Smart Lights

Anthony Chand, James Harrison, Ryad Hafeez, Kevin Patel

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida 32816-2450

Abstract – ***The Smart Lights project is a high-level computer and electrical engineering project which uses various components to have RGB LEDS change depending on what the user request on their smartphone. This paper presents the design methodology utilized to implement the Smart Lights system. This paper will go into detail of the hardware aspects of the project as well as the software side. The Smart Lights system will be controlled by a smartphone which will send data wirelessly which will then be collected by the system through a WIFI card. The Smart Lights will also have a feature which will enable the lights to flash depending on the beat of the music which is being played.***

Index Terms – HTI, GUI, IoT,

I. INTRODUCTION

As technology advances, the way people interact with the world is continuously changing and advancing. Some of these changes don't affect us regularly but some can change the way people live on a day to day basis. Because of the internet of things, many different devices can be connected to the internet. Appliances, security systems, and even coffee makers are increasingly being connected to the internet and controlled by mobile devices. Having smart devices in a user's home will help reduce the amount of time wasted on monotonous day to day task such as turning on the lights. Our smart light system will help achieve exactly that.

Our Smart Light system will allow a user to fully control our lights in the setting of their choice. Using our Android application, a user will have the ability to change the states of the light by turning them on/off, and change the color of the lights. Our system will contain 3 individual lights, all controlled and managed from the application. Each light in our system will also have the capabilities to react to music. There are a few big names in smart home lighting such as Phillips and LIFX, our system has most of the capabilities as the competitors with the added feature of reacting to music.

II. Goal and Objectives

The main objective of the Senior Design project is to showcase that the University of Central Florida has provide the necessary education needed to become a competent engineer through their rigorous four-year degree program. The Smart Lights system implements technology that is starting to enter the homes of many individuals as the market of IoT (internet of things) continues to grow it is essential to gain the skills to compete in this emerging market. Two members of the group are aspiring to become computer engineers and securing jobs as software engineers. While the other two members are aspiring to become electrical engineers. Both electrical engineers are interested in working on different signal and circuit analyses. This project will test the group's ability to effectively work with different engineering disciplines and ability to conduct research and development to engineer a system that work based on the requirements agreed on. Also, pushing the group to use technology they are unfamiliar with to achieve the goals of the specifications.

Another objective of the project is to make this product simple to use because its indented users may not have an engineering or technology related background. The computer engineers will accomplish this goal by implementing an application for android mobile devices. Giving the users of this product an application with a GUI (Graphical User Interface) will limit the amount of technical knowledge these users will need to have. The time needed to learn to use the system and setup the system will be cut down. To make the application user friendly they will design the application using the best practice of HTI (Human Technology Interaction). The electrical engineers will make the system easy to use by making the Smart Light Hub portable and little. They are going to accomplish this with using various compact wireless and LED technologies.

III. System Components

The system components are what defines the product. System components are the physical modules which can be designed or purchased that are built together to create a product. This section will provide a technical introduction to each of the components which are using in the Smart Light system.

A. Microcontroller / WIFI Module

When it comes to any product the Microcontroller is the brains of the product. The microcontroller which is used for the Smart Light system is the ESP 8266 esp-12E. The reason for choosing this microcontroller is because it is also a WIFI module. This allows for a smaller form factor while also reducing the amount of power being consumed by Smart Light system. Some feature of the esp-12e are

- Integrated low power 32-bit MCU
- Integrated 10-bit ADC
- Integrated TCP/IP protocol stack
- Integrated PLL, regulators, and power management units
- Wi-Fi 2.4 GHz, support WPA/WPA2

The Block diagram for the microcontroller is shown in Fig.1. Esp-12e offers a complete and self-contained Wi-Fi networking solution; it can be used to host the application or to offload Wi-Fi networking functions from another application processor. When ESP8266EX hosts the application, it boots up directly from an external flash. It has integrated cache to improve the performance of the system in such applications. Alternately, serving as a Wi-Fi adapter, wireless internet access can be added to any micro controller based design with simple connectivity (SPI/SDIO or I2C/UART interface).

