Multi-Function Hexahedron An Interactive LED Cube

Roberto Amaya, Luis Ferrer, Eury Reynoso, and Julio Romero

College of Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32826, USA

***Abstract* — This paper presents the design methodology utilized to realize circuitry that allows for the driving of 512 LEDs in an LED cube based on previously calculated animations. The necessary modes of operation of the cube fall into several categories: (1) Main Animations Mode (Intricate looping animations); (2) VU Mode (Animations following input signal of vu meter); (3) Accelerometer Mode (Animations following input signals from 3-axis accelerometer). This paper also presents the methodology to realize a more complex version of this cube containing multiple colors.**

***Index Terms* — RBG LEDs, Single Color-LEDs, 3-Axis Accelerometer, VU-Meter, 16-Channel LED Driver, Persistence of Vision, PWM Brightness Control.**

I. Introduction

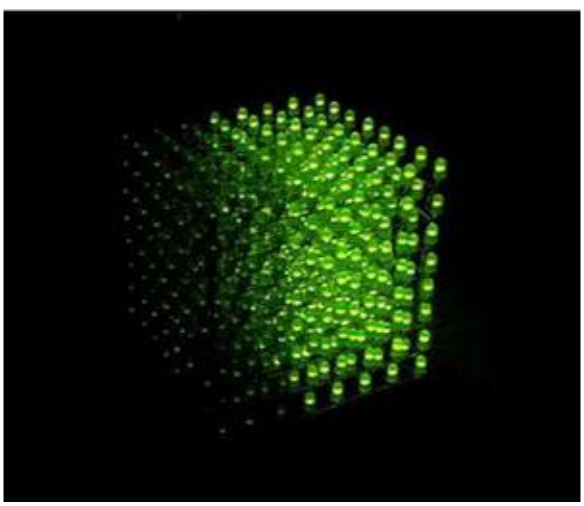

Fig. 1.1 LED Cube by

The Multi-Functional Hexahedron is an interactive electrical system displayed as a three dimensional array of light emitting diodes, which will essentially feature three modes of operation with the overall purpose to entertain its user with a visually appealing light animation.

II. Specifications

The following subsections describe the different specifications (physical, environmental, hardware and software) to which our project abides by.

A. Physical Specifications

The following specifications explain the physical dimensions the cube must follow for this project:

- The LED resolution will be 8x8x8 = 512 minimum, having five visible faces
- RGB light emitting diodes should be used
- Total mass should be approximately 2 lbs. or less
- The size should be approximately one cubic foot
- Electronics should be housed underneath the LED cube structure
- Casing should be made from transparent acrylic/Plexiglas
- FT232 USB cable connection should be accessible from the outside
- Three position switch should be accessible to the user
- PCB encasing will be made out of a light weight material

B. Environmental Specifications

The following specifications explain the environmental guidelines the cube must follow for this project:

- System should work in a standard dry indoor environment
- Should withstand vibrations caused by sound waves at high volumes
- Should function in possible humid/moist outdoor environments
- Should withstand temperatures between 50° F – 100° F
- Should be able to sense acceleration
- Should be able to sense sound

C. Hardware Specifications

The following specifications explain the hardware requirements the cube must meet for this project:

- Components must run off of 5V DC or less
- Must have an on/off switch
- Must have method of changing state of animations of the LED cube
- Must have a method of communicating with PC
- Must have a reliable microphone for registering sound

- Must have a device to measure movement

D. Software Specifications

The following specifications explain the software requirements the cube must meet for this project:

- Program should not be more than the available memory on the microcontroller
- Should 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
- Should be uploaded through a JTAG or FT232 connection

III. Relevant Components

The following subsections describe all of the components involved in this project and how they were used together to make this project possible.

A. Microcontroller ATMega1824p

The selected microcontroller for this project is the ATmega1284P. This is a high-performance controller with low-power usage, reducing heat production.. This section will discuss the features and their use within this project.

Parameters

- 8-bit AVR CPU
- Flash Memory: 128 Kbytes
- SRAM 16 Kbytes
- Pin Count: 44
- Max Operating Frequency 32 MHz
- ADC Channels 8
- Operating Voltage 1.8 to 5.5
- 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 types of animation modes, the selected microcontroller has 128 Kbytes, four times the amount of previous projects as well as the testing microcontroller, the atmega328. A large amount of script will be stored compared to previous projects; therefore this amount should give us plenty of space to perform our complex animations and more.

B. Accelerometer MMA7341L

The MMA7341L[1] 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.

