Keyless Entry

(First Draft)

Don Vo, Justin Couch, Ethan Ahrens and Kevin Rhu

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450 (12 point Times font)

***Abstract* — This project involves the use of modern technology to bring ease and comfortability to entering one's home. This is known as a smart lock and similarly to other products in the market it will have its own unique features. The goal is to provide security and convenience in entering and exiting the home at a low cost using either Bluetooth, fingerprint, RFID, or a PIN keypad. In this paper we will be showcasing how these components will unlock the smart lock and reach the desired goal using a mix of hardware and software to construct an efficient project.**

***Index Terms* — Smart Lock, PIN Keypad, RFID, Fingerprint Sensor, Bluetooth, Wireless, Phone Application, Microcontroller.**

I. Introduction

Every day home security is on the mind of homeowners that want to keep their family and property safe from intruders. The need for these homeowners to have peace of mind has given way to a new form of security that has more versatility and adaptability for security and forms of entry. This form of security is known as the Smart-Lock and is manufactured by multiple different companies. These locks have proven benefits such as greater security and more versatility. Although the physical security mainly comes in the form of a deadbolt, its other features of access and monitoring through the application gives the product a greater security experience than basic locks. The user can have a way of observing who enters and leaves the establishment as well as a way of monitoring attempts that could potentially bring uneasiness.

Keyless Entry is a Smart Lock system that consists of the four main features discussed within the Abstract. The microcontroller within the casing of the device will handle the bulk of the processing. This will be connected to the WiFi chip, RFID sensor, fingerprint sensor, Keypad and Bluetooth chip. There will also be a servo motor connected to the microcontroller and it will receive a signal to turn on whenever an unlocking mechanism is used. This servo motor is to be connected to the door lock. The phone application will receive signals through the WiFi chip whenever one of the components are used. This will be a notification system to notify a user when the Smart Lock is being used and it adds security to the system.

II. System Components

Our overall system is divided into several components that individually run their own tasks when selected in order to operate the overall device. The device will be based upon interrupts and run all of the additional components simultaneously.

A. Microcontroller

The Microcontroller (MCU) will be the integrated circuit that controls all of the additional components within our system. These components can be considered the peripherals of the project. This device will be handling the bulk of the processing throughout our system. Since it is a fair amount of simple tasks, we only require one Microcontroller to handle the processing and distribution of information for our project. It can be observed in the block diagram below in Fig. 1 the flow of both data and power within this system.

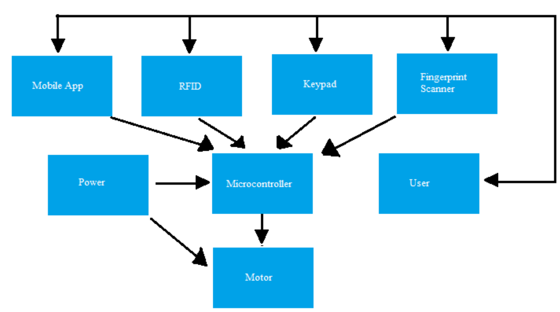


Fig. 1. Overall System Flowchart

The microcontroller chip we will be using is based off the ATMEGA chip. It's processor is capable of handling all of the tasks that will be required to be performed. Through testing, it has been noted that the microcontroller was able to operate all of the unlocking mechanisms at the same time, but this caused the processing to slow down significantly. The chip is also able to choose certain tasks based on choice commands being sent to it, which was a much faster way of processing the data. It is also expected to have our MCU receive tasks from the iPhone app and to perform the tasks that are requested. This component will be what handles the coding to message the servo motor to unlock the door whenever it's necessary.

B. Fingerprint Sensor

We will be using a capacitive fingerprint sensor to allow access with the Smart Lock in our design. This is the type of fingerprint sensor chosen due to the fact that it has better security and durability than the optical fingerprint sensor. This device is based on a set of small capacitors that record the ridges of the fingerprint to create a pixel image that cannot be copied. The optical sensor can easily be surpassed with an image of the original fingerprint, which would be a compromise in the security of the device. This sensor is able to save previous fingerprints, add new ones and delete undesired fingerprints as well from the system.

C. RFID Sensor

The RFID sensor will function similarly to how hotel room doors function, except the sound notification will not be included within our design. The sensor will be located on the exterior of the door alongside the fingerprint sensor and will be operable through the phone application. This will have the ability to accept the valid RFID card and deny any cards that are not registered within the overall system.