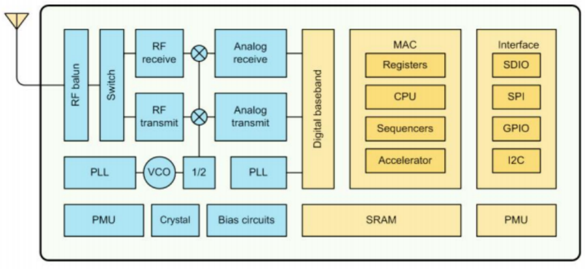

Figure 1:Block Diagram of the ESP8266-ESP12E Microcontroller

B. RGB Leds

The Adafruit neopixel digital RGB led strip is used for the Smart Lights. The benefit of these Leds are it can easily be programmed with the use of an Arduino. It also is power efficient while also being cost efficient. The Led have an operating voltage of 5V and draw around 60mA of current for .65 inch of strip segment. These Leds are also very illuminous, with 30 of these Leds it can easily light up a dark room.

C. Sound Sensor

For this project, the team decided to select the LM393 comparator chip. This module can detect various sounds and sound intensity. The operating voltage for the system is between 3.3-5V. This sound sensor is important for the Smart Lights since it allows the leds to flash on and off depending on what kind of music is playing and at what intensity. This implementation is great for a party environments since it adds another element of excitement. In Fig.3, it shows the schematic of the LM393.

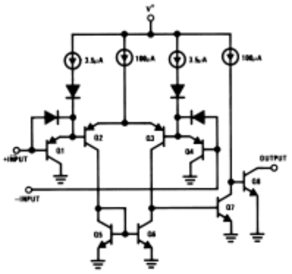

Figure 2: Schematic for LM393 Comparator Chip

While the LM393 comparator chip is the main component, it is not the only component that will be used to detect the sound. Figure 3 shows the overall schematic diagram for the sound sensor that will be implemented into the Smart Light design.

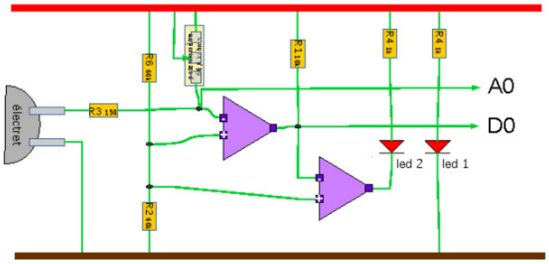

Figure 3: Sound Sensor Schematic Diagram

The two op-amps represent the overall LM393 comparator chip. There are two outputs that are present on the sound sensor schematic. The A0 pin represents the analog out pin. This pin is used to detect sounds being played from a speaker. The D0 pin represents a digital out pin and it is used to read only digital signals. The overall way that this is going to work is simple. From the cellphone, the user will be able to set the light system into music mode. Music mode allows the light system to react to sounds being played which reacts using the Sound Sensor diagram above.

IV. Hardware Design

The hardware design of the Smart Lights consists of the microcontroller/wireless communication, lighting, power supply and the PCB which connects everything together. An overall schematic design can be found in figure 4 below.

