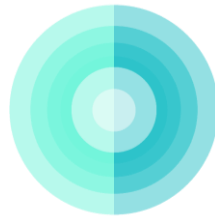




College of Engineering and Computer Science

Senior Design II

Dr. Samuel Richie



Smart Lock

Group A

Damo Park - Electrical Engineering

Gregory Mueth - Computer Engineering

Mhelith Natavio - Computer Engineering

Table of Contents

1.0 Executive Summary.....	1
2.0 Project Description.....	2
2.1 Motivation and Goals.....	2
2.2 Marketing Goals.....	4
2.3 Requirements and Specifications.....	4
2.3.1 Keyless Entry System Specifications	4
2.3.2 Wi-Fi Integration System Specifications	5
2.3.3 Mechanical Specifications.....	7
2.3.4 Indicator Specifications.....	8
2.4 Block Diagram	9
2.4.1 Main Board Diagram	9
2.4.2 Key Fob Diagram	11
2.5 House of Quality	12
3.0 Research.....	13
3.1 Existing Product and Designs	13
3.2 Relevant Technologies	16
4.0 Related Standards.....	19
4.1 Standards	19
4.2 Design Impact of Standards	21
5.0 Design Constraints	25
5.1 Economic Constraints	25
5.2 Time Constraints	25
5.3 Environmental Constraints	26
5.4 Social Constraints.....	26
5.5 Political Constraints	26
5.6 Ethical Constraints	27
5.7 Health Constraints	27
5.8 Safety Constraints	28
5.9 Manufacture Constraints	29
5.10 Sustainability Constraints	30
6.0 Comparison and Design	31

6.1 Keyless Entry System Door Module	31
6.1.1 Key Module	33
6.2 Wi-Fi Integration System.....	36
6.2.1 Wi-Fi Module.....	37
6.2.2 Server Design	40
6.2.3 Mobile Application.....	45
6.3 Mechanics and Power Supply	48
6.3.1 Servo Motor	49
6.3.2 Power Supply	49
6.3.3 Lock Description.....	50
6.4 Status Indicators	51
6.4.1 Speaker System.....	52
6.4.2 LED Indicators	58
6.4.3 Door Jamb Sensor	60
6.5 Power supply and Distribution.....	62
6.6 Microcontroller	63
6.6.1 Peripheral Features.....	65
6.6.2. Microcontroller Schematic.....	66
7.0 Prototype Construction.....	69
7.1 Parts Acquisition	70
7.2 PCB Vendor and Assembly	71
7.2.1 Comparison of PCB Software	71
7.2.2 PCB Layout	72
7.3 Coding Plan	72
7.3.1 Microcontroller Programming Plan	73
7.3.2 Mobile Application.....	76
7.3.3 Server Coding Plan	79
8.0 Prototype Testing.....	82
8.1 Keyless Entry Testing.....	82
8.2 Wi-Fi Integration Testing.....	83
8.2.1 Mobile Application.....	84
8.2.2 Application Programming Interface (API)	86
8.2.3 Microcontroller	86

8.2.4 Security	87
8.2.5 Stress tests	87
8.3 Input and Output Devices Testing	88
8.3.1 LED Light test	88
8.3.2 Door Jamb Sensor test	88
8.3.3 Servomotor testing	89
8.3.4 Speaker System	89
9.0 Administrative Content.....	91
9.1 Milestones.....	91
9.2 Budget and Finance	94
References	A
Copyrights Permission	B

List of Figures

Figure 1 Main Board Diagram 10

Figure 2 Keyless Entry System 10

Figure 3 Revised Main Board Diagram (Left) and Key Entry System (Right)..... 11

Figure 4 Original Key Fob Diagram (Left) and Revised Key Fob Diagram (Right) 11

Figure 5 House of Quality comparing relationships between marketing requirements and engineering requirements. 12

Figure 6 August Smart Lock Exterior Design [3]..... 17

Figure 7 2018 Toyota RAV4 Smart Key 18

Figure 8 Common Lock Types: What can and cannot work with the system [4] 22

Figure 10 350MHz RF receiver and transmitters. 33

Figure 11 Schematic of Key Module 36

Figure 12 The ESP8266 802.11n module 37

Figure 13 Schematic for ESP8266 802.11 module [7] 38

Figure 14: Chart illustrating the programmatic and hardware layout of the LAMP stack [8] 41

Figure 15 an Entity Relationship diagram for the MySQL database to be stored on the server. 45

Figure 16 *(from left to right) is the LogIn, Register and the Main pages of the Android Application..... 47*

Figure 17 Image lock Kwikset lock to be retrofitted 50

Figure 18 Top view of lock to be retrofitted 51

Figure 19 Schematic Diagram of Tone Generator 54

Figure 20 Schematic of RC4589ID 56

Figure 21 Schematic of Audio Amplifier..... 57

Figure 22 Schematic of Door Jamb Sensor 62

Figure 23 Schematic of Power Supply 63

Figure 24 Original Schematic of Microcontroller (Above) and New Schematic Design (Below) 67

Figure 25 Key PCB (Left) and Door PCB (Right) 72

Figure 29 Visualization of user Oauth2 authentication flow. 79

Figure 30 Keyless Entry Testing Flow Chart..... 83

Figure 32 Breakdown of current android operation system statistics. [13] 84

Figure 33 flow chart of the Android application lifecycle. 85

Figure 34 Gantt chart of project milestones. 93

List of Tables

Table 1 Keyless Entry System Requirements	5
Table 2 Lock Hardware and Microcontroller Requirements	8
Table 3 Input/ Output Device Requirements	9
Table 4 August Locks Comparison [1]	14
Table 5 Schlage three locks comparison [2]	15
Table 6 AS 3933 Parameter [5]	35
Table 7 CC 1101 Parameter [6]	35
Table 8 Specifications of Tone Generator.....	53
Table 9 Calculated Frequency with Different Value of Resistor	55
Table 10 Specifications of RC 4580ID and LM 386	57
Table 11 Requirement for LM 386 Amplifier	58
Table 12 LED Lights Specifications Comparison	59
Table 13 Parameter LED lights Will be Used	60
Table 14 Voltage Regulators Specifications	63
Table 15 Comparison Between ATmega328 and MSP430G253	65
Table 16 Required Parts for the Main Microcontroller	70
Table 17 Required Parts of the Physical Locking Mechanism and its Housing	70
Table 18 Required Parts for the Key Fob System.	70
Table 19 Miscellaneous Parts and Equipment Needed for Programming Parts and Running a Self-Hosted Server	71
Table 20 Overall Budget and Finance	95
Table 21 Budget and Finance	95

1.0 Executive Summary

Technology is continually advancing. We as a civilization are constantly creating ways that may improve any aspects of our lives. Electronic devices such as phones, computers, and televisions are some of the few technologies that have considerably changed 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 carry is 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. Our SmartLock will be capable of communicating through a local area network to our self-hosted server. Communicating to a remote web-hosted server then allows us to create a mobile application that will be able to interact with a home door lock from anywhere in the world, so long as you have an internet connection. This is certainly a product that already exists, however, products on the market are prohibitively expensive, which has led to slow adoption of the technology. 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.

The second key feature of our SmartLock is keyless entry. 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. When the user interacts with a triggering device on the exterior of the lock such as a button, the microcontroller inside the lock will send a low frequency wake-up signal to the key fob in the user’s pocket. That key fob then sends a high frequency signal containing the correct key to the door lock, telling it to unlock. Purchasing keyless entry system that operates in a similar was 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 to the smart locks on the market, we hope to prove they can be convenient and affordable for all.

2.0 Project Description

In this section will cover our motivation and goals, marketing goals, requirements and specifications, block diagram, and house of quality. Motivation and goals section explains that how we decided to do the project SmartLock. Requirements and specifications section explains that the requirement of each system or parts. Each module, parts, or microcontroller needs to meet the requirements, so then the SmartLock will work property. Diagram section explains how the module, I/O device, sensor connected. As it shows diagram, it will be able to understand easily. House of quality section shows how the marketing goals related with the engineering goal in one picture.

2.1 Motivation and Goals

Those times where one has to rummage through their backpack or purse, misplacing keys, unidentifiable keys due to owning multiple keys that looks exactly the same or just having both hands occupied with something that one cannot physically unlock or lock their door are the motivations of this project. Nowadays, many offices as well as some private homes uses some type of smart lock. A few uses buttons to input their chosen passcode, some use a key card to either swipe or insert, another type of smart lock uses a low frequency Radio Frequency Identification (RFID) that needs to make direct contact to a sensor to unlock the door. On newer model automobiles, a form of RFID keyless entry is being implemented where the user simply leaves the key in their pocket/purse, interacts with the car in some way, and the car recognizes the key is in close proximity with it and knows to unlock the door. That is the easy to use, modern form of unlocking doors we hope to bring to home/apartment doors. Some companies have already taken a stab at creating their own smart locks, but none with the exact features we plan to implement and all are very expensive. Although this project is not something new, our group will produce a similar product that not only works but will be affordable compare to those products that are in the market today.

In addition to the annoyance of operating physical keys, another common problem with traditional, mechanical locks is that question that sits in the back of everyone's mind at least once or twice a week as they tiredly stumble to their car and drive to work in the morning. "Did I leave the door unlocked?" This constant nagging question can be mitigated in two ways. First, a simple timer that automatically triggers the door to lock itself after a certain amount of time. It's a simple solution that some users may like, and therefore makes absolute sense to implement. However, other, more forgetful users may lock themselves out. As such, it must be an optional feature, able to be switched on and off. The second solution is to interface the door lock with a mobile application that allows the user to view and toggle the locked/unlocked status of their door lock from any location through the internet.

The main goal of this project is to give the user a more convenient and secure way of unlocking and locking their door. Since there are already numerous products on the market that do the same job, we want to create a more affordable version of those. By using cheap yet reliable alternative products we believe that we can successfully carry out our goal. Each member will apply the engineering principles and values throughout the entire time designing to produce the best results.

The three main systems for this project are the: Keyless Entry System, Wifi/Mobile Application Interface and Physical Door System itself (indicators, mechanical components, etc). There are to be three ways of unlocking the door: physically pushing an exterior button with the key fob in range, using a toggle switch included in the mobile application, or by using the backup physical key. Our design will center around retrofitting an existing, traditional lock (the type that can be acquired from any hardware store) with a servo motor and wifi/RFID capable microcontroller.

The Keyless Entry System will have a button that when it is pushed will send a low frequency signal to the key fob to bring it out of low power mode, which then will send a high frequency, encrypted unlock passcode to the microcontroller. Then, the microcontroller will send a signal to the servo motor unlocking the door. Finally, once the door is successfully unlocked the door system will play a tone and the LED light on it will turn to GREEN.

Alternatively, the user will have the option to unlock the door from any distance through the internet by using the mobile application. The mobile application will have a home screen that displays the status of all locks associated with a particular user account. In addition, there will be a toggle switch for each lock allowing the user to change the status. Current statuses of each lock will be stored on a server that is programmed and hosted by this design team. When an instruction to change lock status is received by the server from the mobile application, the server will then push the instruction to the physical lock's microcontroller, which then drives the servo motor and changes status LEDs.

There will be multiple different ways for a user to lock the lock. First, we plan to leave the physical keyed locking mechanisms in place, so simply turning the mechanical lock from the interior/exterior (with backup key) still works. Second, the previously mentioned "lock after a certain amount of time" feature is to be implemented, but able to be switched on and off (through the mobile application) as not every user would enjoy such a feature. Third, the mobile application has a toggle switch to send the lock signal through the internet and our server. And lastly, pressing the exterior button while the door is in the unlocked state will cause the microcontroller to lock the door. Upon the door the LED status indicator will switch from green to red, the lock tone will play, and lock status will be updated on the server. A door jamb sensor is also to be included, to prevent the microcontroller from attempting to drive the servo motor while the door is open or partially closed.

2.2 Marketing Goals

- Keyless entry using a key fob (similar function to automobiles)
- User friendly mobile application both in Android and iOS
- Interface mobile application lock and unlock through Internet
- Mobile application to toggle lock between locked and unlocked
- Mobile application shows current status of lock and logs
- Optional push notifications locking action is completed
- Indicator LED lights for lock/unlock statuses
- Unique tune indicating locking/unlocking the door
- Door automatically locks on closing after certain amount of time
- Unlocks in minimum possible time after user interaction
- Door Jamb sensor for added security
- Must fit existing door frames and be easy to install
- Must not be larger than necessary or overly bulky/misshapen

2.3 Requirements and Specifications

This section exists to codify the exact engineering requirements the project must measure up to. What follows are the design specifications for the entire SmarLock project. The specifications are broken down by which module they relate most closely too, but there is some overlap between the systems. These specifications will be used to select parts that best fit requirements and will heavily influence the design of the project. All engineering requirements and specifications are derived directly from the original marketing goals, or the features the design team set out to achieve from the outset of the project.

2.3.1 Keyless Entry System Specifications

A short range, radio frequency identification unlocking system to be used as the primary door key. When a button on the exterior door lock is pressed, the lock sends a low frequency broadcast to wake up all keys in the immediate area. When a key receives the wake-up signal, it transmits its high frequency unlock signal to the door lock microcontroller. This is similar to how the keyless entry system works on most new automobiles. The target range for both key and door receivers/transmitters is just under one meters. Any shorter range may cause unreliable operation, and a further range would make the door lock ineffective, as it needs to be stored in the user's house/apartment. A long-range key would allow people besides the owner to operate the lock while the user is home.

Keyless Entry System Requirements			
	Requirement	Target	Comment
Max Range	< 2m	1m	Must be short range enough so key can be kept in house
Min Range	> .25m	.5m	Key must function when on user
High Freq Signal	>300MHz	350MHz	High enough frequency to send string with low latency
Low Freq Signal	125kHz	125kHz	Wake-up signal for key fob
Supply Voltage (main)	5V	5V	Microcontroller VCC
Supply Voltage (key)	3V	3V	Powered by coin battery
Transmitting Power	< 40mW	20mW	Regulations
Size on board (main)	6cm x 6cm	5cm x 5cm	Size goals
Size on board	3cm x 3cm	3cm x 3cm	Size goals
Latency	<1 second	1 second	Usability
Battery Life (key)	>168 hours	500 hours	Usability

Table 1 Keyless Entry System Requirements

While building the keyless entry portion of this project, the keyless entry feature was changed from being passive keyless entry. The low frequency wakeup signal was an unanticipated extremely difficult obstacle to overcome. As such, low frequency wakeup portion of the feature was removed and the button was moved to the key fob while the high frequency portion was left in place. This effectively changed the feature from keyless entry similar to new automobiles to regular radio keyed entry.

2.3.2 Wi-Fi Integration System Specifications

The lock microcontroller uses a Wi-Fi module to communicate through the internet with a server running a LAMP stack the design team will host ourselves. A mobile application will then be used to communicate with the server through the internet, allowing the user to view statuses and lock or unlock the door from anywhere they have an internet connection. Upon the server receiving a change status command from the mobile application, it will create a new table for the event (to keep a log)

and pass the instruction along to the lock microcontroller. Although the team is only constructing one lock at this time, the mobile application and server will be designed in such a way that after creation of a user account, that user's account can be associated with zero, one, or many locks. Also, one lock can have zero or many users, as multiple people generally live in the same house/apartment.

Mobile application feature description

The initial design for the mobile application is to it available in both iOS and Android devices but as more research is done, creating an iOS application is much more expensive and challenging than expected. As a result, the iOS application was omitted to the final project. With that being said, the mobile application will be used to control the SmartLock will have both android and iOS versions. It will send and receive JSON payloads to the server in order to interact with the system. From the mobile application, the user will be able to create an account, create an association between the user and a lock, and to view and change the status of locks associated with their user account. The home screen of the application will display each lock associated with the user account along with a toggle switch to interact with it. A settings menu will also be available to the user where they can toggle “lock after a certain amount of time” and “give push notifications” on or off. Instead of having push notifications, the final product will display the current status of the lock on the main screen as well as a pop up that shows the action and when that action was executed whenever the lock/unlock button was pushed.

Server feature description

The server will be self-hosted by the design team. It will be running a LAMP software package. The MySQL database will have tables for Users, Locks, User and Lock Associations, and Locking/Unlocking Events. Each time a lock or unlock instruction is routed through the server, a new row for that event will be created in order to keep logs. The server will use OAuth2 authentication for security with the mobile application. Authentication tokens will expire after 24 hours. Scripting will be done in PHP.

2.3.3 Mechanical Specifications

The physical lock is to be a regular key lock that can be bought from any hardware store. The door lock will be retrofitted with a servo motor, which will be controlled via a microcontroller. The mechanical key portion of the lock will be left intact to be used as a backup in the event that the SmartLock system fails. A switch on the door jamb will be used to determine whether the door is open or closed in order to prevent servo motor turning the lock while the door is still open or only partially closed. This will lower the chance that any hardware is damaged due to user error. Design of the physical lock will largely affect the overall size of the unit to be installed on the door, so the highest torque for the smallest size servo motor must be selected and it must be oriented and geared in such a way that it doesn't extend too far from the door, but still maintains the speed and torque required to operate the door lock in the allowable time.

Lock Hardware and Microcontroller Requirements			
	Requirement	Target	Comment
System Power Supply	4x AA Batteries (6V) (2400mAh)	4x AA Batteries (6V) (2400mAh)	Maximum amount of mAh for their size and cost
Servomotor Torque	10kg-cm	10kg-cm	Strong enough to turn consumer grade deadbolt
PCB Size	7cm x 7cm x 3cm	6cm x 6cm x 2cm	
Exterior Case Size	7.5cm x 7.5cm x 4cm	7.5cm x 7.5cm x 4cm	Marketing goal
Interior Case Size	25cm x 7.5cm x 4cm	25cm x 7.5cm x 4cm	Marketing goal
Battery Life	>168 hours	500 hours	Maximized
Servomotor Speed	>.5seconds	>.25seconds	Latency Requirement

Table 2 Lock Hardware and Microcontroller Requirements

2.3.4 Indicator Specifications

The physical door lock will have two different status indicators. First, a set of red and green LEDs to display the current status of the door lock. There will be LEDs, red for locked and green for open, on both the interior and exterior of the door lock. This will allow the user to know the system worked before they attempt to open the door for entrance as well as allow the user to tell the status of their door from across the room without having to physically check it. Second, an audible, three notes, tune will play when the door is locked or unlocked. If the door is attempted to be locked while still ajar (the door jamb sensor reads open) then a warning tone will play.

Input/ Output Device Requirements			
	Requirement	Target	Comment
Speaker Resistance	8 Ohms	8 Ohms	
Speaker Power	<.2 Watt	.2 Watt	Lowest possible power to produce desired sound
Speaker Sound	>50dB	60dB	Loud enough to be useful as an indicator

Table 3 Input/ Output Device Requirements

2.4 Block Diagram

All electric devices linked between module or sensor to communicate each other. Each module or sensor has specific function, then will be send to others, so then the device will work property. For example, as user writing with keyboard, then the letter will comes out through the screen. As the keyboard receive the input signal, the main board will convert the signal from the input signal to output signal to show the letter through the screen. In this section, Block diagram, will shows that how the system will communicate each other. It will explain how module, sensor, input/output device, or microcontroller connected to work.

2.4.1 Main Board Diagram

Main board is the biggest circuit we have for the project. There is a one microcontroller, Wifi interface, I/O device, sensor, and so on. Microcontroller is used for processing all the signal. As receiving signal from sensor, microcontroller will produce as programed. Wi-Fi module is used to communicate between server and microcontroller. If Wi-Fi module received as signal from microcontroller, it will send to the server. Then, server will communicate with mobile device. Also, there is another system which is called keyless entry system (KES). Through the key fob, the KES will receive a signal, then will send to the microcontroller to know if the key fob is closed to the door. As microcontroller receive signal, it will send signal to the speaker system to generate sound. Speaker system will be needed to generate different tone level and louder sound. The following figures will shows the main board diagram including speaker system, and keyless entry system.

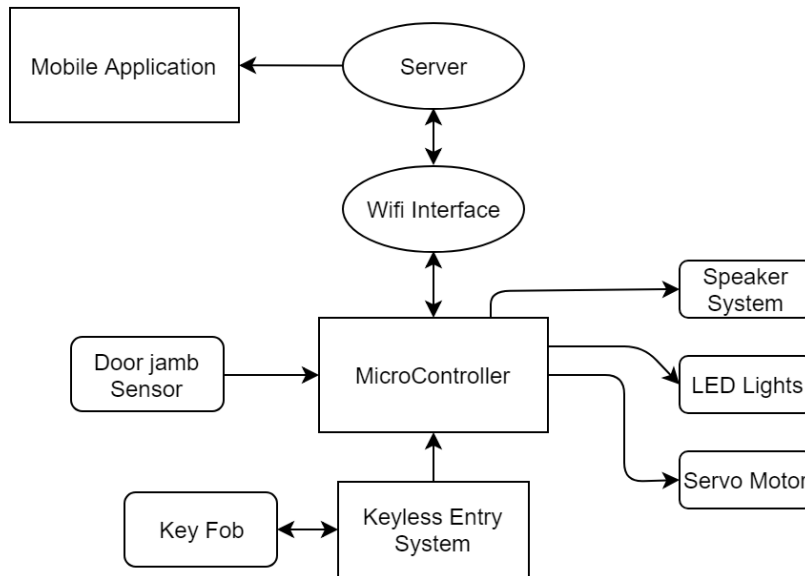


Figure 1 Main Board Diagram

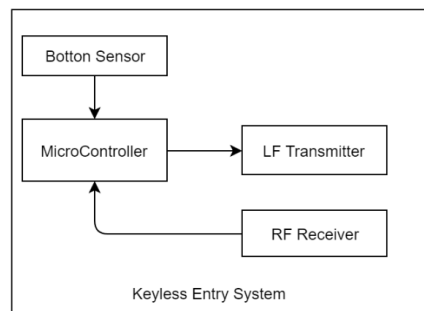


Figure 2 Keyless Entry System

In senior design 2, as we working on our project, some feature has been changed. As initial project, we were calling keyless entry system since we are using low frequency wake up signal. While we working on LFWS, we had a problem, decided to use a button on key fob instead of LFWS. Speaker has removed as well in SD2. Following figure shows new main board diagram and new key entry system below.

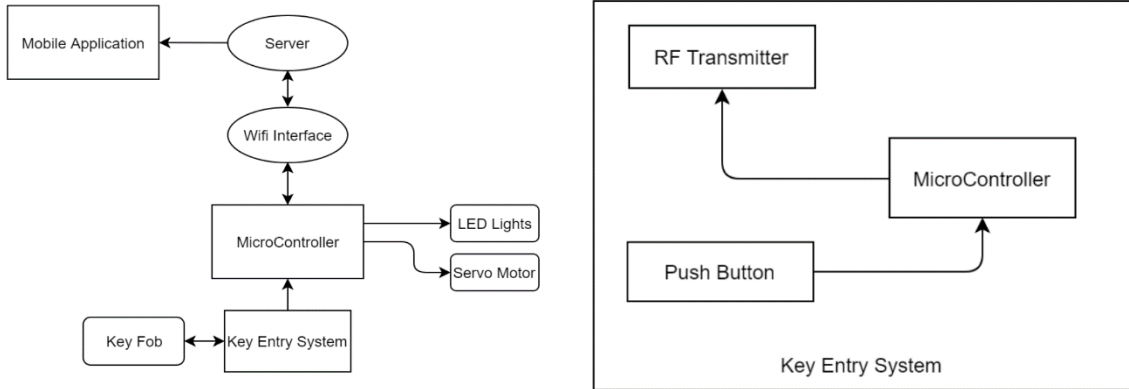


Figure 3 Revised Main Board Diagram (Left) and Key Entry System (Right)

2.4.2 Key Fob Diagram

In this section, key fob diagram, will shows that how the devices are connected each other. Low frequency receiver and microcontroller are connected because the receiver needs to tell microcontroller to send back a signal as RFID. Microcontroller and radio frequency transmitter are connected, so then the signal will be able to send back to the door module. The following figure will shows the key fob diagram.

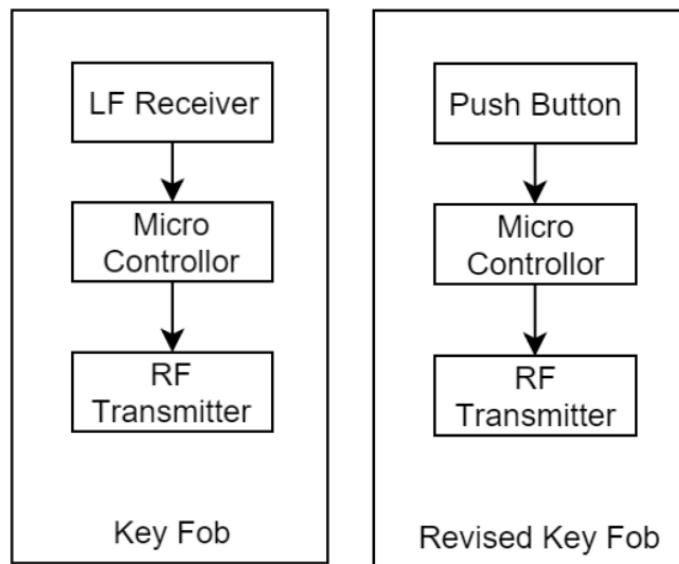


Figure 4 Original Key Fob Diagram (Left) and Revised Key Fob Diagram (Right)

Since the low frequency wake up signal did not work on during the senior design 2, we decided to use push button to send radio frequency to the door module. When the user pushed the button, the microcontroller in the key fob will send radio frequency signal to the door module, the door will be locked or unlocked. In the figure 4 shows the new designed key fob diagram above.

2.5 House of Quality

The house of Quality shown in Figure 5 demonstrates the relationships between features and engineering requirements and the relationships between engineering requirements. Key features and engineering requirements the team desires to implement are all known at this time and present in the House of Quality. If additional requirements become obvious after the design phase has begun, they will be added.