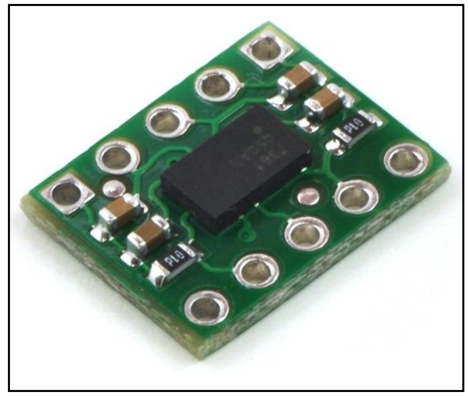

Fig. 3.1 MMA7341L Accelerometer

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 will provide 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 more inexpensive than the DE-ACCM family of accelerometers.

C. VU Meter

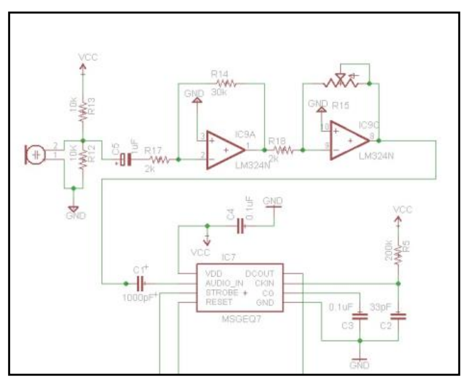

Fig. 3.2 VU Meter Schematic

Table 1
Voltage and Current Operations

	Part Number	**Voltage Range**	**Maximum Load Current**
Microprocessor	ATmega1284P	1.6 to 3.6 V	200 mA
LED Driver	TLC5940NT	3 to 5.5 V	120 mA
Accelerometer	DE-ACCM3D	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	3 mm	3.2 to 3.4 V	20 mA

The selected VU meter model takes audio and divides the spectrum into seven bands. The bands are multiplexed then released. The MSGEQ7 requires an operating voltage between 2.7 to 5.5 volts. The output produced is in a digital format, which meant the team did not need to use the analog to digital converter on the microcontroller as 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). Further discussion of what the source code will do with the data from the MSGEQ7 and the MMA7341L accelerometer can be found in the software design section of this paper.

D. Voltage Regulator

This project has sensitive electronic components, and any power supply implemented will be required to include a voltage regulator to keep the voltage operation point stable. The two most commonly known options are a linear regulator or a switching regulator. To make the best selection, the team followed a guide found online from a published paper by national semiconductor.

The best choice for a specific application can be determined by evaluating each component separately. Table 1 has the parameters considered 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.

E. LED Driver TLC5940NT

The TLC5940NT[2] 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 ranges 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 is anywhere between 3 to a maximum of 5.5 volts, which means this LED driver falls within the needed parameters to operate this LED cube.

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 2.3, 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.

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.

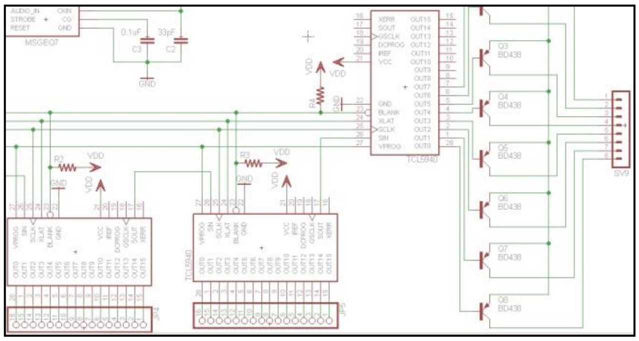

Fig. 3.3 – TLC5940 Connections

F. Power Supply

The voltage input is from a USB "A" type socket and follow the 2.0 specification. The follow resistance and capacitor selection are use in reference of picture 4.9. A maximum of ten micro-Farad is use as bypass capacitor to limited the bus charge drawn to fifty micro-Coulombs and to maintain low alternate current impedance, ensuring only direct current is supply to the circuit.

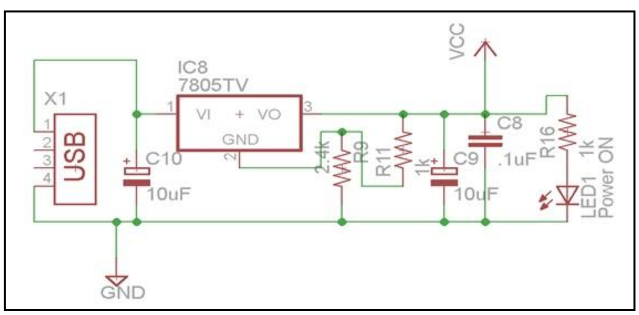

Fig. 3.4 – Power Supply Schematic

The power supply uses the linear regulator 7805 and is connect to make it adjustable. The resistance R11 is chosen to be one kilo ohm and R9 is two point four kilo ohms as a result for the calculation to select three point four volts as the circuit operation voltage supply. R16 is one kilo ohm to reduce the brightness of the LED voltage indicator.

