SmartLock

Gregory Mueth, Damo Park, Mhelith Natavio

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

***Abstract* — SmartLock technology has been around since the "Internet of Things" trend began, yet they have hardly proliferated through the home security market at all. We believe the high cost of these smart locks is what's hindering them from becoming more widespread. A quick look at the home depot smart lock isle will reveal that if a customer desires a WIFI integrated lock with a mobile application, they will be spending $200 or more. This team's SmartLock design retrofits an existing deadbolt for a home door with our own microcontroller and servo. This design includes WIFI integration that communicates though a mobile application through our node.js web server. Additionally, we've designed and built a key FOB able to send a string through 433MHz radio waves for another avenue of entry. We believed and have proven that these two key features can be implemented in an efficient manner that could be marketed towards consumers.**

***Index Terms* — SmartLock, Internet of Things, radio transceiver, WIFI, key fob, security, Android mobile application**

I. Introduction

As technology continue to progress, we as a civilization, are constantly creating ways that can improve any aspects of our lives. Electronic devices such as mobile phones, computers, and televisions are some of the few technologies that have considerably changed and are continuously changing the way we live our lives. One of the common household devices that is being improved today is the door lock. Companies like August Lock, Schlage and Kwikset are changing and improving the traditional lock by incorporating smartphones to access their door. However, the cost these smart locks are extremely high for most people. Smart locks are beneficial when it comes to those times where one has misplaced, forgot, or lost their keys and cannot physically lock or unlock their door. SmartLock is design with those problems in mind while ensuring that the system will be easy to use.

To further push forward the Internet of Things idea, this design team set out to create a SmartLock that not only can it perform the basic features that the other smart locks have but also make it more affordable. The first fundamental key feature that our SmartLock will have are that it will be capable of communicating through a local area network via our remote server. An android mobile application is available to download which is capable of maintaining connection with the door lock anywhere in the world on the condition that the smartphone hosting the application have a working internet connection. The second key feature is the radio key FOB. This key fob operates similar to automobiles. It sends a string from the key to the door module via a 433MHz transceiver.

More importantly, our design seeks to create a SmartLock with features to compete with the largest companies, while still maintaining an affordable cost for the everyday consumer in hopes that this exciting home security device can become more standard in homes everywhere. This is something that has become standard on almost all new automobiles but has been slow to catch on in the home market. We seek to create a low power consuming, reliable keyless entry system by mirroring the methodology used by auto manufacturers. Purchasing keyless entry system that operates in a similar way is currently very expensive, however, we are confident that by building our own, this project will prove that the cost can be drastically reduced, allowing more consumers to have modern home security.

In addition to these new innovations to home security, this design team intends to leave the mechanical locking mechanism in place. This allows the user to still use the mechanical key as a backup should the electronic system fail from low battery or lack of internet connectivity. There are many smart locks on the market that overlook this feature, making their product not viable for a consumer with only one exterior lock such as apartment dwellers. Through these basic improvements from most the smart locks that are currently in the market, we hope to prove that our project is a more convenient and affordable alternative.

II. Goals and Objectives

The goal is to create a project that showcase the knowledge that each of the member acquired throughout the years at University of Central Florida. Consisting of members from both Electrical and Computer Engineering Majors, we want to have a project that can exhibit both the electrical and computer aspect of the group. With this in mind, we decided to go with a project like smart lock that has good computer and electrical foundation.

The objective of the project is to have a low-cost alternative, not bulky and an easy to install smart lock that features a WIFI integration and a remote key. This will be achieved by purchasing components that are low-cost while ensuring that the quality does not drop and compact PCBs. Additionally, an android mobile application will be created in Android Visual Studio which let users to either lock or unlock their door wherever they are as long as the smartphone that has the application is connected to the internet.

For added security, the backend side of the mobile application includes a token that will be included in the header every time the mobile application post to the server. With this included it will be harder for other people to post data on the server. Furthermore, the application is constantly pulling the status of the lock and displaying it on the main page of the application making it easier to the user to determine the status of the lock.

III. Engineering Specifications

Name	Target
Key Fob Range	1m
Signal Frequency	> 350MHz
Supply Voltage	> 5V
Button/ Key Fob Latency	> 1 second
Mobile Latency	> 5 seconds
Battery Life	> 168 hours
Size of door module	25cm x 7cm x 4cm

Table 1: Engineering Specifications

The above table contains a list of specifications the project must meet to be considered a success. The key fob range must be above one meter for the ease of use. In order to achieve a higher range radio identification, a high frequency radio signal must be used, so the signal frequency was decided to be above 350MHz. The supply voltage must be over 5V in order to power our microcontrollers. There are two specifications dealing with input latency. These input latencies are to be minimized to ensure a good user experience. But to achieve these low latencies, it must be balanced with the following specification, battery life. Latency can be minimized by polling the server more often, therefore spending more time out of low power mode. The final specification is the size of the overall project. It must be small enough to comfortably fit on a door that sees daily use.