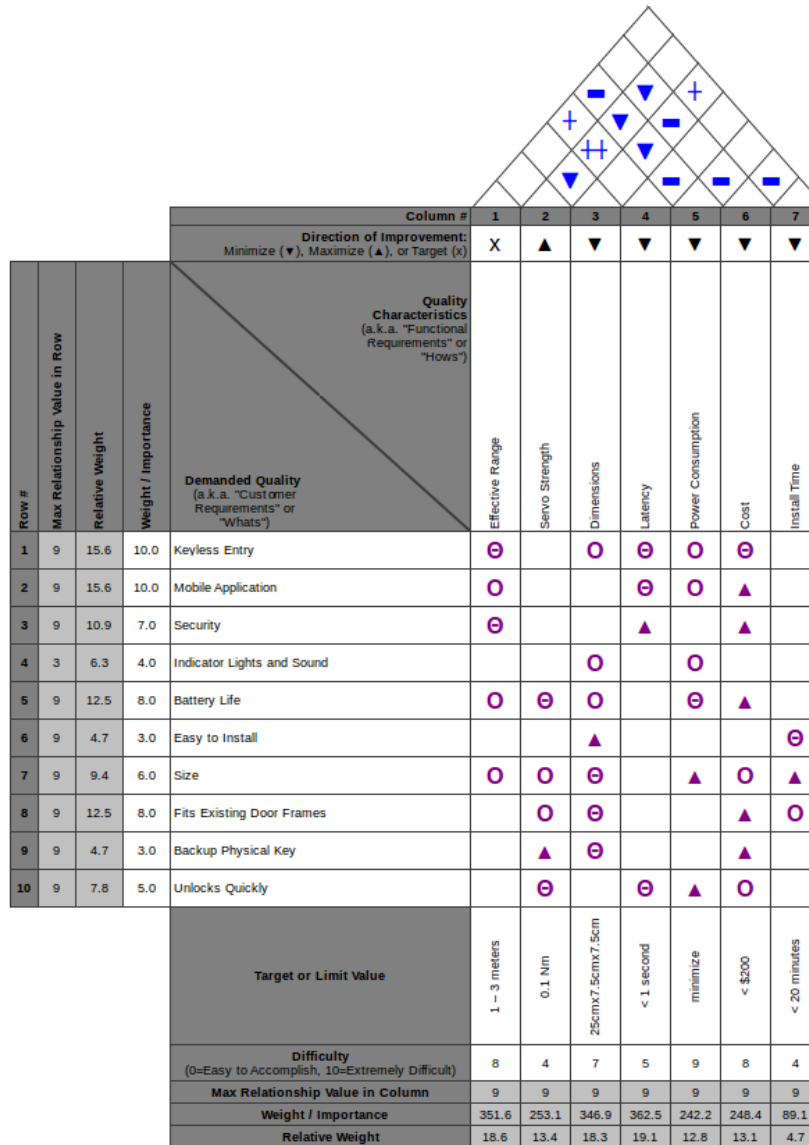


Figure 5 House of Quality comparing relationships between marketing requirements and engineering requirements.

3.0 Research

Before any planning and designing the SmartLock system, researching about every feature of it must be done. This section talks about the existing products, designs, and relevant technologies. The products that are listed on this section influenced the design of the SmartLock system.

3.1 Existing Product and Designs

Smart Lock may not be as sought-after product by private homeowner today, but it is definitely not uncommon as one may think. More and more private home owners are starting to get on this new technology and installing some type of Smart Lock in their home. With how fast our technology is advancing, it is not far to have these types of locks to spread on the market and eventually caught attentions of home owners or even apartment owners. Some of the well-known Smart Lock brand that is out in the market are: August Smart Lock, Schlage, Danalock, Kwikset Kevo, and Yale Assure Lock.

After much research and reading what the consumers says about most Smart Locks, it is not yet a perfect technology. First, most systems cost at least a hundred dollar. The high-end system cost a lot of money and still fail most of their consumers with their products' performance. Second, buying a cheaper system will result in lower functionality. Meaning, cost can affect the functionality; the pricier the system the better performance quality. With these in mind, the goal of the project is to have a reliable working system that cost less. By using cheaper alternative materials without compromising the quality are going to be used to build the system.

Moreover, most of the Smart Lock offers mobile application for users to download and is going to act as a smart key that connects to the actual system. However, the reviews about some of the applications are not outstanding either. Some of the applications is limited to only one software may it be Android or iOS device. Taking notes of these drawbacks that consumers have to deal with these systems inspired one of goal of the of the project— a mobile application that will be available for both an iOS and Android device. A user-friendly application is the goal which means that the application will have simple options pertaining to the tasks that the system can do and an easy to the eyes fonts, font size, and colors. Both application will be in English and can later be updated to have different language giving more accessibility for other consumers that are not English speakers.

To be on par with the existing products when it comes to energy consumption, the goal is to have the overall door system on 4 AA batteries in series totaling to 6V. Additionally, by putting the door system on standby mode whenever it is not in used, energy will be conserve which can result to batteries not being replaced all the time. For the systems to communicate to one another, the main door lock must

be “awake” and can be done by pushing a button located at the exterior body of the door lock. Once that button is pushed the microcontroller will now be sending frequency signal and will be waiting for the key fob to receive that signal. Then the key fob will send back a signal to the microcontroller which will send the command to the microcontroller to either lock or unlock the door.

Table 4 shows the comparison of the two types August Locks. Both systems cost at least a hundred dollars to purchased. The main differences the locks have are shown in the table as well.

Compare August Locks



	 August Smart Lock Pro + Connect <i>The most advanced smart lock.</i>	 August Smart Lock <i>The perfect low-cost smart lock</i>
Compatible Standards	Bluetooth® Wi-Fi™ included Apple HomeKit™ Z-Wave™ Plus	Bluetooth® Wi-Fi ready*
Use your phone as your key	✓	✓
Auto-Unlock your door as you approach	✓	✓
Auto-Lock your door when you leave	✓	✓
DoorSense™ - know if your door is closed	✓	✓
Works with Apple's HomeKit & Siri®	✓	
Works with Amazon's Echo™ & Alexa®	✓	*
Works with Google Home™ & Assistant	✓	*
Remotely control your lock	✓	*
August Connect Wi-Fi Bridge included	✓	
Z-Wave Plus support	✓	
Price	\$279.00	\$149.00

Table 4 August Locks Comparison [1]

Another company that sells smart locks is Schlage, and 5 shows the three types of locks they have available and the advantages of each one. The difference between these locks from the August Locks is that the August Locks does not have any keys/buttons. Schlage offers code access as well as smartphone access. During planning stage, using a keypad was considered as one of the main access

but taking into account that as the users inputs their codes the keys that corresponds to their code will be evidently more worn out than the rest of the unused keys. As a result, it easier for other people to guess the right passcode.

Which Keyless lock is right for you?



	Schlage Sense ™	Schlage Connect ™	Connected Keypad
Installs in minutes with a screwdriver	✓	✓	✓
Illuminated touchscreen	✓	✓	✓
Control while away from home	✓*	✓*	✓*
Access code memory capacity	30	30	19
Built-in alarm sensors; senses potential attacks and can issue an alert	✓	✓	—
Programmable with your smartphone	✓	✓*	✓*
No incremental fees or subscriptions	✓	✓	✓
Low battery warning	✓	✓	✓
Works with Apple HomeKit™	✓	—	—
Works with Amazon Alexa	✓*	✓*	✓**

*Requires a compatible smart home hub or wi-fi adapter

Table 5 Schlage three locks comparison [2]

Similar to August Locks and Schlage locks, DanaLock V3 is another smart lock that is out in the market. The DanaLock V3 and August Locks have many similar features such as the nonexistent keys on the system, fits most single-cylinder deadbolt lock, shareable access code, and an available mobile application to control the lock. Users of this smart lock can share access-keys either through

SMS or email. Access-keys maybe shared as a permanent, recurrent, or temporary access-key. Furthermore, once the access-key is accepted the user will get notified through the mobile application. Much the same as other mobile applications, it includes a push notification whenever the lock was used and by whoever used it. In addition, DanaLock can connect to some smart home systems such as, Apple HomeKit, Z-Wave and Zigbee. All in all, DanaLock has several features that are very similar to August Smart Locks including shareable key-access, notifications about locks activity and being able to connect with various smart home systems.

Smart locks from Yale Assure Lock are very similar to Schlage as it also incorporates keypads on the system. On the other hand, Kwikset Kevo's smart lock is very similar to SmartLock system as it requires user to touch something in their case the actual lock system to unlock it. To differentiate the owner of the lock from others, the mobile application must be connected via Bluetooth connection to the lock meaning the user must be near the device for it to work. Apart from that main difference, Kwikset's smart lock is similar to the previous smart locks that was mentioned. Users can download a mobile application where they can lock or unlock their door and lastly users can receive a notification about the lock's activity.

3.2 Relevant Technologies

Combinations of Smart Locks, Key Less Car System and a simple traditional doorbell are the technologies that inspires the design of the whole system of the project. The conveniences of having a Smart Lock and key less entre will give the user an easy access to their home and by having a doorbell like button to "wake up" the system from standby mode will help conserve power that is desired to accomplished. Finally, setting a condition which is that the key fob be within the set ranged will help the user more secure from unauthorized users locking or unlocking the door which can lead to them getting inside the house.

Figure 6 shows the front exterior design of the August Smart Lock, one of the many Smart Locks in the market. This particular system can be bought from August online store for \$149.00. The difference of this design to the project's exterior design are: the LED lights, key hole and the exterior button. Apart from having small speaker installed in the interior, two LED lights will be added that will display Green LED light indicating that the door is unlock and a Red LED light for when it is lock. It is decided that the key hole will not be removed in the case that the door lock system is not available, and/or the user decided that they no longer want to use the system. Thus, prevent the users from doing extra work of removing and installing a new lock.

The final product, did not include a speaker to save more power, and LED lights were changed to red, green, and yellow. Red indicating that the door is unlock, green when the lock is currently connecting to the server and finally the yellow light to indicate if it successfully connected to the server.



Figure 6 August Smart Lock Exterior Design [3]

As mentioned before, one of the main goal is to have a system that can run on only 4 1.5V AA batteries in series totaling of 6V which means we need to have the system to cut back on power consumption as much as possible meaning only have the system broadcast frequencies when the users want or need to use it. As a result, the system is intended to always be on standby mode whenever it is not being used. To bring the system up and running a button will be installed on the exterior of the system which will need to be pushed. The button will have a similar effect as a Push-Button Start one can be seen in a latest car model, the button is pressed by the user to start and stop the engine. This certain characteristic is very similar to the exterior button that will help the system save power and can wake up the system. Once the button is pressed, the main system will start broadcasting frequency signal and waits to receive a signal from the key fob. Furthermore, since the button will be exposed in different kinds of weather it should a quality similar to a doorbell.

One of the goal from Senior Design I is to have the system only used 4 1.5V batteries this particular goal was not met since the servo motor that is used in the system was changed and takes up more power than the initial component. With that component, the power consumption went up and more batteries were added to fully power the system.

Similar to the Smart Key that a 2018 Toyota RAV4 has, which is shown in Figure 7, the user will have a key fob with them that should be within the ranged when the main door system starts broadcasting frequency signal this way the signal will be received by the key fob and can then sends another signal back to the system. By establishing that connection between the door lock system and the key fob, the microcontroller can then send command to other components connected to it such as the servo motor, LED lights and the speaker. allowing it to determine that it is in fact within in range and can send command to the servo motor to either lock or unlock the door.



Figure 7 2018 Toyota RAV4 Smart Key

The particular August Smart Lock shown in Figure 6 has a mobile application available for both iOS and Android users. Likewise, the project's system will include a mobile application also available for iOS and Android phones. For the application, it will have options/buttons similar to the one seen on the Smart Key that is shown in Figure 7. It will have a toggle button for unlocking and locking the door and an option to have push notifications about the last executed task the system did.

For the prototype design of the mobile application, instructions will be written in English. Due to time constraints and to make testing easier, additional language will not be added until further notice. Additionally, to avoid receiving mass notifications from the system whenever someone tries to spam the lock/unlock button on the mobile application, it will only save the last action after 5 seconds and send that information.

Finally, the user's phone must be connected to a working Wi-Fi for the application to work. Testing for the application will be done by having different users connected to a working Wi-Fi try to send commands to the door system from different locations. Multiple phones—iOS and Android devices, will also be used to check that both types of software are working properly and connecting to the system successfully. This will be done during the prototyping stage of the project and any changes, errors, and solutions done to the code will be documented in order to have programmers' every change monitored in case any of the mobile applications are not working properly or outputting wrong results.

4.0 Related Standards

This section is for all the related standards that the SmartLock system will use during the building. The standards influenced how the final design is going to look like and how it will be tested. Ensuring the system follow the listed standards on this section as without proper standards the system may not be able to work or communicate to one another.

4.1 Standards

Testing how each component reacts to one another will be test throughout the prototyping stage. By checking that each connection is safe only then the building will continue. If by any circumstances that the system is deemed dangerous for the members to continue working on, the prototyping will be stop immediately. Making sure that not only the future users but all the people involved during the building of the product are safe is the number one priority. Members will customize tests that are meant to fail the system, by doing so members will be able to think, research, design and create the appropriate solutions for the problem.

To have a more reliable testing standard, apart from the constructed tests, Standard Performance Evaluation Corporation (SPEC) benchmark will be used in testing the system. By verifying that each component of the system passes the benchmark will ensure that the project is up to standard and is safe for the users. Furthermore, by successfully passing each SPEC benchmark will establish the validity of the SmartLock system.

In addition, certain constructed tests will verify if the system is reliable to produce and to use by humans. IEEE's General Principle which can be found on their website, will be followed during the entire time of building the system. Violating any of the principles will not be tolerated. As a result, any decision or designs that go against the principal will be reviewed and corrected as soon as possible.

Two out of the three main systems— the main door lock system and the key fob will have to communicate by sending frequencies to one another. On the other hand, the mobile application will be communicating with the door lock system through WI-FI. It will follow the IEEE802.11 communication standard, failing to do so will result to problems in the future which will not be acceptable.

The Wi-Fi module will need to use Universal Asynchronous Receiver Transmitter (UART) to communicate to the microcontroller which will then send payloads from the server. Afterwards, sends the command to the servo motor to lock or unlock the door. On top of that, the microcontroller will send signal to the correct LED lights as well as to the speaker as it will play a unique tune according to the task being executed.

One requirement to use the mobile application is for the user to create an account. These informations are necessary to be stored somewhere which is why MySQL database will be use. Apart from user's informations, the lock and unlock events and the SmartLock's information must be stored as well. To create an account, users must provide a unique email address, username, and a password. After creating the account successfully users will have to register the SmartLock on their account for the application to connect to the door lock system. With an account created and connected to the smart lock, user can now control their SmartLock.

Another way to access the door is using a key fob. The key fob must be within 1 meter in order receive the frequency that the SmartLock is broadcasting. Once that is received the key fob will send a low frequency to the lock and will let the door know to either lock or unlock the door. But before the door lock can send frequency it must be activated as it will be on standby mode whenever it is not in use. To activate it a button on the exterior of the SmartLock must be pressed. At any time, this button is pressed the SmartLock will start sending frequencies until it receives signals from the key fob, mobile application or when the set time is over, and it has not received any signal will it go back to standby mode.

Documentations will be necessary every step as it will help oversee anything that is going on during the building of the system. A clear and proper documentation during testing must be followed to allow more valuable informations to be documented. Having knowledge on everything about the production will be beneficial at understanding problems as well as solving them. With this being said each documentation must have at least the following informations; the date it was documented, what system or part of the system was changed, why it was changed, what it was changed to and who changed it. Including the date when it was changed is important as it will be easy to backtrack what was the state of the system before that time. Specifying exactly where or what part of the system was changed must be included as it determines if the changed that was made resulted to other components not working as it was before or if it solved other complications in the system. Adding the reasons of the changed is going to help others review and decide whether it is right or if it the best solution. Finally, putting in the name of the person who made the changes is extremely important as anybody who have questions or suggestion may go up to that person and communicate those concerns to them.

The mobile application will be written in JAVA languages and will only have English language as its display language. Likewise, texts will be written using fonts that are legible for the readers. Likewise, types of fonts that are going to be use will be limited to up to three types in this way the whole page will be clear. The background colors that are going to be used will be neutral color as the page will be easy on the eyes. In addition, brighter background colors may distract users from reading and would result to difficulty navigating the page. Buttons and menus are going to be place somewhere it is visible and is easy to read. Excessive drop-down menus will make it harder for users to access features of the applications.

Mobile applications must be connected to the internet as this is the only way it can communicate to the system. Failing to connect to internet will cause the mobile application to not work.

When it comes to actually coding the program the software, programmers must and will have to include the following informations on top of the code: each members' name, the group number, course number, and the semester it was written. With these informations it will be easier to determine which group wrote and owns the rights to the code. In addition to the informations that was previously mentioned, a short description about the functionality of the code and how to run it is going to be written. Including those informations that was mentioned on the very top of the program will help other programmers who wants to read and run the code as they would have an idea on how to run it and what should be the results when running the program.

When choosing the name of the variables, it must be in camelCase format and must be meaningful which means it must give the reader an idea what the variable is used for. Also, variables must not be a single letter with the exception of a counter variable. Variables must be declared in the beginning. Furthermore, `{}` will be on their own line to have a more comprehensible formatting. Indentation must be done every set of conditions to differentiate it from one another. A one to two sentences description of the functionality of the functions or classes must be written before the actual function. There should also be a two new line every after class this way the classes can effortlessly be distinguishable from one another. Finally, excessive white space is frowned upon and would not be done when coding.

To allow easier access to the code GitHub will be user to share, edit, and comments on each code. Not only does GitHub allow members to edit the code it also tracks the history of it. This certain feature is very useful when it comes to backtracking from previous state of the code. Anyone will be able to go back and reverse back the code to when it was still working, error free, etc. In addition, GitHub can be access using any computer that is connected to the internet. On that note, all members can easily view, edit, comment, run, debug, etc. the code whenever they want. Furthermore, it will be easier to share the code to the professors whenever there are any concerns about it. Lastly, the code can be set to public for people to view and comment without editing the code.

4.2 Design Impact of Standards

One of the main goal of the SmartLock system is for user to easily install the system and it must fit an existing door frame. As a result, the door lock system must fit and work with most single-cylinder deadbolts as shown in Figure 8. By using this type of deadbolts, the user can choose to use the traditional key to open their door. This traditional feature is agreed to not be removed as some particular problems may occur such as misplacing the key fob, a dead mobile phone, very slow internet connection, unable to connect to the internet, batteries of the door

lock system ran out causing it to not connect to neither the key fob or the mobile application or just by simple having a family member that still prefer using a traditional key to open the door.

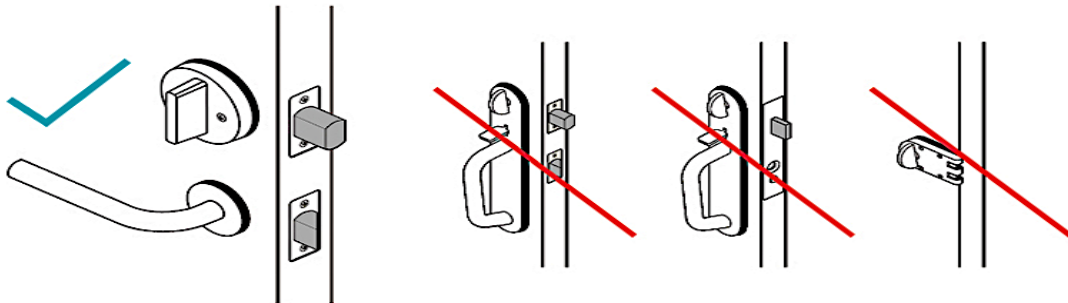


Figure 8 Common Lock Types: What can and cannot work with the system [4]

For the system to be more secure, a set range of one meter is set. As a result, the key fob must be within the set range in order to receive the frequency signal that the SmartLock is broadcasting and for the key fob to send back another frequency to the system in order to execute the command. Furthermore, the set range will prevent unexpected people from opening the door whenever the key fob is around the house. Keeping in mind that users will have to store their key fob inside their homes which means that it will be relatively close to their door. Without the set range, the door will be vulnerable to being unlocked by just pressing the exterior button. The keyless entry was changed for the final design, instead of having an RFID a remote control is used instead. It transmits a string to the transceiver which is then translated and executed.

The door lock system must have the latency of under one second starting from pressing the button until the servo motor fully unlocks the door. Being said that, the connections between the button and the microcontroller cannot be disrupted and each component must execute their work as soon as possible to achieve the latency goal. Moreover, the exterior button must “wake up” the system from standby mode and have to instantaneously send signal to other components such as the LED lights and the speaker as certain LED light would light up and a tune will play through the speaker as the door is being unlocked or lock.

The entire door system must run on four 1.5V AA battery connected in series. As a consequence, the system must conserve power as much as it can and one way to do this is to have the system be on standby mode whenever it is not being used. For this reason, the system must be ‘waken up’ or ‘activated’ every time the user needs to use the system. An exterior button that will be mounted on the exterior of the system will act as a switch to ‘wake up’ the system. The exterior button must be waterproof and should withstand Florida’s weather or temperature. What is more being that the size of the exterior button must be big enough for the user to push small enough to prevent unwanted attention from passerby’s as to not let them be too curious that they would press the button as they see it, similar to a

simple doorbell. As mention from previous section, the power consumption was higher than expected as result two more batteries is added.

For both the mobile applications to work the smartphone where it is installed must be connected to the internet. Failing to connect to the internet will prevent the user from using the application and will result to the phone not being able to communicate to the system. Another condition is to have user create an account, without an account the application will not know which system to connect to since the users must put in the lock where they have. Users must create an account by providing a unique email address, username, and password. Users can then put all the locks that they want to control using the application.

The standard user interface for both SmartLock's mobile applications includes the following elements; a button to unlock and lock the system, some type of status indicator to display the current status of the lock, and the timer status for when the user chose to activate the automatic lock feature. The unlock/lock button will be a toggle button and must be large enough for users to spot easily. Additionally, it will be displayed on the main screen of the application to allow user to get to it effortlessly. The status indicator must also be somewhere on the main screen of the mobile application and would display clearly if the door is lock and unlock for a straightforward information. Likewise, the timer would be on the main screen and would manually display the current time when will the door will be lock. Although, if the door tried to lock itself or when the user tried to lock the door, but the door is not completely close, apart from obvious clues to let the user know that the door is not completely closed is the system will play a unique tune.

There will be a total of four ways that the users can get informed about the status of SmartLock. First, is through the LED lights, upon reaching the door the user can see the color of the LED lights and learn the status of the door. A red LED lights signify that the door is locked. On the other hand, a green LED lights signify that the door is unlocked. This designed is changed to having a red LED that indicated when the door is unlock when it is on and when the door is unlock if it is not.

Another feature is playing a unique tune base on the activity, when the user lock or unlock the door a tune will be played. There is a total of three unique tones each one corresponds to a different situation. One for locking, another one for unlocking and lastly one more for when the user tries to lock the SmartLock, but the door is not completely closed. Since power consumption is a problem, saving power as much as possible without compromising the important features, the speaker was omitted from the final design.

The third way is through the mobile application and it is by looking at the status of the door. It will be on the main page of the mobile application to allow user to get to it really quick. The status will change concurrent to the status of the SmartLock. It will always be available for the users and is very beneficial when users are away from their homes and wondering if they have locked the door or not.

Lastly is the feature of the mobile application which is the notification. This feature requires the user to allow in order for it to work. Once the user allows the mobile application to receive push notification will it then the SmartLock send that information whenever the SmartLock has been unlocked or locked. Considering that the SmartLock can be continuously unlock/lock or vice versa it can cause a problem with the push notification as users may get multiple notifications about the continuous activity. To prevent users from getting numerous notifications, the SmartLock will wait five seconds after the last activity then it would send that activity to the mobile application. The notification was not included to the final design of the application instead a display that shows the current status of the lock is added to the main page.

Regarding to locking and unlocking the door, there are also four ways it can be done. The following are system and or the thing that a user can do in order to lock or unlock their door; using the key fob, toggle switch through mobile application, timer through mobile application and using a key. The timer feature to lock or unlock the door was taken out at the end.

To use the key fob to access the SmartLock, it must be within one meter around the SmartLock and user must activate the SmartLock to send signal to the key fob. A button on the exterior of the SmartLock must be pressed to “wake up” the lock from standby mode and can then send signal within one meter. Once the key fob receives that signal, it will then send back a signal verifying that it is within ranged and to let the servo motor to either lock or unlock it itself. Instead of using a low frequency RFID, a remote control is created to unlock or lock the door similar to a remote garage door.

In the mobile application, users can manually press the lock/unlock button on the main page of the application. The smartphone will then talk to the microcontroller letting it know about the command and simultaneously sends signal to the servo motor to either lock or unlock the door. Another feature that can be found in the mobile application is a lock set timer. Users may choose to either use this feature or not. If the user wishes to use this, they would have to input the time they wish the SmartLock to lock itself after unlocking the door. The timer will be displayed on the main page as it counts down and will changed the status to lock as soon as the SmartLock locks itself.

Finally, the last way to lock the door is to use a key to either lock or unlock it. This feature is decided to not be removed as situations such as lost key fob, mobile phones not connecting to the internet, etc. may happen making the user have to go back to using the traditional key to lock or unlock their door.

5.0 Design Constraints

This section is meant for the all the engineering constraints that this system has to consider when planning, designing, testing and prototyping the system.

5.1 Economic Constraints

As this project is self-finance by each members of the group, the goal is to have a total cost but not limited to around \$600.00. The design has three main components: microcontroller/Printed Circuit Board, proximity sensor and WI-FI interface. The microcontroller/PCB is to be designed and build by the group, while the proximity sensors' parts will be bought from reliable sellers online. Finally, both mobile applications will be coded up using public coding engine and will be tested by installing to each group members' mobile phone. Considering that design is aiming to be affordable, cheaper alternative materials will be bought and will be used to build the whole system. Although the system is going to be cheaper than the ones already in the market standard features of a Smart Lock will still be available. Due to this project being built for a Senior Design Project, production will not cost anything as each member will assemble the whole system. Furthermore, miscalculations are bound to happen during prototyping which means some components will break due to it and not by the actual components. As a result, the same components will have to be bought multiple times increasing the cost.

5.2 Time Constraints

Having only two semesters to plan, research, design and prototype the project, time is one of the problem that was and will be encounter. Due to this, not a lot of advance features are included on the final design. Although, that is the case the standard features of a Smart Lock are still included on the system. Having set scheduled on when tasks should be done and should finished keeps the group in path to completing the design. Additionally, a weekly meeting is set for members to communicate and ask questions pertaining to current tasks or future tasks that needs to be accomplished. The goal is to have a system simple enough to meet the requirements of Senior Design Project but is complex enough that members can learn and even add project to their resume.

The following conditions must be met in order for additional features to be considered; there is enough time to add more features on or to the system, all original requirements and goals are successfully met— meaning each and every components are properly communicating to one another and are all doing the designed tasks it is supposed to do, and more importantly the whole group decides and sees that additional features can be added. However, if Dr. Richie decides that the features are not enough and that failing to add more will result to failing then additional features will have to be added as soon as possible.

5.3 Environmental Constraints

To have our system more environmentally friendly, we will limit our energy consumption by having our entire door system only runs on four AA1.5V battery. Due to more power consumption two more batteries were added to power the system. Furthermore, using rechargeable batteries will considerable reduce waste and cost that the users make and spend. To attain this condition, the microcontroller will be on standby mode whenever it is not being used. This means that it will consume power once the user wakes up the system which is by pressing the exterior button on the system. Moreover, if in any condition the user wish to discontinue using the electrical components they would still be able to keep it and use it as a traditional door lock. Furthermore, parts of the hardware will be installed outside which means that the hardware must withstand different weather and climate patterns. the system will be partially installed outside meaning it should withstand different weather pattern and temperature.

