kit would. Also, the way that this particular cube would be assembled is by making use of the included foam guide, which differs from the group's intention of constructing a wooden soldering jig. Because the group wants its first prototype to model very closely and influence heavily on the final design, that would require setting up the cube in a stand-alone surface, therefore allowing us to implement and test additional features such as the accelerometer and vu meter.

Prototype Decision

Given the above points, the group has decided to disregard the acquisition of an LED cube kit. While the Jameco kit solution proves to be an outstanding asset to the group, as it heavily resembles the final vision of the project, it unfortunately goes over the available budget estimation. The group is much better off gathering each material individually and saving a lot of money for the same quality product.

In the other hand, the Maker Shed solution is very conveniently priced and would only imply a small investment. However, this kit would go out of track as far as the aim of the group goes.

Therefore, it has been decided that our first prototype will be constructed with materials gathered individually to better understand and learn the inner workings of the project.

6.2 3x3x3 Cube Prototypes

With the possibility of the acquisition of a LED cube kit out of the question, the group's next candidate was the construction of a 3x3x3 light emitting diode cube prototype. The group thought that by going for a smaller dimension cube as its first prototype would be the best and most efficient way to learn the way it works, because that would imply that the time spent on this prototype would be smaller, as well as the amount of components used in creating it. These points prove to be most beneficial, due to the time constraints of the overall course.

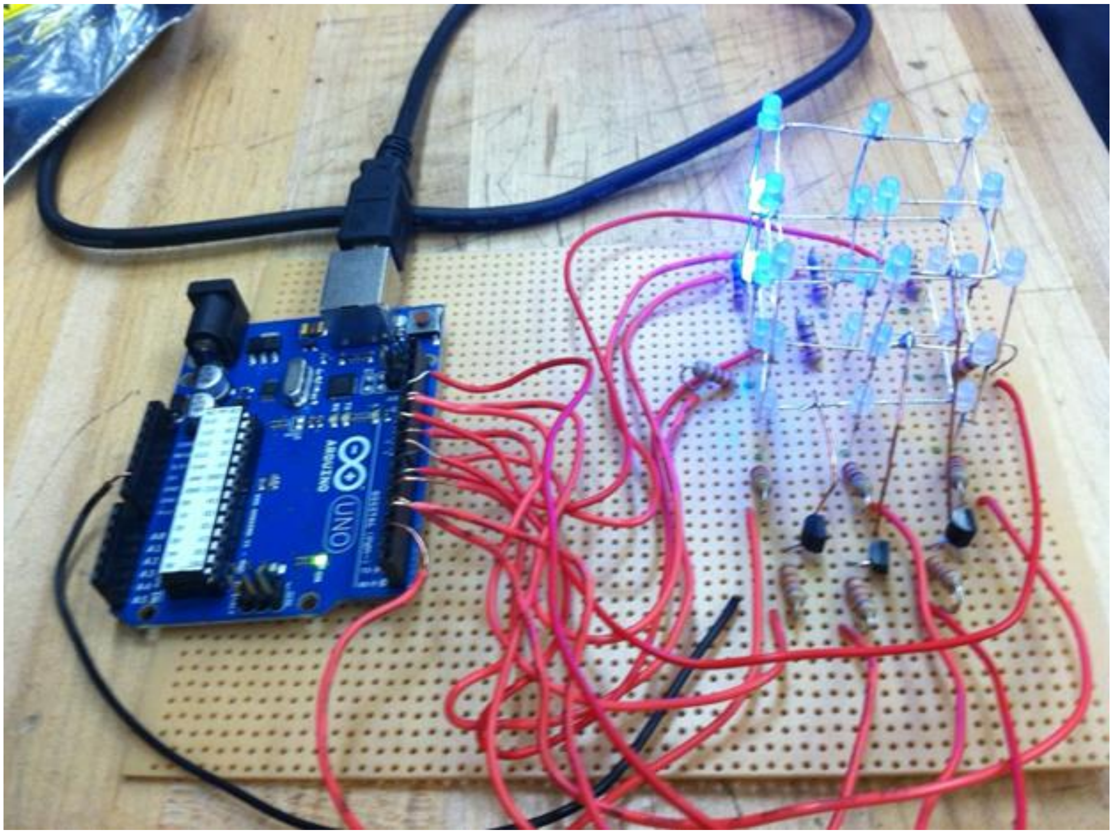


Figure 6.3: 3x3x3 LED Cube Prototype.

The following section will describe in detail the plan of work that will be employed in order to create this small-scale prototype, as well as the final decision regarding the period of construction and deadline specification.

Prototype Objective

The group hopes that, with the implementation of this prototype, a general understanding behind the implementation of the cube is achieved. This implies that all aspects of the prototype are fully functional. That is, the body of the cube is assembled and soldered properly, all light emitting diodes are lighting up correctly, all remaining circuits are constructed in the right way, and the computer-Arduino and Arduino-cube communications are behaving properly.

Besides reaching for the main animation functionality, the group also expects to implement all the interactive aspects of the cube within this first small scale prototype. That is, implementing accelerometer functionality as well as VU meter functionality. This will allow for a smoother transition into the latter prototypes and final design.

Prototype High-Level Design

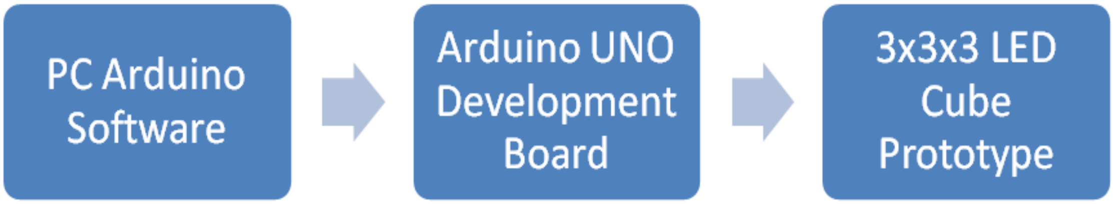


Figure 6.4: Block diagram architecture of 3x3x3 LED Cube Prototype.

PC Arduino Software: The software and drivers provided by Arduino will allow a personal computer to communicate with the Arduino development board. The personal computer that will be used for testing purposes will be required to have this software installed beforehand.

Arduino UNO Development Boards: The compiled data that is provided by the software in the previous step is sent to the Arduino development board, more specifically, to the Atmega microcontroller that is attached to the board.

3x3x3 LED Cube Prototype: With the data already in the integrated circuit, all that is left is to send this data to the individual LEDs in the cube. The cube is a simple yet interesting structure that could be described as an arrangement of LEDs in columns and layers. Each layer is constructed by soldering together the cathode legs of the LEDs, while each layer is then set on top of each other so that each anode leg is connected. This setup will allow for the most optimal distribution of current to each layer, while making sure that the whole cube structure is not damaged in case one of the LEDs malfunctions.

Prototype Construction

The group has decided to start with the construction of this first prototype sometime during the Spring term of the 2013 School Year. All assembly and testing will be conducted in the Senior Design Lab at the University of Central Florida, by all members of the group.

Once the body of the cube is completed and all remaining circuits are constructed, the group plans to test the software in the same environment. When tested in the Senior Design Lab, the group will be able to observe and understand the complex functions of the software implementation and compare it with the hardware behavior of the cube. Following is the plan of work regarding the hardware construction of the first prototype.

The group has gathered beforehand all materials needed for the construction of this prototype. These materials include:

- 9x 220 Ohms resistors
- 3x 22K Ohm resistors
- 1x Perfboard
- 3x NPN Transistors (Model number 2N3904)
- Piece of square wood
- 27x LEDs
- Wire

The group will also make use of many other tools and materials that were already available, such as a soldering iron, solder, power drill, and millimeters.

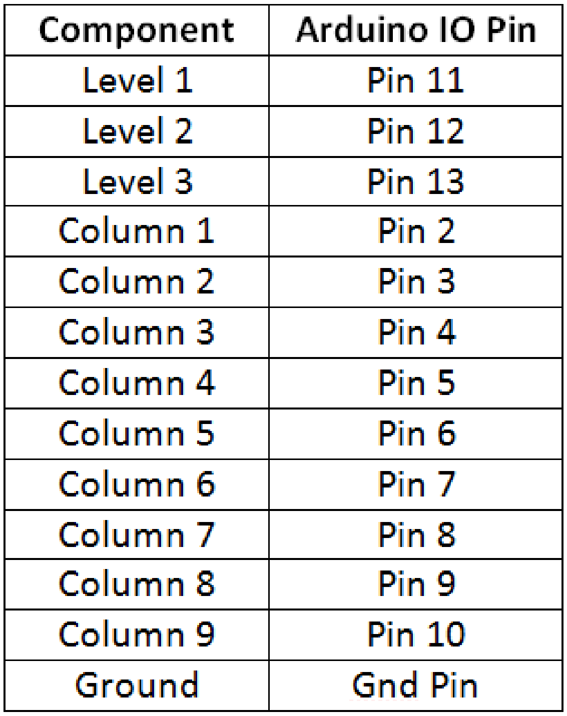
To start with the construction, the group will make use of a drill to make holes in the square piece of wood, in order to make a soldering jig. This jig will hold the LEDs in place and will allow the soldering process to be much easier. The hole spacing will match with the length of the cathode leg of a light emitting diode, because the legs of the LED will comprise the body structure of the cube. Being exact with this spacing isn't critical, there may be a little bit of overhead to make sure that the leg reaches the next LED. The holes will have a tight fit around the LED since any movement of the LED head while soldering is highly undesirable. This method has been employed by many other projects and has shown great results in regards to the visual appeal of the final product.

Once the wooden jig is completed, each LED will need to have its cathode bent in a 90 degree angle, The group will make sure that the cathode is bent in the same direction for each and every one of the LEDs.

After all the light emitting diodes are set up, the group will begin soldering 9 LEDs to make a layer, and concurrently each layer to make the complete cube. For each layer, the LEDs will need to be facing in the same counter-clockwise direction. After all three layers are constructed in this manner; the group will place one layer on top of the next and solder the corresponding anode legs together. At the end of this process, the complete body of the LED cube will be finished.

The rest of the circuit will be assembled in a much similar fashion as the cube project described in section 4.1.3 (Pocket LED Cube by Instructables). Each of the nine LED columns will connect to a digital pin on the Arduino development board through a 220 Ohm current limiting resistor. Similarly, each of the three levels will be connected to ground via three 2N3904 NPN transistors in series with 22K Ohm resistors, when activated by other three IO pins in the Arduino board.

With the circuit now complete, the next step will be to move on to the software. The following table shows the pin connections corresponding to each column and layer of the LED cube:



Component	Arduino IO Pin
Level 1	Pin 11
Level 2	Pin 12
Level 3	Pin 13
Column 1	Pin 2
Column 2	Pin 3
Column 3	Pin 4
Column 4	Pin 5
Column 5	Pin 6
Column 6	Pin 7
Column 7	Pin 8
Column 8	Pin 9
Column 9	Pin 10
Ground	Gnd Pin

Table 6.1: Arduino board pin connections of 3x3x3 LED Cube Prototype.