D. Keypad

The keypad will be a wired component that is also attached to the exterior of the overall Smart Lock system. The microcontroller will have certain pins saved to allow certain users access, and it will also have the ability to delete or add new users to allow flexibility of entry to the user's home. This is a common and efficient feature when it comes to overall Smart Lock designs and it is extremely desirable to have for our project.

E. Bluetooth Chip

The Bluetooth chip will allow the user to have additional access of entry using their phone. With this method of entry they can enter their home with a simple swipe. This allows entry from a considerable distance from the doorway, and due to the fact that we do not need a large range for this project made it easier for us to find a fairly priced Bluetooth chip. This acts as another peripheral for the microcontroller and is simply used as another option of entry for the Smart Lock device.

F. Servo Motor

The servo motor is what will be used to act as the unlocking mechanism for the project. They're extremely versatile and compact, which is perfect for our design due to our size constraint. Servo motors also output a high torque at a very power efficient rate. Whenever one of the peripherals are used to unlock the system, the Servo motor will receive a signal to turn it on and it will rotate in the desired direction. This can work to either lock or unlock the door depending on the task that is requested from the microcontroller.

G. WiFi Chip

The WiFi chip is used to offer a connection between the Smart Lock device and the phone application. This will offer the communication required to notify the user whenever a lock mechanism is used, as well as whichever one it is. It's low powered and it acts as an addition to the microcontroller rather than its own peripheral.

H. Phone Application

The phone application will mainly operate as a notification system for the overall project. This will receive communication through the microcontroller's WiFi chip and receive notifications and send out commands to the microcontroller. These commands include being able to unlock the door and picking one of several options to gain access.

III. HARDWARE DESIGN DETAILS

The overall structure of the hardware of this design involves having the microcontroller in the center so it can efficiently control all of the peripherals in the project. This includes the RFID, Keypad, Fingerprint Scanner, Motor and even the power supply. It will have connector pins to properly distribute the information and power throughout the design. It can be observed in the circuit design below in Fig. 2 the connection of the microcontroller and all of the peripherals that are connected with it.

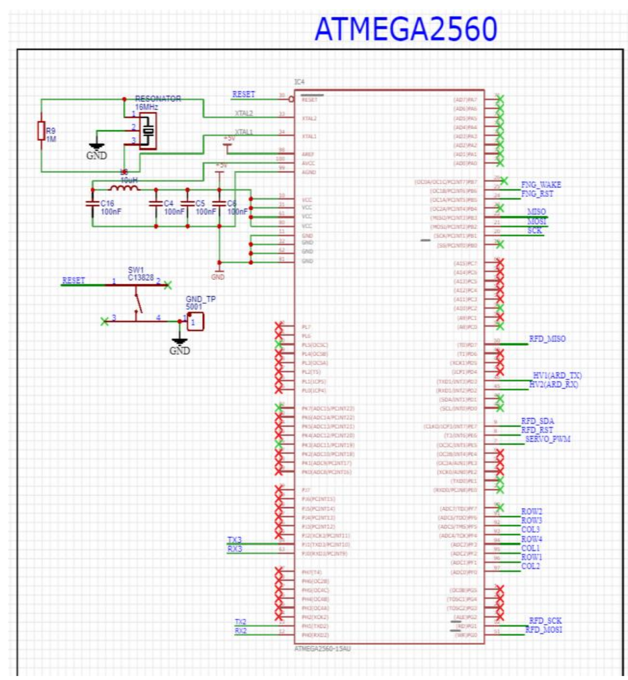


Fig. 2. Microcontroller Circuit Design

A. *Power Flow*

It can be noted that the power supply will primarily be connected to the microcontroller, but it will also supply power to the motor. This is due to the fact that the motor has a higher operating voltage than the other components of this design. There is a voltage regulator and amplifier implemented in order to properly power both the microcontroller and motor without underpowering or short circuiting either device.

The operating voltage values required to use this device vary between 1.8 and 7.2, this lets us know that we will be requiring a voltage regulator in order to properly operate this device. The voltage regulator we will be using is the MAX603 and it can give an output voltage of about 11.5 volts, which is more than enough to power any of the modules in this device. In Table 1. The voltage requirements for the components of this design can be noted.