IV. Hardware Components

In this section, Hardware Specification, will explain what components are used, how the components are used, and why the components were chosen for this project.

A. Microcontroller

In our project, Smart Lock, ATMEGA328 is used as microcontroller for both key module and door module. Microcontroller could be a brain of the smart lock, it will control each component, analyze a receiving signal to command. ATmega328 is high performance and low power 8 bits microcontroller, which combine flash memory, input/output ports, 32 general registers, three timers, 61 internal/external interrupts, programmable USART, and so on. ATMEGA 328 composed with 28 pins dual inline package. The table shows the specification of the microcontroller below.

Specification	ATMEGA328
CPU Type	8 bits AVR
Operating Voltage	1.8 – 5.5 V
Recommended Input Voltage	7 – 9 V
Digital I/O Pins	14 pins
DC Current for I/O Pins	40mA
Flash Memory	32KB
Clock Speed	16MHz
Temperature Range	-40 - 85 °C
Speed Grade	0 – 20 MHz at 1.8 – 5.5V
Power Consumption at 1MHz, 1.8V, 25°C	Active : 0.2mA Power Down : 0.1uA Power Saved : 0.8uA

Table 2: Specification of Atmega328p [1]

B. Radio Frequency Transceiver

To communicate between key module and door module, we decided to use CC1101 radio frequency transceiver. Most of important requirements of the project are power consumption and the size. CC1101 is low voltage power consumption transceiver and the size is compact as well. CC1101 is low power RF transceiver, operates in the 315/433/868/915 MHz frequency band. It also can be programmable output power up to 12dBm for all supported frequencies. The rate data from 0.6 to 600 kbps. In our project, Smart Lock, will be used 433MHz frequency band. The table will show parameter of CC1101 transceiver below.

Parameter	CC1101
Supply Voltage	-0.3 – 3.9 V
Voltage on Digital Pin	-0.3 – 3.9 V
Voltage Ramp-up Rate	120kV/us
Storage Temperature	-50 °C – 150 °C
Current consumption at 433MHz range	13.1mA to 16mA

Table 3: Specification of CC1101 [2]

CC1101 Radio Frequency Transceiver consist of 8 pins to operate with microcontroller, and in our project, we will used only 7 pins will be used to communicate with microcontroller. VDD is for input power, SI is for data input, SO is for data output, GDO0 is for Serial Clock output, SCK is for Clock pin, SS is for Chip Select, and ground.

C. Status Indicators

It is hard to recognized that if the door is unlocked or locked. Since the deadbolt is inside of the door frame it will not be able to see. To solve the problem, we decided to make some indicators such as LED lights. User will be able to recognize the door state by observing the indicator light.

Two pins will be used for the LED light on door PCB. One pin will be connected to microcontroller through 128-ohm resistor, another pin will be on ground. If the door is unlocked, the LED light will be on, if the door is locked, the LED will be off.

Another two LED are used for indicator of WIFI module statue. Since we keep posting the state of the door to the server, we need to make sure if the WIFI is connected to the server or not. If the green LED is on the WIFI is trying to connect to the server, if Yellow LED is on then the WIFI is connected to the server.

A button is installed at the inside of the house to unlock or lock the door from the house. The input button is connected to the WIFI module, once user push the button it will lock/unlock the door. At the same time, the statue of the door will be updated to the server.

D. Servo Motor

The servo motor will be used to unlock or lock the deadbolt. To decide what servo motor will be used for our project, we needed to concern about the power consumption, size, torque, rotation angle. Since the power is supplied by 4 of AA battery, power is pretty much limited, one of our objective of the project is make it as compact size. Also, the servo motor should be able to rotate the deadbolt itself, and it has to rotate enough to unlock or lock the deadbolt. We chose a servo motor, TowerPro MG995R metal Gear Servo. The servo motor consists of 3pins, power, PWM, and Ground. Since we are using 4 of AA size of battery, the power will be supplied as 6V to the servo motor. Table 3 shows the specifications of the servo motor.