Software Implementation

During the research phase performed by the group, many existing code examples to control an LED cube were found all around the internet. However, the open source code provided by the Pocket LED cube tutorial from Instructables.com proved to be of great interest to the group. What was most persuasive about this code implementation is that the software basically mirrors the architecture of the hardware. This implies that each LED location is addressed by column and layer number, instead of the traditional raw port data or <x,y,z> format found in other implementations. This allows the group to easily visualize the code and make a strong connection between the coding and the structure of the cube.

In order to get the code running on the cube, it is very important to first download the library provided by this site, and place it in the libraries folder of the arduino directory. Then the code can be uploaded with ease by making use of the Arduino software.

The group decided to use this open source library mainly because it will be very beneficial to first understand how the code works, in order to then start playing around with different implementations and displaying many different animations.

Prototype Decision

It was decided that it would be in the group's best interest to construct the aforementioned prototype, given specifications and functional requirements described above. A simple, small scale cube that employs the same structure and design as our final project will allow us to better understand all aspects that go into the construction of such a device.

This prototype was successfully built during the Spring Semester of the 2013 School Year, and it was able to display the different animations that were included in the open-source code. However, the group did not have enough time to implement the additional functionality such as the accelerometer and VU meter mode. Therefore, the group has decided to continue with this approach after the Spring semester is over, provided that there is enough time for other prototypes. Otherwise, this particular prototype will be left as is, in order to spend more time in the more involved prototypes, which are discussed below.

6.3 8x8x8 LED Cube Prototype

Once the group is in possession of a fully functional, small-scale LED cube prototype, the next logical step in the prototyping process will be to go full scale with a 8x8x8 LED Cube.

Now that the general procedure and the inner workings regarding the construction of the cube have been studied, this knowledge can be successfully used for the construction of this prototype. It is important to note that the structure of this prototype will heavily resemble the final design of the project. Therefore, the body of this cube can be practically reused in the final design, effectively saving the group time and money that would have been spent gathering components and constructing a whole different piece.

The following section will describe in detail the plan of work that will be employed in order to create this full size 8x8x8 LED Cube prototype, as well as the final decision regarding the period of construction and deadline specification.

Prototype Objective

With the hardware construction and software implementation of this particular prototype, the group truly believes that it will accomplish all technical requirement specifications that are described in section 2.4 of this document.

The group hopes that, with the implementation of this prototype and the previous small-scale prototype, a more complete and thorough understanding regarding the construction of the cube is reached. Similar to the 3x3x3 LED Cube prototype, the objectives that the group hopes to accomplish with this prototype involves that all aspects and functionalities of the cube are fully implemented. By extension, this means that the complete structure of the cube should be fully functional, all LEDs should be lighting up correctly, the cube responds to the different input of the three-way switch, correctly outputs all three different modes that are to be implemented, and correctly responds to different input by the user, such as movement and sound.

Prototype High-Level Design

The following diagram, Figure 6.4, displays architecture of the 8x8x8 LED cube prototype:

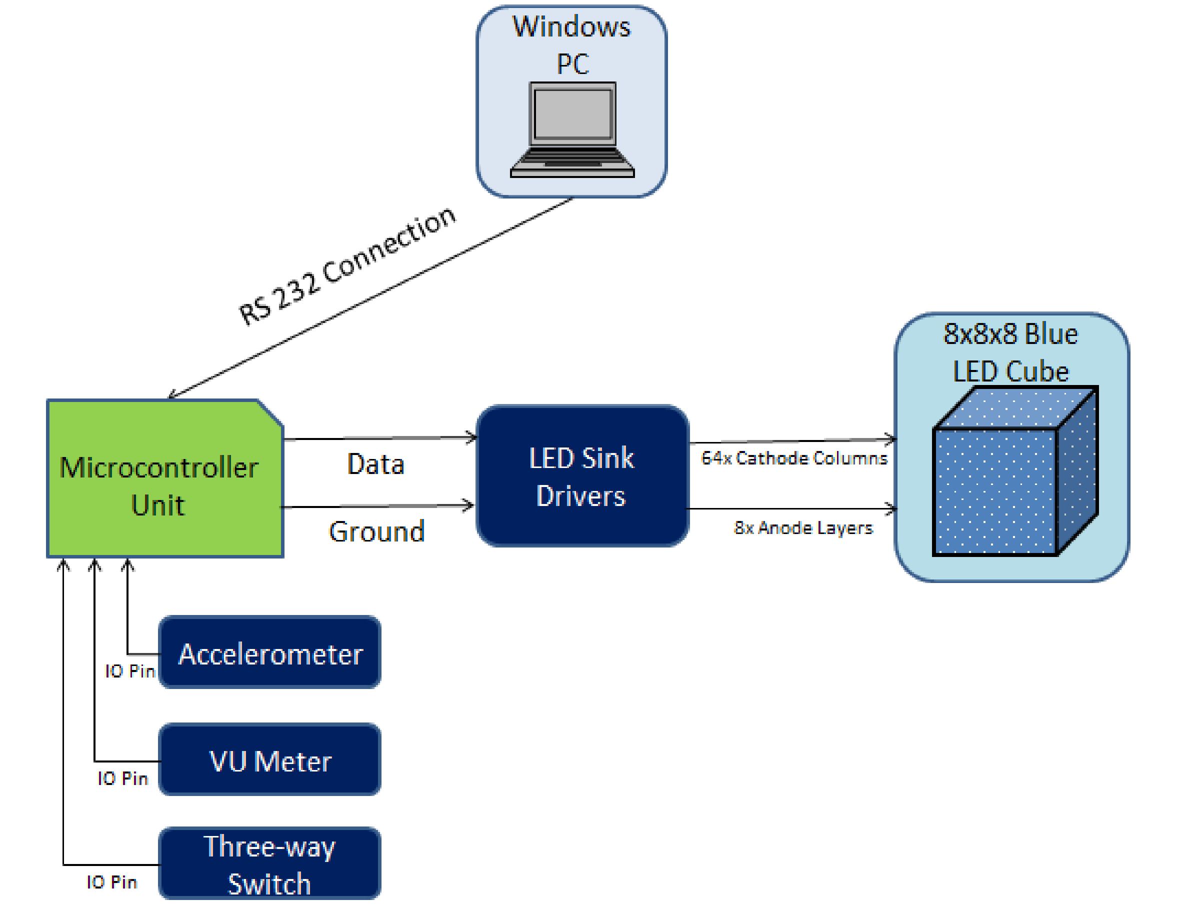


Figure 6.4: Block diagram architecture of 8x8x8 LED Cube Prototype.

Windows PC: A personal computer running the Windows operating system was be provided by one of the members of the group. This computer will be used to implement and compile the animation and functionality software that is to be displayed in the prototype. The personal computer will communicate to the microcontroller unit by making use of a RS232 cable connection.

Microcontroller Unit: The compiled data that is provided by the software in the previous step is sent to the microcontroller unit, in this case the Atmega1284 microcontroller, and along with inputs received from the accelerometer, VU meter, and the state of the switch, it will send corresponding output data to the five STP16CP LED drivers that will be used to light up the LEDs.

LED Sink Drivers: When the microcontroller unit has received sixty-four bytes of information, it passes the data onto the STP16CP LED drivers, one layer at a time. These drivers will be responsible for lighting up the corresponding light emitting diodes in the LED cube.

8x8x8 Blue LED Cube Prototype: One layer is powered using the corresponding transistor and the respective LEDs are turned on using the output given by the STP16CP LED drivers.

Prototype Construction

It was within the plans of the group to perform the construction of this particular prototype during the break days in between the Spring and Summer terms of the 2013 School Year. Similar to the first prototype, it has been decided that the best environment for construction of the prototype will be the Senior Design Lab at UCF, and the process will be performed by all members of the group.

After the body of the cube is constructed, the group will implement the remaining circuit of the cube in a similar fashion. After all phases of construction are complete, it is within the plans of the group to test the structure within the same environment. Following is the work flow that will be conducted in regards to the prototype construction.

All of the components and materials used for the construction of this prototype have been gathered beforehand, and mostly the same tools that were used for the construction of the 3x3x3 prototype will effectively be used for this procedure. The list of materials and components for this prototype are:

- 1x ATmega1284 Microcontroller
- 5x TLC5940 LED sink drivers
- 8x PNP Transistors
- 9x 8-pin headers
- 512x Single color (blue) LEDs
- 1x LED for power status
- 1x DE-ACCM accelerometer

- 1x LM324NE3 Amplifier and Frequency Divider
- 1x RS232 connection
- 1x single pole triple throw switch
- 5x resistors (value to be determined)
- 9x capacitors (value to be determined)
- 1x plank of wood

It is important to note that some of these components are subject to change, be it the model number or the amount of pieces. Also, additional components for the power supply portion of the circuit will be specified. Other than that, the remaining components are the usual materials used in circuit design, such as ribbon wire, solder, multimeters, and such.

Similarly to the 3x3x3 prototype previously constructed, the group will begin by creating the soldering jig. Having the LED anode leg length as the spacing between the holes, the group will use the power drill to make holes that are big enough to place the LED head. Once an eight-by-eight hole grid is created, sixty-four LEDs will be placed in each hole and will be ready to solder together. An important factor to take in consideration is the manner in which each light emitting diode will be bent. The following figure, Figure 6.5, depicts the way the LEDs will be bent for this project:

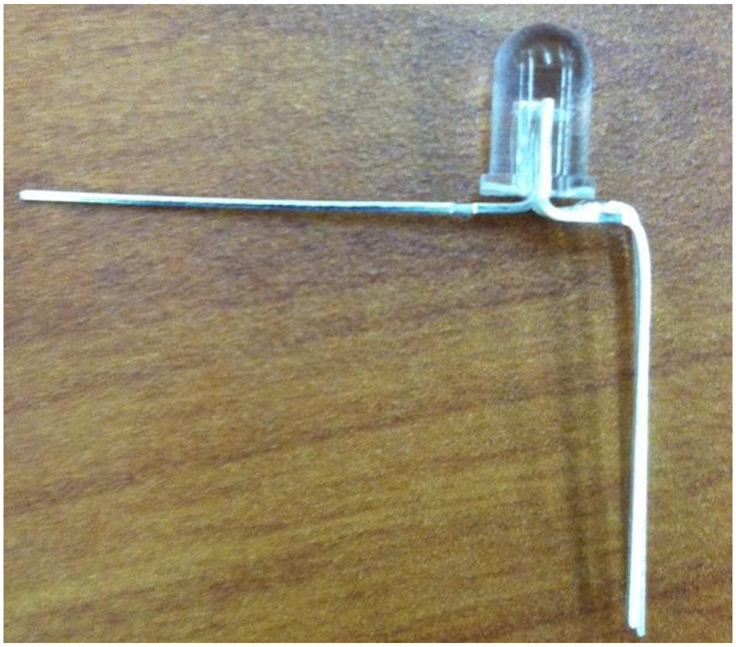


Figure 6.5: Illustration of LED lead bending.