TABLE I
SUMMARY OF CIRCUIT OPERATING VOLTAGE

Component	Minimum Operating Voltage (V)	Average Operating Voltage (V)	Maximum Operating Voltage (V)
Arduino MCU	N/A	5	N/A
Bluetooth	1.8	N/A	5.5
Servo Motor	4.8	N/A	7.2
RFID	3	N/A	5
Fingerprint Sensor	3.3	N/A	5

There will be two AA batteries used to power this system and their connection to each other will require a polarity protection device. We came to the conclusion of using two AA dry cell batteries due to the fact that since there is no liquid inside of them, they can be orientated in any way possible which allows to have flexibility in the mounting of our product. We will also be using non-rechargeable batteries due to the fact that rechargeable batteries may not have the capacity to handle supplying the necessary voltage for this project.

We used a reference design by Texas Instruments called "Battery Powered Smart Lock Reference Design with Cloud Connectivity Using Simplelink™ Wi-Fi®," and it estimates that their system can use four AA batteries to enable 5 plus years of battery life for their system. This was calculated by first calculated the total average system power and then calculating the energy capacity of the batteries. The following Equation 1. was used to find these calculations.

$$\text{Battery Life}_{yrs} = \frac{\text{Energy Capacity of Batteries}(\text{mWh})}{\text{Average System Power}(\text{mW})} \times \frac{1\ \text{day}}{24\ \text{hrs}} \times \frac{1\ \text{year}}{365\ \text{days}} \quad (1)$$

We are using a diode to control the current that is flowing through the circuit. The diode will allow for proper current flow which will keep our components from breaking. This is done through the use of forward-bias where there is a small voltage drop and the diode ends up controlling the current of the design. If the batteries are installed incorrectly, the diode also adds additional protection. This is because the current will pass through the diode and end up shorting it. The diode has a breakdown voltage that is high enough to keep the rest of the components in the design safe through polarity protection. Fig 2. below shows how the system is wired for power using a voltage regulator and a diode.

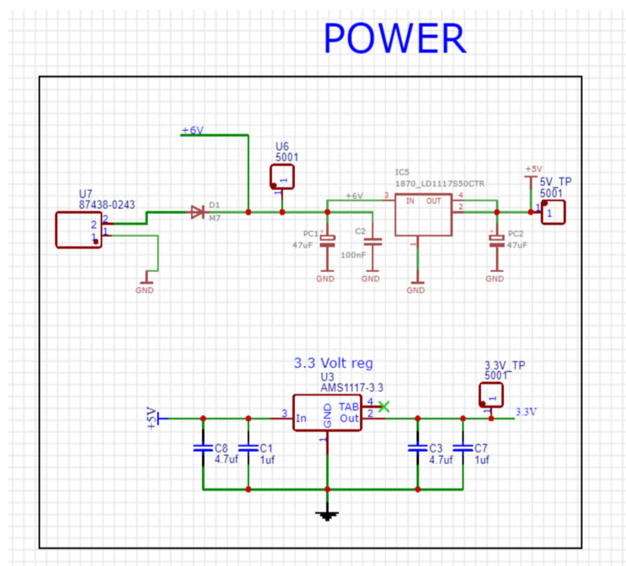


Fig. 2. Circuit Power Design

B. *Servo Motor*

The motor functionality of this device will simply be that due to the connections to both the microcontroller and the power supply, the motor can partially operate on its own. The function is for the motor to only operate when the microcontroller sends it the signal to unlock or lock the door. The motor will rotate approximately 90 degrees clockwise to unlock the door, and 90 degrees counterclockwise to lock the door The microcontroller will send the signal to the motor to tell it which direction to turn to. The microcontroller will also have a timer set that will begin after the door is unlocked and while it is closed.

When the timer reaches about 30 seconds it will send a lock signal to the motor.

The connection for the servo motor involves a simple PWM connection to the microcontroller and a capacitor with the power. We focused on having a 6 V connection in order to power or servo motor, this is a value that we need to reach in order for the servo motor to output the amount of torque necessary in order to unlock a door. If the voltage value ends up being too low, the motor will be unable to perform the tasks required. In Fig. 3. below we can see the connections required for the servo motor and its relation to the microcontroller's schematic.