Specification	MG 995R
Weight	55g
Dimension	40.7 x 19.7 x 42.9 mm
Torque	10kgf·cm at 6V
Operating Speed	0.16s/60° at 6V
Operating Voltage	4.8V to 7.2 V
Dead Band Width	5us
Temperature range	0 °C – 55 °C
Rotate Angle	120° (60° each side)

Table 4: Specification of MG995 [3]

E. WIFI Module

WIFI module is needed to connect server and door PCB. We decided to use ESP32, single 2.4GHz WIFI. ESP32 has serial port monitor, it is really easy to read what it is doing. The table below shows the specification of ESP32.

Specification	ESP32
CPU	Single Core 32bit
Memory	520kB SRAM
Peripheral Interface	12bit up to 18 CH
Current at deep sleep	5uA
Current at transmit	180mA

Table 5: Specification of ESP32 [4]

There are two wires connected between door PCB and WIFI module, one wire is for the signal from WIFI to door PCB, another wire is for the signal from PCB to WIFI. Since WIFI module receive only 3.3V signal and door PCB communicate with 5V signal, we used two resistors to drop voltage which is from door PCB. 2K ohm and 1K ohm used to drop voltage.

F. *Power Distribution*

Power is the fundamental matter to work components. Microcontroller, transceiver, and WIFI module will need power to operate. Our project will need both 5V and 3.3V to supply power to the components. We will have four AA battery, 2CR5 battery, 5-voltage regulator, and 3.3 voltage regulator. Since we have same microcontroller and transceiver for both key module and door module, same voltage regulators will be needed. Due to servo motor consume pretty much of power, the power does not supply to the WIFI module properly. When the servo motor is rotating, WIFI module reset itself due to shortage of the power. To solve that problem, we decided to supply power for servo motor separately. 4 AA battery is used to supply servo motor, 2CR5 supply power for MCU and WIFI module. Since servo motor and WIFI module supplied power separately, WIFI does not turn off when servo motor is rotating. The figure bellow shows how the power distributing on PCB.

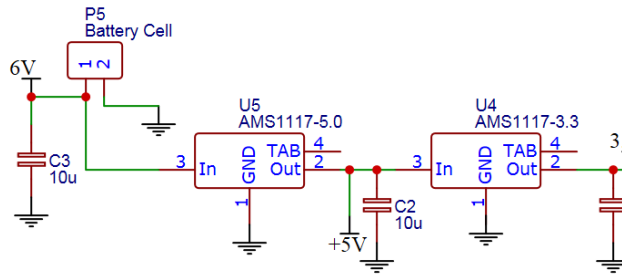

Figure 1: Power Distribution

To supply proper voltage to microcontroller and transceiver, we sued two voltage regulators. 2CR5 is connected to the 5-voltage regulator, the output of the 5 voltage regulator supply power for WIFI module and MCU. Another extra pin will be used to supply power for 3.3 voltage regulator. 3.3 voltage regulator supply power for transceiver.

Since the key fob should be easy to carry, we cannot use AA battery pack to supply power. Coin battery will be used for power of key fob. Two CR2032 battery are used for key fob. it is convenient to carry it since the battery is small and light. The power distribution of key fob will be same as door module PCB.

G. *PCB*

PCB, printed circuit board, connects all electronics and components using conductive tracks or pads. Also, PCB has one than one layer. Components are soldered onto the PCB to connect with PCB stably. There are plenty of PCB software in website, EasyEDA is the one what we used to design PCB. The reason we chose EasyEDA to design PCB is it is really easy to use. We do not have to download the software to design PCB, also it is really easy to share to each member. Moreover, a lot of user upload their own designed components, most of component what we need to use are exist.

There will be two PCB layout for the project Smart Lock, Door main board and Key module PCB. For our PCB, it will be only one layer and front and back printed for both key module and door module. The figure below shows PCB for both key module and door module.