With previous knowledge of the 3x3x3 prototype, the group noticed that the column construction was a bit problematic, due to the fact that the LED leads are placed directly below the LED head, in the center to be exact. This implies that, in order to solder together LED cathodes to make a columns, each cathode lead will have to be bent at the end in a forty five degree angle so that it can make contact with the next cathode. This poses an issue because the final result will not be a straight structure, and will not be as visually pleasing as the group desires. The way that the group plans to handle this issue, in this prototype and the next, is by making use of the bending logic illustrated in Figure 6.5.If we bend each LED cathode in such a way that it ends up outside the perimeter of the LED head, this will allow for a much easier time in the soldering process. In this manner, when all the cathodes are connected together, the resulting column will be a close to perfectly straight line, and thus enhancing the visual appeal of the final cube.

After one layer of sixty four light emitting diodes is constructed, this step will be repeated seven more times, resulting in eight complete 8x8 layers. All that is left to do at this point in time is to place each 8x8 layer on top of the next, and solder each cathode column together. At the end of this tedious process and with a great deal of patience the complete body structure of the LED cube will be finished.

The rest of the circuit, however, will be more involved to deal with, due to the fact that this is a more complete prototype when compared to the previous 3x3x3 prototype, and the Arduino board is out of the question for use. For a more detailed description of the circuit architecture, refer to section 4.0, where a circuit schematic of the 8x8x8 LED cube is laid out. Basically, the thinking behind the group's design is to use four LED drivers to control the sixty four LEDs in each layer, and the fifth driver along with eight transistors to control the layers. It is important to remember that these LED driver chips are designed to sink currents, not to supply them. Therefore, the group has to make sure that, opposite to the previous prototype, the cathodes (negative legs) of the LEDs are connected to form a column, while the anodes (positive legs) of the LEDs are connected to form a layer and supplied via the transistors. Also, the accelerometer and VU meter will be connected to their respective pins in the microcontroller unit.

With the circuit complete and all the components included, all the software and coding requirements will be discussed in the following section.

Software Implementation

As far as software implementation goes, there were many different takes coming from various 8x8x8 LED cube projects that were found online. Some of the more interesting ones used some form of coordinate logic in order to control each individual light emitting diode.

The way that the group handled the animation logic was by making use of arrays and LED buffers. Although this method implied a more computationally demanding process, it allowed at the same time for a more visual representation of the animations as they were written. For instance, if the light emitting diode at position (x,y,z) = (2,0,8) is to be turned on, the corresponding value in the cube array position [2,0,8] was set to 1. As far as animations go, the group took time to figure out and research many algorithms that ended up in eye-pleasing animations. Some of these came from mathematical functions, such as the sine wave function, while others were 3D letter projections. The number of animations implemented was relative to the amount of time that was available after the construction of the cube was done and the remaining circuit was fully functional.

The main classes that were implemented during the coding section were Main, Animations, Accelerometer, and VU meter. More information about the implementation of these classes can be found in the Software Design section of this document (section 4.2).

Prototype Decision

Given that this prototype is crucial to the learning and understanding of the final product that the group has set out to create, it is the group's best interest to go ahead and construct this piece. Within the expected milestones of the group, the implementation of this device will begin right after the spring semester of the 2013 school year is done.

Depending on the outcome of this prototype, the group took into consideration the many advantages and disadvantages of the construction process that were followed, and put to good use the many concepts and experiences learned from this practice in order to achieve a complete project. The group also reused several of the components that were included in this prototype for the final device, given that everything occurs as expected.

6.4 8x8x8 RGB LED Cube Prototype

The group expected to have many difficulties and roadblocks when constructing the single color LED cube. There existed many factors in the construction process that would have subjected the project to some type of malfunction or unexpected behavior. However, in case that every aspect of the previous prototype went as planned and there was still a considerable period of time left, the features of the project would be expanded to include a variety of colors for the LED animations. That is, the group switched from the single color LED cube to a Red-Green-Blue version of the cube. By expanding the functional requirements of the project at hand, the group accomplished a broader learning experience that will be put to good use when needed.

The following section will describe in detail the workflow that was followed to create this RGB light emitting diode cube prototype.

Prototype Objective

With the completion of this prototype and the previous single-color prototype, the group expected to expand on the technical requirement specifications to obtain a very detailed and thorough understanding of the differences that imply creating a multi-colored LED cube, as opposed to making a single-color cube.

As with the previous prototypes, the group expected to implement the three different modes of the cube: animation mode, accelerometer mode, and VU meter mode. Additionally, these three modes had an increase in difficulty, due to the fact that in addition to these modes, a variety of colors were needed to be displayed in each mode. This implied an increase in components used, and therefore a bigger room for error and unexpected behavior.

Prototype High-Level Design

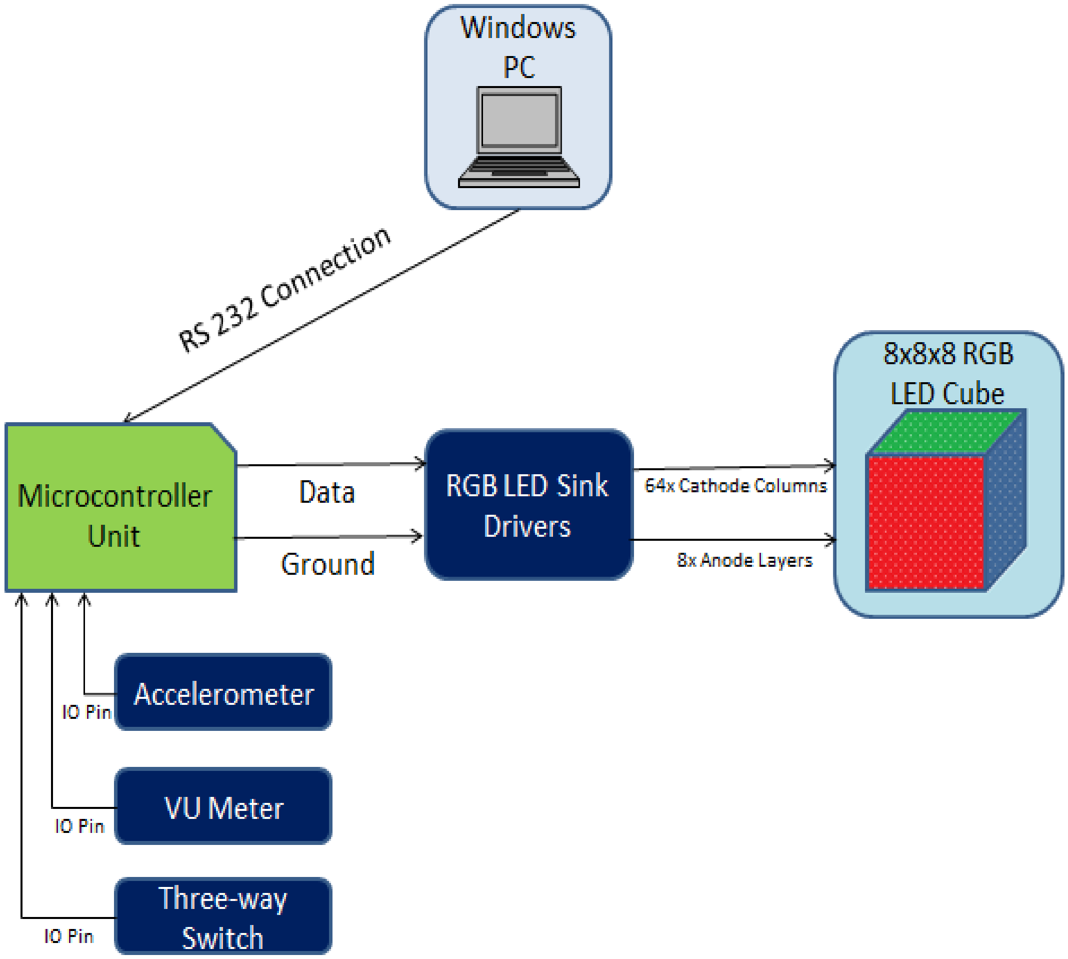


Figure 6.5: Block diagram architecture of RGB 8x8x8 LED Cube Prototype.

Windows PC: Similar to the single color light emitting diode cube, the group made use of a personal computer running the Windows operating system in order to implement and compile the data that was sent to the cube. However, the coding process was more involved, due to the fact that a whole range of colors was taken into consideration.

Microcontroller Unit: The compiled data was sent to the AtMega1284 microcontroller unit, in a much similar fashion as with the single color LED cube. Also, depending on the input by the three-way switch, the AtMega1284 integrated circuit was responsible for deciding which device to listen to. Be it the accelerometer, the VU meter, or the plain animation sequences, the integrated circuit then sent the corresponding output data to the twelve TLC5940 LED sink drivers.

RGB LED Sink Drivers: Given that the TLC5940 LED sink drivers were also capable of handling multi-color light emitting diodes, these same drivers were used for the red-green-blue prototype as well. The only difference was that seven more of these integrated circuits were required, since the red-green-blue light emitting diodes had two more leads and therefore required more IO lines. When the microcontroller unit had received data, it passed it onto each LED driver, one layer at a time. The drivers were responsible for lighting up LEDs in the corresponding color.

8x8x8 RGB LED Cube Prototype: Just like the single color light emitting diode cube, the persistence of vision concept was used to light up one layer at a time, and the corresponding LED lit up in the color specified by the TLC5940 LED drivers.

Prototype Construction

Given that every aspect of the single-color light emitting diode cube is working, and enough time is still available to the group, the construction of this RGB LED cube will start sometime in the early to middle weeks of the Summer term of the 2013 School Year. As with the previous prototypes, this particular cube’s construction will be performed at the Senior Design Lab in the University of Central Florida, and all members of the group will be present to design and test the prototype. Following is the plan of work that will be employed for the implementation of the RGB LED cube.

The first step of the procedure will be to create the body of the cube. Since the soldering jig for the 8x8x8 single-color cube has already been created, the group can make use of this piece to solder together the red-green-blue light emitting diodes. Once again, sixty four LEDs will be placed in each hole for the soldering process. The leads of each RGB light emitting diode will be bent in the same manner as the previous prototype, given that this bent allows for maximum efficiency when soldering layers together which has been a tedious process in the past.

Once all eight layers are put together, one layer was placed on top of the next, and every cathode that was on top of another was soldered together, forming a column. After doing this process seven more times, the complete body structure of the cube was finished. Another method for constructing the body of the cube was also in consideration by the team, and was discussed briefly in section 5.3. This method involved constructing the cube by columns first, instead of layers. First, eight LEDs were put in a line in such a way that the three anodes of an LED were aligned with the next three anodes; then, the eight LEDs were soldered together, forming a column, and the process was repeated until a total of sixty four columns were achieved. Afterwards, the resulting eight columns were placed in a two-dimensional array of eight by eight dimensions, and the cathodes of each LED were soldered to form an eight by eight LED layer. The process was repeated until eight LED layers were achieved. The only disadvantage that comes with this method was the fact that a different soldering jig will need to be constructed.