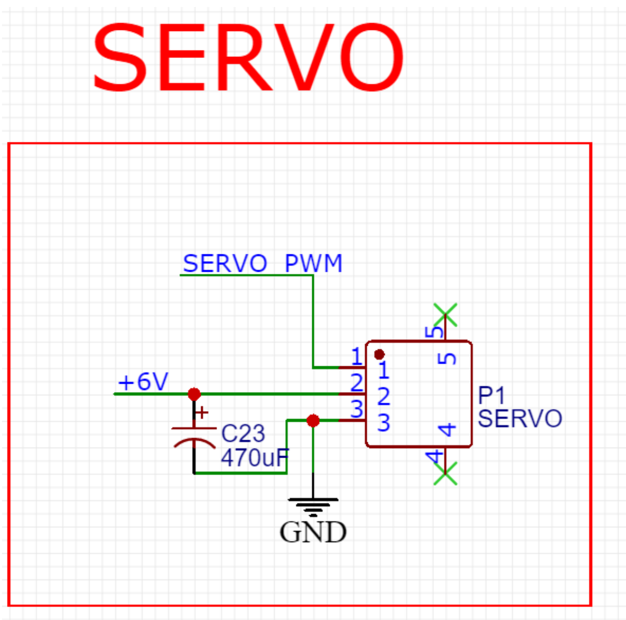


Fig. 5. Connections for the Servo Motor.

C. *RFID Sensor*

When it comes to the regular connections of this project, they are fairly simple. They're connected through pins and are noted in the microcontroller schematic. The main focus for these components is that they receive the voltage required to operate. This is due to the fact that some devices end up becoming corrupt due to a lack of voltage. The RFID is an example of a sensor that if the voltage isn't maintained between 3 and 5 volts, the programming of the scanner will become corrupted. The following image in Fig. 4. shows the other peripherals and their connections.

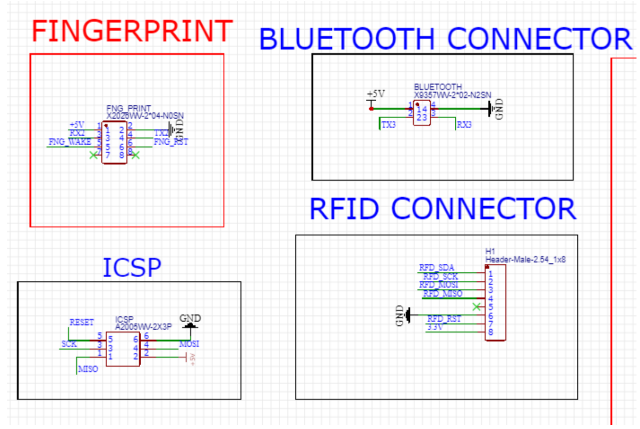


Fig. 4. Connections for the Fingerprint Sensor, Bluetooth Connector, ICSP, and RFID Connector.

D. *Fingerprint Sensor*

For the capacitive fingerprint sensor it was a straight forward connection just like the RFID sensor. The component calls for UART connections, which we have for our system due to using an Arduino based microcontroller. The images will be received, stored and evaluated through these pins and will serve as the main source of communication between the microcontroller and the fingerprint scanner.

E. *ICSP & Bluetooth Connector*

The connections for the ICSP is an SPI connection and it is based on the master/slave connections. This offers a direct connection and control over the system. This is used as an extension to the microcontroller. This is the in system programming and it will be where the code for the microcontroller will be stored. This is what will allow our microcontroller to operate and perform with all of the connected peripherals.

The connections for the Bluetooth connector are simple TX and RX connections, which is similar to the way our fingerprint sensor is connected. This allows us to store and deliver information. The storage is an important factor because it allows us to save the phone that is being connected through Bluetooth to operate the unlock feature. As noted in Fig. 3. this component requires 5V to operate which is similar to the other components that we will be using such as the fingerprint sensor.

F. Keypad

The keypad connection of the system may be the simplest of the connections. All of the pins will be going into the ADC section of the microcontroller and it doesn't even require an operating voltage. This device acts as a resistor, so no operating current or voltage will be required to use this component. This will also lighten the load on the power supply. In Fig. 5. below the connections for the keypad can be noted as well as the connection with the overall microcontroller.