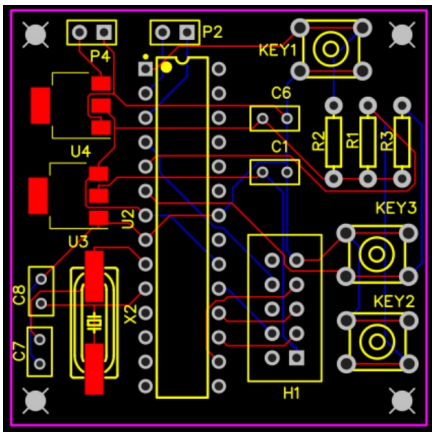

Figure 2: PCB Key Fob

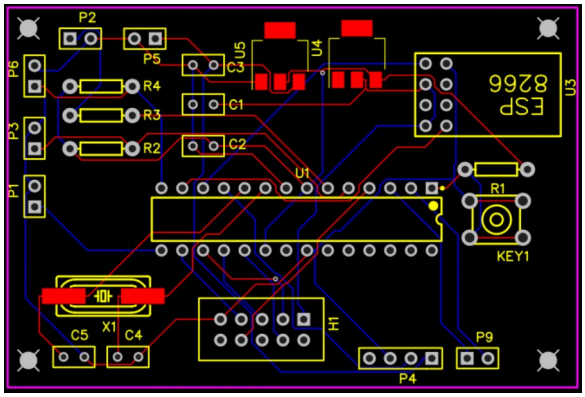

Figure 3: PCB Lock Module

Top PCB is for door module and bottom PCB is for door module. The red line is for top layer and blue line is for bottom layer. As you can see in the figures, a microcontroller is located at center for both key and door

module. Since we are using ATmega328, a 16MHz crystal is needed as oscillator to create an electrical signal with a precise frequency. One button installed on both PCB to reset microcontroller. Two 22pF capacitors will be needed for crystal as well. Two voltage regulators are located, and 10uF capacitors are used to supply stable DC current, female header is installed on PCB so that we can easily mount or connect components to PCB. Push buttons are installed on key fob which will be used to unlock or lock the door.

V. Mobile Software Specification

The application is designed and written in JAVA using Android Studio. The goal is to have a simple yet appealing user interface. To make the application as simple as possible without omitting the main features there are only three screens that the user can go to— the login screen, the register screen and finally the main page. It also features fonts that are easy to read, color contrast that are not hard on the eyes, elements are placed nicely where it can easily be seen and accessed. The flowchart below shows how the application flows.

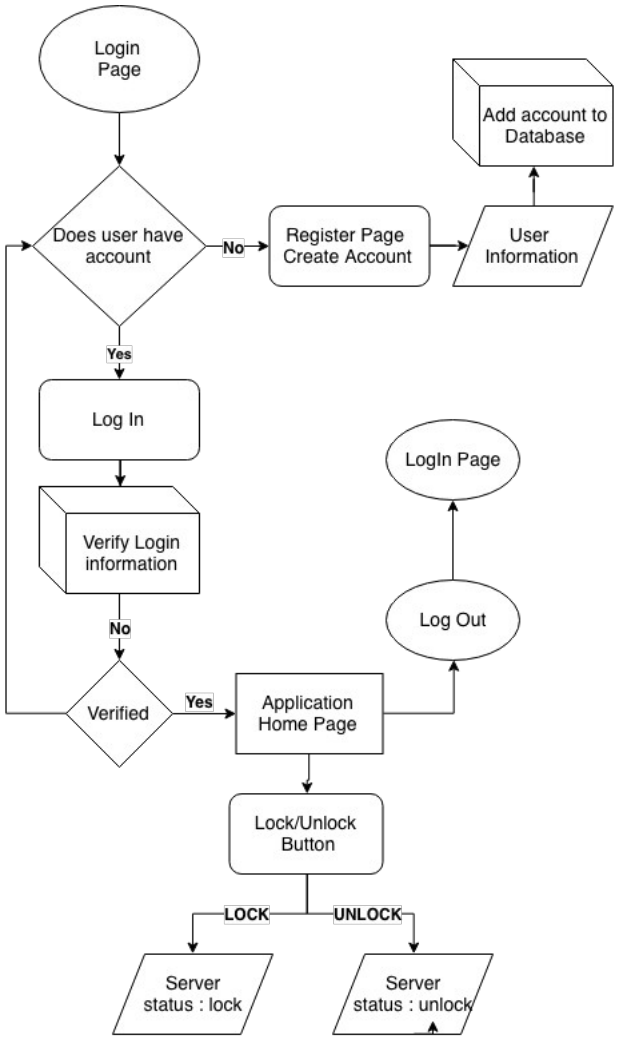

Figure 4: Android Mobile Application Flowchart

It starts at the login page shown in the Figure ___, where a user can either login straight away with their registered email and password or they can choose to register a new account.

If they chose to register, they will be taken to the register page where it will ask them to input their name, email and their chosen password. To avoid account duplicates, the backend will check if the account is already in the database and if so, it would not save the account information to the database and an error message will appear below the screen. If this is the case the user can either return to the login page by clicking the “Already have an account” link to login with the information or they can put a new information. With this new information put in the appropriate fields, the following steps will be executed— the account will be saved to the database, a message saying that account has been registered

successfully will appear at the bottom of the page and then it will send the user back to the login page.