The rest of the circuit was very similar to the single color LED cube prototype, the only difference being an increase in the number of TLC5940 LED drivers used. A definite advantage about these LED sink drivers was that they could be run in cascade to control as many light emitting diodes as needed; in other words, the LED drivers were perfect for this prototype and project. Also, the accelerometer and VU meter that were used in the previous prototype can be connected to their respective pins in the Atmega1284 microcontroller.

Once the whole electric circuit was complete, it was time to move on to the software implementation of the cube.

Software Implementation

Similar to the single color LED prototype, there were a couple of different implementations of RGB LED cube software found online. Based on the research already performed by the team, it was found that the best logic to go about when implementing LED cube functions is by the use of a coordinate system.

Just like the single color LED prototype, the group handled the animation logic by making use of arrays and LED buffers. As far as animation implementation goes, the team made use of already available open-source animations and modified them, as well as came up with new, intuitive, and eye-pleasing animations.

The main classes created during the coding section were Main, Animations, Accelerometer, and VU meter. More information about the implementation of these classes can be found in the Software Design section of this document.

Prototype Decision

The construction and implementation of this RGB light emitting diode cube prototype was a highly desirable phase in the project's development cycle.

Depending on the outcome of previous prototyping phases, and given that it is still not too far into the Summer term, the team took on this challenge to expand on the features of the project.

This concludes the workflow that will be used during the prototyping phase of the project. The team expects to take from this phase all the required knowledge and experience to put together a final design that meets all the functional requirements specified, and more importantly, a final design that will awe and captivate its target audience.

6.5 Lessons to Learn

During the construction and testing of the three by three prototype constructed during the Spring Term, some lessons were definitely learned, and they were of great help when starting with the construction of the latter prototypes. For better understanding, the selection of the parts were discussed first. In Orlando, there is a lack of a full size specialized electronic store. Therefore, one of the group members drove down to Miami and was able to find all the parts that were needed. However, the group still had to use half a watt resistors. Therefore, we learn that all the electronic parts, no matter what simple is, will have to be order online. The first step into construction was to do holes in a two by four piece of wood to form and hold the square shape of the cube for later place the LED inside the hole and solder together forming the layers. To do the hole we use a small four volt, quarter inches screwdriver. We start at the center mark and every time the drill vibrate making the hole in different position and some of the holes were bigger than the LED due to the imprecision of the drill. The lesson learn in this part is to use a more powerful drill, or even better a press drill, with a new drill bit that will not misplace holes. The second step into the construction was to put the LED's together. The soldering of the LED have to be by two people, one person to solder and another person to hold the LED's on places until cool down and form a strong connection. Soldering the different layer was not as easy as show in the instructional video because is hard to keep the different layer at the same high. We have to find a better way to have the same high throughout the final eight by eight by eight projects. The size of the final prototype looks smaller than the one shown in the video, but it has better portability.

After finishing the inserting of all the parts, all the connections were made to the Arduino. There were too many wires and we had to double check for loose connections. For the final design, ribbons were used to connect between the LED's and the LED's drivers. The prototype helped us to understand in full all the functions of the electronics and the codes provided in the in the instruction video. After the construction, the prototype was left in the locker and the next week we tried to do more testing but the upper layer did not work. After close examination, we discover some cathode was loose and one anode for another layer was loose too. With that fail we learned the importance of having very strong soldering and also to add more wire, that way the cathode will have a ring type path. With that type of connection, if a fail occurs, the current will have another path to go and the final circuit will not be compromised. Also, we saw the necessity of having better support for all the layers.

7.0 Project Prototype Testing

Testing is a very integral part of any project since at this time; problems can be discovered and addressed. Since the Multifunctional LED cube was constructed from many components, it was important to test each and every part to ensure minimum number of issues. Testing was divided into two phases, each phase made up of multiple tests. Phase one involved testing the components in the 3x3x3 LED cube. A modified Sandwich Integration testing methodology was used to make sure that components work in unison. The idea was that each component would be tested individually at the lowest level. As different components passed testing, they were then tested together and were slowly assimilated in the system as a whole. This allowed us to begin testing at an early stage. The objective of Phase 1 was to gain understanding of the LED cubes to facilitate the construction for the final prototype. Phase 2 included similar integration testing in with the 8x8x8 LED cube. Due to the nature of this project, hardware and software was tested simultaneously.

7.1 Phase 1 3x3x3 Cube Testing

The 3x3x3 LED cube was mainly used to understand how the components and coding worked when properly implemented. Information gathered from these tests facilitated the processes of constructing the final prototype. The breadboard layout of the cube constantly changed to accommodate the test taking place. Testing time length varied due to layout formatting and software writing. Majority of this testing was done with the Arduino development board and then repeated with the ATmega1284 to ensure consistency.

Test #1

Setup:	The Cube pins will be directly connected to the pins on the development board as seen in **Table 6.1**in the 3x3x3 Prototype section of this document
Description:	LED cube will be powered off, then turned on by initiating the Arduino development board
Initial Condition:	The cube will be inactive since the board is inactive
Steps to run test case:	1) Run the development board from the user's PC 2) Wait for cube to light up
Expected Behavior:	Upon uploading and running the code to the development board, the LED cube will turn on. It will then commence to display patterns.

Test #2

Setup:	The Cube pins will be connected via the TLC5940 LED sink driver which will be connected to the development board
Description:	LED cube will be powered off, then turned on by initiating the Arduino development board
Initial Condition:	The cube will be inactive since the board is inactive
Steps to run test case:	1) Run the development board from the user’s PC 2) Wait for cube to light up
Expected Behavior:	Upon uploading and running the code to the development board, the LED cube will turn on. It will then commence to display patterns.

Test #3

Setup:	Accelerometer will be setup on a separate bread board and connected to the ADC pins
Description:	LED cube once powered on, will wait for input from the accelerometer to display pattern
Initial Condition:	The cube will be inactive since the board is inactive
Steps to run test case:	1) Run the development board from the user’s PC 2) Wait for cube to light up 3) User picks up and translates or rotates the accelerometer board
Expected Behavior:	Upon uploading and running the code to the development board, the LED cube will turn on. It will then wait for user interaction. The cube will change display based on accelerometer movement.

Test #4

Setup:	Only the VU Meter will be setup on the breadboard and connected to the development board
Description:	LED cube will be turned on. User will produce noise by any means to make the cube react
Initial Condition:	The cube will not be plugged in, therefore not receiving any power and remain off
Steps to run test case:	1) Run the development board from the user’s PC 2) The cube will light up for a second 3) User may produce noise directed at the cube
Expected Behavior:	Upon uploading and running the code to the development board, the LED cube will turn on. The cube will light up for a second, and then wait for user input. LEDs will light up producing a pattern according to the noise input from the user

Test #5

Setup:	The VU meter and accelerometer will share the same breadboard, The SPTT switch will be used to communicate with the development board
Description:	LED cube will be turned on. User will switch mode of the cube. Choosing between accelerometer mode, display mode or VU meter mode. Cube will respond by switching to requested mode
Initial Condition:	The cube will be in one of three modes, performing usual routine and standing by for user input
Steps to run test case:	1) Run the development board from the user's PC 2) The cube will light up and behave in the mode it is currently switched to 3) User may proceed to work in this mode 4) User may then change the mode through moving the switch 5) The cube will light up and behave in the mode it is currently switched to 6) Repeat steps 2 through 5
Expected Behavior:	Upon uploading and running the code to the development board, the LED cube will turn on. The cube will perform a predetermined mode until it receives input by the user to switch to a different mode. At this point the cube will begin to behave in one of three manners according to the switch.

7.2 Phase 2 8x8x8 Cube Testing

The setup for these tests was consistent on a large breadboard with the TLC5940 LED sink drivers in use. When testing the accelerometer, the cube itself was not lifted, the board containing said device was. Manipulation of the actual cube was not done until the PCBs were installed on the final prototype.

Test #1

Description:	LED cube will be powered off, then turned on by the user
Initial Condition:	The cube will not be plugged in, therefore not receiving any power and remain off
Steps to run test case:	1) Plug in the power supply to wall outlet 2) Wait for cube to light up
Expected Behavior:	Upon plugging in the power supply, the LED cube will turn on. It will then commence to display patterns.

Test #2

Description:	LED cube will be turned on. The user will toggle the switch to change modes
Initial Condition:	The cube will not be plugged in, therefore not receiving any power and remain off
Steps to run test case:	1) Plug in the power supply to wall outlet 2) Wait for cube to light up 3) Cube will begin to display patterns 4) User activates switch 5) Cube will light up the top face and maintain it 6) User activates switch 7) Cube will return to displaying patterns
Expected Behavior:	Upon plugging in the power supply, the LED cube will turn on. It will then commence to display patterns. The user will be able to switch the mode in which the cube will stop displaying patterns and light up the top face. If the user chooses to flip the switch again. It will return to its previous mode of displaying patterns

Test #3

Description:	LED cube will be turned on. User will handle the cube and it will respond to input from the accelerometer
Initial Condition:	The cube will not be plugged in, therefore not receiving any power and remain off
Steps to run test case:	8) Plug in the power supply to wall outlet 9) Wait for cube to light up 10)User picks up the cube 11)User translates or rotates the cube
Expected Behavior:	Upon plugging in the power supply, the LED cube will turn on. No patterns will be displayed, waiting for user movement. LEDs will illuminate according to the movement input of the accelerometer by the user.

Test #4

Description:	LED cube will be turned on. User will produce noise by any means
Initial Condition:	The cube will not be plugged in, therefore not receiving any power and remain off
Steps to run test case:	4) Plug in the power supply to wall outlet 5) The cube will light up for a second 6) User may produce noise directed at the cube
Expected Behavior:	Upon plugging in the power supply, the LED cube will turn on. The cube will light up for a second, and then wait for user input. LEDs will light up producing a pattern according to the noise input from the user

Test #5

Description:	LED cube will be turned on. User will switch mode of the cube. Choosing between accelerometer mode or display mode. Cube will respond by switching to requested mode
Initial Condition:	The cube will be in one of two modes, performing the usual routine and standing by for user input
Steps to run test case:	1) Plug in the power supply to wall outlet 2) The cube will light up and behave in the mode it is currently switched to 3) User may proceed to work in this mode 4) User may then change mode through moving the switch
Expected Behavior:	Upon plugging in the power supply, the LED cube will turn on. The cube will perform a predetermined mode until it receives input by User to switch to a different mode. At this point the cube will begin to behave in one of two manners according to the switch.

Test #6

Description:	LED cube will be turned on. User will switch mode of the cube. Choosing between accelerometer mode, display mode or VU meter mode. Cube will respond by switching to requested mode
Initial Condition:	The cube will be in one of three modes, performing usual routine and standing by for user input
Steps to run test case:	7) Plug in the power supply to wall outlet 8) The cube will light up and behave in the mode it is currently switched to 9) User may proceed to work in this mode 10) User may then change the mode through moving the switch 11) The cube will light up and behave in the mode it is currently switched to 12) Repeat steps 2 through 5
Expected Behavior:	Upon plugging in the power supply, the LED cube will turn on. The cube will perform a predetermined mode until it receives input by the user to switch to a different mode. At this point the cube will begin to behave in one of three manners according to the switch.