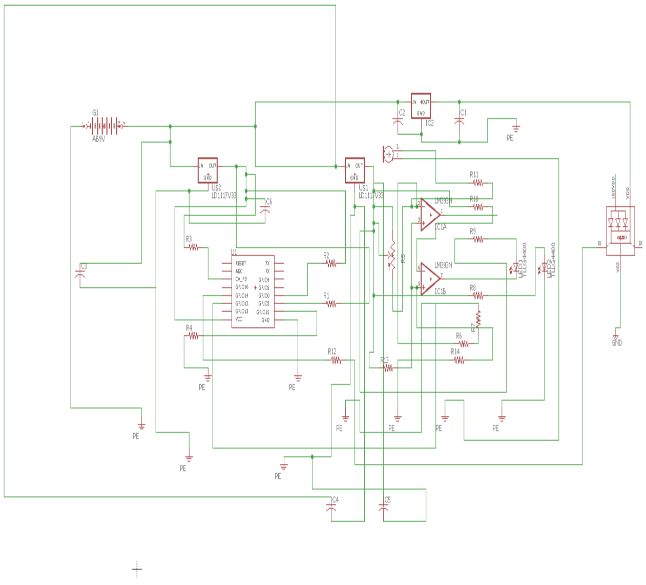

Figure 4: Overall Hardware Schematic Diagram

Looking at the figure above, there is a power source that represents the overall power that will be supplied to the system. There are three voltage regulators in this schematic diagram. There are two 3.3 voltage regulators which are utilized to power the microcontroller and the LM393 comparator chip. To ensure that there is no substantial voltage being supplied from the output of the voltage regulators the team implemented two 10 uF capacitors which would keep the output voltage to 3.3 volts.

For the schematic design shown above, it only shows one LED. The reason for this is because the LEDs come connected as one strip. The team decided however that it would be of best interest to put the LEDs into a circle. So since the LEDs come in a straight line, it would be hard for the LEDs to bend in a circle. Because of this, the team has decided to cut the LEDs one by one and wire them individually into a circle. Doing so would make the design of the LEDs seem more like an actual light bulb.

Now with the sound sensor and the LEDs being wired into the microcontroller, they can not be wired directly into the microcontroller. If that were to occur, the LEDs would fry and so would the ADC pin on the microcontroller. The ADC pin on the microcontroller can only handle a maximum voltage of 1.0 volts. Because of this, the team designed a voltage divider which takes two resistors and drops the voltage to 0.8 volts. In terms of the LEDs, to ensure that the digital signal from the microcontroller turns on the LEDs, the team implemented a 470 ohm resistor. This acts as a voltage protection for the LEDs in the event that the LEDs were to burnout.

The next set of resistors are for the microcontroller. In order for the ESP8266-ESP-12E to function correctly, there needs to be a set of resistors implemented between certain pins on the microcontroller. On the ESP-12E, the Chip select pin needs to be connected to 3.3 Volts. Before the connection can be made however, there needs to be a 10 kΩ resistor needs to be implemented in between the connections. The reason for this is because the team wants to protect the microcontroller from frying since there will be multiple pins on the ground port and on the 3.3 Volt port. Besides connecting the ground pin to ground on the ESP-12E, there needs to be another connection to ground in order for this module to function correctly. The other pin that needs to be grounded is GPIO15 on the microcontroller. As for the reason why this pin needs to be grounded however, there was no source online that could explain the reasoning behind the grounding of this pin. However when the team tested the microcontroller, the only to program this or to have it function is to have that GPIO pin grounded.

The next figure that will be explained below is the Hardware block diagram. This can be used to better illustrate the overall hardware schematic diagram below.

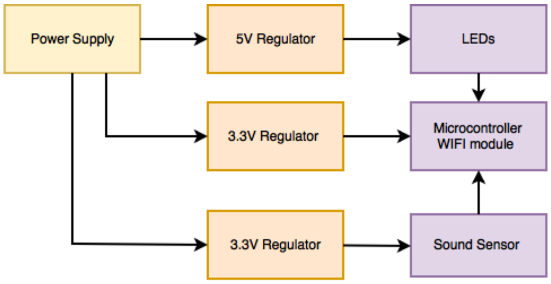

Figure 5: Block Diagram for Hardware Schematic

In the block diagram above, it clearly illustrates that the power supply is going to be supplying voltage to three different components. Each of these components that are getting voltage from the power supply are the three voltage regulators. The voltage regulator going to the LEDs is the LM7805 linear voltage regulator. This voltage regulator is required because the overall neopixel LEDs requires a 5 Volts dedicated voltage. Because of this the team selected this regulator to take the voltage from the power supply and outputs to 5 volts.