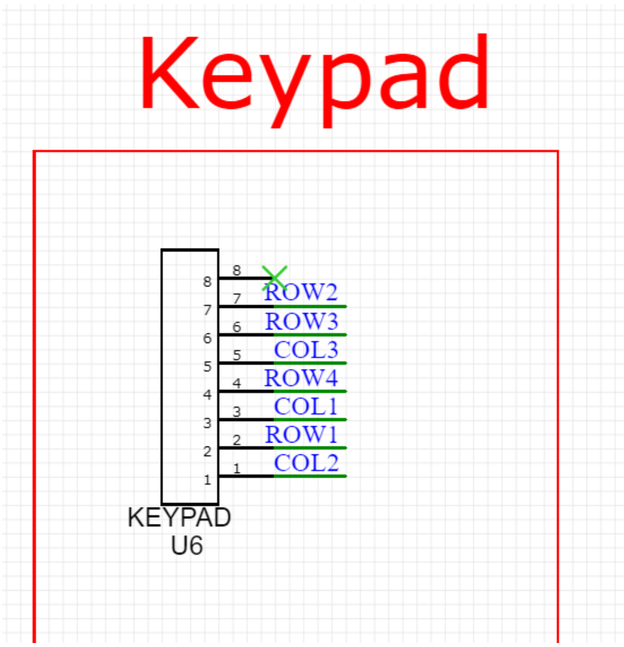


Fig. 5. Connections for the Keypad.

G. WiFi Connection

The connection for the WiFi was the most complex of the connections we needed to make for the project. This requires two logic level shifter connections and UART connections in order to operate. There is one Logic Level Shifter connected to each UART connection and this module still requires a 3.3 voltage connection in order to operate as intended. In Fig. 6. below the connections for the WiFi can be easily noted.

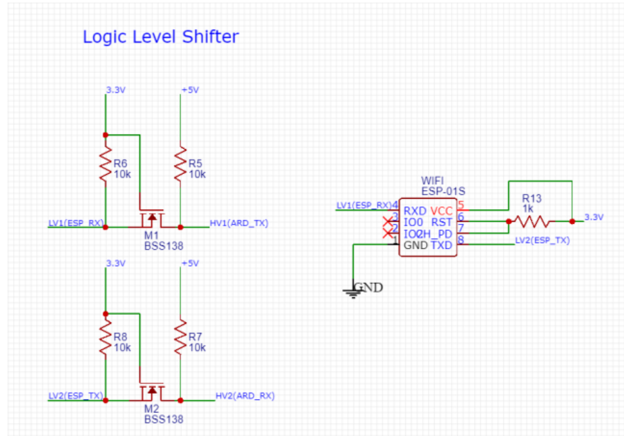


Fig. 6. Connections for the WiFi module.

IV. Software Design Details

The key goal of the software for our project is to get each of the components : fingerprint sensor, Bluetooth module, RFID sensor, keypad and WiFi module to run in parallel with the same goal of turning the lock. Each component also had their own unique Arduino library that we had to implement the code. We had to make replacements as we ran into issues while we were programming the Arduino.

The first replacement that we had to make was changing our development board from Beaglebone Black Wireless to the Arduino Mega. The reason behind this replacement was due to the fact that the WiFi module for the Beaglebone was not operating as it was intended. This caused us to choose the Arduino Mega 2560 and a separate WiFi module. We experienced the added benefit of the abundance of information on how to implement the modules that we planned on using in our smart lock.

The second replacement that we made was for our RFID sensor. The reason for this is that the DLP-RFID2, which we originally planned on using, required a special development board to program the component that would have made us go over budget. Once we realized this fact, we decided to search for a new one. This led to us finding the Mifare RC522 RFID module. The benefit of this new component was that it was compatible with our first replacement, the Arduino board. We were also able to benefit from the wealth of knowledge that is available online for programming those two components together.

The final component that we replaced was the Bluetooth module. The CY8C4128LQI-BL543 manufactured by Cypress was the Bluetooth module that we originally planned on using. The change occurred the moment we noticed that the component was a surface mount part as well as there was a lack of knowledge for us to begin programming it. Once a change was agreed upon for the component, research began again and led us to the CC2540 Serial Wireless module which is manufactured by HiLetgo. The new component, similarly to the second replacement, had a wealth of knowledge online that we could draw from to conduct our software programming.