5.4 Social Constraints

The door system is designed to fit and to be easily by both apartment owners and homeowners. To ensure that the requirements set in the beginning are being completed and are working properly, members are required to test every changed and document the results. During the initial production of both iOS and Android mobile applications, all settings will be set to English as the its language. Nevertheless, it will not prevent other users from using either applications as both are designed to be user friendly. For example, to unlock or lock the door there will be a toggle button that will display the words unlock or lock depending on the user's command.

5.5 Political Constraints

At this moment in time, as a group we do not intend to sell either our product or design. However, if the group or one of the members decide to expand on the design and sell the product, it will have to be first approved by the National Recognized Testing Laboratory (NRTL) to ensure that all requirements by both the Construction and General Industry Occupational Safety and Health Administration Laboratory (NRTL) are met to avoid any complications in future production. In the meantime, to avoid any political drawbacks components that will be use must and will be obtain from a trustworthy company and members will confirm that the products obtain from the seller are all approved by the right agencies and/or organizations.

5.6 Ethical Constraints

To have positive feedback from the consumers, the final product will be designed to fit an existing door lock and can be effortlessly installed by the user. By designing the overall door system to fit the traditional door lock saves the user time and money. They do not have to search for a door frame that will fit the system and would not have to buy it, which is a big downside if a user will still have to buy extra products just to use the actual system. Additionally, instructions on how to install the system will be provided to guide users during installation.

Furthermore, a mobile application available in both iOS and Android devices will be created and are both free download. The mobile applications will have a simple design such as an easy to read font and neutral color background to ensure that users with hard time reading words due to small fonts and/or bright color backgrounds will have no problem using the applications. Users will have an option if they would like to receive push notifications about the door last activity and a timer that automatically locks the door.

The exterior button will also be installed in a place that is easy for the user to spot. Dimensions of the button will be big enough that an adult person can press it without difficult and also small enough to not attract unwanted attention from passerby's.

5.7 Health Constraints

Since the system will have to send and/or receive frequencies to be able to communicate to one another, following the conditions for safe operations identified in ANSI/IEEE C95.1-2005 – IEEE Standard for Safety Levels to Respect to Human Exposure to Radio Frequency Electromagnetic Fields, 3kHz to 300 GHz. Due to the system itself using power by using 4 AA 6V batteries that may cause shortage of the system and can eventually make fire. To avoid such thing from happening, calculation of power consumption must be done properly, and materials is required to be purchased from reliable company that ensures that all components are approved by the Underwriter Laboratory. materials mu

Even though, the group decided to buy cheaper alternative materials it is decided that materials will only come from companies/people that are certified to sell the products. Without proper certification, members will have no knowledge how the materials are produced, that materials are compatible with the rest of the components and specially if it is safe to use. Just by knowing that all of the materials are certified and past the standard testing makes it safer to use.

5.8 Safety Constraints

For safety purposes, the door system will have a different tune played as the door is being lock and unlock as well as different colored LED lights will be displayed on the door system notifying the users on what the current status of their door. Besides from the door system unique hardware notifications, the mobile applications will also have an option that allows electronic notifications. In addition, users can choose the option to have a set timer using the mobile application that allows the door system to lock itself once the timer is done. Finally, a set distance will limit the ranged of how far the key fob should be in order to receive and send back signal to the microcontroller.

Just by walking up to their door, the users can find out whether their door is locked or not with the help of LED lights indicator. Two LED lights— Red and Green, is one of the hardware feature that allows user to be inform of what is the current status of their door. Red LED light will be ON once the door is locked and will remain ON until the door is unlocked. By that time the Red LED lights will turn OFF and a Green LED lights will turn ON and will remain green until the door is locked. This feature will be very useful for those users who are hearing impaired. LED lights were changed to red, green and yellow. The red LED indicates if the door is unlocked— the red LED is on or if it is lock— it is off. The green LED to show that the system is currently connecting to the system and a yellow light to show if it successfully connected to it

The second hardware feature that can inform users about the door, that is also very helpful for those visually impaired is through playing different tune for different tasks. There will be a total of two tunes that is going to be played through a small speaker inside the door system. One for when the door was locked and another tune for when it was unlocked. Additional tune may be added for when a user tries to lock the door, but it is not properly closed. Although, this feature is useful for other, it had to be taken out as it adds power consumption to the overall system.

For those people always wondering if they remembered to lock their door, there is a feature to solve that problem. An option for push notification can be turn on by the user through the mobile app. This way every time the door was locked or unlocked the user will get notified and can stop worrying about going back just to check if they indeed locked their door. Also, if they indeed forget to lock their door, the users can use the mobile application to lock the door. Another helpful feature which can be found in the application is set timer. Since some users forget or cannot check their phone right away, they can set the timer ON using the mobile app. By having the timer ON, it allows the door to automatically locks itself once the timer is done.

Finally, having a set ranged for the key fob to be in able to receive and sends signal back to the microcontroller helps unauthorized people unlocking the door. Since the users are going to be placing the keys inside their home, a two-meter ranged

is appropriate. A farther ranged than that will then the key fob become useless since whenever the user place the key inside their homes it can still receive and sends signals just by someone pressing the button on the door system.

5.9 Manufacture Constraints

Hardware and software requirements that gives the users a more convenient and secure way of unlocking and locking their door are established in the beginning which all the member must have to comply with throughout the whole time designing and creating the whole system. The three main components of the system are: main door lock system, key fob and mobile applications available in both iOS and Android phones.

The main door lock system is consisting of the following: low frequency transmitter, a high frequency antenna, servo motor and a printed circuit board (microcontroller). The PCB will act as the brain of the system which will have all the components connected to it. On the other hand, the key fob will have a high frequency transmitter, low frequency antenna and a waterproof exterior button. Lastly, both mobile applications will be coded using JAVA language. More detailed information about each of these can be found under Requirements and Specifications section.

To allow users to use the traditional door frame, the final product's dimensions must not be greater than: exterior: 7.5cm x 7.5cm x 7.5cm and interior: 25cm x 7.5cm x 7.5cm. Also, a lower power consumption is desired as the door lock will be using four AA 1.5V battery. For that reason, it will continue to be on standby mode until the user push the button located at the exterior of the door lock frame that will wake up the system. Materials that will be used to build the system can be obtain from reliable sellers in a reasonable price so that whenever the users feel like replacing any of it, they will be able to do so.

By having the microcontroller/PCB on standby mode whenever it is not in used, the system will conserve more energy. To “wake up” the main system and have it received and send signal a button installed on the exterior of the door must be pushed. Seeing that the button will installed outside, it must have the quality similar to a door bell withstanding the rain and the heat outside. Researching on how the exterior and interior of a doorbell will help achieved the qualities mentioned.

In order to accommodate more mobile users there will be a mobile application available in both iOS and Android phones. The applications will be written in JAVA and will allow users to connect the main door system with a working WIFI network. Through the applications, users can create their own account for the Smart Lock System and can essentially use their phones as a key. They can unlock or lock their door, check the status of it, and if they wish have push notifications send straight to their phone about the last action the door has executed.

5.10 Sustainability Constraints

To assure that all working components are remained working even after changes or additions to the systems, a constant system test system test is required to be done before and after the changes are made making sure that those changes be documented. The main tasks that needs to be accomplished are: exterior button “wakes up” the main door system, main door system sending and receiving signals from key fob and mobile application, microcontroller sending the right commands to LED lights, speaker, and servo motor, mobile application can communicate to the microcontroller and the set timer is doing its job of letting the microcontroller lock the door

The exterior button must be tested and confirm that it does starts and keep the microcontroller broadcasting signal. Multiple variety of two types of simple tests can be done to assure that this is working: pressing the button with the key fob in ranged to receive the signal and by having the key fob outside of ranged. Examining how the microcontroller would react to the test can confirm if the button is indeed doing its job properly. By doing these test, users can confirm if the key fob is communicating with the microcontroller. The test will check the following, what the button is doing to the microcontroller if the microcontroller is sending/receiving signal and if the key fob is receiving/sending signal.

One option that a user may opt for is a timer that automatically locks the door. This feature is extremely convenient for when users are hurrying out the door and completely forgets to lock it. To keep this feature, user just have to leave the timer option in the application set to ON. Similar to the exterior button, this feature should also be checked to make sure that that part of the mobile application is working properly. User can set the timer the timer and on and wait for the timer to end and checks the door indeed lock itself. The opposite of that should be test as well, not setting the timer on and checking if the door is not locking itself.

Finally, updating the mobile application can helpful and assure that the software remains compatible. Being said that, a simple upgrade can also wreck the communication of the mobile application to the microcontroller. Upgrading the system should initially be done by the programmer and using a door lock tester to test it on. Guaranteeing that every change on the code will only help the system and not to stop it from working. Reading the reviews online about the mobile applications on some of the Smart Locks this is a common occurrence. The system upgrades their software which leaves the mobile user unable to use the mobile application or even worse cannot access it. Avoiding the same thing to happen to the mobile system, a proper and thorough testing must be done before upgrading the whole user system.

6.0 Comparison and Design

The design section of the documentation is primarily used to compare and choose valid parts/designs for each facet of the project. This chapter of the documentation will cover each subsystem individually and in depth, as well as software design decisions and their justifications. Designs for each system will be explained, comparisons between valid part choices will be made, and parts that fit the design specifications best will be selected. A look into how the parts will all function together within their respective systems will be touched upon, but only for the purpose of justifying part choices. An in depth look at how components will exist on a circuit board in the final project can be found in the prototyping section, section 7.0.

6.1 Keyless Entry System Door Module

The keyless entry feature of the SmartLock is meant to be the primary way the user interacts with the lock, and as such, it must operate smoothly every time and be low latency. Also, both the door lock micro controller and the key FOB portions of the design are to be battery powered, meaning the passive power consumption must be extremely low. To achieve these marketing constraints, the keyless entry system operates similarly to that of a modern automobile. The door module contains a 315MHz receiver, a 125kHz emitter, and a button for the user to interact with the door lock. During normal operation, the user will have the key FOB on their person or within one meter of the door. The user will press the button which triggers the door lock micro controller to come out of low power mode, send a 125kHz wake up signal to any key FOBs in range, and wait to receive a 315MHz response. If the door micro controller receives the correct high frequency response, it knows to send the unlock signal to the servo motor.

This keyless entry design affords the user the most convenient experience possible while still maintaining lower power consumption. It is unfortunate that the user will have to interact with the door in order to activate the system, but the power consumption trade of for a system that say, uses OpenCV and a camera to detect if a person is near, or a system that uses IR sensors does not seem worth the power consumption and build complexity trade-off. Other alternative design included things such as the user pressing a button on their key. This, while being easy to implement, doesn't meet the requirement of "keyless entry" and is already prevalent with garage door openers and has been standard on every car for decades. Another possible design was to have the key FOB broadcast the high frequency unlock signal after a predetermined, short amount of time (under 5 seconds). This idea however, would require the door micro controller to always be listening, leading to a high power consumption for it. And it would also, obviously, cause a high power consumption for the key FOB, to have to constantly be broadcasting a 315MHz signal. The design the team ended up settling on, with both low and high frequency systems seems to be the happiest middle ground,

between ease of use for the user, ease of design for the design team, and it minimizes passive power consumption for a keyless entry system.

Constructing our own low frequency emitter for the door lock micro controller would more than likely be incredibly time consuming, more expensive, and less reliable than if the team simply purchased a module to integrate into our design. As such, rather than reinventing the wheel, the low frequency transmitter will be a KGEA-BFCAM. It is an emitter typically used in passive entry for automotive applications. It has a low profile, making it easy to place in the door lock's housing. The built in connection is simple. With only two wires, it is easy to integrate into our design. It's read distances are acceptable, perhaps even too far (as we want the max range of the keyless entry system to be under 1 meter).

The second portion of the keyless entry system on the door lock micro controller, the high frequency receiver, will also be purchased module rather than designed by the team. The HiLetgo Transmitter and Receiver modules that were developed for hobbyist Arduino/Raspberry Pi users will be used. They were initially selected because they were already owned by the design team prior to the start of this project, giving us a certain familiarity with how the modules operate. However, in addition to this familiarity, the modules are excellent choices for this project, as they have a low profile, allowing the parts to fit inside the housings and smaller size is preferable for the final design of the project. Additionally, the working current of these modules is a current range, allowing the design team to adjust the transmitting power of the modules and dial in the desired range of the key FOB, which is entirely necessary, so the key will only have a transmitting range desired by the team. This was an important engineering requirement for the project, being that it is a house key. The range cannot be too far or else anyone would be able to unlock the user's front door if the user is home and the key is located somewhere in their house. The module's max transmitting power of 25mW could certainly be far enough to achieve this. It can transmit up to 90m in an open area.

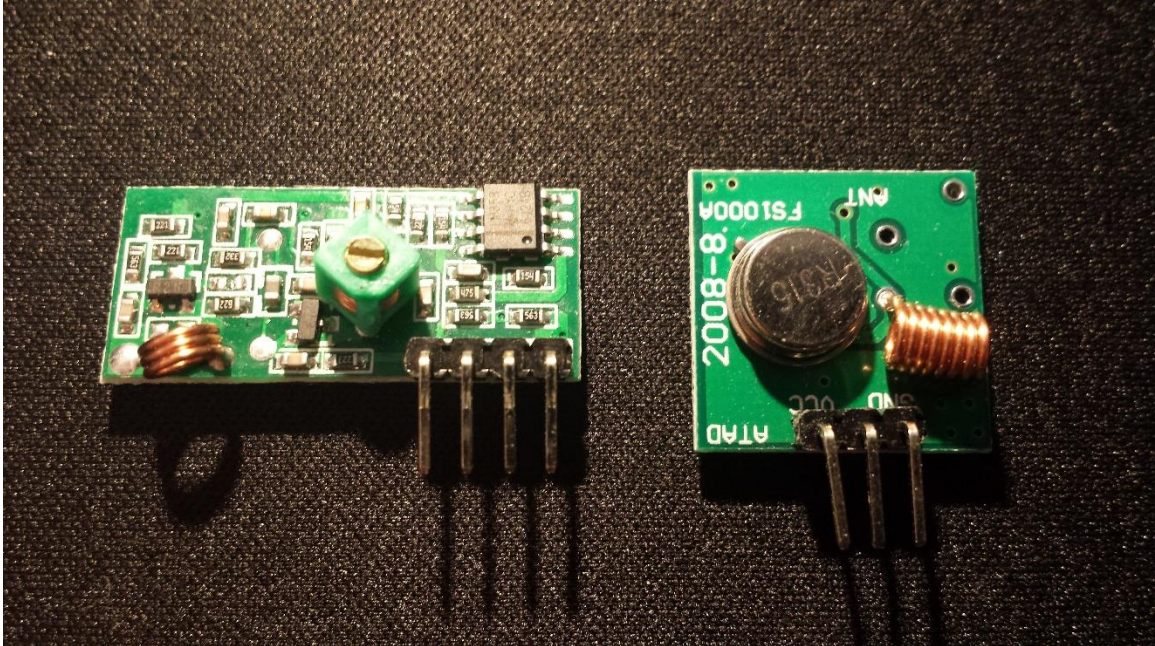


Figure 9 350MHz RF receiver and transmitters.

In addition to the sending and receiving modules, there needs to be a way for the user to interact with the door lock module in order to bring the micro controller out of low power mode, send the wake up signal, and listen for the unlock signal. A weather resistant button will be placed on the exterior of the lock to fill that role. The micro controller will be in low power mode in its default state, with all unnecessary systems shut off. When a circuit is completed by pressing the button, an interrupt will be triggered and the unlocking subroutine will begin. The through hole modules to be used in prototype design can be seen pictured above in figure 10.

In senior design 2, Keyless entry system and the door module system has been changed. Since the low frequency wake up signal did not work, we decide the use only radio transceiver to communicate with key and door. Push button at the key fob will be used to send radio frequency to unlock or lock the door. For radio frequency transceiver, CC1101 will be used instead of 350MHz RF receiver and transmitters.

6.1.2 Key Module

To brief the keyless entry system, there is key fob which will communicate with the main door module. Once the door module send signal around the door module, the key fob will receive the signal if the key fob located close to the door module. After receiving the signal from the door module, the key fob will generate radio frequency included the identification, which is called RFID. If the key fob and the door module are located close to each other, then door module will be able to

receive RFID from the key fob. The smart door will unlock the door. Since our project power supplied by battery, we need to save power as much as possible.

Key module composed with low frequency receiver, radio frequency transmitter, microcontroller, and analog front end. Low frequency receiver will be used to get 125kHz signal from door module. Since key module also has limited power supplied by battery, low frequency receiver will be used for wake up key module. After receiving low frequency from door module, microcontroller will work on to generate Radio Frequency Identification (RFID). For our keyless entry system will use 315MHz radio frequency for RFID. It is very important using RFID, because the smart door should not open the door with any other frequency. The RFID will be sent through the radio frequency transmitter to the door module's radio frequency receiver. The figure 4 in section 2.4.2 will shows the diagram of the key module.

In senior design 2, the design has been changed. As it is mentioned above, initial project design was keyless entry system. Since we having trouble with low frequency wake up signal, we decided to use a button to lock or unlock the door. When the button is pushed on key fob, key fob will send a signal to lock or unlock the door.

6.1.2.1 Microcontroller for Key Module

As the key fob is also intelligent key, microcontroller will be needed. In this project, MSP430G2553 will be used as microcontroller for the key fob. It is ultra-low-power microcontroller, consists of several devices, different set of peripherals for various applications. MSP 430G2553 has 24 input/output capacitive-touch enabled pins. The detail of parameter of the microcontroller will be explained in the section 6.6, Microcontroller.

However, the design has been changed during the working on senior design 2. MSP430G2553 was the best option for power consumption, we decided to use Atmega328P. We all are used to programing Atmega 328p instead of MSP430. Also, Atmega328P has bigger library then MSP430, and coding would be efficiency.

6.1.2.2 Low Frequency Receiver for Key Module

As low frequency receiver, AS3933 will be used. AS3933 is 3D low frequency wakeup receiver, has three channel low power receiver. It also can be used for the detection of programable 16bit and 32 bits wake up pattern. As 3933 composed with 16 pins, three digital input, four digital output, three pins of supply pad, and six pins of analog input/output. The following table will shows the parameters of the AS 3933.

Parameter	Value
DC Supply Voltage	-0.5V to 5V
Input pin Voltage	-0.5V to 5V
Input Current	-100mA to 100mA
Storage Temperature	-65 degree to 150 degree
Total Power Dissipation	0.07mW

Table 6 AS 3933 Parameter [5]

As the table 6 shows that the supply voltage is low as the purpose. Consuming of the current low as well, the total power dissipation is only 0.07mW. The key module as limited power supply, battery, the AS3933 is fit in our requirement as low power consumption.

In senior design 2, however, since we had a problem with crystal oscillator and soldering pin, we decided not to use for this project. Button is installed for the key fob, the door can be opened or closed by pressing the button on key fob.

6.1.2.3 High Frequency Transmission

As sending back to the door module to response, CC 1101 is used for the key module. CC 1101 will be used for the door module as well. CC 1101 is low power RF transceiver. CC1101 transceiver composed as 20 pins. The transceiver operates in the 315/433/868/915 MHz band, can be used for wireless alarm and security systems, or industrial monitoring and control. CC 1101 is integrated with a configurable baseband modern. It also can be programmable output power up to 12dBm for all supported frequencies. The rate data from 0.6 to 600 kbps. The following table will shows the parameter of CC 1101 transceiver.

Parameter	Value
Supply Voltage	-0.3V to 3.9V
Voltage on Digital Pin	-0.3V to 3.9V
Voltage Ramp-up Rate	120kV/us
Storage temperature range	-50 degree to 150 degree
Current Consumption in 433MHz	13.1mA to 16mA

Table 7 CC 1101 Parameter [6]

As the table shows above, the supply voltage is low as our requirement. As the key module, the supplied power cannot be high, the CC 1101 works with low power supply and able to generate programmable high frequency signal. The current consumption is little higher than expected value, but since the high frequency transmitter will only needed when the key fob receive the signal.

6.1.2.4 Schematic of Key Module

In this section will shows how three chip will be connected to communicate each other.

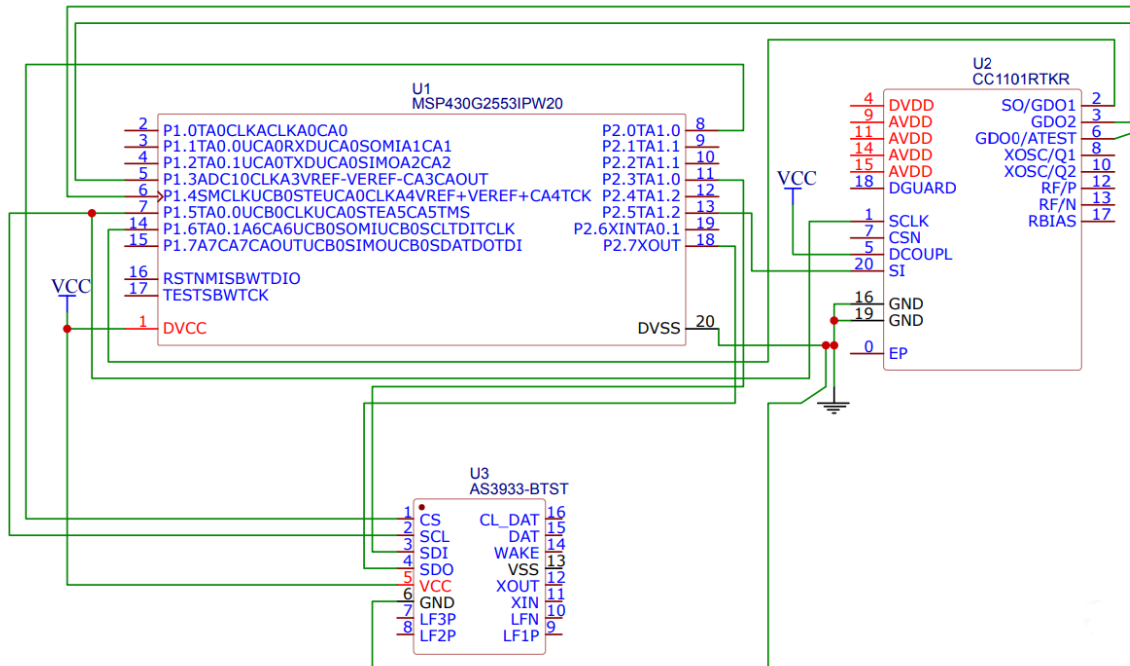


Figure 10 Schematic of Key Module

The figure 11 shows the schematic of key module. A microcontroller is placed between low frequency receiver and high frequency transmitter. Since only one channel will be used to receive wake up low frequency, only one antenna installed at AS 3933. CS, SCL, SDI, SDO ports are used to communicate with microcontroller, MSP430G253. This schematic does not show in detail for the CC1101 because high frequency transceiver will be purchased for key fob. There are four digital pin of microcontroller will be used to communicate with CC 1101 module.

Since the AS3933 has removed during the senior design 2, only CC1101 and Atmega328P will be connected. Since the microcontroller also changed, the pin layout will be changed as well. GDO0 will be connected to pin 4, CSN will be connected to pin 16, SI will be on pin 17, SO will be connected to pin 18, SCK will be connected to pin 19 of MCU. CC1101 will be supplied power as 3.3V.

6.2 Wi-Fi Integration System

One of the key features of the SmartLock is its integration with a mobile application. This feature requires its own system with subsystems to be designed. The following sections will discuss the design of every facet of the Wi-Fi integration

system. This includes the Wi-Fi module to be attached to the embedded system on the door lock, the mobile application that is to be used to control the door lock through the internet, and the server which will maintain a database and act as an in between for the mobile application and the embedded system.

6.2.1 Wi-Fi Module

In order to send packets of data between the server and the door microcontroller, an 802.11 module is required. Designing one would be carry a high cost in time, labor, and funds. Therefore, the design team will purchase a self-contained Wi-Fi module. The ESP8266 has extensive documentation, is low power, has a small profile, contains a built-in network stack and low power microcontroller to run it, has a built-in voltage regulator, and carries a very low unit cost. The ESP8266 is used in many hobbyist microcontroller projects, which has led to a large amount documentation and tutorials on its setup and operation as well as regular firmware updates for the network TCP/IP protocol stack included. The module itself is very small and will have no problems fitting inside the door lock microcontroller housing dimensions specified in section 2 of this document. The particular module that the design team is implementing has been discontinued in favor of Wi-Fi modules with more powerful micro controllers and more GPIO pins, which has driven the purchase cost of our module all the way down to under four dollars. The more powerful microcontrollers and additional GPIO pins on the newer, more expensive modules are not required for our project, as we only need the Wi-Fi module to send and receive packets. All processing will be done with the main microcontroller the module is connected to.

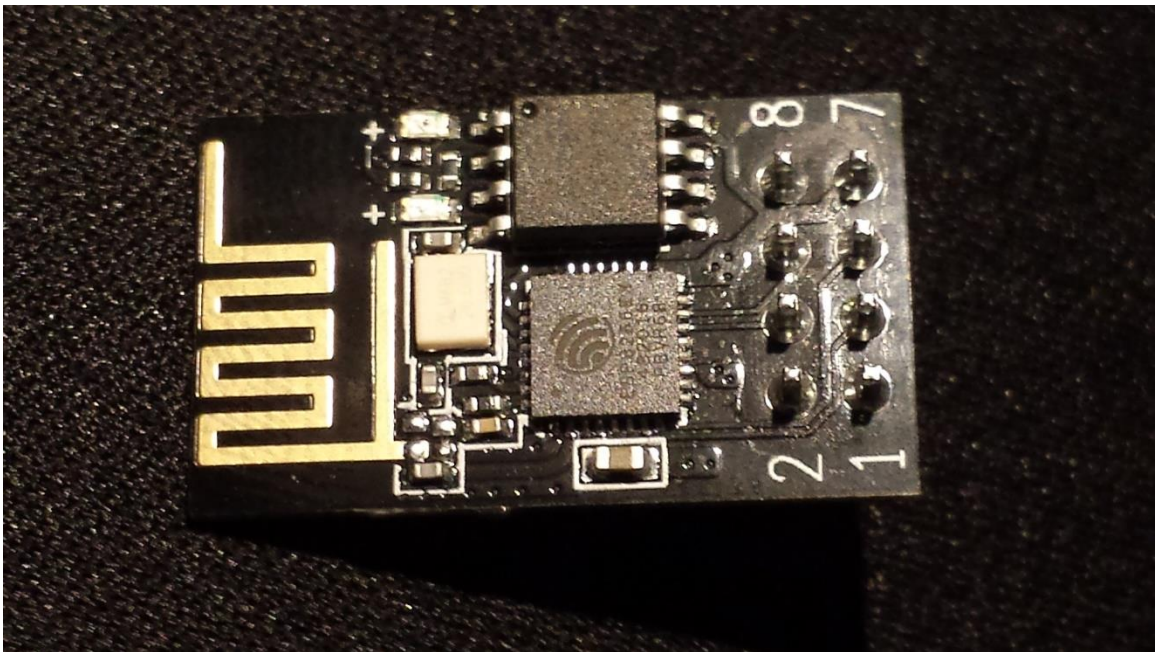


Figure 11 The ESP8266 802.11n module

The ESP8266 module is powered by 3.3V VCC. It has TX and RX pins to be used for serial UART communication with the main micro controller. The two general purpose input/output pins will not be used in the final design. The RST pin is used to reset the module and may be used for as a way to fix bugs related to the module that can be solved with a hard reset. The CH_PD pin is used for shutting off the module entirely to save power, but the SmartLock design requires the ESP8266 to be in sleep mode. While in sleep mode, an external interrupt can be used to trigger an interrupt and wake the module. The through-hole ESP8266 module to be used during initial prototyping can be seen above in figure 12.