IV. Architecture

The following diagram, figure 3.5, displays the overall architecture of the RGB LED cube system. There are three main subsystems: sensing, processing and display. The sensing subsystem houses the accelerometer as well as the vu meter. This subsystem is in charge or containing the sensing components used to interact with the user of the cube and send input signals to the processing subsystem. The processing subsystem then takes the inputs and calculates the different animations to display on the cube's display subsystem. This subsystem also includes the FT232 board which allows for communications between the computer and the microcontroller. The display subsystem consists of the LED drivers which control the 512 LEDs of the cube. The LED drivers take the outputs from the microcontroller and properly illuminate the LEDs based on the previously calculated animations.

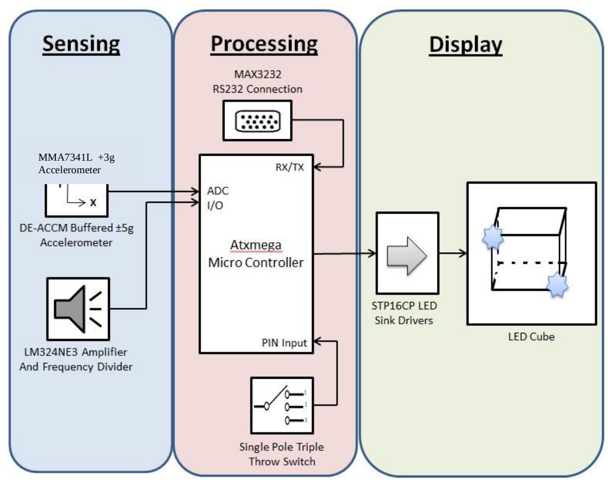

Fig. 3.5 – Overview of Subsystems

IV. Software Design

There will be four basic classes to manage during the coding section of this project. The four classes are Main, Animations, Accelerometer and VU Meter. The code can either be written in C or C++ based on what the team feels comfortable with (most likely C). It would be beneficial to do this task in C, since many resources are available online are in regards to the C language and embedded programming. Also, the university's curriculum bases many of their teachings in C programming language for embedded systems. Therefore making the team experienced with C as of now. This will be written using the Atmel Studio 6 development platform, which is free of charge as mentioned before.

The Main class with initiate 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 will also handle the output method for sending signals to the LED sink drivers. The clearing of the cube array will constantly be taking place as cube is changed from state to state. A series of IF statements or switch cases will be used to alternate between the different modes, calling the required classes to implement the given action. Also the main class will take charge in directing though the signaling of buffers and interrupts.

The Animations class will probably be the largest out of the four. This is due to storing all the varying animation functions. Each given animation will have its own function and specified run time. The more animations, the longer the code in turn the more memory gets used up. Memory management is a bridge that will be crossed when there, since the current objective is to get the cube built and performing simple tasks. The instantiation function will set all the ports and enable switches. Functions to delay time before sending byte data will be implemented

The Accelerometer class will handle the data input and calculations in regard to said device. There will be functions that will calibrate and setup the required ports. A function to take measurements and link results to the cube as well. Gathered and modified data will be sent to the Main class for further manipulation and then sent to the out function for visual manifestation. In retrospect this will probably be the smallest class in comparison to the other three. The VU Meter class will be set up similar to the Accelerometer class. But obviously perform different tasks. There will be functions that will calibrate and setup the required ports. A function will be required to read the spectrum values and modify for cube use. A function to calculate the spectrum into visual form will be called upon and then sent out to the Main class for true visualization. It is currently unclear as to how the volume animations will be displayed. For testing and simplification purposes, the VU meter output display will be set to an outer face of the cube, in which a function must be created for. Depending on how successful this turns out, more complicated VU meter designs can be coded.

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.

V. Housing Unit

The housing unit will be holding the PCB and LED cube. It will be divided into two parts. The box unit will be made out of a light wood material, possibly balsa wood. 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 will have rubber stubs to stand on and prevent it from slipping. The SPTT switch will be mounted in the front, while the power input and RS232 connecters will be in the back. 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 will be placed over the LED cube, enclosing it for protections. It is still undecided if the acrylic casing will be removable or permanently seal the LEDS. The acrylic casing will be slightly bigger than the LED cube itself with about an inch between the LED and the wall of the. The color of the balsa wood container may be painted black for special affect. Everything previously mentioned would be applied if an RGB LED cube was constructed. 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 5.1 is a mock up representation of the finished product.