The first step once the replacement was done was to begin creating individual programs for each of the component of our project that the users would be interacting with in order to unlock the door. The first program written was for the interaction between the Arduino and the RFID sensor. The goal of the RFID program was so that the sensor can read a tag and output the code on the screen. This test ended up being a success.

After the RFID sensor program was written we moved onto the fingerprint sensor program. This fingerprint sensor program started off with some difficulty due to the fact that source material for the fingerprint sensor we intend on using was in a completely different language. Thankfully, we were still able to troubleshoot the component to get it to function with our Arduino board. Once the troubleshooting of the component was completed, a program was written so that the sensor could save, read, and verify fingerprints.

Subsequently, individual programs were written for each of the components that used their own Arduino libraries. The program for the Bluetooth module was challenging at first, then we were able to find the solution by sending a string of code through the Bluetooth module to gain the desired output. The individual programs that we completed allowed us to integrate them all into one program that is used to turn the servo motor that is used as our locking and unlocking mechanism for the Smart Lock project.

V. Prototyping

During the prototyping of this project we had to go through trial and error when it came to the state and function of our PCB design. Through the process of testing and checking the coding for this project we had to shift through different components and manufacturers. The concept for our microcontroller began with the Beaglebone, but due to its incompatibility with other components and the level of difficulty of implementing its ICSP into a future PCB design. It was later decided to switch to the Arduino board due to how much easier it was to code, how cost effective it was, and how easy its ICSP was to incorporate to a PCB board. This involved moving from our Arduino board that we used for code testing earlier in the project to the Arduino based microcontroller that we had connected and set up in our new PCB board.

A. *Printed Circuit Board*

The final version of our printed circuit board was finally decided on after making our final choices on all of the components. To create our PCB board we used a free software called Easyeda. We went through several software until we found one that had a library with all of the components we were working with. We ordered the PCB board and also had it go through another manufacturer in order to properly incorporate our ICSP into the design and perform the functions that we require. In Fig. x. the PCB design for the top layer is shown and in Fig. x. the design for the bottom layer is also displayed.

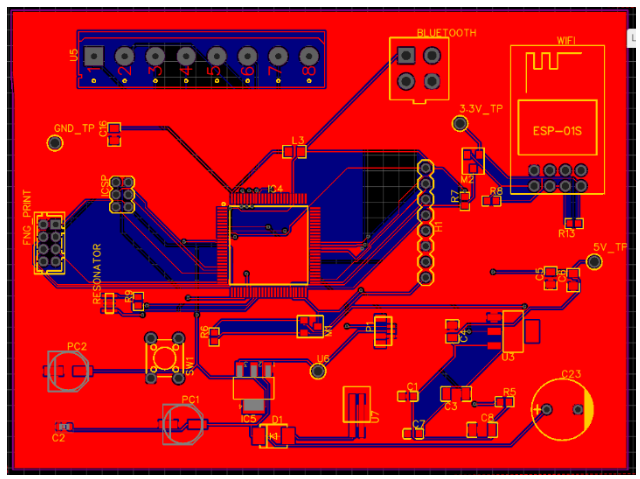


Fig. x. Top layer of the PCB Design.

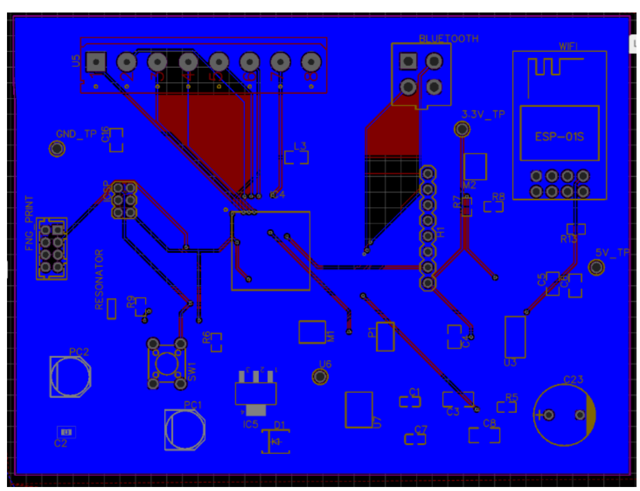


Fig. x. Bottom layer of the PCB Design.