The manufacturer of the ESP8266 provides firmware updates. These updates are available for download on their website. In order to update firmware, a USB to serial converter must be utilized to serially communicate with a desktop computer. Using a terminal program such as Hyper Terminal or PuTTY, the commands to update firmware and network SSID and password can be programmed to the module. This is acceptable for the prototype version of the SmartLock, but a more user friendly way of connecting to a local area network must be developed for later versions.



Figure 12 Schematic for ESP8266 802.11 module [7]

The pinned ESP8266 may be idea for early prototypes and troubleshooting, however, for a lower profile on the printed circuit board and for a cleaner final

product, the surface mountable ESP8266 shown in Figure 13 preferable. Like its pinned counterpart, the surface mountable ESP8266 has a full TCP/IP stack. It is its own, self-contained, low power microcontroller that is capable of bringing the system out of sleep mode by receiving packets, which is perfect for a battery powered microcontroller where battery life is a chief concern. It is a Wi-Fi module designed for hobbyists, meaning that documentation is extensive. Also, the hobbyist targeting of this module has made the documentation exceedingly easy to read and comprehend. There are many tutorials available to help a less experienced design team integrate it into our final project.

While developing the prototype of the SmartLock system, the plan to implement Wi-Fi outlined above was found to be inadequate. The ESP8266 comes with built in firmware and an instruction set of AT commands that can it receives via a UART connection with the main MCU. These AT commands were found to be extremely unreliable. After conferring with other senior design groups that were implementing Wi-Fi for their projects, this design team opted to abandon use of the ESP8266 and switch to using a development board powered by the ESP32.

This switch came with a couple distinct advantages. Firstly, the ESP32 development board has a built in serial port. This allowed us to program it and view its outputs on a terminal directly. The advantages of this cannot be understated. Debugging a project involving communication with remote servers is much, much easier when outputs can be printed to a terminal and when the programming process is streamlined by built in features like a serial port. The second major advantage was power consumption. The ESP32 has an extremely low powered sleep mode, consuming less than micro amps while asleep. This low powered mode was used in conjunction with a timer interrupt to wake up and poll the server after a certain increment of time.

Although, the ESP32 did come with some negative tradeoffs. First, was cost. The ESP32 development board cost roughly twelve dollars, which is four times more expensive than the ESP8266 the group originally planned to use. It is not a budget breaking expense, but still had a fair impact on the final per unit cost of the project. The second main tradeoff is size. The ESP8266 is a relatively small piece of hardware, while the ESP32 development board would have increased the PCB size of our project by roughly 50%. Luckily, the ESP32 could function with the same pin connections allocated for the ESP8266, so when the decision to change Wi-Fi module was made, the PCB design was left unchanged and wires were soldered between the main PCB and the ESP32 Wi-Fi module. This allowed us to stash away the Wi-Fi module wherever the size difference would have less impact on the final look and dimensions of the project.

Comparison of 802.11 Standards

The SmartLock will allow the user to lock/unlock the lock or see its status using mobile application through the internet. For this to be accomplished, the lock PCB

will need a WiFi module to that is capable of communicating with most home networks. There are four main 802.11 standards in use home networks today.

All 802.11 standard releases are backward compatible with previous releases. Meaning that any older standard used in our design will still be compatible with wireless networks using the newer standards. However, if we were to select a standard such as 802.11AC, wireless networks older than 2013 would not be compatible with the SmartLock.

Every current standard has an acceptable range for the purposes of our design. The benefit of the doubled range of 802.11N would be negligible for both the prototype or marketed product. The majority of the population will have their wireless router within 125 feet of their door. Most 802.11 standards operate on 2.4GHz, which is an unregulated frequency and very prone to interference from other home devices such as microwaves. Using these 2.4GHz frequencies in a design would mean possible packet losses that need to be compensated for in software. For the purposes of our design, any of the current standards have a sufficient data rate. The SmartLock will only need to send small JSON payloads, meaning network speed is almost irrelevant. Passive power consumption is one of the largest concerns in our design and the higher 5GHz frequencies of 802.11AC and 802.11N will mean a higher power consumption with little tangible benefit.

6.2.2 Server Design

One of the key features of the SmartLock is its 802.11 integration and the user's ability to access the lock's status or interact with it from any distance over the Internet. Such a feature requires a remote database be connected to a hosted server for both the door lock microcontroller and a mobile application to interact with. Any door lock must be in constant communication with the server to be able to feed its status and receive instructions. As stated previously, the server will utilize a LAMP stack (Linux, Apache, MySQL, PHP) and will be self hosted by the design team. A visual representation of the network stack can be seen in figure 14. This stack was chosen because it is extremely prolific, with nine out of ten of the top websites using this stack. Such proliferation of a standard has led it to be extremely well documented, errors are patched in quickly, and debugging resources can be found with ease. In addition to the extensive documentation, members of the design team have previous experience setting up this stack, making it an obvious choice.

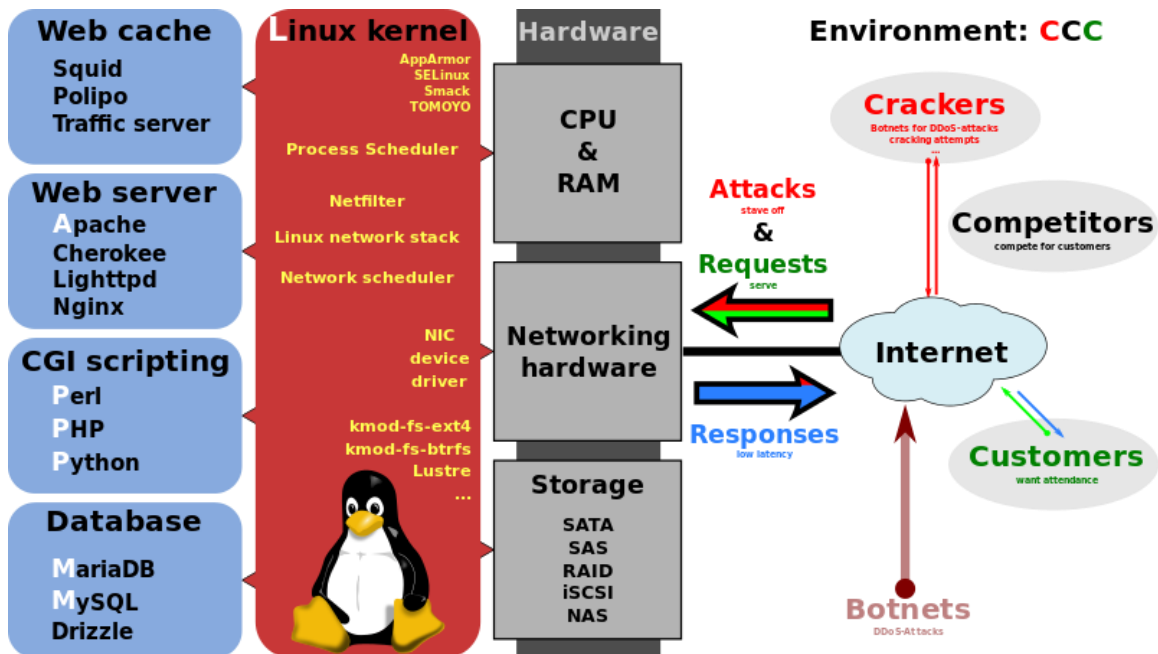


Figure 13: Chart illustrating the programmatic and hardware layout of the LAMP stack [8]

To briefly explain the functionality of a LAMP stack, all server programs run inside a Linux kernel. The Apache web server acts as a network listener and the network stack. The database, in our case, will be MySQL for reasons outlined below. Server side scripting is done using PHP. These scripts allow for communication between the database and the web server.

The team will be self-hosting the server. This will both cut down on final cost and ensure there is no dependency on an outside organization's services for a relatively small upfront cost and low maintenance costs. The server hardware required for the prototype version of the SmartLock is not process, storage, or bandwidth intensive enough to justify outsourcing the server. The bandwidth overhead for the mobile application portion of the SmartLock is expected to be extremely small, with only a few JSON packages being sent with every operation of the lock, so residential Internet speeds and latency should be acceptable.

As with other aspects of this project, when it came time to implement the server portion, better design options presented themselves. The server API for the SmartLock system was implemented using the Node.js with Express framework and a MongoDB database. The decision to switch frameworks was not based on any difficulty or roadblock the team had with implementing a LAMP stack, however. The decision to switch was based on one team member's skillset. Between beginning the project design and actually implementing any part of the design, one team member became employed writing Node.js back ends for mobile applications. After spending a large amount of time gaining proficiency in the

Node.js with Express framework, it seemed silly to develop the back end for the SmartLock using a LAMP stack.

The decision to switch frameworks was based on more than just familiarity. Node.js is an emerging framework that is quickly gaining popularity. It is incredibly powerful and scalable. The node package manager contains many different useful packages that add features to the framework such as error logging, unit testing, data validation, and much more. For our API, we used Joi for data validation. Joi acts as a line of defense to stop bad data from querying the database. If a bad request is sent to an API endpoint, Joi will allow us to catch the bad request and kick it back to the front end before anything is queried or posted. This is very useful for maintaining a stable server. However, no API is perfect and sometimes bugs or holes in programs are overlooked. We implemented Winston as our error logger for this reason. With a middleware function that catches all uncaught exceptions and by using Winston to log those errors to a file, a robust API can be made. It is equivalent to having the whole API inside of one big try/catch block. If an uncaught exception is picked up by the middleware, the server will no longer crash. Also, the uncaught exception can be logged to a file so the team can patch the code causing the exception in the future.

6.2.2.1 Server API

Server-side scripting will be done using PHP, which is short for Hypertext Preprocessor. It is one of the most widely used server-side scripting languages and complements well with Linux, Apache, and MySQL. Each of the following paragraphs represent a PHP file present on the server to be called to fulfill their described function. These PHP files are to be called by one overall index.php. As the SmartLock is a security device, security from attacks is a large concern. JSON packages sent between the mobile application and the server must be encrypted. User passwords must be hashed in the User's table. And the mobile application must be given a method of verifying if a user has permissions to be viewing the information he/she is requesting.

Create account: The user is able to create a new account using the mobile application. LoginID, user email, and a hashed password are received by the server via a JSON payload. A new table is created for that user account. If account creation is successful, a confirmation message is sent back to the application in a JSON. If unsuccessful, the appropriate error message is sent instead (ie: LoginID already exists, internal server errors).

Verify login: Login verification is done by the user sending the LoginID and hashed password via a JSON payload to the server through the mobile application. Access to the rest of the application is contingent on a successful login. Once login info is received, the script checks the database for accuracy and responds appropriately. Either, access is allowed to the rest of the application that the user has permissions for, or the appropriate error message is returned via a JSON package. There are

error messages for “LoginID does not exist” and “Incorrect password” as well as any possible server errors.

Add lock to account: All manufactured locks will have a unique idLock. The database will be preloaded with a table for each lock. A section of the mobile application will allow the user to associate a lock with their user account using this unique idLock. As with the other sections, a JSON of the userID and lockID will be received by the server, which will then create an associative table between the two in the database. If the lock was added to the account successfully, it should appear on the mobile application home screen for the user.

View status of lock: In order to populate the user’s home screen with locks associated with their account, the PHP script will search the MySQL database by LoginID for UserLockAssociation tables with that particular user’s identification. After all locks associated with the user are identified, their status can be found in the table for each lock. These lock names and statuses are sent to the mobile application with a JSON payload.

Change lock status: When a change lock status request is received from the mobile application as a JSON, the server must complete three operations: change the status of the lock in the Lock table, send the instruction to the lock, and send the confirmation to the mobile application. All locks should have their own threads open with the server at all times, so it’s simply a matter of sending a JSON to the lock microcontroller. An acknowledgement that the operation was successful should be received back from the microcontroller or an error message if the operation failed. Failures could include the lock not being found or the door jamb sensor is open. After this acknowledgement or error message is received by the server, it is sent to the mobile application.

6.2.2.2 Hardware

A Raspberry Pi running raspbian (a debian based Linux distribution) will be used to act as our server, for the mobile application and door lock to interact with. It was selected for multiple reasons. The Raspberry Pi has extremely low power consumption (under seven Watts) meaning that the team member that must leave it running in their house all day won’t see a noticeable increase on their electric bill. It comes with a preinstalled 802.11n wireless module, is HDMI enabled, and has a per unit cost of under \$40. The 1.2GHz ARMv8 CPU, 1GB of RAM, and 32GB of removable storage is more than enough to be an effective server for a prototype version of the SmartLock.

This section is no longer relevant, as the decision to self-host the server code was changed to hosting on a Linux virtual machine on Digital Ocean. There are several reasons for this change, but one of the biggest factors was just increased knowledge of web frameworks between senior designs one and two. While in senior design one, it seemed appealing to host on our own Linux machine and

have complete control over the environment, but the exact same result can be achieved by using a virtual machine on a remote server. For only five dollars a month, Digital Ocean will reliably host a virtual machine with one gigabyte of memory and breakneck internet speed. This way, we are not relying on the DSL internet connection from one of our apartments, but we still have a remote hosted Linux system that we have full control over.

6.2.2.3 Database

The database will be created using MySQL as it is open source and has extensive documentation, allowing for a shallow learning curve. It is extremely prolific. Major websites such as facebook, twitter, and wikipedia rely on MySQL. MySQL performs well at all scales, so if the prototype were to be put into production, the database would not suffer performance issues from having many more users and locks added. The database for the prototype project will only have one lock, but the back-end software is to be designed as if there were thousands of locks, allowing for expansion of the product.

Our database will consist of four separate tables. A visual representation can be seen in figure 15. The “Users” table fields has a one to many relationship with the UserLockAssociation table and has different values unique to each user, a unique userID, a unique LoginID, a user email and a user password. The “Lock” table only contains its unique ID and the current status of the lock. Because one user can have many locks associated with his/her account, and one lock can be associated with many users, an associative table between the two is necessary. The associative table’s fields are simply its unique primary key and the two foreign keys of the User and Lock creating the relationship between them. It exists for the server side scripting to compile a list of all the locks associated with a user account to verify which locks the user should have access to. When the user triggers an unlocking event through the mobile application through the web server, an Event table is created with fields for the time, date, and user that triggered the unlocking event. This Event table is used to keep a log of all user interactions with the lock that occur through the mobile application.

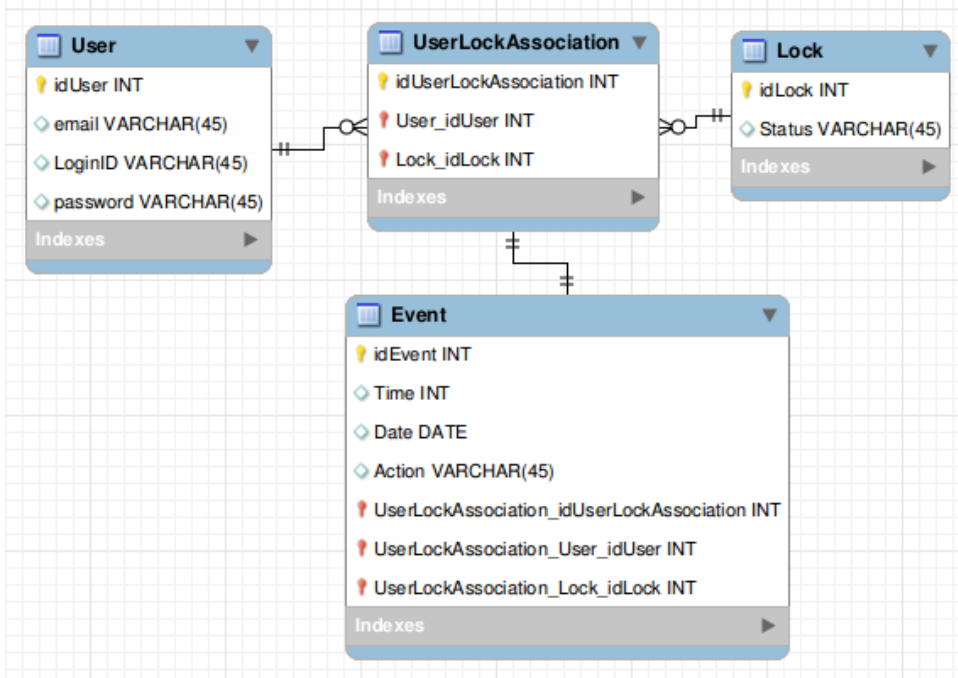


Figure 14 an Entity Relationship diagram for the MySQL database to be stored on the server.

While building the back end for the SmartLock project, MongoDB was used in lieu of MySQL. Generally, with Node.js frameworks, NoSql databases are used. MySQL is fully supported, but the standard is to use NoSql. Because there are no large dependencies on associations between database entities (there is just the user/lock association) it was very simple to use a NoSql database and draw the associations manually by adding ObjectIds inside of the User and Lock models.

6.2.3 Mobile Application

To accommodate more mobile devices, the mobile application will be available in both iOS and Android devices. Both will be programmed in JAVA language and will have English as the display language. The main user interface for both applications will be closely similar in a way that both mobile application will have the same fonts, font size and placements of button and menus. Finally, both applications will be free to download from Google Play or Apple Store.

For the reason that this project is self-finance by the members of the group, it is decided that to not spent more than the set budget, it is decided that the iOS mobile application will not be developed. To developed an iOS application, developer must pay a fee of \$99. Additionally, compare to developing an Android application, an Android application is easier to debug, cheaper to produce and is open source which makes problem solving much easier.

The integrated development environment (IDE) that will be used for the Android and iOS applications are Android Studio and Xcode. Both IDE's are available online and can be downloaded for free. In addition, numerous tutorials are found online that can provide information on how to start coding the application. With that being said, the cost of creating a mobile application will be close to free as it would still need to be determine if either of the application store require applications to be published and for the publisher to pay any fees.

After successfully installing the mobile application, user will be required to create an account which allows them to use their phone to access the door lock system. To create an account, users must provide the following; a unique email address, username and a password. If any of the three information provided is not unique—another user has the same information, the user who is currently trying to make the new account will not be successful at creating it. After creating the account users must provide the information of their SmartLock as it will be use as connection between the two.

For both the mobile applications to be more user friendly, they would have to have the following standard designs such as fonts, font sizes, background colors and placement of button and menus. Despite the fact that the types of fonts, font sizes, and the placement of the menus are to be decided, Figure 16 shows the basic design of the main page of the mobile application on an iOS device as well as an Android device. Additionally, both figures show a toggle switch for the unlock and lock button. It will be place on the middle of the page for an easy access.

There will only be a total of three different types of fonts that will be used to allow clarity throughout the whole mobile application. The types of fonts that will be used are fonts that are simple and are certainly readable by the users. Combination of different font size will give priority to the information being presented for example the important the information the bigger the font of it.

Regarding the colors of the fonts and the background colors everything will be standard and are neutral. Text colors will definitely be in the dark shade like black while the background colors will be in neutral shade. This would give good contrast between the text and the background color resulting in higher resolution giving more clarity to the texts.

With regards to the placements of the buttons and menus, the main unlock/lock toggle switch will be place on the center of the screen to allow easy access. Important information will also be on the same page as the toggle switch such as the status of the lock and the timer. Figure 16 shows the ideal placements of the features that was mentioned.

There will be two features included in the mobile application that will benefit the users when verifying what is the status of the door. First is the real time Lock Status feature, which will be placed right underneath the toggle switch, it will display

whether the door is lock or unlock. The last feature is the push notification feature, which needs the user's confirmation in able to use it. To turn it on, the user will have to click the three horizontal bars located on either top right or top left of the screen as seen in Figure 16.

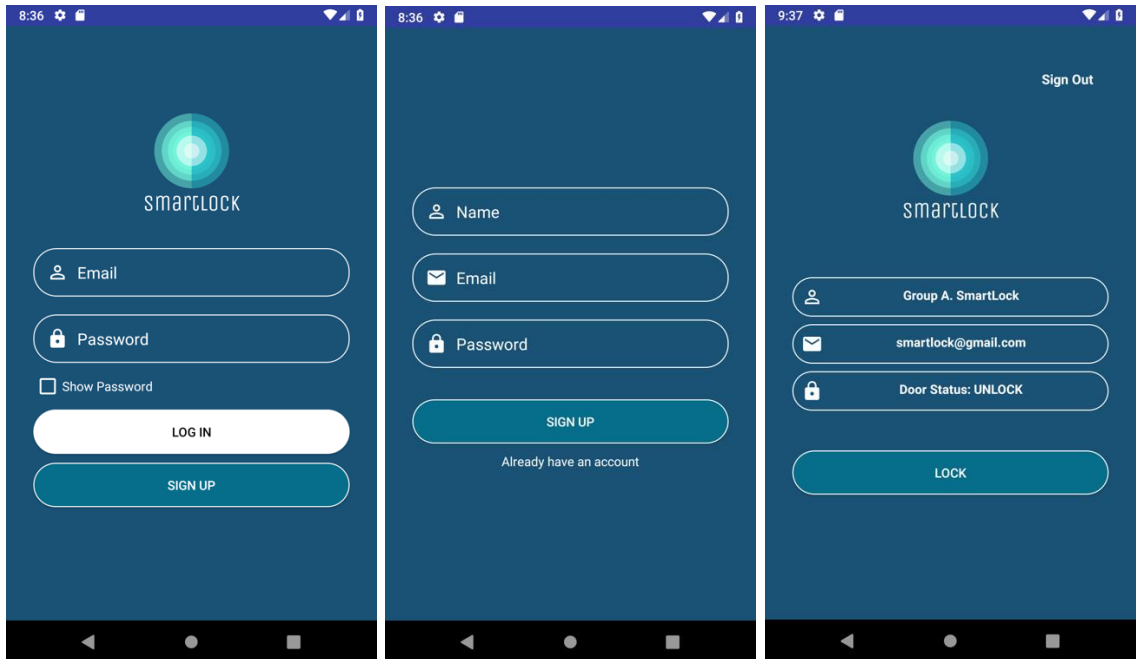


Figure 15 (from left to right) is the Login, Register and the Main pages of the Android Application

After clicking it, a drop-down menu will appear and an option that display notification will be listed. User must click the notification menu to go to the notification page where they can turn on or off the push notification. Once that is set to on then the users will start receiving information whenever the SmartLock has either locked or unlocked by someone. As mentioned before, the notification is omitted to the design, nevertheless, a popup message will appear everytime the unlock/lock button is pushed. It will have the name of the user who did the action and when the action was sent to the server.

Both mobile applications will have to be connected to the internet to be able to have the application communicating to the SmartLock. Failing to connect to the internet will prevent both SmartLock and the mobile application from sending and or receiving signals to one another. A poor internet connection may also result from the situation previously mentioned or can delay the SmartLock and the mobile application from sending and or receiving signals to one another.

One of the problem that was discussed when designing the push notification feature is that users can spam the SmartLock and can locked or unlocked or vice versa the door in a short period amount of time which will result to multiple notifications being sent to the mobile application. In order to not have this problematic situation, SmartLock would wait five seconds after the last activity to

push the notification to the user. With that being said, the last activity will be the one that will be push instead of all the activity within that five second window.

As previously mentioned, to use the mobile application users are required to create an account and register their SmartLock. To store all the information about the users a database is needed. Being said that, MySQL database will be used to store the user's information such as their email address, usernames and password. In addition to that, lock and unlock events are necessary to be stored as well as the information of the registered SmartLock. Initially, there will be three tables; user table, event table and SmartLock table, more tables will be added as seen necessary.

One of the condition which is one of the most important condition when using the mobile application is for the smartphone which has the mobile application must be connected to the interned failing to do can result to problems from simple as delay of execution to mobile application is not doing what is supposed to be doing.

If the user's internet connection is slow it can cause signals from the mobile application to get delayed. With a slow internet connection signals would buffer and would get delayed resulting for the SmartLock to receive that signal later than it is supposed to. In addition, connection may get cut off due to slow connection meaning it would try to get the signal once again until it properly receives it then it will try to send signals back to the SmartLock which previous situations might happen again.

Failure to connect to the internet will without a doubt result to signals not sending and or receives by both smartphones and SmartLock. Since the smartphone is not connected to the internet, which is the only connection between the mobile application and the microcontroller will result in no signal being transmitted between them.

6.3 Mechanics and Power Supply

The mechanical door lock subsystem includes the deadbolt, the exterior keyhole, the interior locking/unlocking lever, the servomotor and its associated parts, and the power supply for the servomotor. The lock microcontroller will also be powered by this higher voltage power supply. This portion of the design must not only work correctly every time, but time and effort must be put into user experience. The user experience is easily the most important portion of a customer facing project such as this one. There may be brilliant engineering and programming going on behind the scenes, but if the user experience is suboptimal, the project will be judged harder by the common person. As such, extra time and effort must be put into making the mechanics of the lock not just be secure and reliable to the user, but also appear that way to the observer.

6.3.1 Servo Motor

The main design concerns for the servo motor are power consumption, size, and speed. The module is battery powered, so active power consumption is a concern. The servo motor must also be small enough to fit inside of the door lock housing. One of the initial design constraints was a low latency from user input to the door being unlocked. Thus, a higher torque servo motor is required. After extensive research, the motor the design team chose is a TowerPro MG995R Metal Gear Servo. With dimensions of 4cm x 2cm x 4.2cm, it will easily fit inside any lock housing. At 6V, the motor has 10kg-cm of torque and takes .16 seconds to move 60 degrees. The cost for this motor is slightly higher than comparable motors, but the higher cost is justified with how well the part fits design constraints.