When logging in to the application, user must put their email and password into the designated fields as shown in Figure __. If any of the following cases are done error message will appear at the bottom of the screen: not having an email, password or both information and account is not registered. Moreover, an option to view the password is available for user to prevent them from getting an error from putting the wrong password. After logging in successfully, the user will now be taken to the main page of the application.

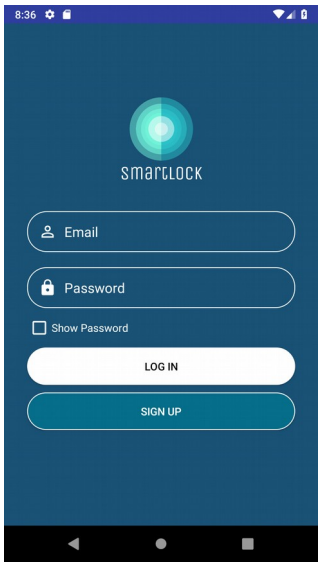

Figure 5: Android Mobile Application LogIn page

On the main page, as shown in the Figure __the user will instantly see the SmartLock logo placed at the top center of the page. Below are the users informations which are the username, email and the current status of their lock. The application is pulling the status of the lock every second. Just below all those informations, slightly separated from the rest is the button that can change the servo motor to either lock or unlock. As mention before the application is pulling the status of the lock from the server every second, meaning the displayed status will change only after a second pass when the servo motor is changed state. Finally, a logout link is located at the top right corner of the page that will instantly send back the user to the login page.

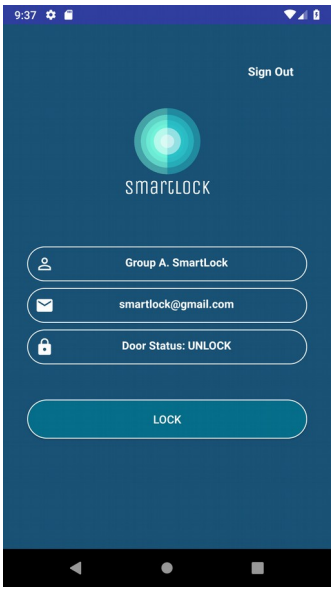

Figure 6: Android Mobile Application Main page

In the future, the group would like to improve the mobile application by adding more features such as adding more locks, deleting locks and having it push notification for every time the state of the servo motor was changed.

VI. Server Software Specification

For the mobile application to be able to communicate with the embedded system on the door lock through the internet, a remote hosted server must be utilized. This is accomplished by using Node.js with Express framework. Node.js is a fast executing JavaScript engine with a wealth of libraries and modules. Adding the express framework on top of it allows for easy handling of HTTP requests. Originally, the group had decided to use a LAMP stack with PHP for the backend, but after one of the design team members spent and an internship writing backends for mobile applications in node.js, the group's confidence in producing a quality product using node was much higher than PHP, so the decision to switch was made based on experience and familiarity.

The database our team chose is MongoDB, which is currently the most popular database choice for node.js web development. It is a NoSql database, meaning that by default it is non-relational. However, for simple applications, it can be made relational by embedding forms inside of one another. In our case, we have only two database collections, users and locks. Because the lock is receiving PUT requests on a regular basis to change its current state from locked to unlocked (or vice versa), and

one lock can be associated with many users, embedding the full schema of the lock into each user is not an ideal approach. Instead, the lock id must be stored in the users form and referenced with a separate operation. Performance suffers a negligible amount with this approach and it is worth the tradeoff as the only other alternative is to embed the full lock schema in each user it is associated with and update the lock item in all user forms after every lock/unlock event.

An additional reason for choosing node.js with express for the mobile application's backend is that there are node packages that make setting up JSON web tokens simple. To ensure secure access to locks in the database, and for a cleaner user experience, JSON web tokens must be used. The basic idea behind tokened user authentication is that after the user logs in, the server issues them a token that contains their account information including the user's id and access level. This token is then stored on the user application side and is included in the header of all subsequent HTTP requests. The benefit of this system from a frontend perspective is that the user doesn't have to log in each time they open the mobile application, creating a cleaner user experience. From a backend security perspective, access to operations is completely secured.