B. Casing

The original idea for the casing of our project was to either repurpose the casting of a smart lock from a company that we researched, or 3D print our own case design. As the semester progressed and we worked on our project, we saw during development that it became apparent that neither of these options were a viable solution. This is simply due to the fact that we wouldn't be able to design a case that would fit our specific needs. We also noted that we could not properly 3D print it with the resources and skills that we had at our disposal. Repurposing the casings from other designs would have been a problem because of copyright reasons and because their cases were custom made for the company's product. We came to the conclusion that attempting to maneuver our project into the case would most likely damage it.

Once we realized that these were not plausible solutions, we discussed how we should change the design of the case and it was then suggested that we make our case out of wood. When the decision was agreed upon by everyone, we researched other affordable methods to construct and design the casing for our project. We then found a popular website that was used for wooden casing construction and we realized that this would be perfect for the prototype of our Smart Lock. We were able to obtain the file with the design for the wooden case and it was submitted for construction.

Construction of the box began once the design was completed and wood was acquired from Ace hardware. The wood chosen for the box is a 1/4 inch MDF because it is ideal for laser cutting design. The establishment where the box was cut and assembled was in UCF's innovation lab. It can be noted in Fig. x. and Fig. x. the design and final model of the casing.

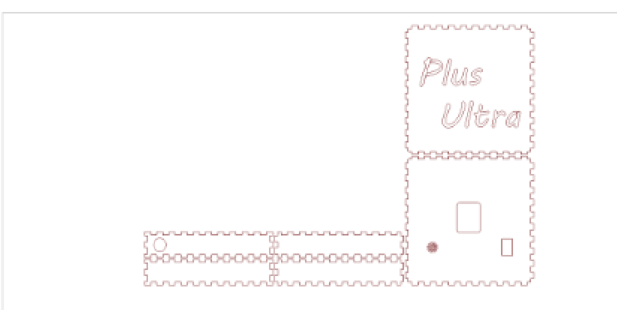


Fig. x. Bottom layer of the PCB Design

Fig. x. Bottom layer of the PCB Design

VI. References

[1] COM-14662 Sparkfun Keypad. (2018) Retrieved November 8, 2019 from https://media.digikey.com/pdf/Data%20Sheets/Sparkfun%20PDFs/COM-14662_Web.pdf

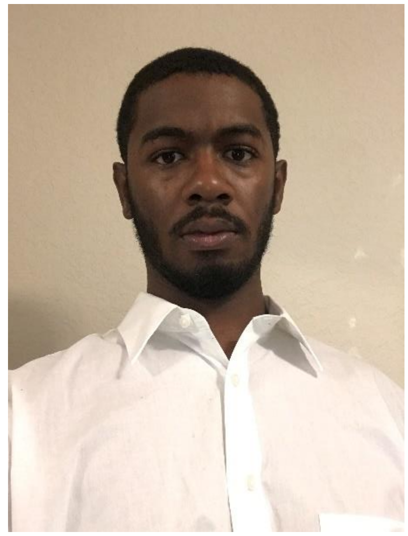
[2] Waveshare Capacitive Fingerprint Reader User Manual (2017, November 13). Retrieved from November 8, 2019 from https://www.waveshare.com/w/upload/8/82/Capacitive_Fingerprint_Reader_User_Manual_EN.pdf

[3] MAX603 Voltage Regulator (2017 April) Retrieved November 8, 2019 from https://datasheets.maximuntegrated.com/en/ds/MAX603-MAX604.pdf

[4] TI Smartlock With 4x AA Batteries (2018 June) Retrieved November 8, 2019 from http//www.ti.com/lit/ug/tidubv3c/tidubv2c.pdf

VII. THE ENGINEERING TEAM

Don Vo is a senior graduating in December 2019 with a Bachelor's degree in Computer Engineering.. He is a two-year member of the Robotics Club at UCF with a focus on sensor interfacing.



Justin Couch is a senior graduating in December 2019 with a Bachelor's degree in Electrical Engineering. He has accepted an offer to work for NAVSEA in Virginia.

Ethan Ahrens is a senior graduating in December 2019 with a Bachelor's degree in Electrical Engineering. He has accepted a job offer to work at Electrical Power Systems Inc. in Florida.

Kevin Rhu is a senior graduating in December 2019 with a Bachelor's degree in Electrical Engineering. He is currently an intern at Parsound focusing on Electro-acoustical engineering designs.