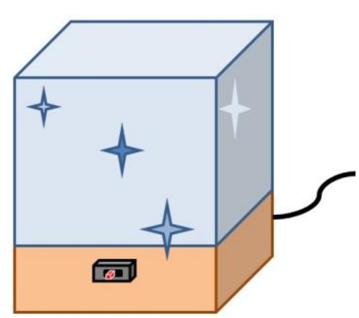

Fig. 5.1 – Casing Mockup

A. 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.

VI. Design Elements

The following subsections describe the different design elements of the project.

A. Maintainability

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

B. 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 will be done. In the final design, the software will gradually be added until a final product is achieved.

C. Performance

The circuit has to run at a fast rate to ensure the persistence of vision principle is carried out. The LED drivers will provide the same brightness for every color and every RGB LED. The final design changes mode based on an input from a triple pole switch mounted on the circuit board. The switch is accessible from the outside after the casing is mounted on the LEDs. Within each mode, the animations displayed change accordingly and do so depending on the predetermined time in the programming.

D. Portability

The final design of the cube closely resembles our initial dimensions of 8 inches tall by eight inches wide by eight inches deep. Unfortunately, due to the amount of wires involved in the RGB LED cube, the cube had to be modified to have more spaces between each LED. This was done to be able to have more viewing room and achieve the 3-dimensional animations of the cube.

E. Safety

The final casing of the design will be 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 will be securely attached to the circuit board and also to the casing. Overall, the final product will be impact resistant and safe to use.

II. ANIMATIONS

As the functional specifications describe, the RGB LED cube design will include three different modes. The switch on animations will be done by a single pole triple throw switch. The specifications for the RGB cube modes are as follows:

A. Main Mode

The Main mode will include different animations. For instance, one of the animations will turn on multiple LED on the top layer and the LEDs on the lower layers will light up acting as raindrops falling. In this mode, only blue LEDs will be used to enhance the illusion of water drops. The other animations will use the different colors in a variety of different ways, but the specifications are yet to be discussed. If time allows, letter patterns will be displayed on the front faces and will move counter clock wise.

B. VU meter

The VU meter will react to ambient sound, and will display according animations in three different patterns. In the first pattern, only one layer will be lit up all the time, and the LEDs in the remaining layers will light up and down as the sound wave increases or decreases; essentially, they will follow the patterns shown on the dot diagram. The second pattern will have the bottom layer always be on as well. However, the different layers will light on and continue as the surround sound wave increases, and they will decrease brightness as the sound also decreases; in this case, they will follow the linear diagram. The third pattern will have only the front faces of the cube interacting. Each column will sequentially light up as the ambient sound increases or decreases. The only difference between the single-color diffused LED cube and the RBG LED cube, as far as the implementation of the VU meter goes, is that in the RGB cube the bottom two layers will turn on blue, the next three layers will turn on green and the upper three layer will turn on red.

C. 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.

D. Cube Construction

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

VII. CONCLUSION

We have successfully completed the research, design, testing, and prototype of the (eight by eight by eight) single-color LED Cube 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), and software animations 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.

ACKNOWLEDGEMENT

The authors wish to acknowledge Dr. Samuel Richie for his assistance and support through the duration of this project.

Julio Romero 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. He is currently interning with HD Supply as a software developer. 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 Amaya is currently a senior attending the University of Central Florida, with expected graduation of August 2012 with a Bachelor of Science degree in Computer Engineering. He is currently 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 currently attends the University of Central Florida, seeking a degree in Electrical Engineering. He is currently working in engineering department in Marriott Grande Vista as technician II. Eury plants to graduate in August 2013. Upon graduation, Eury hopes to obtain a position 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 is currently a senior attending the University of Central Florida, with expected graduation of August 2012 with a Bachelor of Science degree in Computer Engineering. He is currently employed by LiveTV working as a junior testing engineer scripting automated tests of systems on-board airplanes. Upon graduation, Luis hopes to continue growing in his current position in hopes of climbing the corporate ladder.

REFERENCES

[1] Pololu Robotics & Electronics. "Product Description." *Pololu MA7341L 3-axis Accelerometer +-3g/ 11g*. Pololu Robotics & Electronics, n.d. Web.

[2] Texas Instrument. "TLC5940(ACTIVE)16-Channel LED Driver W/EEprom DOT Correction & Grayscale PWM Control." *LED Driver*. Texas Instrument, n.d. We