The motor will be retrofitted to a standard deadbolt using a 3D printed gear system. A 1:1 gear ratio will be used, as the motor has sufficient torque to turn the lock without gearing up, and doing so would unnecessarily add to the unlock time of the module. Gearing down for a faster unlock time would lead to a less reliable system, since doing so also significantly lowers the torque output of the servomotor.

The motor has three wires to be connected. An analog signal wire, a power supply and a ground. The power supply and ground will be connected directly to the battery power supply of the door module and the ground. No voltage shifting is necessary, as the power supply is six volts, which is also the maximum voltage the servo motor is rated for. The signal wire will be connected to an analog general purpose input/output pin directly on the main micro controller.

6.3.2 Power Supply

The main power supply for the door module will be powered by four AA 1.5V battery which will be 6V. This is a sufficient amount of batteries to power both the door micro controller, the low frequency transmitter, and the servo motor. The maximum voltage for the servo motor selected by the team is 6V. The micro controller used in the final design will be powered off.

Powering the door micro controller with batteries makes passive power consumption will be one of the largest constraints in designing, but the portability of a battery powered design makes the tradeoff worth it. The importance of having a modular design that is not dependent on external power cannot be understated. A door lock that must be plugged into the wall is not marketable in the slightest.

Initial project plan was using 4 AA battery for the door module. Since the servo motor drain huge power the new WIFI module, ESP32, and MCU kept reset due to shortage of power. Because of the power issue, 4 AA battery supply power only for servo motor, 2CR5 which is 6V will be supply the power for MCU and WIFI module.

6.3.3 Lock Description

The final prototype of the SmartLock is to be neatly packaged in a plexiglass housing. This is not a secure way to construct an actual door lock, however, for our prototype, it will be secure enough to keep the parts in place and give a clean final look. Using plexiglass will allow the internal working of the lock to be seen while in operation. It can be cut and shaped by the team members with tools we already have on hand. It is also a very cost-effective way to construct a housing. As computer and electrical engineers, we lack the material science and mechanical background to safely construct a home security lock housing out of metal or a stronger material, so it seems best to create a prototype that is clearly not a security device, but demonstrates the security systems we set out to achieve. The main purpose of the final prototype is to demonstrate operation, not to be a marketable product.

The keyed exterior portion of the retrofitted door lock is to be left intact to be used as a backup to the electric system. Since we are retrofitting a lock rather than constructing our own, leaving the existing mechanics in place for the lock is actually easier than fabricating our own to work with the servomotor. The small exterior keyed portion will be sunk into a plexiglass housing along with the exterior button. The interior housing will be a much larger plexiglass housing and contain all of the electronics and power supply for the system. The mechanical deadbolt will be taken from the lock to be retrofitted. A single shaft runs through the lock housing from the keyed portion, through the deadbolt, to the internal lever used to lock/unlock. This shaft must be turned up to 120 degrees to extend or contract the deadbolt. Simply adding a gear to the shaft will allow the servomotor to accomplish this.

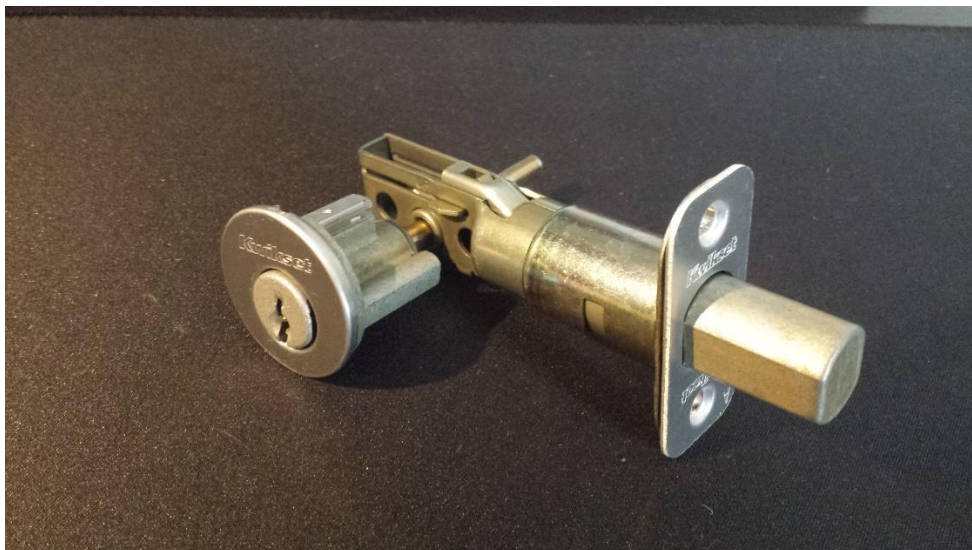


Figure 16 Image lock Kwikset lock to be retrofitted



Figure 17 Top view of lock to be retrofitted

By viewing figures 17 and figure 18, it becomes evident that attaching a servo motor to a standard door lock while still leaving the exterior locking mechanism intact is a straight forward task. A gear must simply be added to the shaft on the interior side for the servo motor's own gear to mesh with. The deadbolt takes very little torque to turn, allowing the team to choose a low powered, small, 6V servomotor. The shaft only needs to rotate over a range of 100 degrees total for the lock to fully function, therefore the servomotor selected with a 1:1 gear ratio will both have sufficient torque and range of motion to have a fully functional automated deadbolt.

6.4 Status Indicators

As we open or close the door, the lock need to detect if the door is closed or opened. There is a door jamb sensor which is located at the door frame. So, when the door is closed then the door jamb sensor will be folded, the smart lock will know that the door is closed. On the other hand, when the door is opened, the door jamb sensor will not be folded. Then, the smart lock will know that the door is open. There are several indicators if the smart lock is locked or unlocked. We decided to use different Patten of sound and different LED lights as status indicators. As the door lock is closed, the red LED light will be on. On the other hand, if the door is open, the green LED light will be on. Also, if the door is not closed completely, the smart door will not be able to lock the door, then the red LED light will be blinked as waring sign. As the LED light indicator, three different pattern of sounds will be sued for opening, closing, and warning.

During the senior design 2, most of status indicators have been changed as well. Since our project requires easy installation, door jamb sensor was not able to use.

6.4.1 Speaker System

The speaker system is designed with the different tone generator, amplifier, and output which is speaker. The speaker system will receive signal from the microcontroller if the door is opened, closed, or fail to close. Microcontroller will send a signal to speaker system in different order, the signal will be passed through the tone generator, then send to the amplifier, then will be arrived at the speaker.

6.4.1.1 Tone generator

There are three different patterns if the door is opened, closed, or not able to close completely. the tone generator makes three different tone level. Tone generator will have three input which will make different tone level for each input. To generate different tone levels, the tone generator will generate different frequencies. There are two different sample of tone generator are listed.

During the senior design 2, the pins were shortage since WIFI module also connected to digital pins. Instead using three pins, we just decided to use only one pin for the speaker and generate different frequency from MCU.

6.4.1.1.1 Comparison of Tone Generator

As the smart lock need different sound tone, tone generator will be needed. As microcontroller send out the output signal to the tone generator, the tone generator will be able to make different tone sound levels depend on what output signals are. Tone generator will generate different frequency wave form which will be different tone level sound. There are well designed tone generators such as LM tone generator and 555 timer tone generator.

LM 3909 Tone Generator

The tone generator used with LM 3909 can generate different frequency oscillation. LM 3909 is designed as oscillator, so it is usually used for the flashing LED light, or beep sound. The maximum power of the chip is 6V. There are 4 resistors as external components, and one capacitor. The power dissipation is 500mW.

555 Timer Tone Generator

The tone generator used with 555 timer can generate a range of the output frequency. The rise or fall time of output is 100ns which is quick. Voltage high will be same as V_{cc} , voltage low is about 80mV. The maximum power consuming of the 555 timer is 1180mW. Two resistors and two capacitors exist as external components. Maximum supply voltage is 15V. The following table shows that the comparison of the specifications for each designed tone generator.

Feature	LM 3909	555 timer
Supply voltage	5V to 25V	4.5V to 16V
Power dissipation	500mW	1180 mW
Operating temperature	-25°C to 70°C	To 70 °C
Output low	N/A	200mV
Rise time of output	N/A	100ns
Fall time of output	N/A	100ns
Frequency range	Typ. 1.1kHz	500kHz to 2MHz

Table 8 Specifications of Tone Generator

- Supply Voltage: supply voltage is the required voltage to run the chip. The voltage is supplied through V_{cc} terminal.
- Power dissipation: The power is needed to run.
- Operating temperature: the temperature range that the chip will work properly. High temperature will cause error or disfunction.
- Output low: As the output signal is oscillation, high voltage and low voltage exist. Output low is the voltage at low.
- Fall time/ rise time of output: the time takes to fall to high or high to fall.
- Frequency range: the frequency range that the chip can supply.

Tone generator what we are going to use, should be able to generate the frequency clearly in low voltage input. Also should have wide frequency range. The big requirement is power consumption. Since our project's power will be supplied with the battery, power consumption has to be low. LM 3909 power consume is lower than 555 timer, but the tone generator with the 555 timer is still fit in our requirement. The power consuming of 555 timer is bearing pass 1W. Moreover, there are only 4 components as external components, two resistor and two capacitor. Also, only one resistor will be used for discharging capacitor. Since there are few components exist as external, power consumption will be low. 555 timer tone generator also can generate the square wave clearly in low input voltage. To generate the square wave clearly, the low voltage should be close to zero, so then you can tell this is high voltage or low voltage. 555 timer generator's low voltage is 0.08V which is really closed to zero. So the generator will be able to supply clear output in low power input. The output of 555 timer generator will be able to control the duty cycle. By changing the two resistors and one capacitor, the frequencies of the high voltage and low voltage can be adjusted. It will be flexible to adjust the tone level for the future.

6.4.1.1.2 Design Tone Generator

555 timer is composed 8 pins, ground, trigger, output, reset, Vcc, discharge, threshold, and control. It is composed with three resistors which are same value, and two voltage comparators, SR Flip-Flop, and transistor as switch. To brief, the voltage comparator has two inputs and one output. If the voltage at the positive input is bigger than negative input, the output will be 1, if not output will be 0. For the SR flip-flop, if S is 1, Q is 1, if R is 1, Q is 0. If both are zero, Q will not be changed. Three resistors in the 555 timer used as voltage divider. The positive input of first voltage comparator is threshold, and the negative input is connected to the voltage divider which has $\frac{2}{3}V_{cc}$ V. the positive input of second voltage comparator is connected to the voltage divider which has $\frac{1}{3}V_{cc}$, and negative input is trigger. The transistor is used as switch. the base of transistor is connected to inversed output of SR flip-flop. If the base of transistor is 1, switch is closed. If not, the switch is opened. The output stage works as NOT-Gate.

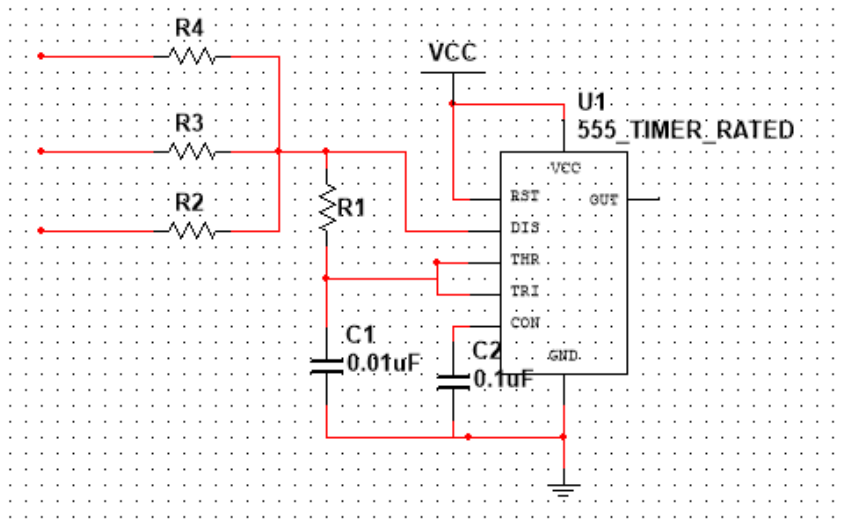


Figure 18 Schematic Diagram of Tone Generator

Figure 19 is schematic diagram of tone generator. As you can see, the threshold and trigger are connected. There are three inputs have different values of resistors. Therefore, the output frequency will be generated differently. The output of the tone generator will be square waves which have different frequencies. Once the 555 timer is connected to the power, the capacitor, C1, will be charged through resistors, R1 and R2. While the capacitor is charging, the voltage across of the capacitor will be low. at the same time, the voltage between C1 and R1 will be low as well. So then the output of the second voltage comparator which is has trigger at the negative terminal will be 1. The output of the comparator is connected to the S of SRFF, the inverse of the output will be 0. As the inverse output is 0, the transistor switch will be closed. Since the inverse output is connected to the NOT-

gate, the final output will be 1. Once the capacitor, C1, is charged between $\frac{1}{3}V_{cc} < V_c < \frac{2}{3}V_{cc}$, The output of two voltage comparators will be 0, the output of Flip-flop will not be changed. When the voltage across capacitor is greater than $\frac{2}{3}V_{cc}$, the output of voltage comparator will be 1, the output of Flip-Flop will be 0. At the same time, the transistor switch will be closed, the power of capacitor will be consumed by the resistor and the transistor.

The calculation of frequency of astable mode of LM 555 timer

High voltage output time $t_{high} = 0.693(R + R_1)C_1$

Low voltage output time $t_{low} = 0.693(R_1)C_1$

The total time period is $T = t_1 + t_2 = 0.693(R + 2R_1)C_1$

The table below is the calculation summary for each input signal.

	Input R2	Input R3	InputR4
R value	1000Ω	2000Ω	4000Ω
High voltage time	0.000014	0.000021	0.000035
Low voltage time	0.000007	0.000007	0.000007
Total time period	0.000021	0.000028	0.000042
Frequency	47.619kHz	35.714kHz	23.809kHz

Table 9 Calculated Frequency with Different Value of Resistor

However, due to shortage of pin out, we are using only one pin for the speaker and generate different frequency to make melody.

6.4.1.2 Amplifier

Since the signal from the microcontroller will be low voltage, the tone generator will not be able to generate the tone property. To be make sure, using the amplifier, we can amplify the signal, so then the tone generators will be able to generate the tone property. Since the signal voltage from the microcontroller is between 0V to 5V the gain has to be around 5 gain will be needed to make enough sound level.

6.4.1.2.1 Comparison of Amplifier

Once the door is unlocked or locked, the smart lock will let user know if the door is locked or unlocked with sound and LED light. As the microcontroller is low voltage output, the smart lock will not be able to generate enough sound level. As we need louder sound system, amplifier will be used for sound system for smart lock. There are well designed amplifier circuit which are amplifier with RC4580ID, and LM 386.

RC4580ID Amplifier

One of the candidate designing amplifier is RC4580ID, audio amplifiers dual audio oper amplifier. This is designed for tone controlling audio application. RC4580ID can be used for high gain band width, high output current, low noise, or low harmonic distortion. Using this dual audio amplifier, some electronic device could be made such as preamplifiers, active filter, or measurement equipment. Since RC4589ID is low voltage consumption, it also can be used as low power voltage application which is fit our project requirement.

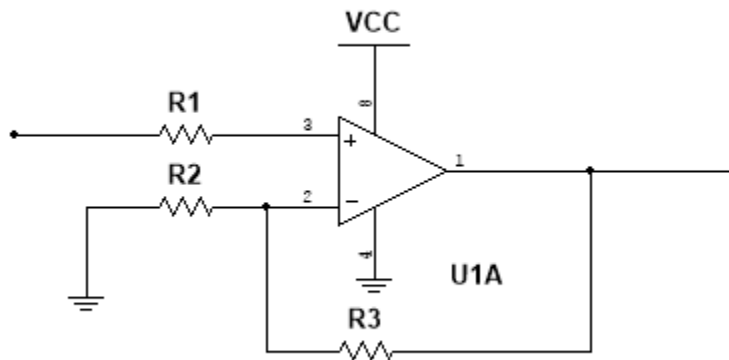


Figure 19 Schematic of RC4589ID

The figure 20 shows non-inverting amplifier schematic design. As the figure shows, there are only three resistors are used for non-inverted amplifier. To the positive terminal input, the input of signal will be received through resistor, negative terminal input will be connected to the ground through another resistor. As the designed amplifier is non-inverted amplifier, negative terminal and output will be connected with one resistor. Since there is no negative input signal, the voltage limit will be between Vcc and ground. The gain will be calculated as $A_v = 1 + R_3/R_2$.

LM386 Amplifier

LM386 chip also can be designed with audio amplifier. LM386 is the power audio amplifier which is designed for low voltage consumption application. This amplifier usually used for AM-FM radio amplifier, intercoms, TV sound system, small servo driver, or power converter. The biggest advantage is the gain could be between 20 and 200 with consuming low voltage. LM386 composed with 8 pins, two gain, power, ground, positive input, negative input, bandpass, and output. The schematic of the LM386 audio amplifier is below. The following figure 21 shows designed audio amplifier.

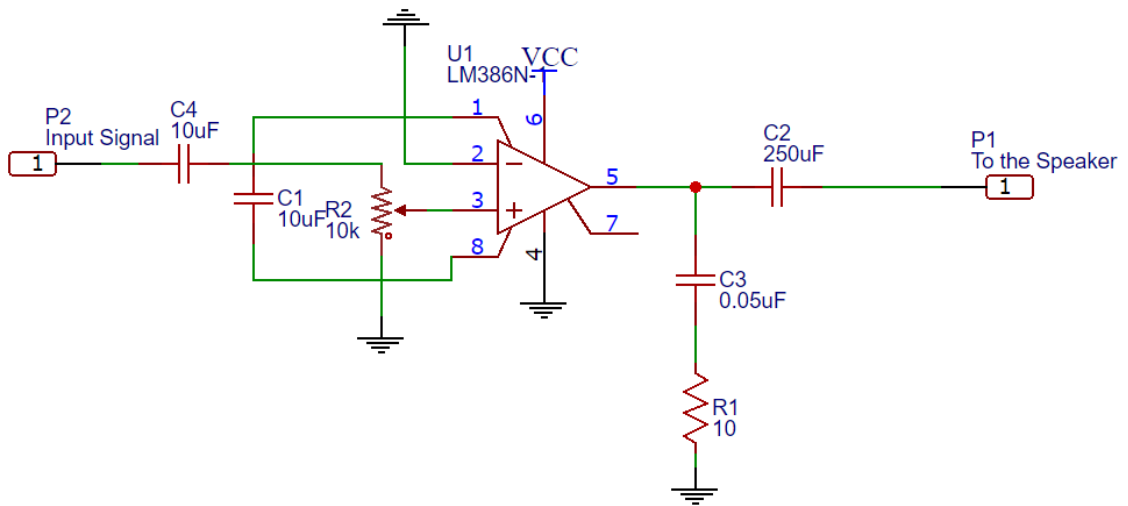


Figure 20 Schematic of Audio Amplifier

As the figure 21 shows above, four capacitors are used, and two resistors are used for the audio amplifier. The power supplied through pin 6, Vs. two gain pins are connected through capacitor. Other three capacitors are used for the stability of DC power. The resistor at the positive input terminal will be used to adjust gain. Bypass will not be used for this design. Table shows the comparison of two different amplifier specification below.

	RC4580ID	LM 386
Quantity of components	3	6
Max Power consuming	N/A	0.73W
Voltage gain	Depends on resistors	26dB
Supply voltage	2V to 18V	4 to 12V
Output power	N/A	325mW
Output current	50mA	N/A
Gain bandwidth	12MHz	300KHz
Range of temperature	-40 C to 125C	0 C to 70C

Table 10 Specifications of RC 4580ID and LM 386

Table shows that the summarizations of specifications of both RC4580ID amplifier and LM386 amplifier. It shows that the number of external components and the max power dissipation, voltage gain range, supply voltage range, and output power. RC4580ID amplifier does not require a lot of component which will be less cost for designing. However, LM 386 will need more components then the RC 4580ID amplifier. As the data from the datasheet for each amplifier, LM 386 power consumes around 0.73W, but RC 4580ID datasheet does not show power consuming. The operating supply current is around 3mA, and the operating supply voltage is between 4V and 32V. So, the power consumption could be calculated, and it will be around 60mW. The power consumption needs to be measured. LM 386 amplifier voltage gain is 26dB if the supply voltage is 6V. 26dB is 20 gain of

voltage. Both RC4580ID and LM386 amplifiers do not need huge voltage supply. The output power of the LM386 audio amplifier is 325 mW if supply voltage is 6V. Therefore, the efficiency of the audio amplifier is $\frac{325}{730} = 44.52\%$. By comparing both RC4580ID and LM386 audio amplifier, LM 386 audio amplifier is well designed for amplifying of audio signal. However, RC4580ID designed well for amplifying of signal which means it may not work with audio signal property. Even though RC 4580ID could consumes less power than LM 386, since LM 386 audio amplifier is designed well for our project requirement, LM 386 will be used for sound system amplifier.

6.4.1.2.2 Design Amplifier

LM 386 composed as 8 pins to connect. There are two gain, negative or positive input, bypass, Vs, Vout, and ground. Vs is power supply voltage, bypass is bypass decoupling path. The gain of amplifier will be decided how the pin 1 and 8 are connected. Both pin 1 and 8 are gain pin. If they are connected together through the capacitor, 10uF, the gain will be 200, if the pin 1 and 8 are connected through one resistor, 1.2KΩ, and one capacitor, 10uF, the gain will be around 50. If the pin 1 and 8 are not connected each other, it will be 20 gain as default.

As the figure in the section 6.4.1.2.1, one capacitor is used between pin 1 and 8. Pin 6 is connected to the voltage supply, pin 2 and 4 will be connected to the ground, and pin 3 will be connected to the input signal. Pin 7 will not be used for this design, pin 5 is output. The schematic circuit is designed as gain 200. The requirement of the design is flowing below.

Design Parameter	Required Value
Load Impedance (Speaker Resistance)	4 ohm to 32 ohm
Supply Voltage	5 V to 12 V

Table 11 Requirement for LM 386 Amplifier

The resistance of the speaker what will be used for this project is 8 ohm mini speaker, and supply voltage will be 5V, the designed amplifier should be able to work well.

6.4.2 LED Indicators

Other statue indicator is LED indicators. The smart lock will tell the user if the door is locked, or unlocked completely, or the door was not able to be closed. In the LED indicators, red LED and Green LED will be used to let user about the statue of the smart door. If the door unlocked successfully, green light will be on, if the door locked, then red light will be on. If smart lock was not able to lock or unlock,

then red light will be blink. For designing LED lights, the time of LED lights on, and blinking LED light will be controlled by the microcontroller.

In senior design 2, some designs are changed for LED indicator. Red LED will be used to know if the door is locked or unlocked. Green LED and Yellow LED are used if the WIFI module is connected to the server.

Table shows that the specifications of LED lights which will help to design circuit.

6.4.2.1 Types of LED Light

There are three candidates LED lights what will be LED indicators for this project. First is 5.9mm RGB LED light. Since we need two colors which are red and green, we need to buy two LED lights. If RGB LED is used, however, only one LED will be used which is benefit for cost and space for design. There are two 5mm LED lights as different type. Single LED light will be needed two LED lights for Red and Green. The cost of LED light will be cheaper, but it will be needed two and will take more space than RGB LED light. The table will show the specifications for each LED light below.

Product Attribute	5.9mm RGB LED	5mm LED	5mm LED_2
Illumination Color	RGB	Red or Green	Red or Green
Wavelength_Red	625nm	630nm	642nm
Wavelength_Green	525nm	573nm	527nm
Luminous Intensity	1100mcd	8200mcd	20000mcd
View Angle	60 deg	15 deg	55deg
Forward Current	20mA	20mA	20mA
Forward Voltage	2.2V, 3.3V	2.1V,2.2V	3.2V
Length	5.9mm	5.8mm	5.5mm
Width	5.9mm	5mm	5mm
Max Temperature	85 C	100 C	95 C
Min Temperature	-40 C	-40 C	-40 C
Cost	\$2.05	\$0.15	\$0.25

Table 12 LED Lights Specifications Comparison

The table shows that the different specifications between three LED lights. RGB LED is very useful for space. However, since luminous intensity is only 1100mcd which is pretty weaker than other LED lights, it will be hard to see the light in the daytime. Moreover, the cost is much higher than other LED. By comparing second LED and third LED, the biggest difference is the forward voltage. Second LED consume 2.1V or 2.2V, but third LED consume 3.2V which is bigger than second LED. It will cause more power consumption. According to the data, second LED will be used for LED indicator. The brightness is enough to see in the daytime, and it is really cheap.

6.4.2.2 Design Circuit for LED Indicator

As 5mm LED is chosen in the section 6.4.2.1, the circuit needs to be designed as the LED light parameter. All LED lights has different value of forward voltage, forward current, reversed voltage, and so on. The parameters of 5mm LED light what have chosen in the section 6.4.2.1 are shown table below.

Parameter	Red 5mm LED Light	Green 5mm LED Light
Forward Voltage	2.1V	2.2V
Forward Current	20mA	20mA

Table 13 Parameter LED lights Will be Used

As the table shows, the forward voltage is little bit different between Red LED light and Green LED light. Forward voltage means that the voltage across the LED light needs to be that value. As the forward voltage is given as 2.1V and the output voltage from microcontroller is 5V, LED light cannot be connect to the microcontroller directly. A resistor needs to be used to drop voltage, so then the dropped voltage will be equal to forward voltage of LED light. The forward voltage of Red LED light is 2.1V which means that other voltage, 2.9V, needs to be dropped by the resistor. Forward current is 20mA. According to Ohm's law, resistor value will be calculated as $R = \frac{V}{I} = \frac{2.9V}{20mA} = 145\Omega$. Same method will be applied to Green LED light to calculate resistor value which will be $R = \frac{V}{I} = \frac{2.8V}{20mA} = 140\Omega$. Since LED light is diode, the current has to flow property. There are two pins at the LED light, longer pin is anode and shorter pin is cathode. Anode is positive and cathode is negative. Therefore, longer pin will be connected to the resistor what has calculated and shorter pin will be connected to ground.

6.4.3 Door Jamb Sensor

The last indicator of the smart lock is door jamb sensor. Door jamb sensor is input indicator which sends signal to the smart lock. Door jamb sensor tells that if the door is closed or opened. The door jamb sensor is located at the edge of the door. So, if the door is closed, then the door jamb sensor will be closed and let the smart lock starts lock the door. There are two different type of door jamb sensor.