The other two voltage regulators used in this project are both 3.3 voltage regulators. The purpose for these two regulators are to power the microcontroller and the sound sensor. These two regulators are called the LD1117V33 voltage regulators. The sound sensor is capable of handling between 3.3-5 volts but the team decided that it was better to supply 3.3 volts to the sound sensor because the overall sensitivity of the sound sensor yielded better results at 3.3 volts. The other 3.3-volt regulator is used to power the microcontroller. The microcontroller is only capable of handling a maximum voltage of 3.3 volts. If the microcontroller is supplied with the power supply, it would easily burn out.

In the block diagram above, there are two components listed, the LEDs and the Sound Sensor that are connected to the microcontroller. The LED and Sound Sensor receive data from the microcontroller which tells the components to either turn on the LEDs or to enter a 'music' mode which enables the LEDs to react with the sound of music being played. As to what will be sending the data to the microcontroller, the team decided to utilize an android device to send the data over to the microcontroller. The reason why this was selected was because the team felt that the integration of android software to the microcontroller would be more smoother than integrating the microcontroller on an iPhone device.

The power supply for this system is designed to take an input voltage of 120 Volts AC power and convert it to a 12 Volts DC power. The overall schematic design of the power supply can be found below.

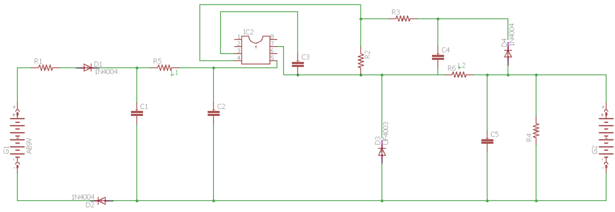

Figure 6: Power Supply Schematic Diagram

In the above schematic diagram, there are two voltage sources. The voltage source on the left represents the input AC voltage going into the power supply. The voltage source on the right represents the DC voltage source that is being outputted. The input voltage that is going into the system is 120 Volts AC and the output voltage coming out of the system is 12 Volts DC.

In the schematic diagram above there is a chip. This chip is the LNK304PN AC/DC converter chip. The overall purpose of this chip is to take the input voltage and convert it to a DC voltage that can be used for the three smart light systems. The reason why this chip was selected was because the voltage that this chip can handle is a maximum of 265 Volts. Even though the system would never hit 265 Volts AC, the team needed an AC/DC chip that is capable of handling the 120 Volts of AC power from a standard wall outlet.

The overall use of the inductors, capacitors and resistors for the power supply is needed because supplying an input voltage of 120 Volts AC is dangerous. If the power supply is not properly wired, then the power supply is capable of being unstable. Because of this, the team took extra precautions in wiring the power supply and utilized capacitors, inductors and resistors that had a maximum voltage rating of 400 volts.

V. Software design

A major component of the software was the Android application. Through this application, a user could fully control the lights in this system. Before any code could be written for this system, the application was planned thoroughly. Unified Modeling Language (UML) is used in object oriented software engineering these UML diagrams are used to model the behavior, processes and the structure of an application.

A. Android Application

One of the main diagrams used to display the functionality of an application or software is a flowchart. In a flowchart, the different symbols represent a different aspect of the system, for the legend for these symbols is located at the bottom left of the flowchart. For example, the cylinder represents an action taking place which involves the database. The arrows generally indicate the flow of data or the flow the user will go through.