Test #7

Description:	The user will power on the LED cube. The proper functionality of all colors of each single-color LED in the cube will be tested as well as the PWM brightness control. The PWM brightness control should smoothly blend the dimming between the brightest and dimmest setting of the LED.
Initial Condition:	The cube will be on the off state. There will be no power being supplied to the cube because the cube will not be plugged into the power outlet
Steps to run test case:	1) Plug in the power supply to wall outlet 2) Wait for cube to display main mode animations 3) Verify proper display of all the colors in the LEDs. 4) Verify proper dimming of LEDs from brightest to dimmest
Expected Behavior:	Upon plugging in the power supply, the LED cube will turn on. The cube will perform a predetermined mode until it receives input by the user to switch to a different mode. At this point the cube will begin to behave in one of three manners according to the switch.

7.3 Phase 3 8x8x8 RGB LED cube testing

The setup for these tests was very similar to that of the single color LED cube. They took place over a few breadboards connecting all chips (accelerometer, vu meter, LED drivers, etc) to the LED cube. When performing tests of the accelerometer, the LED cube was not lifted; instead the board containing the accelerometer was lifted. The animations were still displayed on the cube though it was not moving.

Test #1

Description:	The user will power on the LED cube. The proper functionality of all colors of each RGB LED in the cube will be tested by running through the main mode of the animations (continuous patterns).
Initial Condition:	The cube will be on the off state. There will be no power being supplied to the cube because the cube will not be plugged into the power outlet
Steps to run test case:	1) Plug in the power supply to wall outlet 2) Wait for cube to display main mode animations 3) Verify proper display of all the colors in the RGB LEDs.
Expected behavior:	The cube should properly display all of the colors in each of the RGB LEDs.

Test #2

Description:	The user will power on the LED cube. The proper functionality of all colors of each RGB LED in the cube will be tested as well as the PWM brightness control. The PWM brightness control should smoothly blend the transition between colors in each RGB LED as well as the dimming between the brightest and dimmest setting of the LED.
Initial Condition:	The cube will be on the off state. There will be no power being supplied to the cube because the cube will not be plugged into the power outlet
Steps to run test case:	1) Plug in the power supply to wall outlet 2) Wait for cube to display main mode animations 3) Verify proper display of all the colors in the RGB LEDs. 4) Verify proper dimming of LEDs from brightest to dimmest 5) Verify proper blending between colors of the LEDs
Expected behavior:	The cube should properly display all of the colors in each of the RGB LEDs. The cube should dim the colors

	of the LED smoothly (Without skips in light or sudden jumps in brightness) as well as blend the transitions of colors.

Test #3

Description:	The user will power on/off the LED cube
Initial Condition:	The cube will be on the off state. There will be no power being supplied to the cube because the cube will not be plugged into the power outlet
Steps to run test case:	1) Plug in the power supply to wall outlet 2) Wait for cube to display main mode animations
Expected behavior:	After the cube is plugged into the wall outlet, depending on the mode that the cube is on, the LEDs will display the according the animation patterns.

Test #4

Description:	After powering on the LED cube, the user will be trigger all available modes through the three state switch
Initial Condition:	The cube will be on the off state. There will be no power being supplied to the cube because the cube will not be plugged into the power outlet
Steps to run test case:	1) Plug in the power supply to wall outlet 2) Wait for cube to display main mode animations 3) Cube will display the current mode animations 4) User switches state through switch 5) Cube recognizes state change and illuminates the top layer 6) User switches state once again through switch 7) Cube will begin to display current state animations
Expected behavior:	After the cube is plugged into the wall outlet, depending on the mode that the cube is on, the LEDs will display the according the animation patterns. Once the user switches modes the cube will recognize the state change and begin the newly selected set of animations

Test #5

Description:	The user will power on/off the LED cube. The user will interact with the cube in the manner that will interact with the accelerometer on the controller board. The animations of the cube should follow the input from the accelerometer
Initial Condition:	The cube will be on the off state. There will be no power being supplied to the cube because the cube will not be plugged into the power outlet

Steps to run test case:	1) Plug in the power supply to wall outlet 2) Wait for cube to display main mode animations 3) The user will switch to accelerometer mode 4) The user will pick up the cube and rotate and lean the cube in the direction of their selection
Expected behavior:	The cube animations should follow that of the input received from the accelerometer based on the actions performed by the user on the cube.

Test #6

Description:	The user will power on/off the LED cube. The user will interact with the cube in the manner that will produce that the vu meter on the control board will perceive
Initial Condition:	The cube will be on the off state. There will be no power being supplied to the cube because the cube will not be plugged into the power outlet
Steps to run test case:	1) Plug in the power supply to wall outlet 2) Wait for cube to display main mode animations 3) The user will switch to accelerometer mode 4) The user will produce the noise of their choosing
Expected behavior:	The cube animations under the vu mode should follow the input perceived by the vu meter based on the noises produced by the user. If no noise there will be a static illumination patter on the cube. If there is noise, the cube should illuminate dynamically based on the noise.

Test #7

Description:	The cube will be powered on. The user will interact with the cube in the manner that will interact with the accelerometer on the controller board. The animations of the cube should follow the input from the accelerometer
Initial Condition:	The cube will be on one of the two modes part of this particular test. The cube will be awaiting the next input from the user
Steps to run test case:	1) Plug in the power supply to wall outlet 2) Wait for cube to display main mode animations 3) The user will switch to accelerometer mode 4) The user will decide to stay in this mode and interact or switch to another mode via the switch
Expected behavior:	The cube animations should follow the current active mode and should await the input of the user. If user decides to switch to other mode, the cube shall display that modes animations

Test #8

Description:	The LED cube will be powered on. User will switch between the vu mode and the accelerometer mode. The cube will respond to the switching between these two modes
Initial Condition:	The cube will be on the on state. It will be in any of the three modes awaiting user input to change between modes
Steps to run test case:	1) Plug in the power supply to wall outlet 2) Wait for cube to display main mode animations 3) The user will switch to accelerometer mode 4) The user will pick up the cube and rotate and lean the cube in the direction of their selection or select to switch to vu meter mode 5) The user will switch to vu meter mode 6) The cube will change to the current set mode of animations
Expected behavior:	The cube animations should follow that of the input received from the accelerometer based on the actions performed by the user on the cube.

7.4 Testing Environment

The testing and construction of the project took place in the Senior Design Lab located in Engineering I room 456, at the University of Central Florida. It was mainly handled by the team members during this time to ensure that fault is not placed on bystanders if an incident were to occur. Room temperature of operation will be between 60 to 76 degrees Fahrenheit. At times it was required to run test in a dark to low-light room due to the nature of this project. A traditional 120 volts 20 amps power outlet was used to supply power to the device. Since portability was a major specification, the testing locations changed from time to time, to ensure ease of transport. Other locations simulated the equivalence of the design lab conditions.

8.0 Administrative Content

Naturally, a monetary cost will be associated with every carefully designed electrical project. With the high-level and detailed design of project fully laid out, the group will describe in its entirety the administrative factors that will come into play regarding the implementation of our project. Within the confines of this section, information about group management, distribution of work, milestones and financial information will be detailed.

8.1 Management

To certify the proper completion of this project in a timely manner, the team decided to make an outline of the final senior design documentation, which contained all of the sections needed for a finalized paper. Each team member was allowed to work on any particular section they desired and informed the other team members of their selections. As research and work was done, this outline was updated with the names of each person working on their selected section, as well as how many pages were created. The finalized date to have all of the sections and pages of the document complete was Sunday, April 21st.

The team decided to choose their sections in a chronological order. This meant that research was the first section to be covered. In the research section, team members chose from various LED cube projects done in the past as well as the components to go along with them. After researching all the major components for this project as well as similar previous project, the team went ahead and begun planning and designing the inner workings of the project (Hardware Architecture, Software Architecture, etc.) From this phase, the team decided to build a small scale LED cube to become acquainted with the circuitry and programming of an LED cube. Parts were purchased between the team members to begin assembling of the small scale three by three by three LED Cube. After the team created the small scale LED cube, they began the designing of the LED cubes.

The team decided to choose their sections in a chronological order. This meant that research was the first section to be covered. In the research section, team members chose from various LED cube projects done in the past as well as the components to go along with them. After researching all the major components for this project as well as similar previous project, the team went ahead and begun planning and designing the inner workings of the project (Hardware Architecture, Software Architecture, etc.) From this phase, the team decided to build a small scale LED cube to become acquainted with the circuitry and programming of an LED cube. Parts were purchased between the team members to begin assembling of the small scale three by three by three LED Cube. After the team created the small scale LED cube, they began the designing of the LED cubes.

The team decided to design the version of the LED cube that was single-color as a backup plan. As for the RGB LED cube, we decided to design the cube as well, with the constraint that if the team is not successful at building it, the single color LED cube will be successfully completed and available for display. As the design was finalized, the team documented the different component designs developed as well as test environments for both the small scale LED cube, the eight by eight by eight single-color LED cube as well as the RGB LED cube.

In terms of roles, the Team was divided into the general position taken by each team member. Team member, Roberto, assumed the position of leader/manager as he proposed the idea and worked on most of the more detailed work in design and research. Both team members, Julio and Eury provided specific sections (Accelerometer Mode and VU mode respectively) of the cube design as well as provided extensive research on all components and previous components as well as design prototyping and project prototype testing. Team member, Luis provided research on certain components as well as a few similar projects and provided extra support to sections that required filling.

8.2 Milestone Discussion

The team decided to treat the milestones a bit differently. Instead of keeping a record of the time taken in a particular section, we decided to break off each general topic by date. This meant that every general section had a deadline that needed to be met, after this deadline was met, the next general section was started and finished on its respective deadline. If the section was not finished, however, the team decided to finish that section in the free time slots of the next sections time frame. Overall, though not the most ideal set of milestones, the goals were met in the end and the team was able to create a complete set of documentations for the LED cube project.

The following outline shows the division of the general sections with their respective starting dates as well as the team member who was assigned to said section. The separate sections will have a check mark highlighted in the same color highlighting the team member's name who completed it:

8.2.1 Milestones for Senior Design 1

The following table describes the milestones for senior design 1. The tables were split by week of completion. Each week has a list of goals to complete and the name of the person who completed such goal.