During working on senior design 2, as project has changed, door jamb sensor also changed. Instead using door jamb sensor, we installed another feature which key fob can lock or unlock the door. Using mobile app or a button inside of house can also lock or unlock the door as well. Having door jamb sensor can be more complicated to install which violate our project goal.

6.4.3.1 Type of Door Jam sensor

Magnet door jamb sensor

Magnet door jamb sensor composes with one magnet and one switch. So, the switch will be installed at the door frame, and the magnet will be installed at the door. So, when the door is closed, the magnet will be closed to the switch at the door frame. Then one of the iron will be closed to the magnet, the switch will be closed. On the other hand, when the door is opened, the magnet will be far way from the switch. the iron will be back to the initial position by the spring. The switch will be opened.

Physical door jamb sensor

Other door jamb sensor is door jamb sensor works physically. The sensor will be installed in the hold at the edge of door. There is ON and OFF switch at the sensor, if the door is closed, the switch will be pushed by the door frame, the ON/OFF switch will be closed. On the other hand, If the door is opened, the switch will be back to open since there is nothing to push ON/OFF switch.

Both door jamb sensor, magnet and physical works based on same concept. If the ON and OFF switch is closed, then the sensor will be closed circuit which can tell microcontroller that the door is closed or opened. However, we need to make a hole inside of the door to install the physical door jamb sensor, the installation will be much harder than magnet door jamb sensor.

Magnet door jamb sensor is much easier to install than the other door jamb sensor. For magnet door jamb sensor only need to attach a magnet on the door side, and the switch on the door frame side. Also, the price is not even expensive. Because of the two reasons, we decided to use magnet door jamb sensor instead of using the physical door jamb sensor.

6.4.3.2 Schematic of Door Jamb Sensor

As the door jamb sensor is opened or closed, microcontroller should be able to receive the signal. Since the door jamb sensor does not have electrical signal, it is designed as switch to make open or close circuit. The following figures 22 shows schematic of door jamb sensor.

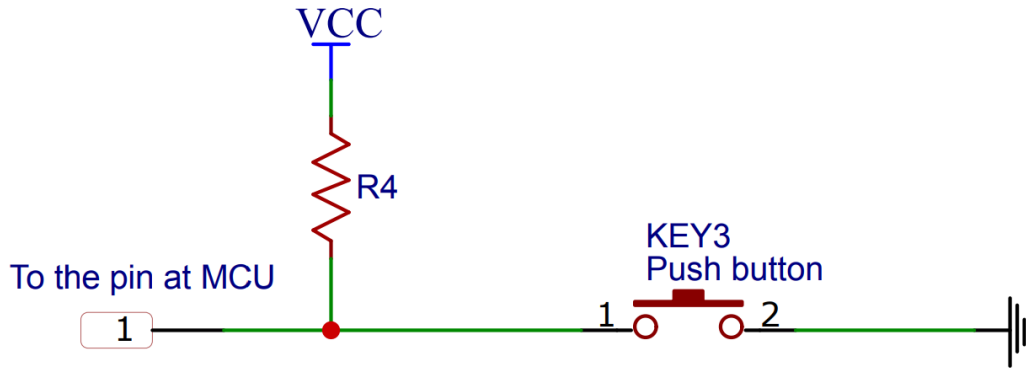


Figure 21 Schematic of Door Jamb Sensor

As the figure 22 shows that if the door jamb sensor is opened, the circuit will be open circuit, which means that the pin which is connecting to MCU does not connect to the ground, the voltage value of the pin will be 5V. However, if the door jamb is closed, button is pushed down, the circuit will be close circuit which means that the pin will be connected to the ground, the voltage at the pin will be 0. By getting the voltage value at the pin, microcontroller will know that if the door is closed or not.

6.5 Power supply and Distribution

Since the project, smart lock does not consume a lot of power, we decided to use battery to supply the power. It is hard to supply stable power to the microcontroller. Any leak of power might cause of failure of processing, or extremely high voltage might cause burns chip. As microcontroller is electronic devise, it is required to receive stable DC power voltage.

To supply stable DC power voltage, voltage regulator will be used. Voltage regulator supplies stable power source to any electrical device within acceptable limits. For example, if it is 5V voltage regulator, the output of the voltage regulator will be 5V, no matter what input voltage is. In the main board, there is microcontroller, ATmega328p, and wi-fi module. Microcontroller need to supply as 5V and wi-fi module works well in 3.3V. Since two different voltage needs to supply, two different voltage regulators will be connected as parallel. Voltage regulator, AMS 1117-5.0, is used for 5V supply, and AMS 1117 – 3.3 is used for 3.3V supply. The table shows the specifications of two voltage regulator.

Parameter	AMS 1117-5.0	AMS 1117 – 3.3
Drop out voltage	1.1V	1.1V
Output current	1A	1A
Line regulation	0.2% Max	0.2% Max
Load regulation	0.4% Max	0.4% Max

Temperature range	-40 C to 125C	-40 C to 125C
-------------------	---------------	---------------

Table 14 Voltage Regulators Specifications

As the table shows above, the maximum output current around 1A. since the system is designed as low power consumption, it does not necessary to supply bigger amount of current. The following figure 23 is the schematic of power supply in the system.

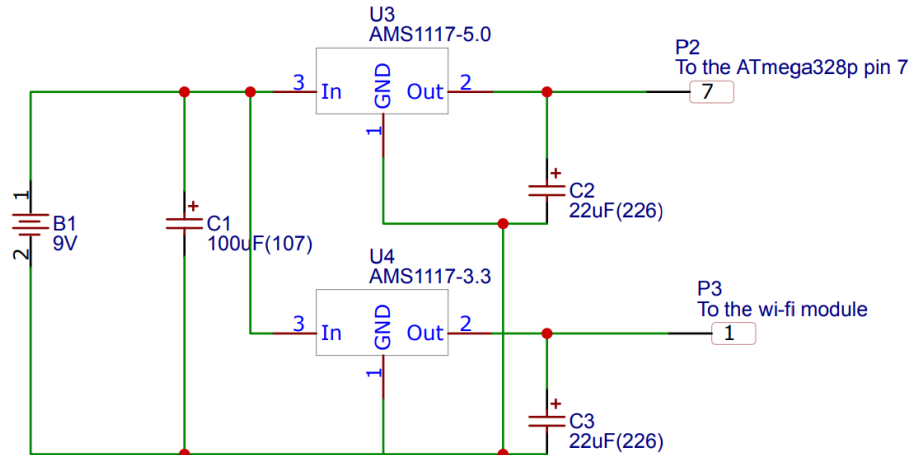


Figure 22 Schematic of Power Supply

As the figure 23 shows, two voltage regulators are used to supply two different voltage. Also, there are three capacitors are used as well. the capacitor is needed for stable voltage. The capacitor makes the ripple voltage much smoother, which will supply much stable voltage to the MCU. Since the voltage regulator connected as parallel, only one capacitor is used, two capacitors are used for each output of voltage regulator.

During the senior design 2 project, the 3.3 voltage regulator connected to the output of 5 voltage regulator.

6.6 Microcontroller

Microcontroller or MCU is small chip which is programmable, single integrated circuit (IC). Microcontroller composed with one or more than one CPUs with memory and programmable input or output peripherals. Microcontroller are designed for embedded system application [9].

Our project, smart lock, also require the microcontroller to control smart lock. There is keyless entry system, Wi-Fi module, input/output indicators, and output device which is servo motor. There are two modules which need to communicate with microcontroller. Wi-Fi module is used for communicating with server and microcontroller to keep update if the door is opened or closed. Also, keyless entry system needs to keep communicating with microcontroller due to open the door if the key fob is closed to the door module. Moreover, there are several input or

output device. There are two input sensors which need to connect to the microcontroller and four ports of pin will be used for output devices. One port will be used for speaker system, two ports will be used for LED lights indicator, and one port will be used for servo motor.

ATmega 328/P

ATmega328/P is high performance and low power 8 bits microcontroller, which combine flash memory, input/output ports, 32 general registers, three timers, internal/external interrupts, programmable USART, and so on. Atmega 328/P composed with 28 pins dual inline package. ATmega 328/P has two 8 bits timer with separate prescale and compare mode, and one 16bit timer as well. There is programmable watchdog timer and interrupt and wake up on pin charge. It has 14 digital input/output pins can be used, and pinMode(), digitalWrite(), and digitalRead() functions will be used for programming. Also, analog to digital converter. [10]

MSP 430G2x

MSP 430G2x is microcontroller which is designed for low cost and low power consumption. One of the powerful feature of MSP 430 is low power consumption. The current drawn in idle mode can be less than 1uA. There are six different low power modes. Also the response time is less than 1ms. MSP 430 composed with 14/20pin DIP socket, and 16KB flash, 512B RAM, and 16MHz CPU. There are integrated peripherals which are ADC, timers, serial communication UART, and so on. MSP 430 also has five power saving modes. Table shows the comparison between ATmega328/p and MSP 430G2x below. [11]

Feature	ATmega328/P	MSP 430G2x
CPU type	8 bits AVR	16 bit RISC
Operating Voltage	1.8-5.5V	1.8 – 3.6 V
Recommended Input Voltage	7-9V	1.8 – 3.6 V
Digital I/O pins	6	24 Input/output capacitive-touch enabled pins
Analog Input pins	6	
DC Current for I/O pins	40mA	48mA
Flash memory	32KB	16KB
SRAM	2KB	512B
EEPROM	1KB	
Clock speed	16MHz	16MHz
Temperature Range	-40-85 °C	-55-150°C
Speed Grade	0-20MHz at 1.8 – 5.5 V	8MHz at 1.8-2.2 V
Low power Consumption @ 1MHz, 1.8V	Active: 0.2 mA	Active: 230uA Standby: 0.5uA

	Power-down Mode: 0.1uA Power-save Mode: 0.75uA	Off: 0.1uA
--	---	------------

Table 15 Comparison Between ATmega328 and MSP430G253

By comparing with ATmega 328\P and MSP430G2x, MSP 430G2x consume less power than ATmega328\P. the operating voltage is lower on MSP430G2x as well. Also, the active mode, standby mode, or power save mode, the MSP430 is lower than ATmega328\P. For this project, the microcontroller does not need to be waken up for all time, the active mode of power consumption will not important. The microcontroller will be on standby mode until the push bottom is pressed by user. For smart door lock project, ATmega 328\P will be used for microcontroller. Active mode consumes more power, but since the microcontroller will be down for long time, ATmega328\P will be better power saving since power down mode is really small. Also, it is fit with the input/output pin quantity, and bigger process than MSP 430.

6.6.1 Peripheral Features

There are so many different peripheral features depends on what microcontroller is. In the section 6.6, few peripheral features which are going to be used for the project, smart lock. It also will explain how the peripheral features of the microcontroller will be used and where the peripheral features used for.

The main peripheral features of ATmega328p are the three different bit of timers, real time counter, PWM channels, programmable serial USART, 2 wire serial interface, programmable watchdog timer, analog comparator, and interrupt and wake up on pin change.

6.6.1.1 Sleep and wake-up peripheral

One of the main register of ATmega 328p is sleep mode control register. The sleep mode control register will manage the microcontroller what mode it will be. 3 bits of register will be used for sleep mode select between five available sleep modes, idle, ADC noise reduction, power-down, power-save, standby, and extended standby.

As in previous section which have discussed about the project smart lock, most important thing is the power saving since the power is supplied with battery. Because of the reason, the microcontroller cannot be waken up for all the time. One of the peripheral feature which is interrupt and wake-up on pin change will be used.

6.6.1.2 Timer/Counter with PWM peripheral

ATmega328p has 8 bit or 16bit timer/counter peripheral. It is a general timer module with two independent output compare unit and PWM support. The timer/counter normally used for two independent output compare units, clear timer on compare match, phase correct pulse width modulator(PWM), frequency generator, or three independent interrupt sources.

In the project, smart lock, servo motor will be used to lock or unlock the door. Servo motor is rotary actuator so then it will be able to control of angular position. The motor will be controlled by sending a signal, pulse width modulation (PWM). As the servo motor is controlled by the PWM, the timer/counter with PWM peripheral will be needed to control servo motor. According to the pulse width, the servo motor position will be decided. For example, if 1.5ms pulse width makes the servo motor to rotate 90 degrees, shorter than 1.5ms of pulse width will make servo motor to rotate 0 degree. If pulse width is longer than 1.5ms, the motor will rotate 180 degrees.

6.6.2. Microcontroller Schematic

In the section, microcontroller schematic, will shows that how the sensor or modules will be connected to the microcontroller. In the chip ATmega 328p has 28 pins, 13 digital pins, 5 analog input pins, power, reset, and so on. it will be important to connect to right pin with any sensor or module. Depends on how to connect to the pin, will be caused to build different code as well. The figure 24 below will shows schematic of microcontroller

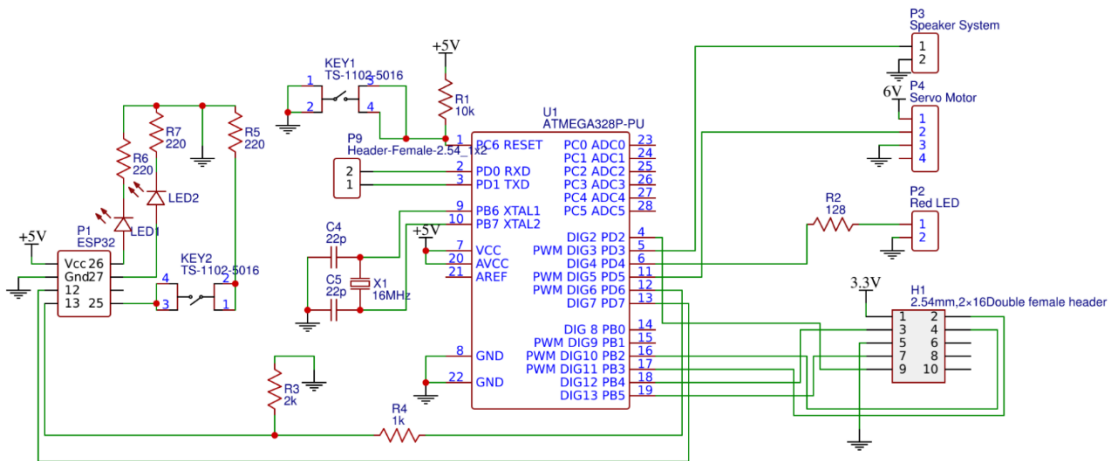
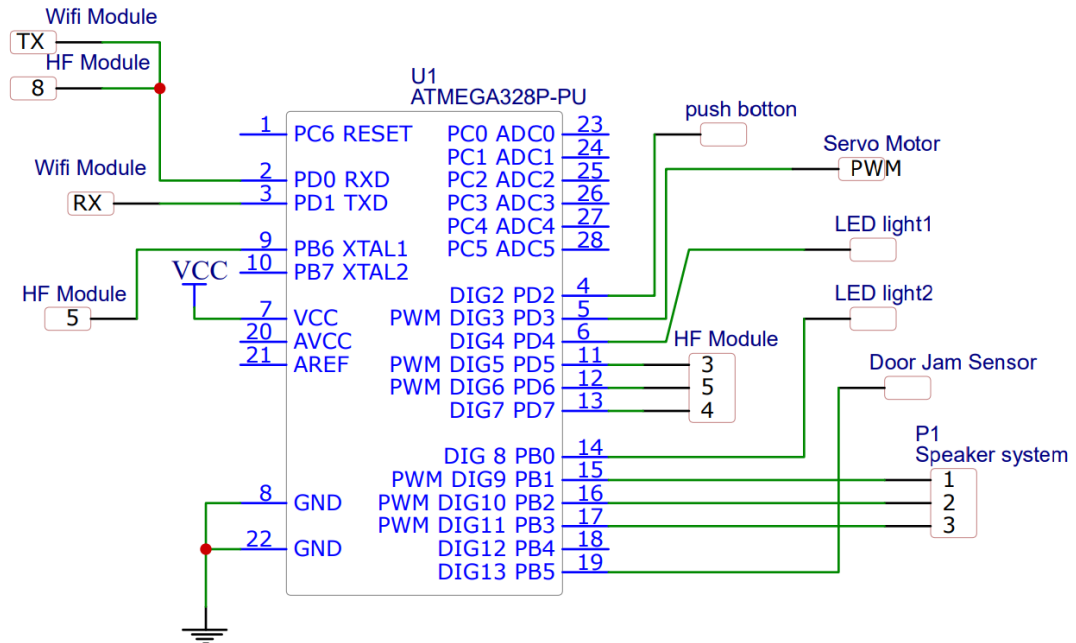


Figure 23 Original Schematic of Microcontroller (Above) and New Schematic Design (Below)

Connect the push button between the 5V power and the ground from microcontroller. Use one resistor to connect the push button. Then connect one of the input/out pin to the node between resistor and the push button. Since the push button open unless the button is pressed, it will be open circuit which means that there is no current goes through. Therefore, the signal at the pin will be 0V. if the push button is pressed, then the current will go to the ground, the signal will read the voltage.

In the new design of the microcontroller schematic, it shows that the connection between WIFI module and microcontroller. Also, with CC1101 as well. As mentioned one Red LED will be used and crystal oscillator is used for the microcontroller since the MCU was not working without crystal.

7.0 Prototype Construction

Prototype construction of the SmartLock will be done in several phases with varying levels of complexity. The thought process behind this is that because the team is relatively inexperienced in creating our own electronic systems, it would be too much too great an undertaking to jump straight to laying out parts on a printed circuit board, ordering the circuit board, soldering everything to it and serially programming it. Therefore, construction of the final prototype will take place in three phases. The zeroth phase will simply use an Arduino Uno development board alongside the extra modules chosen. Parts will be connected via a breadboard and programming will be done using the serial port on the Arduino. This phase, while it counts towards nothing as far as final design is concerned, will give the team valuable experience in working with new embedded peripherals as well as serve as a proof of concept that the design is indeed feasible. The first prototype will be similar in spirit to the zeroth, except rather than using a full Arduino Uno, the individual components of the microcontroller will be purchased and on a breadboard. Programming for this phase will be done using an Arduino Uno as an in system programmer and the breadboard microcontroller will be very similar to the design of the Arduino Uno, but will only include the components needed for the SmartLock to function. The purpose of this phase of prototyping is to build confidence in among the team that the design is valid, as well as giving the team an easy to troubleshoot microcontroller for the first prototype. If any circuit pathing is not valid, it is much better to discover on a breadboard where the design can be changed quickly without any hassle. This prototype is the goal for the team to reach before the first presentation in Senior Design 2.

After this breadboard first prototype is fully functional, the next logical step is to move the components from breadboard to printed circuit board. By this phase in development, we will have proven that the design is fully functional and the jump from breadboard to PCB should be an easy one. However, the team was told to give a preference to surface mountable parts over through hole components, meaning that many of the bread boardable components that worked well for the first prototype can no longer be used. From both a financial and design standpoint, this isn't a very far leap for the team to make. All components cost around five dollars, so double purchasing through hole and surface mountable parts is acceptable for the amount of time it will save in troubleshooting. The pinout of the through-hole components is, in most cases, identical to that of the surface mountable parts, so no design change is required and only a small extra cost is taken on by prototyping in phases.

Prototype construction can take place anywhere for the early two phases, however, tools in the senior design lab such as the soldering irons, oscilloscopes, function generators, and more will be required for construction of the final prototype that will be using surface mountable parts.

7.1 Parts Acquisition

This section will mostly consist of the current known parts required to build the prototypes outlined in section six. They are broken down by their respective systems they will be attached to. There are duplicates of some parts due to the first prototype being one hundred percent breadboard while the final prototype will consist of almost entirely surface mounted parts on a printed circuit board. This double purchasing of parts has been accounted for in the budget and was decided the time saving of troubleshooting on a breadboard as well as the certainty of having a viable circuit outweighed the small losses of buying both through-hole and surface mountable parts.

Main Microcontroller Parts		
Description	Model Number	Status
Microcontroller	atmega328P	to be acquired
Wi-Fi Module	ESP8266 SMT	to be acquired
Wi-Fi Module	ESP 8266	Acquired
Transmitter	KGEA-SMD-B-0162J	to be acquired
HF Receiver	CC 1101	Acquired
AA battery	Duracell	to be acquired
AA battery holder	PartsExpress 140-972	to be acquired

Table 16 Required Parts for the Main Microcontroller

Lock Hardware Parts		
Description	Model Number	Status
Plexiglass Housing	n/a	to be fabricated
Deadbolt	DL61-F	owned
Servomotor	MG995R	to be acquired
Gear	n/a	to be fabricated

Table 17 Required Parts of the Physical Locking Mechanism and its Housing

Key Fob Parts		
Description	Model Number	Status
Microcontroller	MSP430g2553	to be acquired
Transmitter	CC1101	to be acquired
Receiver	AS3933	to be acquired
Battery Holder	PartsExpress 140-760	to be acquired
Battery	CR2032	to be acquired

Table 18 Required Parts for the Key Fob System.

Miscellaneous		
Description	Model Number	Status
Launchpad	Arduino Uno	owned
Launchpad	MSP430g2553	owned
Server	Raspberry Pi v3b	owned
Capacitors	n/a	to be acquired
Resistors	n/a	to be acquired

Table 19 Miscellaneous Parts and Equipment Needed for Programming Parts and Running a Self-Hosted Server

Table 16,17,18,19 are show what parts will be used for our project. Since projected has changed, some of the parts are added or removed. In table 16, WIFI module changed as ESP32, Transmitter is removed since we do not use low frequency wake up signal, 2CR5 battery is added for door module. In table 18, microcontroller is used as Atmega 328P, low frequency Receiver is removed. Due to MSP430 is not used for this project, launchpad will not be used as well.

7.2 PCB Vendor and Assembly

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 will be two PCB layout for the project Smart Lock, Door main board and Key module PCB.

7.2.1 Comparison of PCB Software

To design PCB for our project smart lock, PCB design software will be needed. There are so many PCB design software on website. One of the famous PCB software is Eagle. Eagle PCB offers 3D design, engineering software and services. Features of Eagle PCB is it is easy to use for the schematic editor. To design schematic was not hard to use. There are accessible library which is already designed. So, it is easy to access to there to use designed schematic. There is routing engine makes it possible to spend through the complex layouts with modern PCB routing tools. One of the strongest feature is after making the schematic, it is easy to make PCB design. Modular design blocks that are synchronized between schematic and PCB.

Another PCB software is EasyEDA. EasyEDA is a web-based EDA, which means that there is no install needed for EasyEDA. It is available for schematic capture, spice circuit simulation, and PCB layout tool as well. EasyEDA is also available to import design from Eagle, Altium, KiCad, and LTspice. The libraries is open source, so that any component or chip can be found easily.

By comparing these two PCB software, Eagle and EasyEDA, Eagle PCB will be better to use to design PCB layout. There were pretty much of problems occurred

while designing schematic circuit in EasyEDA. Also, there were some limited function since EasyEDA is web-based program. Eagle PCB is pretty convenient to design schematic for our circuit. Components were able to found easily. The most powerful feature is any schematic that I have designed can be easily transferred to the PCB layout. It will be really easy and save time to design PCB layout.

7.2.2 PCB Layout

As we designed the circuit printed circuit board will be used for our final project. Two PCB will be used for key fob and door module. The following figure will show our PCB for this project.

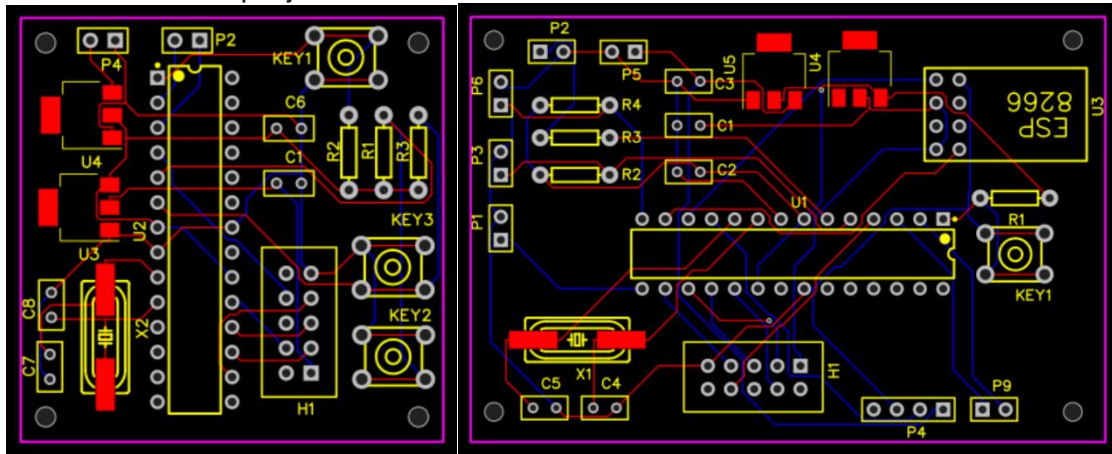


Figure 24 Key PCB (Left) and Door PCB (Right)

Figure 25 shows the both key and door PCB for our project. Since key fob should be able to carry easily, we made in half of door PCB for our project. Both PCB has reset button in case the MCU is not working property. ISP port also made to program easily. We installed a female header for the CC1101 transceiver, so that we just mount on our PCB which looks better and more stable. WIFI module changed as ESP32, only two ports will be needed for WIFI module from door PCB.

7.3 Coding Plan

The SmartLock project involves several different electronic devices that must work together in unison. There is a mobile application that must communicate with a web server that must communicate with an embedded system using TCP/IP protocols. Also, that embedded system must communicate with a second embedded system using radio waves. As any programmer knows, the largest obstacle to getting systems of this type, with many different parts, working is to get all of the electronics to reliably talk to each other. As such, special care must be taken to develop coding plans for each component. The server API must be

easy to utilize for the mobile application developer as well as the embedded programmer and so on. Additionally, the embedded systems to be programmed will not have serial ports to debug in the traditional way taught in university classes. As such, special in system programming or ISP procedures must be developed for each embedded component. These software development plans will be discussed in detail in this section.

7.3.1 Microcontroller Programming Plan

The SmartLock will have two separate embedded systems. One for the physical door lock itself and the other to be used as a key fob for sending signals to unlock the lock. As such, a comprehensive plan for development of software and firmware for the SmartLock must be created. Rather than adding a serial port to both embedded systems, an Arduino development board will be used in conjunction with the Arduino Software integrated development environment (IDE) to in system program (ISP) the micro controllers for the SmartLock. In order to convert the Arduino to a programmer, a couple of tools must be used. The Arduino IDE has an ArduinoISP sketch built into it. After programming that built in sketch to our Arduino Uno, we can use the board as a programmer for the door lock embedded system that is utilizing the atmega 328-9 microcontroller.