The SmartLock API uses the https protocol for its encryption, for integrity of data, and for authentication. The project is centered around home security, so transferring data over regular http would have been inexcusable in 2018. By encrypting data transmissions and preventing man in the middle attacks, as well as utilizing tokened user authentication and controlling access to administrator routes in the API, we can be confident that there will not be security breach from the mobile application and Wi-Fi integration portion of our project.

The node package jest allows for simple creation of unit and integration tests. Its libraries streamline the code required to write tests and it has a tool that automatically runs all unit and integration tests every time it detects a change in the code, streamlining the development environment. It also has code coverage tools and reports, which shows what portions of code down to the individual line that are not currently covered by unit or integration tests. Because of our use of this node package, the SmartLock API has 100% coverage of test cases and as such, can be refactored or other changes can be made with complete confidence that they will not interfere with the operation of the rest of the program.

The SmartLock API is currently deployed on a Digital Ocean Linux virtual machine. A url was purchased from Go Daddy and pointed to the node project running on the virtual machine. The node package 'forever' ensures the server restarts any time the virtual machine is reset or the server crashes for any reason.

VII. Conclusion

The SmartLock has four main components— the remote-control key, the door module, the mobile application and the server, that communicate through each through a server.

The remote-control key has a button that can be used to lock or unlock the servo motor. This works by fetching the status of the lock from the server and ultimately change the status by posting back to the server when the button is pushed consequently updating the status of the lock on the server. By updating the status of the lock, the other two components of the project can fetch that status from the server and execute the proper command. Once the remote-control sends that signal to the door module, the PCB will execute the command and will change the state of the servo motor as well as the LED light.

For security reason, LED lights are added to the door module which indicates the whether the servo motor is turn locked or unlocked. There are two LED for the WIFI module— a yellow LED indicated whenever it is currently connecting to the sever and a green LED to indicate when it successfully connects to the server. Lastly, a lit red LED means that the door is unlock and turns off when the door is lock. The third component of the project is an android mobile application that can change the status of the lock and displays the current status of the lock as long as the smartphone is connected to a working WIFI or a working internet connection. It is written in JAVA language and developed in Android Studio. Before anything, the user will have to register their account, once all the informations are sent to the server and is valid, the account will be added to the database. Afterwards, the user can login to the mobile application using the registers email address and password. To avoid getting errors from misspelled password, a checkbox is available that can show the password to the user. Password can be easily viewed of hide by checking or unchecking the "Show Password" checkbox.

Lastly, the components that allows the other three components previously mentioned is the server. The server handles the authentications and adding account to the database. To restrict other users online from freely posting unsecured data to the server, a token is used. The token must be included whenever the mobile application post to the server. The encoded token will be sent to the application as they successfully login. Once, inside the application the token is then decoded to get the users

information essential to getting more information about users. Then, those information are used to post on the server.

VIII. SmartLock Team

Greg Mueth is a 28-year old United States Marine Corps veteran and a Computer Engineering student at the University of Central Florida with an expected graduation date of Fall 2018. He currently works for a tech startup creating RESTful APIs to be the backend of mobile applications. After graduation, Greg plans to be working in the Orlando area in software engineering.

Damo Park is a 26-year old Electrical Engineering Senior at University of Central Florida. He will be graduating in Fall 2018 with Bachelor's Degree of Electrical Engineering. He is pursuing to obtain Professional Engineering License after graduation. Also, working with company in Electrical Engineering field.

Mhelith Natavio is a 23-year old Computer Engineering Senior at University of Central Florida. She will be graduating in August 2018 with Bachelor's Degree of Computer Engineering. Her goal is to work for a company that can enhance her skill coding skills.

Acknowledgement

The authors wish to acknowledge the assistance and support of Dr. Samuel Richie, Dr. Lei Wei and the University of Central Florida.

References

[1] Atmel.(2009,Feb.)8-bit Microcontroller with 4/8/16/32K Bytes In-System Programmable Flash[Online] Avilable: https://www.sparkfun.com/datasheets/Components/SMD/ATMega328.pdf [Accessed Nov.27,2018]

[2] Texas Instruments(2018)Low-Power Sub-1 GHz RF Transceiver[Online] Avilable: http://www.ti.com/lit/ds/symlink/cc1101.pdf [Accessed Nov.27,2018]

[3] TowerPro.MG995 High Speed Metal Gear Dual Ball Bearing Servo[Online] Avilable: https://www.electronicoscaldas.com/datasheet/MG995_Tower-Pro.pdf [Accessed Nov.27,2018]espressif .(2018)

[4] ESP32 Series Datasheet.[Online] Avilable: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf [Accessed Nov.27,2018]