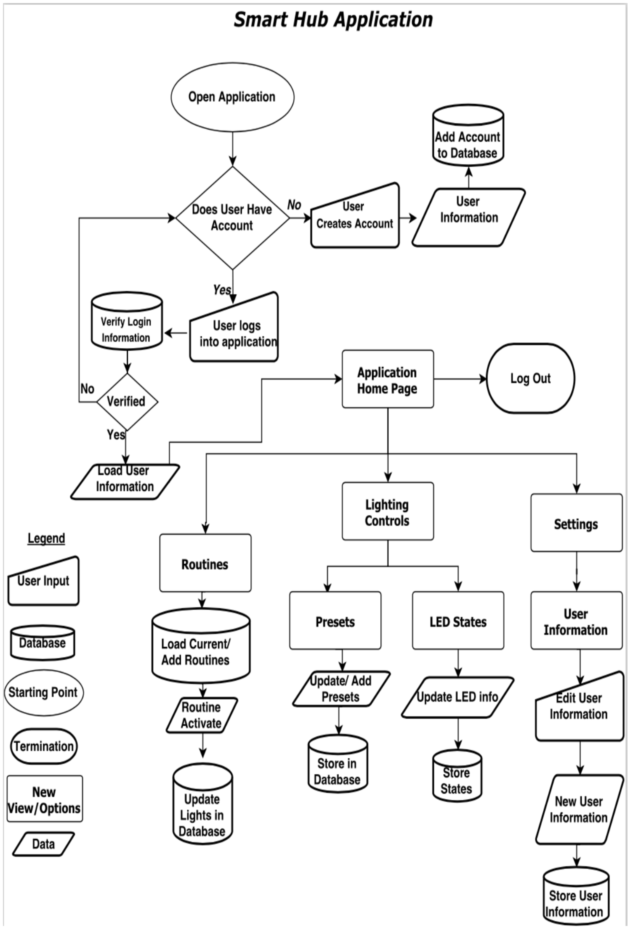

Figure 7: Android Application Flowchart

This flowchart is a representation of the capabilities and process the user will go through when using the smart lights application for the proposed system.

Once a user enters this application for the first time they will have to create an account, the information manually entered by the user will then get stored in a Firebase Database. Now anytime the user enters the application they can log in with the credentials used to create the account, the Firebase database will then verify the credentials, if these credentials match what is in the database, the user will be taken to the home screen of the application. The home page of the application, as shown below, is where the user will spend the most of their time. From this screen, a user can change the state of the lights (turn them on/off), change the color using the color picker shown below, and load the full light information to change the name as well. Tapping the music icon in the top right of the screen puts the lights into music mode enabling them to react to music.

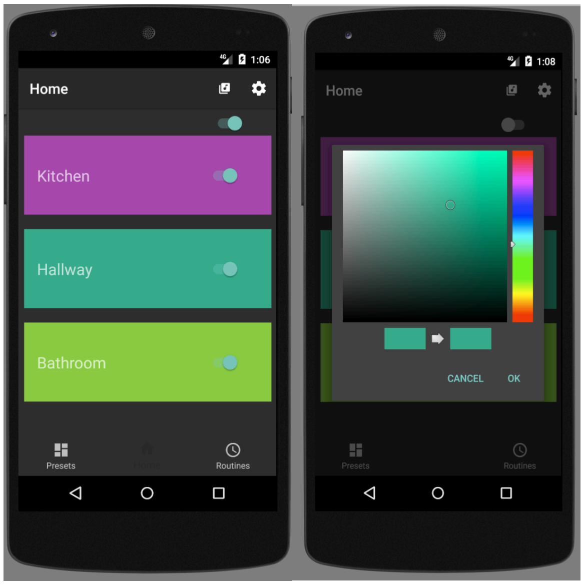

Figure 8: Home Page and Color Picker

The user can select the “Presets” icon in the bottom left of the screen, this will take them to the presets page. A preset is a light configuration which is created by the user, this configuration will have a saved name, and colors from this page a user can see all their created presets from this view.