The hardware connections between the programming microcontroller to the atmega 328 are as follows, with the programmer microcontroller going to the microcontroller to be programmed. Pin13 to pin19, pin12 to pin18, pin11 to pin17, pin10 to pin1. Since these pins are needed for programming, the PCB design will include pads to make these connections. While programming, the main power source for the microcontroller to be programmed will come from the programming microcontroller.

The Arduino IDE is to be used as a development environment for the programs related to SmartLock's main embedded system for several reasons, namely ease of use, familiarity, documentation, it is open source, has baked in ISP tools, and the atmega 328p bootloader can be programmed using it. It is a very simple to use, straight forward piece of software, but also contains advanced features for more experienced users. Because of this, several of the design team members have already have a base familiarity with the Arduino IDE, making it an easy choice. The Arduino IDE is free and open source, and if needed, can be modified to fit the needs of the design team. In addition to this, Arduino's core mission is to make micro-computing more accessible to everyone, so their documentation is both extensive and easy to use. For the reasons stated above, the IDE is widely used by the microcontroller community. This proliferation has led to numerous online tutorials to be made, walking the programmer through almost every eventuality that one could come across.

The selected Wi-Fi module, the ESP8266, can be connected to a LAN and have its firmware updated prior to installing on the lock microcontroller by using a USB to TTL converter in conjunction with a serial port terminal. Then, a file with current firmware for the module can be downloaded from Espressif's website and programmed to the module using a serial port terminal.

The Amtel atmega 328p will not necessarily come from the factory with a bootloader, the firmware responsible for loading the operating system, installed. This can be done simply through the Arduino IDE while using an Arduino Uno as a programming microcontroller to go between the IDE and the microcontroller to be programmed while the boards are connected in the fashion stated above.

Inputs for the door lock microcontroller include the unlock button, the door jamb sensor, the Wi-Fi module, and the high frequency receiver. The door jamb sensor is a simple high/low signal used to determine whether or not the lock function is safe to execute. The unlock button, on the exterior of the lock, will trigger the wake-up signal functions. The Wi-Fi module is able to trigger an interrupt and is an input device to allow the microcontroller to receive JSON payloads from the server. The high frequency receiver inputs strings received from key fobs in the area.

Outputs for the door lock microcontroller include the tone generator, the LED status indicators, output logic to the Wi-Fi module, and output to the low frequency "wake-up" antenna. Each of these outputs will have their own output pin on the microcontroller. The LED status indicators and tone generator both have very simple logic. When an interrupt is triggered and a physical locking/unlocking action is completed by the microcontroller, the appropriate signals will be sent to these output pins. The Wi-Fi module receives JSON payloads from the microcontroller to be sent to the server. The low frequency antenna sends the low frequency wakeup signal to keys in the area.

7.3.1.1 Button press interrupt

When the exterior lock button is pressed, the button press interrupt is triggered. This brings the microcontroller out of low power mode. If the door is in locked mode, the wake-up signal function is called. If the door is unlocked, the locking function is called.

7.3.1.2 Wi-Fi module interrupt

The Wi-Fi module takes up very little power in standby mode, but is still able to be woken up through the Internet. This allows us to use it to trigger interrupts for locking and unlocking events sent from the mobile application through the server. When the interrupt is triggered, a JSON will be received with instruction to call either the lock or unlock function. An acknowledgement is sent to the server that the action was completed by either of the function calls. The mobile user that

triggered the action is recorded on the server side and is unnecessary for the microcontroller to receive.

7.3.1.3 Locking function

The locking function will only be called if the door jamb sensor reads as closed, so it doesn't need to be checked for in this function. When called, the locking function sends the appropriate signal to the servo motor to lock the door. A JSON payload is then sent to the server updating the status of the lock in the database. The LED status indicator is changed to red, the lock tone is played. Then, the microcontroller is put back into low power mode.

7.3.1.4 Unlocking function

The unlocking function takes the string received from the key fob as an input. It sends a signal to the servo-motor to turn the mechanical door lock. Then a JSON is sent to the server through the Wi-Fi module updating the status of the door lock in the database as well as creating an unlocking event with a string identifying the key that caused the event. The LED status indicator is changed to green. The unlock tone is played. The microcontroller is then put back into low power mode.

7.3.1.5 Verify key function

The verify key function takes a string received by the high frequency receiver as an argument and simply compares strings with correct strings preloaded into the memory. If a match is found, the unlocking function is called. If no match is found the microcontroller is put back into low power mode.

Wake-up Signal function: The wake-up signal function uses the low power antenna to send the low frequency signal to wake up key fobs in the immediate area. Then the microcontroller will listen for a high frequency signal to be received by the high frequency receiver. If no signal comes, the microcontroller is put back into low power mode. If a signal is received, the verify key function is called.

The key fob microcontroller is in low power mode by default. The low frequency antenna is always listening for a low frequency signal to be received. Once the signal is received, the send signal function is called. The send signal function sends the string preloaded in the key which identifies it to the door lock microcontroller. After the signal is sent, the key is put back into low power mode.

The key fob will be programmed using an existing MSP430g2553 development board as an in system programmer, similar to the programming method used with the Arduino. By connecting the appropriate pins of the MSP430 to be programmed to the MSP430 development board, the g2553 can be used as an in system programmer. This method allows for fast programming with equipment the design

team already owns, as the MSP430g2553 development board is part of the required curriculum at the university. The added benefit of using a development board as an in system programmer is the design team's familiarity they already have with the system.

7.3.2 Mobile Application

The integrated development environment (IDE) that will be used for the Android application will be Android Studio while for the iOS application it will be Xcode. Both IDE's are available online and are free download. For additional guidelines when coding, plenty of helpful and dependable online tutorials are available and are mostly free which will be extremely beneficial. For the reason that the application will be in both iOS and Android device, one member will be the lead programmer of the iOS and another one for the Android device. GitHub will be used to host both codes as it allows members to share, edit, review and manage it. Communication between the lead programmers must be constant as it would allow the applications to be similar. Apart from taking out the development of the iOS mobile application, the way the Android application was developed was not changed.

The fact that most of the members does not have any experience creating a mobile application, helpful online tutorials will be one of the primary resources to learn how to do it. Also, since both IDEs are free to download and are two of the most common IDEs out there are some of the advantages since many people have already dealt with the issues that may arise during coding the program. Additionally, since it will be written in JAVA a language that both lead programmers have already been used before and is familiar with it is going to make it easier to understand how functions/classes should be written.

Each lead programmers will work on one of the program and will assist anyone who works on it. They will oversee everything and make sure that everything is done properly and on time. Both lead members will have to communicate with each other about the statuses of their codes. Making sure that any problems is addressed and solved as soon as possible. In addition, they will have to make sure that neither one is slacking and to keep them on track. Communication will be the key in order to finish the job in a timely manner. Without proper communications many complications can arise and can slow the testing process. Coding the mobile application should not take longer than the building the hardware of the system. Failing to complete this part of the system on set schedule can result to delaying the testing. On top of that, testing that the mobile applications work with any internet connection is impossible if the application itself is nonfunctional.

To allow more successful result and prevent misunderstandings both lead programmers must always be on the same page when it comes to the design of the applications. Since the goal is to have both applications to look similar to one another designing the applications must be done and agreed by both lead

programmers. At least one physical meeting per week is scheduled to make sure that both applications are being completed at the same time. This will also be the time where any suggestions and/or problems should be talked about. Suggesting any new ideas or changing initial ideas are best to communicate during the physical meeting. This way, any questions about it can be answered with more clarity. Furthermore, showing where the problem in the code is the best way to solve it. All in all, a physical meeting would be scheduled every week to talk about any concerns about everything— how the code is written, problems running it, concerns about the designs, or any suggestions.

Being consistent on how each code is written will be extremely beneficial to all the members because it will be easier to understand and to edit if both codes are written with the same format. This means that formatting of the code must be the same for both and should remain that way throughout. For example, both should have the same outline when naming variables, declaring variables, adding comments and more,

All codes must have the following in the beginning: names of each group member, group number, semester it was written, the application that would use the code, short description about what the code is supposed to do and how to run it. The descriptions will help whoever tries to run the code and can limit questions about how to run it and what it should do. When it comes to naming the variables the camelCase format will be followed. Variables must not be a single letter with the exception of counter variables and must be meaningful as it should give an idea what that variable is used for. Required libraries must be written right after the information that was mentioned above. Additionally, declaration of the variables must be done in the beginning of the program and appropriate comments must be added. Each function/class must have at least 1-2 sentences of description before it describing what is the purpose of it. Although it is recommended to have comments excessive and unnecessary comments are frowned upon and would not be done. For each curly bracket, {}, it must be on their own line to give more clarity to the programmer. Conditions must be indented to the left as it will be easier to determine which conditions or statements are with each other.

Deciding on the fonts to use will be a group decision and will be done in person as it would prevent miscommunication. Fonts must be simple enough to understand and must be large enough to read. The minimum types of fonts that will be used is two and the maximum is four. Using fonts that are different in design will result in inconsistency and can lower the readability. Background colors are going to be neutral as to not distract and prevent users from reading the texts. Colors that are harsh on the eyes will not be used as well as fonts that are too fancy to read. Placing texts so close to one another will not be done as it will look congested and make it harder to read. Buttons will be placed on spots that are easy to see and access. Additionally, it will be large enough that users will not have a hard time using it.

To make it easier to share, edit, and comments on each code, GitHub will be use. GitHub allows each member to not only edit the code, but it also show all the changes that have been made. With this feature, anyone one can go back to previous state of the code. For example, if the code was changed without any knowledge of the lead programmer and is tried to run but is not working properly then the lead programmer can go back to the history and revisit all the changes that have been made until to the point where the code stops running correctly. Moreover, GitHub can be access using any computer that is connected to the internet limiting the issues of not being able to access the code whenever and wherever.

When running the code and warnings have occurred, as long as the warnings is not stopping the code from running it, it will be temporarily ignored since some of the instance that warnings may occur is when a declared variable has not been used in the program. Whereas, if any errors have occurred, testing will be stopped, and errors must be solved immediately to not cause any more problems in the future. If in an instance that an error cannot be solved even with hours of researching for a solution or asking other members for help, the programmer must seek help from professors preferable a Computer Science professor as they would likely know what the issue and can help solve it.

To make sure that both applications are working and are doing what its intended purpose, it will be tested using iOS devices and Android devices. Conveniently, each member has either type of smartphones and will be available to use for testing. With those smartphones already acquired as well as free internet access the cost for testing the mobile application will be zero. Connecting to multiple Wi-Fi's is one of the condition on checking if the mobile application is properly working. This means that before testing the application should already be completed and are working properly in terms of creating an account and registering the SmartLock. With the smartphones connected to the internet will test if the mobile application is communicating to the microcontroller. Multiple tests are going to be done to ensure that the functionality is working.

Since users will have to create an account where they will have to register and associate their account to the SmartLock system, a database is needed to store that information. MySQL database will be used to store the user's information such as username, password, and email address, SmartLock's information and the events. There will be a table for the users, the SmartLock, and for the lock/unlock events.

7.3.3 Server Coding Plan

To interface the SmartLock's embedded system with a mobile application through the internet, a webserver which is separate from the two is required. A more detailed breakdown of how the server API functions can be viewed in the design section of this document (6.2.2). This section will focus on design methodology. This includes software used, milestones to be achieved, and an overview of the intention behind different sections of the API rather than a rehash of what each portion of the API does and how it interacts with other systems in the SmartLock design.

The webserver will be created on a Raspberry Pi running Raspbian, a version of the Debian distribution created for use on the low power Raspberry Pi hardware. It has a full desktop graphical user interface and several USB inputs, an HDMI output, and a Wi-Fi card. This means that development for the webserver and any troubleshooting that needs to be done can be done directly on the Raspberry Pi itself.

The many jobs the server must complete make starting development of it a daunting task, but by breaking it down by its main functions, it becomes straight forward to create a design plan. The three main jobs of the server are to receive inputs and give outputs to/from the mobile application, to maintain tables in a database, and to control the microcontroller through JSON payloads. By looking at each of these tasks one at a time, a large task is broken up into several smaller ones.

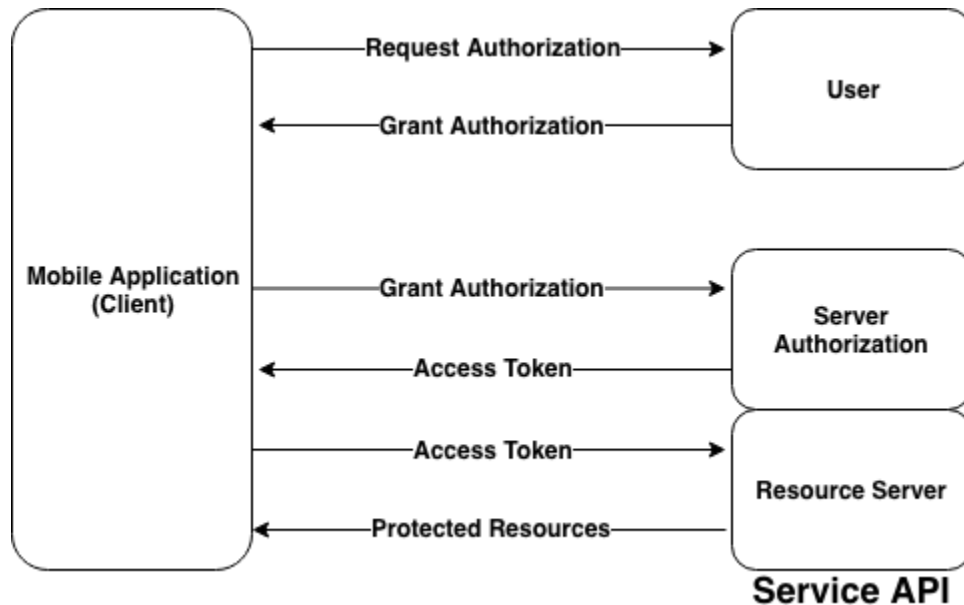


Figure 25 Visualization of user OAuth2 authentication flow.

To ensure security mobile application will use OAuth2 authentication. The basic flow of this is as follows. The user enters their name and password on the mobile application. A POST request is sent from the mobile application to the API containing the username and password. The server validates credentials and creates a token that will expire after some amount of time. The token is sent back to the mobile application and is used as an API key which gives the user of the mobile application the appropriate permissions. After the token expires, the user is prompted to enter their login information again [12]. A visualization of this process can be seen in figure 28. This authentication design allows for a secure connection to the server while also minimizing the amount of data that needs to be sent back and forth, as the token only needs to be sent once. This is a standard way of user authentication for mobile platforms.

Interaction between the server and the mobile application will require the server API to accommodate both GET and POST requests. After authenticating, the only GET request would be for status of the locks associated with that user account. This GET request should include the authentication token as well as the unique username. This is enough information to retrieve the locks and their statuses from the database to be returned to the mobile application as a JSON. The user is also able to toggle the status of a lock associated with their account through the mobile application, meaning a POST request will have to be utilized. The post request, in addition the user's id and authentication token, will require the lock to be modified and the status to change it to represented by either a one or a zero. A JSON will then be sent to the mobile application with the updated lock status if the requested operation is successful.

The microcontroller's lock status, either locked or unlocked, is dependent on itself, not the server, so the server will not need to serve GET requests from the microcontroller. Every time the lock status is changed, by either physically turning the key or by using the key fob, the lock microcontroller will send a POST request to the server. This POST request will contain the lock's unique identification number and the status of the lock. Server-side PHP scripting will use the information passed in this POST request to update the appropriate table in the database. No confirmation JSON to be returned to the microcontroller is necessary, as the status indicators of the lock are dependent on the microcontroller and not the status of the server.

In addition to receiving POST requests from the microcontroller to update status of a lock, the server must also send JSON payloads to the microcontroller to bring it out of low power mode and change its state from locked to unlocked or vice versa. This is what allows the mobile application to interact with the lock. The door microcontroller will be able to decode JSONs given to it, so all that is necessary is the requested status from the server. After the JSON is received by the lock and the status is changed, logic in the door microcontroller will send its regular POST request described in the previous paragraph.

There are several options for how development environments to create a database. Some use graphical interfaces while others are command line driven. For this project, the database will be developed using MySQL Workbench for the following reasons. It has a graphical user interface, leading to a more user-friendly experience. Command line driven development environments, while good for many purposes, tend to have a much steeper learning curve than software with a user interface. In addition to ease of use, members of the design team already have a level of familiarity with the MySQL workbench from previous courses taken at university. Also, MySQL Workbench includes tools to graphically model database elements, such as the ER diagram that can be seen in figure 14 back in the design section for the server. The combination of all this, makes MySQL Workbench a solid choice for development of the database used by the server for the SmartLock project.

The server-side scripting will all be done in PHP. This is because it is a very prevalent programming language that can interact with databases. It is free to use, quick to learn and well documented. For any issues that arise that cannot be resolved by referencing PHP documentation, there are more than likely hundreds of forum posts detailing solutions because of its prevalence in the web programming community. An advanced integrated development environment is not required to write PHP scripts. They can be developed using a simple text editor.

Rather than deploying the webserver, for early prototyping, a tunneling service such as ngrok can be used. Essentially, ngrok will connect our locally hosted webserver to their cloud services. This allows us to relay traffic through their servers to ours, letting us test our webserver without fully deploying. They provide a stable address to be able to test our API with minimal work on our part. With their paid services, it is even possible to declare our own domain names, but the free services suffice for testing the prototype SmartLock. When self-hosting a server like we are, it is easiest to use a tunneling service such as ngrok. The alternative is to make the ip address of one of our design team members static, which could many internet service providers will charge extra for.

Another alternative to self-hosting our own server would be to outsource it to a free or paid web service such as Amazon Web Services, Digital Ocean, Heroku, or GoDaddy. These web hosting platforms each come with their own baggage. Some have a high monthly cost such as GoDaddy. Others, such as Heroku don't natively support MySQL, which would create hours of extra work to either convert a MySQL database or learn a new database language all together.

8.0 Prototype Testing

As we have final designed project, we need to check if the project works property as we expect. The project, smart lock, composed as two microcontrollers, one speaker system, LED output indicator, servo motor, two input switch, and wifi-module. If the project does not work property, there is no way to find out what the problem is.

The testing phase of development, while often overlooked, is extremely important. Proper testing of each individual system as well as how they interact with each other must be done to ensure the final product is in good working condition, does not fail during regular operating conditions, and has an acceptable failure rate when placed outside its regular conditions. Any failure that is encountered must be remedied in either hardware or software so that when the system fails or gets close to failing, it will fix itself. One example of this could be if the door lock microcontroller triggers too many interrupts and begins using too much of its memory, the module will shut itself down or stop receiving instructions until it can clear the current stack of instructions. It may be ambitious to have a perfectly smooth operating project, but if adequate time for testing is built into the final schedule and deadlines are met, software and hardware bugs can be at least, kept to an acceptable minimum. The design team's testing facilities will predominantly be the senior design lab, as it has required equipment. This equipment to be used for testing and troubleshooting includes, but is not limited to, an oscilloscope, a function generator, a personal computer that can be used to serially communicate with components, soldering and de-soldering equipment, breadboards, and extra components such as resistors and capacitors.

This section will explain how the project will be tested. Since there are pretty much of modules, or input/output devices, the project will be tested one by one. As keyless entry system is main function of the project, it will be tested if the door module would be able to recognize the key fob. Also, power supply will be tested if the power will be able to supply stable power to the microcontroller, input/output devices will be tested if there are works property as the command of the microcontroller. Microcontroller and Wi-Fi module will also be tested as well.

8.1 Keyless Entry Testing

Once the user pushes the button at the door, the door module system will send signal to key fob, then key fob send ID signal to the door module. Then the door will be open. To test keyless entry system, there are several conditions will be given to test keyless entry system. As the figure shows below, there will be three tests for keyless entry system.

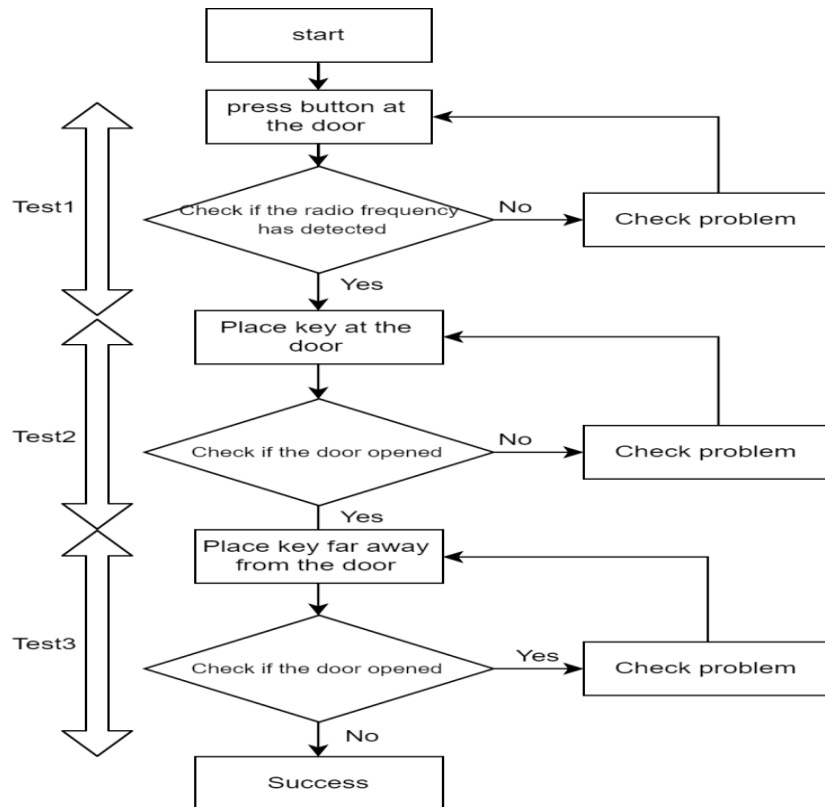


Figure 26 Keyless Entry Testing Flow Chart

First test is if the door module sends radio frequency to wake up the key fob. It is really important to check if the door module is working properly. Even though the key fob works properly, the keyless entry system will not work if the door module does not send wake up signal.

If the spectrum analyzer detects a signal, there is a peak at the specific frequency with amplitude of the signal. So, if the button pushed, the door module should be able to send the 125KHz low frequency. Then there will be peak at 125KHz from the spectrum analyzer. Second test is if the door is opened if the user holding a key fob. If the key fob is closed to the door module, the key fob will send back the ID signal to the door module. Once the door module detect the ID signal, the door will be opened. On the other hand, if there is no key fob closed to the door module, the door should not be opened. Because there is no ID signal to open the door. While doing the test 2 and 3, it is important to measure the distance between key fob and door module. It is not necessary to have long distance to communicate each other. Since the user needs to press button, the key fob will be really closed to the door module. Also, if the keyless entry system will be able to communicate under the long distance condition, it will consume more power which is not met our requirement of design.

8.2 Wi-Fi Integration Testing

To ensure proper operation of the SmartLock's Wi-Fi integration system, a rigorous test plan must be developed. Test cases will be developed to test each component of the Wi-Fi integration system individually, such as the mobile application, the server API, the microcontroller, general security of the system, as well as some stress testing that will cause the system to operate beyond what is normally expected of it. A prototype device that passes all our test cases will have proven to be sufficiently reliable, as the test cases are designed to pressure edge cases and common oversights that often occur with a networked embedded system. It is not expected that the device will pass all test cases initially. The purpose of these test cases is to expose bugs in the system and provide the design team with enough feedback that they can be patched in an effective manner.

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.3%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.4%
4.1.x	Jelly Bean	16	1.7%
4.2.x		17	2.2%
4.3		18	0.6%
4.4	KitKat	19	10.5%
5.0	Lollipop	21	4.9%
5.1		22	18.0%
6.0	Marshmallow	23	26.0%
7.0	Nougat	24	23.0%
7.1		25	7.8%
8.0	Oreo	26	4.1%
8.1		27	0.5%

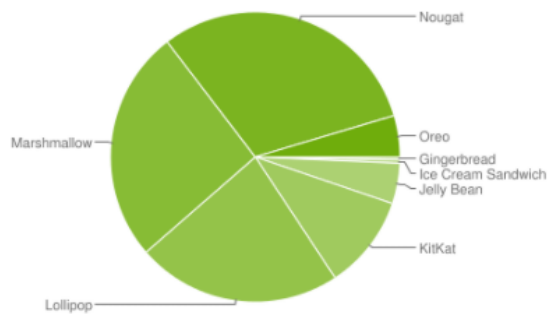


Figure 27 Breakdown of current android operation system statistics. [13]

As can be seen in the figure, nearly 95% of android users are using a device that runs Android 4.4 or newer. As such, KitKat is an excellent target for android application development.

8.2.1 Mobile Application

The target operating system for our mobile application is Android 4.4. As can be seen in Figure 31, 95% of the Android userbase is using KitKat or newer. As such, mobile application use case testing will center around those versions that are most commonly used. Firstly, to ensure the mobile application's constrained layout is

properly implemented, an emulator shall be used to test the appearance of the user interface on many different screen sizes. Proper resolution images must be utilized. No overlap between user interface elements should be observed. Both portrait and landscape orientations must be checked on many different screen sizes. All user interface elements should load properly.

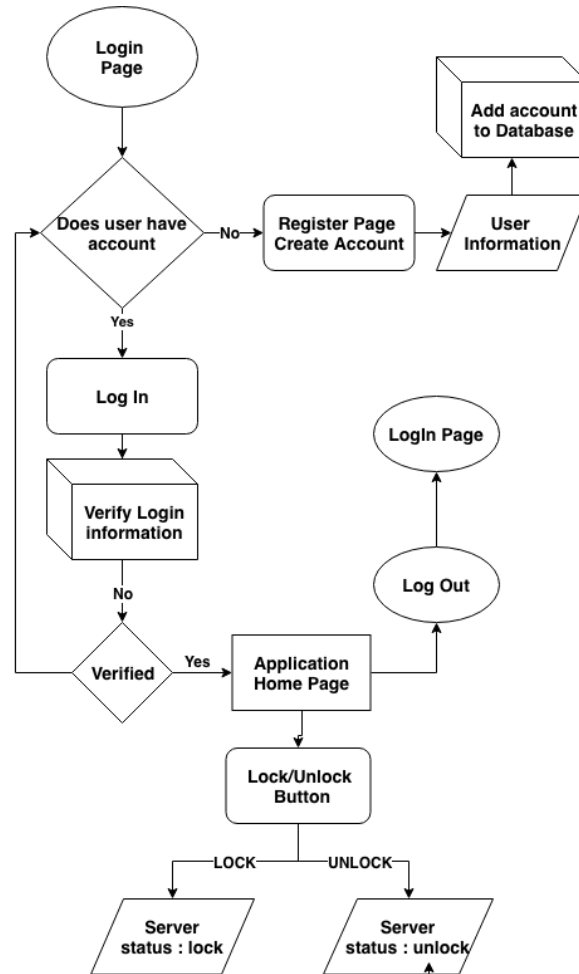


Figure 28 flow chart of the Android application lifecycle.