2-19-13	**One Week to describe Project Requirements**	
	Executive Summary	Julio
	Project Description	Roberto
	Introduction	Roberto
	Project Motivation	Roberto
	Goals and Objectives	Roberto
	Project Requirements & Specifications	Roberto
	Physical Specifications	Roberto
	Environmental Specifications	Roberto
	Hardware Specifications	Roberto
	Software Specifications	Roberto
2-26-13	**Two and one-half weeks of component research** **Research Related to Project Definitions**	
	Existing Similar Projects and Products	Roberto
	8x8x8 Instructables LED Cube	Roberto
	University of Portland Senior Design Project	Roberto
	3x3x3 Pocket LED cube	Luis
	3D RGB LED Cube	Eury
	How Not To Engineer – Single-Color LED Cube	Roberto
	How Not To Engineer – RGB LED Cube	Luis
	Relevant Technologies and Component Research	Roberto

	Microprocessor	Roberto
	Accelerometer	Julio
	VU Meter	Eury
	Power Supply	Eury
	LED Sink Drivers	Luis
	LP3943 RGB/white/blue 16-LED Driver	Luis
	Latches/Demultiplexers	Julio
	RS232 Communication	Roberto
	Light Emitting Diodes	Roberto
	RGB Light Emitting Diodes	Luis
	Switches	Roberto
	Control Boards	Luis
	Architecture and Related Diagrams	Roberto
	Hardware Architecture	Roberto
	Software Architecture	Roberto
3-15-13	**Three weeks to design and build prototypes**	
	Project Hardware and Software Design Details	Roberto
	Hardware Design	Roberto
	Processing Subsystem	Roberto
	Sensing Subsystem	Roberto
	Display Subsystem	Roberto
	Printed Circuit Board	Roberto
	Software Design	Roberto
	Design Summary of Hardware and Software	Luis
	8x8x8 Single Color LED Cube	Luis
	Design Issues	Luis
	Single Color LED Cub	Luis
	RGB LED Cube	Eury
	Animations	Eury
	Cube Construction	Eury
	Project Prototype Construction and Coding	Julio
	Purchasable Light Emitting Diode Cube Kit	Julio
	3x3x3 Cube Prototype	Julio
	8x8x8 Cube Prototype	Julio
	8x8x8 RGB Cube Prototype	Julio
	Lessons Learned	Julio
4-5-13	**One week to finish prototype testing of all models**	
	Project Prototype Testing	
	Testing for 3x3x3 Cube	
	8x8x8 LED Cube Test Environment	
	Hardware Test Environment	
	Software Test Environment	

	Team Biographies	All
4-12-13	**Since most work was finalized, the team decided to finish administrative content of the paper which included the following**	
	Administrative Content	Luis
	Management	Luis
	Milestone Discussion	Luis
	Budget and Finance Discussion	Luis
	Distribution of Work	Luis
	Conclusion	Luis
4-21-13	**Last day for each team member to have finalized all 30 pages of work assigned.**	
4-22-13 - 4-24-13	**Documentation is revised and corrected by all team members. Once revision is completed, finished document is sent to binding**	
4-25-13	**Completed documentation is submitted to Dr. Richie**	

8.2.2 Milestones of Senior Design 2

4-28-13 – 5-11-13	**Two weeks to complete the 8x8x8 Single-Color LED Cube prototype. At this time, the team will finalize all animations for single-color LEDs as well as finalize the design of the controller board. The board will be purchased the first week of the two-week period. Second week of the two-week period will involve integrating all components to board to have a fully functioning prototype.**	
5-12-13	**One Week to finalize design for the 8x8x8 RGB cube as well as begin the ordering process of the control board through RushPCB if new customer deal still applies. If the team's budget allows for a new updated price, the board will still be ordered from RushPCB. If not, the board will have to be ordered from BatchPCB, which offers the design at a lower price but a higher turn around time. Its crucial to have the design finalized by the end of this week in order to have enough time to deal with a high turn around time from the manufacturer of the control board, assuming the team cant purchase the RushPCB board.**	
5-19-13	**Assuming the control board is ordered from RushPCB, it will**	

- 6-2-13	arrive during this two-week period. In this period, the building of the cube will commence. Specifically, the soldering of all the RGB components will take the majority of the team's time during this week. This two-week period will also contain the purchasing of any materials the team might have missed to purchase as well as modify any designs as the cube is being built. In addition, the first initial presentation of the cubes will presumably be around this time period.
6-9-13	One-week period to implement all the animation software as well as driving software required to have a properly functioning eight by eight by eight RGB cube. During this time, the codes will be tested on the computer before implementing on actual controller processor.
6-16-13 - 7-19-13	Remainder of time before last day to have everything finished (7-19-13) will be dedicated to testing and document writing. During this time, every testing environment developed by the team will be implemented on all the finished prototypes of the cubes (the eight by eight by eight single-color LED cube and the eight by eight by eight RGB cube). The senior design 1 documentation will be updated to present tense and will include all of the corrections to the design of all the cubes as well as new testing environments, updated architectures, lessons learned, updated software code, etc. Alongside to this documentation, the team will be preparing the condensed, technical conference paper based on the IEEE standards. In addition, the team will be preparing their final presentation including a project board, PowerPoint, etc.
7-19-13	Last day to have every finalized prototype of every LED cube to be demonstrated in Senior Design Presentation. Second Set of documentation must be finished by this time as well
7-19-13 - 8-2-13	Presentation Period. If for some reason, any of the planned parts of the above milestone path do not work or need to be revised, this will be the extra time to fix them.

8.3 Budget and Finance Discussion

Fortunately, the construction of an LED cube is not a project that requires an unreasonable amount of funding to complete. The team decided to design and build three different prototypes due to the reusability of components and their low cost. The team members bought the components arbitrarily and, at the end, once there was a total cost, it would be divided amongst all team members and compensation would be distributed.

Most of the components the team needed for the initial building and testing of prototypes were already purchased by some team members. These components, such as the Arduino One Microcontroller, Soldering Irons, Breadboards, Leads, etc., were not included in the total cost distributed amongst the team members. Other components, such as the TLC5940 LED drivers as well as the LP3943 RGB/white/blue 16-LED Light Driver were available as samples on the Texas Instruments website, free of charge including free shipping.

The following tables display all of the components that were purchased by the team for each individual cube.

3x3x3 Single Color LED cube

Part	Price	Quantity	Total Price
22-Gauge Solid Hookup Wire	$2.83	3	$8.49
2N3904 Transistor	$1.49	3	$4.47
220 Ohm Resistor	$0.30	10	$2.98
22k Ohm Resistor	$0.30	5	$1.49
Perforated Bare Phenolic Prototype Board	$4.75	1	$4.75
Ribbon Cable	$3.46	1	$3.46
Dual Row Header	$1.23	1	$1.23
Arduino UNO	$19.99	1	$19.99

8x8x8 Single Color LED cube

Part	Price	Quantity	Total Price
3mm 2-pin Round top Diffused LED	$0.02	1000	$24.20
22-Gauge Solid Hookup Wire	$2.83	3	$8.49
2N3904 Transistor	$1.49	8	$11.92

220 Ohm Resistor	$0.30	10	$2.98
22k Ohm Resistor	$0.30	5	$1.49
Ribbon Cable	$3.46	1	$3.46
Dual Row Header	$1.23	1	$1.23
MAX3232 RS232 to TTL Converter Module	$2.79	1	$2.79
Amico 3 Position Switch	$1.99	1	$1.99
Boards	~$20 - $40	1	~$20 - $40

8x8x8 RGB LED Cube

Part	Price	Quantity	Total Price
3mm 2-pin Round top Diffused LED	$0.02	1000	$24.20
22-Gauge Solid Hookup Wire	$2.83	3	$8.49
2N3904 Transistor	$1.49	8	$11.92
220 Ohm Resistor	$0.30	10	$2.98
22k Ohm Resistor	$0.30	5	$1.49
Ribbon Cable	$3.46	1	$3.46
Dual Row Header	$1.23	1	$1.23
MAX3232 RS232 to TTL Converter Module	$2.79	1	$2.79
Amico 3 Position Switch	$1.99	1	$1.99
Boards	~$20 - $40	1	~$20 - $40
Microcontroller ATmega1284	$7.62	1	Free Sample
MSGEQ7 Equalizer	$1.89	1	$1.89
Electret Microphone	$0.58	1	$0.58

LM324 Operational Amplifier	$0.48	1	Free Sample
7805 Voltage Regulator	$0.53	1	On Stock
10k Potentiometer	$0.27	1	$0.27
Plexiglas	$12.99	1	$12.99
USB Class “A” Connector	$0.66	1	$0.66

Table 8.1: Summary of Expenses for 3x3x3 LED Cube, 8x8x8 Single Color LED Cube and 8x8x8 RGB LED Cube

8.4 Distribution of Work

As mentioned in the Milestone Section, each team member decided to pick the sections they desired and communicated with the other team members which sections they were doing. Though this separation of work was not the most optimal for even distribution of work, in the end, each team member was required to have at least 30 pages of work involving research and design. In terms of physical work, the cubes were built and tested by all members.

The following chart displays a visual display of the approximate percentage of work performed by each member of the team. The total representation in the final documentation, however, is an even split of 30 pages per work mentioned in the previous paragraph.

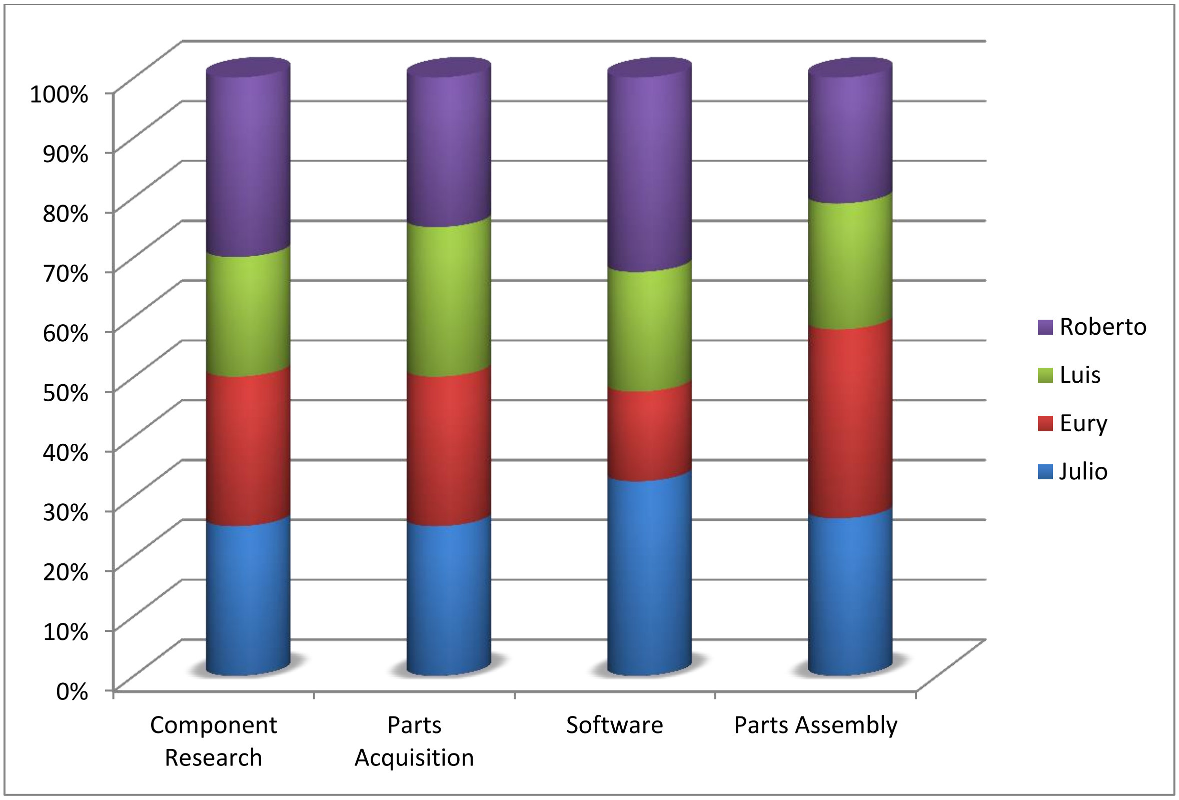


Figure 8.1: Percentage of Work Performed by Each Member of the Team per Section

Though the team was successful at fulfilling the requirements of this design paper, they did come across some difficulties. Due to the lack of a concrete milestone path, It was difficult for the team to properly set deadlines and divide work evenly amongst the team members in an orderly fashion. Another main difficulty was the lack of communication of one of the members of the team, which lead the team to some confusion and to fall behind on the project documentation. In the end, however, the team managed to fulfill the 120 page requirement for this documentation as well as the research, design, and implementation of the eight by eight by eight LED cube as well as the eight by eight by eight single-color LED cube.