If a user clicks on the “Routines” icon on the bottom right of the navigation bar they will be taken to the Routines page. From this view, a user can have light settings which activate based off time.

The user can also choose to adjust the settings of their application, the main setting in this application would be the users personal account information. From this page, the user can edit all information such as name, email, username, and password. This data is then passed to the database to be stored. Firebase updates in real-time so the user will automatically be able to use their new account information. From inside the settings page the user can log out of the application. This will then allow a new user to log in.

B. Wireless Technologies

For user authentication and the real-time database this application implemented the user of Google Firebase. Firebase is a NoSQL cloud hosted Real-time Database owned by Google. JavaScript Object Notation (JSON) is used to store the data in a Firebase Database. Data in a Firebase database is also updated in real-time for all users connected. Firebase allows users to share one real-time database, instance across multiple platforms, such as iOS, Android and JavaScript.

From Client-side code secure access to the Firebase database is granted. This data is available even when the application is offline since it is persisted locally. Although to give the end user a responsive experience real-time events continue to execute. The updates that took place while the application was offline are not lost, once the application re-enables its online connection the real-time database synchronizes the changes and resolves any conflict with the data. Firebase provides a scalable real-time experience, and allows quickly executing operations thanks to the design of the Real-time Database API.

Firebase has many capabilities and benefits the most notable being the data synchronization because of the Real-time Database. The responsiveness of the applications even when they are offline. As well as the firebase database being accessible from any web browser or mobile device. Firebase also offers notifications, cloud messaging, crash reporting and more.

For our project, we used a service to communicate to are individual systems called PubNub. PubNub offers a wide array of services such BLOCKS, publish/subscribe messaging, online presence detection, access management, data stream controller, storage & playback and mobile push notifications. For the purpose of our project the main service that was utilized is the publish and subscribe messaging.

To communicate with the hardware, we used a service called PubNub. PubNub is real-time publish subscribe messaging service that is deployed globally. The PubNub service is gaining a lot of popularity for IOT applications. This was an essential service for the smart light system because we needed a way to communicate to are devices from anywhere in the world. This service makes it easy for us to connect and scale are project if needed without having to make or maintain a real-time infrastructure.

C. Embedded Design

The heart of the project is driven by the embedded software. when the system is powered on startup goes through an initialization/setup process. when completed the system will create a temporary Wi-Fi network that will be used to connect to the local Wi-Fi that the user specifies. Once connected to the internet the system will start to listen to the appropriate PubNub channel waiting for a message to be sent.

Now that the user has connect the device to the internet the system will start to listen. If a payload is not received in ten seconds of the communication channel being opened it will repeat the listening process. When a JSON payload is received from the user it must go through a pre-process step.

The system has successful received the payload which is a JSON object. The JSON object contains important information that the system need in order to complete the correct action. The format of the payload is the following {“0": R, "1": G, "2": B, "3": state, "4”: music mode}.
R, G and B have values of 0-255 to indicate the insanity of the color. while state and music mode should contain a value of 0 or 1 to indicated if they should be on or off.

After all the information collected and stored we can start the appropriate functions to change the color of the system turn it off or on and finally turn music mode on or off. Once the task has been accomplished it loops back in the listening section of the code. Figure 8 below give high level overview of the all the step decided above to give a visual representation of how the system works.