To understand many common bugs the lifecycle of an Android application must first be understood. This lifecycle can be seen in Figure 33. When the user first starts an application, onCreate() is called and the initial setup is completed. Because the Android operating system must save as much computing power as it can, when a user navigates away from a running application, onPause() is called and the application stops running. If the state of the session isn't correctly saved here, there will be nothing for onResume() to rebuild the session with when the user navigates back to the application and the application will start over from onCreate(). Therefore, extra care must be taken to avoid introducing software bugs at this junction. The addition of having an OAuth2 session with the server makes

it doubly important. A common oversight with new Android developers related to the Android lifecycle is when screen rotation is enabled, the Android operating system will call `onPause()` prior to rotating the screen. If the `onPause()`, `onResume()` methods are not properly initialized, the application will restart from `onCreate()` upon the user rotating the screen.

8.2.2 Application Programming Interface (API)

The operations required of the server are relatively simple, but any time computer systems must communicate with one another, hundreds of opportunities for bugs arise. Fortunately, there are developer tools to help the design team better understand what went wrong, should the system not work. Android studio's logcat can be used to ensure the OAuth2 token is being received from the server. Firstly, the database tables can be monitored to ensure that all database requests made by PHP files are being executed correctly. Incorrect or duplicate entries can be traced back to the PHP file that caused the error. Additionally, given that the lock microcontroller will be battery powered and depend on a wireless internet connection, if an action requested by a user on the mobile application fails to happen or if the server simply cannot find the lock microcontroller, a built-in error message to be displayed to the mobile application user must be integrated. User authentication is one of the most important jobs the of server API. Extra care must be taken to ensure that all mobile application functions require the OAuth2 authentication token. More on user authentication will be covered in the Security section of this section.

8.2.3 Microcontroller

Use of a microcontroller to communicate through Wi-Fi to an internet hosted server introduces the possibility of both hardware and software bugs to be introduced to the system. Also, as the microcontroller will be functioning as an embedded system without any form of output display, software bugs will be much harder to detect and compensate for after programming is completed. The best the design team can do is to complete the design as early as possible, attempt to introduce as many edge cases that could cause failure as can be, and adjust programming accordingly. One condition that could possibly cause a system failure of the microcontroller is an unexpected restart due to low battery power. The design team must ensure that the microcontroller has networking information stored in non-volatile memory so that after a restart, the microcontroller may reconnect with the local area network and re-establish connection with the server. Another area of concern is the ability of the ESP8266 to connect to many different wireless networks. This can be tested for simply by taking the module around to different wireless networks and attempting to establish a connection with the server through that wireless network. Doing this may be especially difficult on wireless networks that require a second authentication, such as the UCF WPA2 wireless network that requires student login information to make a connection. This can be compensated

for programmatically, but will take extra time and effort. Failure to connect to the university's Wi-Fi would make a major feature of the SmartLock unusable during any on-Campus demonstrations of the project and is unacceptable.

8.2.4 Security

As with any networking application, security must be a concern and precautions must be taken on the developer's side to ensure safe operation of the system in the face of potential attacks. One of the most basic and important precautions to take is to prevent SQL injections. Any application that allows user input also opens the door to possible attacks. A SQL injection could allow an outside party run MySQL statements to make modifications to the database. They could drop all the tables or worse, download them all. Fortunately, all that must be done to prevent an SQL injection is to scrub all quotation marks and semicolons from the user input fields. Without these characters, a malicious person cannot run SQL code on the server. Another basic precaution to take is to hash user passwords. User data is being sent through the internet, and sending unencrypted passwords is not advisable. It is simple enough to check that password hashing is working correctly, as they should be hashed even in the database. Additionally, the API token system must adequately protect users. The tokens must be sufficiently random and too large to brute force. Also, the design team must ensure that the tokens expire at their expected time. A buildup of unused tokens that should have expired is a glaring security loophole. Altogether, there are not many areas in which security is a concern, and basic precautions will mitigate most possible security risks.

8.2.5 Stress tests

To ensure the system will work properly under all conditions, the design team must think of edge cases that might cause the system to break. A possible cause point of contention is what might happen if the mobile application sends many instructions at once from one device. The user is capable of flipping the lock/unlock switch as fast as they can click it, we need to investigate how that will affect the entire system. With the current design in mind, the server should execute all of its commands as fast as they come in, as it has relatively few instructions to execute and then it will pass them on to the microcontroller. The microcontroller receiving a spam of inputs will trigger an interrupt for each input. It will eventually run out of memory and crash. One possible way to mitigate this is to simply restart the microcontroller if a certain condition is achieved. Possibly if it detects too much memory is in use, or if the Wi-Fi module detects too many incoming packets in a short amount of time, it can assume something went wrong and reset itself.

Another possible point of failure is if the device receives multiple inputs from different devices at once. Will the key fob be prioritized over instructions from the server? It would be best to design the system in a way that after the key fob triggers the door to lock or unlock, there is a cooldown timer before the mobile application

is able to cause a change. Any interrupt triggered by the key fob should override the mobile application.

It is difficult to account for every stressful event that could cause the wi-fi integration system to fail at this phase in the design. The best the team can do is build in time to the schedule for troubleshooting and try to think of conditions that would cause failure while developing. Without having written any code for this project and having done only vaguely similar projects in the past, many possible failure inducing conditions could exist without us even being aware. Every device that is added to a networked system of software adds its own hardware and software failure points to the system. The possibilities of bad code or hardware ruining the system become exponentially higher with each addition. Edge cases will always exist and the team can only do their best to predict them at this juncture and design around possible points of failure. More bugs will certainly make themselves known during development and testing. For this reason, several weeks must be built in to the development schedule to allow for testing and to hopefully prevent an embarrassing demonstration experience at the product's conclusion.

8.3 Input and Output Devices Testing

There are several input and output devices exist in the project. As all of IO devices works differently such as blinking light, different tone sound, or detecting. In this section, all of input or output device including sound system will be tested.

8.3.1 LED Light test

To test LED light, DC voltage generator and multimeter will be needed. The purpose of LED light is indicator if the door is opened or closed. Set the DC voltage to generate 5V because the output of microcontroller will be 5V. Connect to where the microcontroller's pin will be connected. Generate the power, then the LED light will be up. Make sure that the Red LED light and Green LED light should have similar brightness. Make sure that the voltage across LED lights should be close to the forward voltage, and the current flowing through the LED should be close to forward current as well. If the LED light is too bright, measure the current flowing across the LED and then adjust the resistor value to decrease current flowing. If the LED light does not turn on, the voltage across to the LED should be lower than forward voltage. Then, replace the resistor as lower value to make similar voltage and current as the LED is designed. Both LED light should have 20mA current flowing through the LED, and 2.1V to the Red LED light and 2.2V to the Green LED light.

8.3.2 Door Jamb Sensor test

Testing door jamb sensor is simple. By closed or opened the door jamb sensor, the circuit will be decided if the current going through the circuit or not. As testing door jamb sensor, multimeter will be needed to test voltage. First of all, open the door and let the door jamb sensor be opened. test the voltage where the pin will be connected. The voltage will be 5V while the door is opened. On the other hand, if the door is closed, the voltage should be 0. If the voltage is different than as we testing, look up the circuit if somewhere connected wrong.

8.3.3 Servomotor testing

To testing servo motor, it needs connected with power and pulse width modulation (PWM). As connecting with servo motor and microcontroller, servo motor needs to get signal from microcontroller. The motor will only work 90 degree to one side and another 90 degree to other side, total 180 degree. There is a minimum pulse, maximum pulse, and a repetition rate. PWM will be sent to the motor to move the position of motor. Then the servo mother keeps as neutral position if the pulse width is 1.5ms, less than 1.5ms of pulse width, the motor will rotate as counter clock wise, longer than that will rotate as clockwise.

To test servo motor, send the pulse signal which is 1.5ms of pulse width. See if the motor moves as counter clockwise. Send another signal which pulse width is less than 1.5ms. see if the motor rotates as counter clockwise. Send another PWM signal which is longer than 1.5ms and see if the servo motor rotates as clockwise. Check if the motor rotates as desire direction. Nonorally, the minimum pulse width is 1ms and the maximum is 2ms of pulse width. And the period of PWM signal is 20ms. The servo motor will be tested with minimum pulse width and maximum pulse width. By sending the signal for 20ms with 1ms pulse width, the motor should be rotated as 90 degree as counter clockwise. See if the servo motor rotates. Also, the servo motor needed to be measured what degree the servo motor should be rotated to make the lock is locked or unlocked. If the lock only needs 45 degree of rotating servomotor, design the PWM signal again as longer pulse width. After finding the angle of the servo motor, will be designed again for other direction of rotation. For example, if the motor needed only 45 degree of rotation, and 1.25 ms of pulse width used for the signal. Also, the neutral position of the motor was 1.5 ms of pulse width. Other side of pulse width will be calculated as pulse width = 2 * neutral pulse width – measured pulse width =
 $1.5\text{ms} * 2 - 1.25\text{ms} = 1.75\text{ms}.$

8.3.4 Speaker System

The speaker system will have three different input, and according to where the signal is from, the tone sound will be changed. In this testing only DC voltage generator and oscilloscope will be needed for the testing. Testing the first input, generate 5V DC voltage and feeding to the first input connection. See if any sound comes out from the speaker. Attach a probe from oscilloscope at the speaker, one

for positive, another one for negative. The wave form will be created at oscilloscope. Measure the frequency to compare with others. The frequency should be same as what we calculated in the table 9 in section 6.4.1.1.2. Feed the 5V to the second input. Measure the frequency again. Do same measuring as before for third input. If the frequency is measure as what we calculated, and the tone sound level comes out property, then the test is done.

If the tone level comes out differently as expected, the tone generator needs to be reviewed. As the figures 19 shows in the section 6.4.1.1.2, there are four resistor is used. By changing value of the resistor, the frequency of the output can be changed. There are the calculation for the frequency in the same section. Using the equation, re-calculate the resistor value as needed.

If the tone sound is too loud, then the amplifier needs to be modified. The designed amplifier's gain is 200, so it should be enough to hear the sound. However, if it is too loud, we need to make it smaller sound. There is a capacitor between two gain pin, pin 1 and pin 8. Add 1.2k ohm resistor between the one of the pin and capacitor, and test again. After adding a resistor at the gain pin, the sound level should be low because the gain will be changed as 50 after adding resistor. Check the sound level if it is enough to hear the sound. If the sound is still too loud, then make the gain smaller by removing all the component between the two gain pin. The gain will be changed as 20.

9.0 Administrative Content

Having gone through the design, prototyping, and testing phases of the project, this section will focus on milestones administrative, organizational aspects of the project such as milestones, timelines, expectations, and budget. While nothing exceptional will be added to the project design in this section, it is necessary to keep the team on track towards achieving project goals, maintaining clear expectations of what should be achieved by what point in time, and the all-important sources of funding. This section will serve as a guide for dividing labor and cost amongst team members, ensuring deliverable deadlines are met by using in depth scheduling of milestones, and ensuring quality deliverables are made by clearly outlining expectations of each deliverable along the way of the development process.

9.1 Milestones

This is a comprehensive list of all milestones to be completed for senior designs one and two. The syllabus for Senior Design Two has not been made available to us at this point in time, so this list may not contain all milestones related to it. Senior Design One is predominantly writing documentation, so the milestones are all centered around keeping the design team on track with their writing. Once the research phase of the semester wraps up, the table of contents is created, and roles are assigned, then the team has self-imposed deadlines for each section of writing. Five pages must be completed and turned in to the group's cloud storage by Sunday at midnight. If these self-imposed deadlines are adhered to, then the design team will be comfortably on schedule to make the forty-five and ninety-page submissions. This list of milestones is fluid, and will be updated constantly as new obstacles arise or assignments are made known to the team.

Senior Design 1 Milestones

- Divide and Conquer initial document completed
- Divide and Conquer second submission completed
- Table of Contents completed
- Divide up table of contents and assign roles
- Group five page per member deadline
- Group ten page per member deadline
- Group fifteen per member page deadline
- 45-page submission
- Group twenty page per member deadline
- Group twenty-five page per member deadline
- Group thirty page per member deadline
- 90-page submission

Senior Design 2 Milestones

- Acquire all components for construction of final prototype
- Back end of server completed
- Mobile application alpha completed (basic functionality)
- Initial, fully functional prototype using Arduino completed
- Presentation number one
- Mobile application beta completed (full functionality, basic UI)
- Finalized PCB design completed and ordered
- All part mounted on PCB
- Presentation number two
- Mobile application complete (full functionality, UI completed)
- Testing completed
- Senior Design Showcase

In the divide and conquer sections, all preliminary research was done and project requirements were drawn up. After that, a table of contents for the rest of the Senior Design 1 documentation was drawn up and divided among the team members. From there, each member of the team was responsible for five pages of documentation per week until the end of the semester and the Senior Design 1 document was due. This system was designed to break up a very large project into small achievable pieces.

The design group had all agreed at the start of the semester to skip the summer 2018 semester and finish senior design in the spring, in hopes of completing a final summer internship each before graduation. During this summer semester, no senior design work is planned, but obviously is not discouraged. Without knowing the team's full work schedules, or even what cities we would be interning in, it seems best to not plan for design work during the summer at this time. The group will have adequate time to complete the project during the spring semester. This decision may be revisited after the team's summer schedules are finalized.

Senior Design Two is difficult to plan for at this time. Currently, the prototype building phase is planned to be split into two large portions. There will be an initial prototype using development boards already on the market. The primary purpose of this phase is to ensure the server, mobile application, servo motor, and mechanical lock all work together adequately. The second prototype phase will be when the design team moves all parts to a PCB and placed in its final housing.

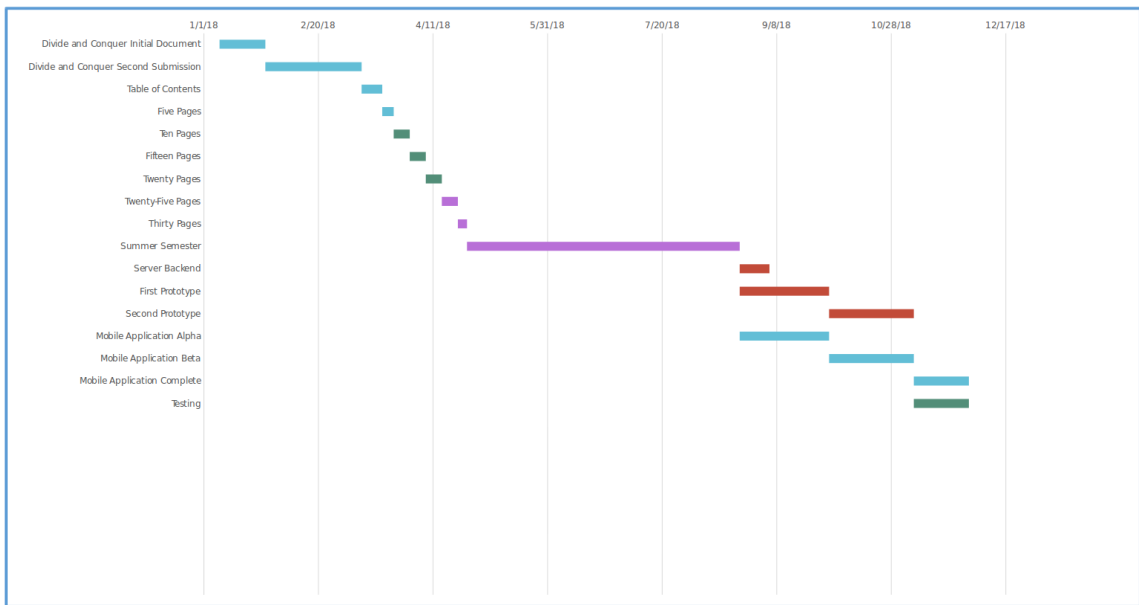


Figure 29 Gantt chart of project milestones.

Goals for Prototype Zero

1. Servomotor and Wi-Fi module connected to Arduino Uno
2. Core functionality of mobile application available (lock/unlock feature)
3. Backend fully functional with database connected and appropriate PHP scripts. Not required to be hosted on the internet.
4. Wi-Fi module receives packets through local area network.
5. Servomotor controlled using mobile application through local server.
6. Key fob created using development board able to both send and receive appropriate signals.

Goals for Prototype One

1. Arduino Uno completely removed from design to breadboard.
2. Server hosted on the internet. Mobile application and microcontroller able to communicate with it through the internet.
3. Servomotor to be used in final design connected to microcontroller and powered by external battery pack.
4. Microcontroller powered by battery pack.
5. Mobile application functionality expanded upon. Allows for additional users, creation of user accounts, creation of user/lock association creations.
6. Servomotor controlled using mobile application through internet hosted server.

7. Key fob design finalized, able to receive strings. Encryption possible implemented.

Goals for Prototype Two

1. All parts removed from breadboard to printed circuit board using surface mounted parts wherever possible.
2. Servomotor connected to door lock via a 1:1 gear system
3. Entire system mounted in plexiglass casing and able to be installed on any consumer doorframe.
4. Mobile application has full functionality as well as a complete front end with user experience in mind.
5. Testing of all subsystems and system integration completed.
6. Wi-Fi module able to connect to any domestic local area network.
7. Key fob on microcontroller moved to printed circuit board.

9.2 Budget and Finance

The SmartLock project will be self-financed by the design team. Much of the expensive equipment is already owned by members of the team, so costs can easily be kept low. However, if design cost ends up running higher than expected, it won't be a problem between team members. All additional costs will be divided up even among group members. One of the chief goals of the project is to prove that a more affordable smart lock can be made, to further increase adoption of the technology. This doubly gives the team incentive to keep costs low. Table 20 shows the total expenses including all the materials that were used during prototyping. On the other hand, Table 21 shows how much one will spend to build one SmartLock.

Purpose	Description	Model Number	Per Unit Cost (\$)	Quantity	Total Unit Cost (\$)
Main Board	Microcontroller	atmega328P	4.35	1	4.35
	WiFi Module 1.0	ESP8266 SMT	6.9	1	6.9
	WiFi Module 2.0	ESP 32	13.22	1	13.22
	Transceiver	CC1101	7.49	1	7.49
	AA Battery	Duracell	0.8	4	3.2
	Battery	2CR5	13.46	1	13.46
Lock Hardware	AA Battery Holder	PartsExpress 140-972	0.8	1	0.8
	Servomotor	MG995	9.9	1	9.9
Key Fob Parts	Microcontroller	atmega328P	4.35	1	4.35
	Transceiver	CC1101	7.99	1	7.99
	Battery Holder	PartsExpress 140-760	0.7	1	0.7
Miscellaneous	Battery	CR2032	3.8	1	3.8
	Launchpad	Arduino Uno	19.9	1	19.9
	Launchpad	MSP430g2553	10.3	1	10.3
Demo Door	Door Frame	Wood	5	1	5
	Deadbolt	n/a	11.98	1	11.98
	Hinge	n/a	2	2	4
Server	Digital Ocean	n/a	5	1	5
	Domain Name	n/a	10	1	10
PCB	PCB 1.0	n/a	8.98	3	26.94
	PCB 2.0	n/a	15.3	3	45.9
Components	Components 1.0	n/a	31.28	1	31.28
	Components 2.0	n/a	20.67	1	20.67
Total					267.13

Table 20 Overall Budget and Finance

Purpose	Description	Model Number	Per Unit Cost (\$)	Quantity	Total Unit Cost (\$)
Main Board	Microcontroller	atmega328P	4.35	1	4.35
	WiFi Module 2.0	ESP 32	13.22	1	13.22
	Transceiver	CC1101	7.49	1	7.49
	AA Battery	Duracell	0.8	4	3.2
	Battery	2CR5	13.46	1	13.46
Lock Hardware	Servomotor	MG995	9.9	1	9.9
	Microcontroller	atmega328P	4.35	1	4.35
Key Fob Parts	Transceiver	CC1101	7.99	1	7.99
	Battery	CR2032	3.8	1	3.8
PCB	PCB 2.0	n/a	15.3	1	15.3
Components	Components 2.0	n/a	20.67	1	20.67
Total					103.73

Table 21 Budget and Finance

References

- [1] [Online]. Available: <http://august.com/keyless-entry/>. [Accessed 25 4 2018].
- [2] SCHLAGE, [Online]. Available: <https://www.schlage.com/en/home/keyless-deadbolt-locks.html>. [Accessed 25 4 2018].
- [3] August, [Online]. Available: https://store.august.com/products/august-smart-lock-3rd-generation?utm_medium=cpc&utm_source=googlepla&variant=44223846915&gclid=CjOKCQjwnqzWBRC_ARIsABSMVTNVJ6WIKP8Y7g-ueXymaC0zagKBA68GxicIAkXuQbY0cF9cvTpC5jAaAv1NEALw_wcB. [Accessed 25 4 2018].
- [4] August, [Online]. Available: <http://august.com/smart-lock-compatibility/>. [Accessed 25 4 2018].
- [5] "Austriamicrosystems," [Online]. Available: <http://www1.futureelectronics.com/doc/AUSTRIAMICROSYSTEMS/AS3933-BQFT.pdf>. [Accessed 25 4 2018].
- [6] "Texas Instruments," 2018. [Online]. Available: <http://www.ti.com/lit/ds/symlink/cc1101.pdf>. [Accessed 24 04 2018].
- [7] adafruit, [Online]. Available: <https://www.adafruit.com/product/2491>. [Accessed 4 2018].
- [8] S. C. O. Traian, Wikipedia, [Online]. Available: [https://en.wikipedia.org/wiki/LAMP_\(software_bundle\)](https://en.wikipedia.org/wiki/LAMP_(software_bundle)). [Accessed 24 04 2018].
- [9] "Wikipedia," 20 02 2018. [Online]. Available: <https://en.wikipedia.org/wiki/Microcontroller>.
- [10] "Atmel," 2016. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf. [Accessed 23 4 2018].
- [11] "Taxax Instruments," 2017. [Online]. Available: <http://www.ti.com/lit/ds/symlink/msp430g2453.pdf>. [Accessed 24 4 2018].
- [12] okta, [Online]. Available: <https://stormpath.com/blog/the-ultimate-guide-to-mobile-api-security>. [Accessed 24 4 2018].
- [13] Developers, [Online]. Available: <https://developer.android.com/about/dashboards/index.html>. [Accessed 4 2018].

Copyrights Permission

August Home i

Good Morning, I am a student at University of Central Florida and am currently working on our Senior Design Project to create a smart lock system. I am requesting permission to use the figures and table from the following websites on my research paper.

 August Smart Lock Image
https://store.august.com/products/august-smart-lock-3rd-generation?utm_medium=cpc&utm_source=googlepla&variant=44223846915&gclid=Cj0KCQjwnqzWBRCARIsABSMVTNVJ6WIKP8Y7g-ueXymaC0zagKBA68GxicIAkXuQbY0cF9cvTpC5jAaAv1NEALw_wcB

 Compare August Locks Table
<http://august.com/keyless-entry/>

 Common Lock Types Image
<http://august.com/smart-lock-compatibility/>

 Thank you for your time.

 Respectfully,
 Mhelith Natavio

12:42PM

Mhelith you are welcome to use any of the information provided on our website. Good luck!

Type a message... 📎 😊 GIF 🗣️ 📷 👍

Schlage ✓ i

NOV 10TH, 10:10AM

Good Morning, I am a student at University of Central Florida and am currently working on our Senior Design Project to create a smart lock system. I am requesting permission to use the table from the following website on my research paper.

📎 <https://www.schlage.com/en/home/keyless-deadbolt-locks.html>

 Thank you for your time.

 Respectfully,
 Mhelith Natavio

Smart locks, Smart door locks, Touchscreen | Schlage

No more fumbling for your keys or wondering if you left the door unlocke...

schlage.com

Hi Mhelith,

Thank you for your message. For immediate assistance, please call Customer Support at 888.805.9837 or email at consumer.schlage@allegion.com. If you would provide your phone number or email, we will be in touch with you soon.

NOV 12TH, 2:02PM

Thank you! You're welcome to cite in your report. Good luck! -Laura

Type a message... 📎 😊 GIF 🗣️ 📷 👍

Schlage ⚙️

Options ▼

🔍 Search in Conversation

Manage Messages

Rate Experience

🔔 Notifications

Messenger Link ▼
m.me/SchlageLocks

B



Utkarsh Sinha <utkarsh@utkarshsinha.com>
Tue 7/31, 4:15 PM



Sure - go for it. It would be nice if you can cite the website as a source. Also, could you share your report / design / presentation with me? I'm just curious about how/where you're using servos :)

I hope this helps!

Utkarsh



Damo Park
Thu 4/26, 11:54 AM
utkarsh@utkarshsinha.com ↕



To whom it may concern,

Hello, I am Damo studying for Electrical engineering at UCF.

I would like to use one of the picture what you posted on website, and I would like to ask if I can use for our senior design class.

here is the link I went:

<http://www.aishack.in/tutorials/servo-motors/>

Thank you

Damo Park
dpark3@knights.ucf.edu

If you would like to see an item in the Adafruit store, let us know! Please send suggestions along with a description of the item, and why it would be good to stock! **Please note we do not offer custom products or manufacturing services at this time.**

URL/Website of suggestion:

<https://www.adafruit.com/product/2491>

Description of item:

To whom it may concern,

Hello, I am Damo studying for Electrical engineering at UCF.

I would like to use one of the picture what you posted on website, and I would like to ask if I can use for our senior design class.

here is the link I went:

<https://www.adafruit.com/product/2491>

Thank you

Damo Park

dpark3@knights.ucf.edu



Adafruit Industries <support@adafruit.com>

Fri 4/27, 5:02 PM

Damo Park



Reply all



Inbox

all good, please do.

On Fri, Apr 27, 2018 at 3:24 AM, damo park <support@adafruit.com> wrote:

contactname : damo park

email address : dpark3@knights.ucf.edu

title : requesting permisson using figure

urllofitem : <https://www.adafruit.com/product/2491>

message text : To whom it may concern,



Utkarsh Sinha <utkarsh@utkarshsinha.com>

Tue 7/31, 4:15 PM



Sure - go for it. It would be nice if you can cite the website as a source. Also, could you share your report / design / presentation with me? I'm just curious about how/where you're using servos :)

I hope this helps!
Utkarsh

...



Damo Park

Thu 4/26, 11:54 AM

utkarsh@utkarshsinha.com



Sent Items

To whom it may concern,

Hello, I am Damo studying for Electrical engineering at UCF.

I would like to use one of the picture what you posted on website, and I would like to ask if I can use for our senior design class.

here is the link I went:

<http://www.aishack.in/tutorials/servo-motors/>

Thank you

Damo Park

dpark3@knights.ucf.edu