8.5 Group Member Backgrounds

Julio Romero

Julio was born October 26 1990, in Portovelo - Ecuador, and moved to Florida in 2005 along with his family in hopes of obtaining a better education and lifestyle. He attended Cypress Creek High School in Orlando Florida, and upon completion, he moved on to Valencia Community College to start his career in General Engineering. He is currently a Senior attending the University of Central Florida, with expected graduation of December 2012 with a Bachelor of Science degree in Computer Engineering. One of Julio's professional experiences include a Software Engineer Internship at the Walt Disney Company, where he was responsible for the development and maintaining of web applications within the company's website. During this Summer, he will be interning with HD Supply, also as a software developer. Julio's main technical strengths include C# and Java programming, as well as Structured Query Language (SQL) knowledge.Upon graduation, Julio hopes to obtain a full-time opportunity with a cutting-edge company, where he can put to good use his technical and analytical skills.

Roberto E. Amaya

Roberto was born September 26 1989, in Brownsville Texas and moved to Florida at a young age where he has been until now. He attended Southeast High school in Bradenton Florida partaking in the International Baccalaureate Program as well as the Visual and Performing Arts Academy. Spending much of his free time playing video games on the computer and home consoles, this sparked an interest in technology. This eventually led him to pursue a degree in engineering. Roberto currently attends the University of Central Florida, seeking a degree in Computer Engineering. Not wanting to focus on neither the software or hardware aspect of a CpE, he has become the "jack of all trades, master of none". He has taken quite an interest in Artificial Intelligence, dedicating majority of his elective classes to the study of. During this Summer semester He will be interning with Stanley Security Solution doing embedded programing for security devices. Upon graduation, Roberto hopes to obtain a fulltime position in an Engineering firm within the country.

Eury Reynoso

Eury Reynoso was born December 15th 1983, in Santo Domingo the capital of Dominican Republic. He spent his childhood in Guaricano, a town on the north side of the city. He graduated from the Instituto Politecnico Cardenal Sancha, a catholic technical high school. During high school he enrolls into the technical program to graduate as an Electrical Technician specialized in installation and maintenance. In 2002 was accepted in the Universidad Autonoma of Santo Domingo in the Electrical Engineering program. While still in the junior year, he moves to the United State to pursuit a better life. After settle in Florida he start to learn the second language at MidFlorida Tech. Then, enroll in Valencia Community College where he graduates as Associate of Arts in General Studies August, 2010. Right after college; he was accept in the University of Central Florida in the electrical engineering program.

He is currently working in engineering department in Marriott Grande Vista as technician II. He passes through hard time while working and attending to school. Eury plants to graduate in August 2013. He wants to work in the power distribution sector. His dream job is working in a company that specializes in building power generator or research for renewable energy.

Luis Ferrer

Luis Ferrer was born January 28th 1991, in Puerto Cabello – Venezuela. After the unfortunate changes the country was going through, his family decided to relocate to Miami, Florida in December 1998 in order to obtain a more free and secure lifestyle. He attended School for Advanced Studies in Miami Florida, and upon graduation in 2009 came to University of Central Florida seeking a bachelor’s degree in computer engineering. In his first semester at UCF he was hired as a computer support technician for College of Engineering and Sciences where he has been working for the past 4 years. After marrying his high school sweetheart in August 2012, planning for internships or jobs have changed for him drastically to include his new family. Luis plans to actively search for a degree specific professional opportunity in the summer semesters where he will have the most amount of free time to devote to it.

9.0 Conclusion

We have successfully completed the research, design, testing, and prototype of the (eight by eight by eight) single-color LED Cube project as well as the final project. The majority of the teams' time was spent on performing extensive research and becoming knowledgeable of some unfamiliar topics regarding our project. Through this extensive research, the team was able to produce a system design and properly select hardware and software that were appropriate based on the teams technical, financial, and educational constraints.

Though there were initial significant difficulties with a particular team member's lack of communication and effort, the team was eventually successful in developing a system design as well as a proper prototype for the proposed LED cube project. Through these difficulties, the team members have learned to manage time properly as well as prioritize the combination of work, school, and family. In the long run, the team members showed the skills necessary to complete the design of an LED cube, the professionalism in properly preparing their assigned sections, and their team unity by completing a cohesive final design document.

In choosing to develop a potential commercial product and applying the team's engineering and technical problem solving skills in order to acquire a solution, the team feels that its formal education in the field of Electrical and Computer Engineering was challenged to the maximum of its ability to apply the concepts and methods learned to solve a real life scenario in physical reality.

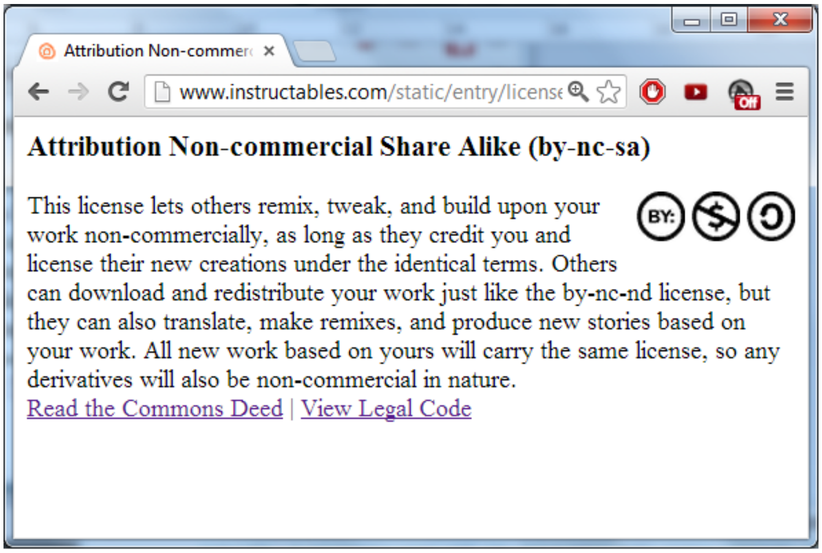
Designing the three different LED cube systems made the team more knowledgeable over certain topics that went beyond the scope covered by formal education. This forced the team to use outside resources and become well versed in methods of researching components, design and implementation for projects at a professional level. Through the process of the design phase of this project, the team became knowledgeable in specific areas pertaining to light emitting diodes, integrated circuitry (LED drivers), software animation using LEDs, integration of microcontrollers as well as physical construction of electrical devices.

Overall, this project really helped the team members gain hands-on skills and experience necessary to go out and design, implement and build engineering projects in the workforce. In addition, these skills as well as the project can be annexed to our resumes, aiding us in having an advantage over other entry level engineers searching for a profession.

Appendix

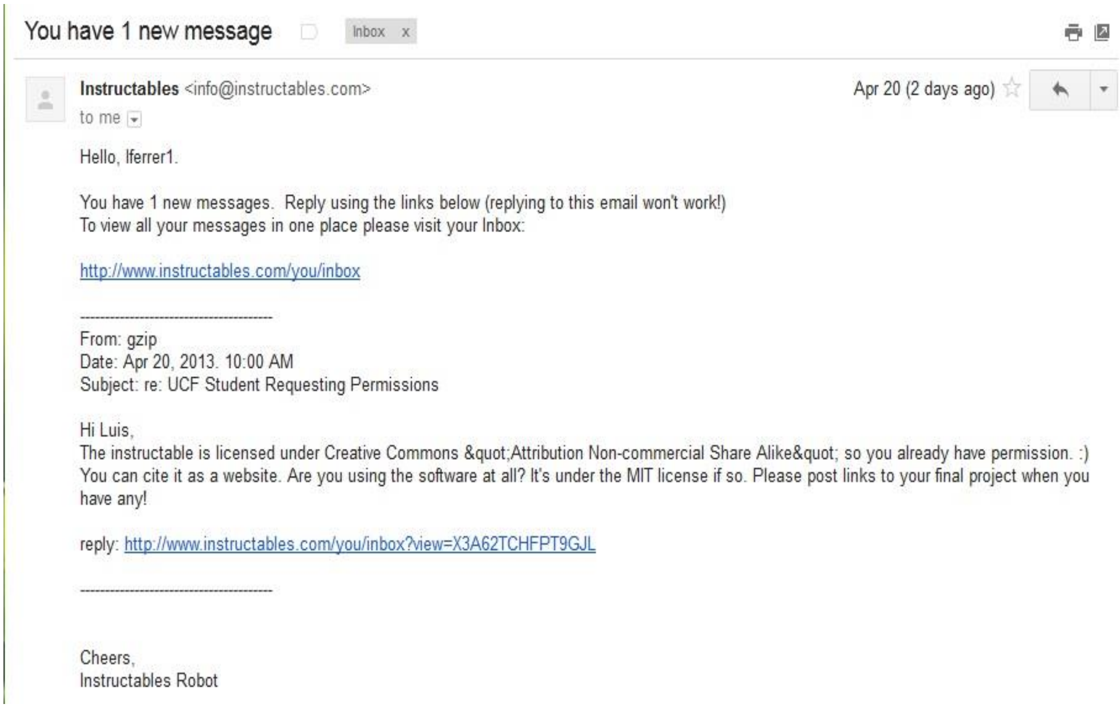
Appendix A: Copyright Permissions

[1] Instructables.com


Attribution Non-commerc ×
www.instructables.com/static/entry/license

Attribution Non-commercial Share Alike (by-nc-sa)

This license lets others remix, tweak, and build upon your work non-commercially, as long as they credit you and license their new creations under the identical terms. Others can download and redistribute your work just like the by-nc-nd license, but they can also translate, make remixes, and produce new stories based on your work. All new work based on yours will carry the same license, so any derivatives will also be non-commercial in nature.
Read the Commons Deed | View Legal Code



Instructables <info@instructables.com> Apr 20 (2 days ago)
to me

Hello, lferrer1.