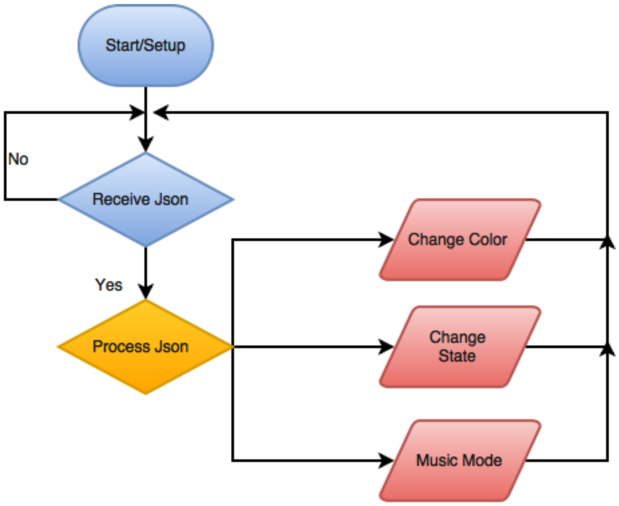

Figure 9: Embedded Flow Chart

D. Software Testing

Before software is deployed extensive testing should take place. Ensuring not only that the software can complete the necessary objectives, but also to ensure that any fatal errors are not found in the software. The proposed Android application will need to allow users to login, the content each user can access will be loaded from the database, the application should then be send a signal to the smart hub to change the states of the LED’s. There are many errors which could occur in this process, such as the states of the lights not updating, so the objective of this testing is to find these errors such that the application can function completely.

The testing was broken down into two main portions, testing just the application functionality, this can be done during development and then testing the communication between the software application and the hardware smart hub.

The testing of the application was done using the Espresso Testing Framework. This framework is provided in the Android Testing Support Library, this framework provides application programming interfaces (API's) for writing user interface(UI) test. These UI test can simulate user actions, such as tapping a button, or typing keyboard input.

The test script which were written were primarily used to ensure the applications main functionality was stable. The most important test written were creating a new account, logging in to an account, updating the current light information, creating and loading a preset and creating a routine. With these tests run frequently, the applications functionality was more stable.

V. Conclusion

The internet of things has opened many possibilities for technology, this new technology is continuously reducing the amount of monotonous task people must do on a day to day basis. IoT simplifies the life of its users, whether it's starting your coffee on your phone or starting your car. Simplifying the life of a user is one of the main goals of this group, our smart light project accomplishes this by allowing users to control lights without even having to walk to a light switch. A user can fully control their lights right from their Android smart phone. They can turn on/off, change the color, create preset light settings for quick selects by importing an image and selecting colors or manually selecting the colors. The user can also set routines within the application, these routines will change the lights based off the time set by the user, helping the user wake up and fall asleep. This system is also useful in situations such as parties because each light reacts to music. Will a little more time the group will trying to refine the design so they can compete with the leaders in the industry. This project has been a great learning experience for all the members and with a project with multiple people there are problems that will arise. Each member has strengthened their troubleshooting and problem solving skill that will translate directly into their future endeavors.

BIOGRAPHIES

Anthony Chand is a 21-year-old senior Computer Engineering student at the University of Central Florida. He will be graduating August 2017 with a Bachelor's of science in Computer Engineering. He has accepted a job offer with Lockheed Martin as a Software Engineering Associate. He plans on pursuing his Master's Degree in Computer Engineering after a year of full time work experience and is interested in machine learning.

James Harrison is a Computer Engineering student at the University of Central Florida. He will be graduating with his bachelor's degree in August of 2017. He specializes in web and mobile development. His career goals are to work for a company which has large room for growth and allows him to learn many different types of technologies.

Kevin Patel is an Electrical Engineering student at the University of Central Florida. He will be graduating in December of 2017 with a Bachelor's of science of Electrical Engineering. His plans are to work for a company which are in the fortune 500 and then climb up to become a manager or even CEO.

Ryad Hafeez is a 22-year-old Electrical Engineering senior at the University of Central Florida. He will be graduating in August 2017 with Bachelor's in Electrical Engineering. He specializes in electrical circuit design and radio frequency testing. He plans on pursuing a professional engineering license soon and hopes to go back to complete his Master's in a year. His career goals are to work for a company that enables him to learn more about the electrical engineering field. He is interested in learning more about power design and electrical design.

ACKNOWLDGEMENT

The author wish to acknowledge the assistance and support of Dr. Lei Wei, and the university of Central Florida.