You have 1 new messages. Reply using the links below (replying to this email won't work!)
To view all your messages in one place please visit your Inbox:

http://www.instructables.com/you/inbox

From: gzip
Date: Apr 20, 2013. 10:00 AM
Subject: re: UCF Student Requesting Permissions

Hi Luis,
The instructable is licensed under Creative Commons &quot;Attribution Non-commercial Share Alike&quot; so you already have permission. :)
You can cite it as a website. Are you using the software at all? It's under the MIT license if so. Please post links to your final project when you have any!

reply: http://www.instructables.com/you/inbox?view=X3A62TCHFPT9GJL

Cheers,
Instructables Robot

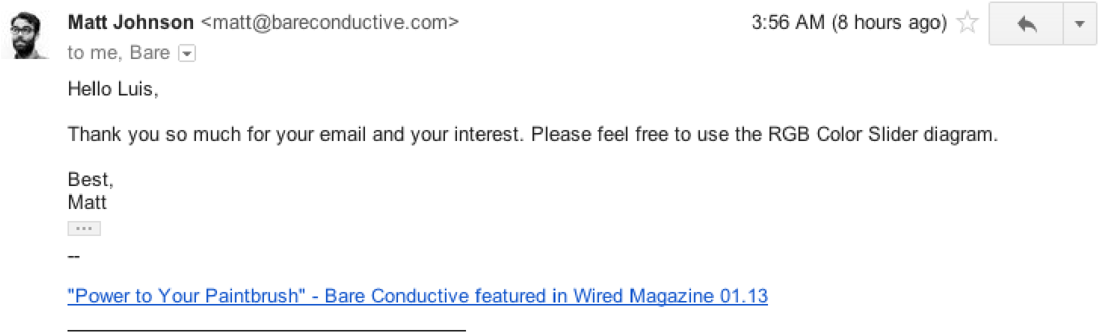
[2] HNTE.com

Permission pending from HNTE.com

[3] MSI

Permission pending from MSI.com

[4] Bare Conductive



Matt Johnson <matt@bareconductive.com>

to me, Bare

3:56 AM (8 hours ago)

Hello Luis,

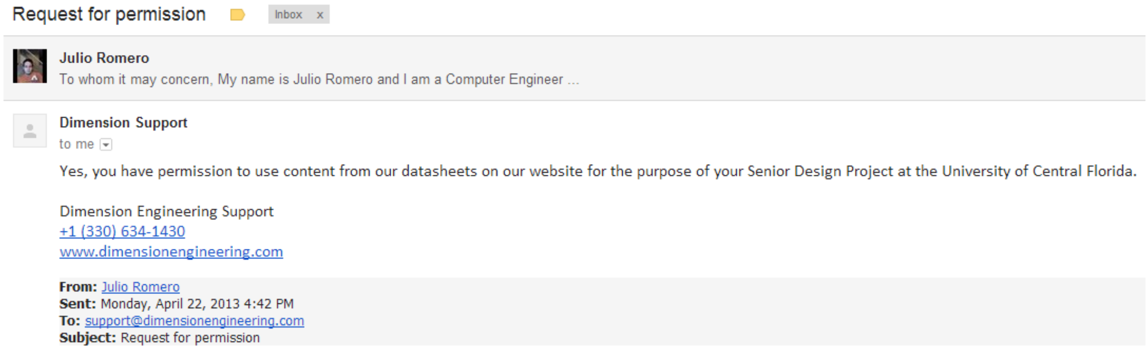
Thank you so much for your email and your interest. Please feel free to use the RGB Color Slider diagram.

Best,
Matt

--

"Power to Your Paintbrush" - Bare Conductive featured in Wired Magazine 01.13

[5] Dimension Engineering



Request for permission

Julio Romero

To whom it may concern, My name is Julio Romero and I am a Computer Engineer ...

Dimension Support

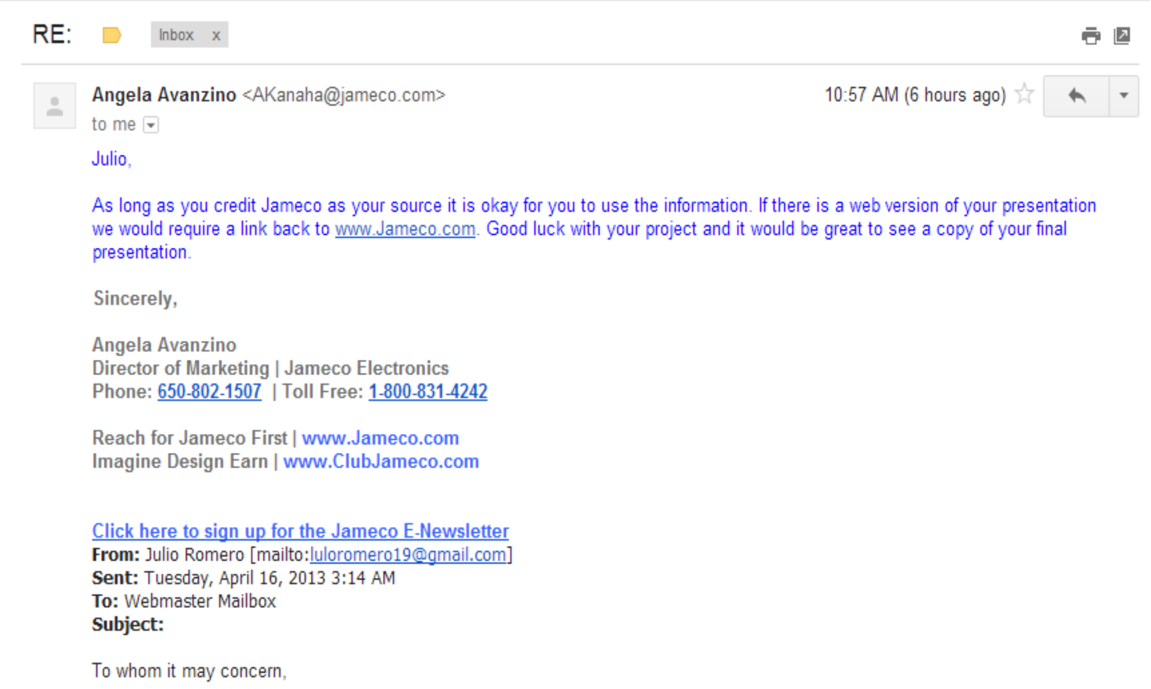
to me

Yes, you have permission to use content from our datasheets on our website for the purpose of your Senior Design Project at the University of Central Florida.

Dimension Engineering Support
+1 (330) 634-1430
www.dimensionengineering.com

From: Julio Romero
Sent: Monday, April 22, 2013 4:42 PM
To: support@dimensionengineering.com
Subject: Request for permission

[6] Jameco Electronics



RE:

Angela Avanzino <AKanaha@jameco.com> 10:57 AM (6 hours ago)

to me

Julio,

As long as you credit Jameco as your source it is okay for you to use the information. If there is a web version of your presentation we would require a link back to www.Jameco.com. Good luck with your project and it would be great to see a copy of your final presentation.

Sincerely,

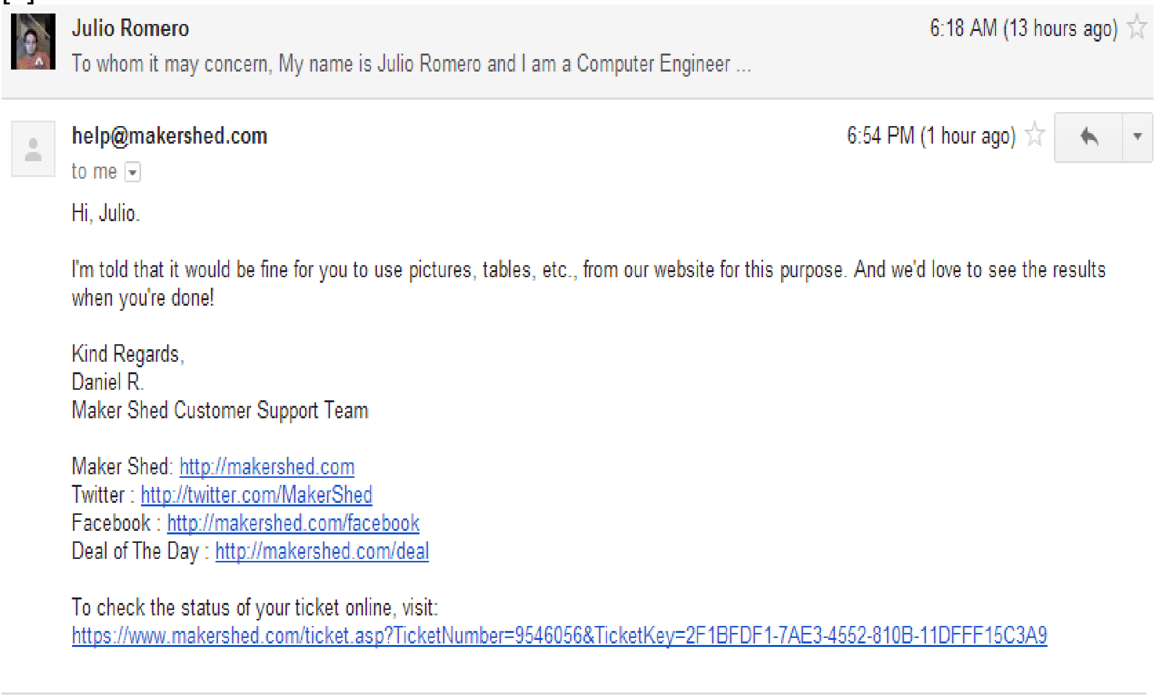
Angela Avanzino
Director of Marketing | Jameco Electronics
Phone: 650-802-1507 | Toll Free: 1-800-831-4242

Reach for Jameco First | www.Jameco.com
Imagine Design Earn | www.ClubJameco.com

Click here to sign up for the Jameco E-Newsletter
From: Julio Romero [mailto:luloromero19@gmail.com]
Sent: Tuesday, April 16, 2013 3:14 AM
To: Webmaster Mailbox
Subject:

To whom it may concern,

[7] Maker Shed



Julio Romero 6:18 AM (13 hours ago)

To whom it may concern, My name is Julio Romero and I am a Computer Engineer ...

help@makershed.com 6:54 PM (1 hour ago)

to me

Hi, Julio.

I'm told that it would be fine for you to use pictures, tables, etc., from our website for this purpose. And we'd love to see the results when you're done!

Kind Regards,
Daniel R.
Maker Shed Customer Support Team

Maker Shed: http://makershed.com
Twitter : http://twitter.com/MakerShed
Facebook : http://makershed.com/facebook
Deal of The Day : http://makershed.com/deal

To check the status of your ticket online, visit:
https://www.makershed.com/ticket.asp?TicketNumber=9546056&TicketKey=2F1BFDF1-7AE3-4552-810B-11DFFF15C3A9