

**University of Central Florida**

**Department of Electrical and  
Computer Engineering**

**Final Project and Group Identification Document**  
Senior Design 2 Final Documentation

**Smart Door Security System**

**Advisor: Dr. Samuel Richie**

**Group 15:**

**Adam Stefanik, Electrical Engineering**  
**Moisess Rodriguez, Computer Engineering**  
**Reham Hammad, Electrical Engineering**  
**Zachary Jackovich, Computer Engineering**

# Table of Contents

|   |    |
|---|----|
| <b>1. Executive Summary</b> .....                               | 1  |
| <b>1.1 Motivation</b> .....                                     | 1  |
| <b>1.2 Goals &amp; Objectives</b> .....                         | 3  |
| <b>1.3 Functionality</b> .....                                  | 4  |
| <b>2. Project Description</b> .....                             | 5  |
| <b>2.1 Requirements and Specifications</b> .....                | 5  |
| <b>2.2 Demonstration</b> .....                                  | 6  |
| <b>2.3 House of Quality Diagram</b> .....                       | 6  |
| <b>2.4 Smart Door Security System: Diagrams</b> .....           | 8  |
| <b>2.4.1 Diagram Status:</b> .....                              | 8  |
| <b>2.4.2 System Block Diagram:</b> .....                        | 8  |
| <b>2.4.3 Software Use Case Diagram:</b> .....                   | 9  |
| <b>2.5 Software Design Overview</b> .....                       | 10 |
| <b>2.5.1 Software Flow (version 1)</b> .....                    | 10 |
| <b>2.5.2 Software Flow (version 2)</b> .....                    | 18 |
| <b>2.5.3 Software Flow (version 3)</b> .....                    | 23 |
| <b>3. Research and Investigation</b> .....                      | 27 |
| <b>3.1 Existing Products</b> .....                              | 27 |
| <b>3.1.1 US:E Camera Smart Lock by Elecpro Group Inc.</b> ..... | 27 |
| <b>3.1.2 Camera Smart Lock by Gate labs</b> .....               | 28 |
| <b>3.1.3 FL1000 by ZKT ECO</b> .....                            | 28 |
| <b>3.1.4 August Wi-Fi Smart Lock</b> .....                      | 29 |
| <b>3.2 Similar Designs</b> .....                                | 29 |
| <b>3.2.1 Smart Lock, Fall 2019</b> .....                        | 30 |
| <b>3.1.2 Keyless Entry, Fall 2019</b> .....                     | 30 |
| <b>3.3 Technical Research</b> .....                             | 31 |
| <b>3.1.1 Wireless Communication Technology</b> .....            | 31 |
| <b>3.1.2 Machine Learning</b> .....                             | 35 |
| <b>3.3.3 Facial Recognition</b> .....                           | 37 |
| <b>3.3.4 Voice Recognition</b> .....                            | 40 |

|  |    |
|--|----|
| 3.3.5 Power Technology .....                     | 42 |
| 3.3.6 Printed Circuit Board .....                | 51 |
| 3.3.6 Low-Speed Communication Interfaces .....   | 53 |
| 3.3.7 Electrical Relay .....                     | 60 |
| 3.4 Components Research .....                    | 61 |
| 3.4.1 Occupancy Sensor .....                     | 61 |
| 3.4.2 Microcontrollers .....                     | 63 |
| 3.4.3 Other Integrated Circuits: .....           | 66 |
| 3.4.4 Central Processing Unit (CPU): .....       | 68 |
| 3.4.4.4 Arduino Uno .....                        | 73 |
| 3.4.5 LED Lights .....                           | 74 |
| 3.4.6 Accelerometers .....                       | 77 |
| 3.4.7 Electronic Locks .....                     | 79 |
| 3.4.8 Webcam .....                               | 80 |
| 3.4.9 Speaker/Intercom .....                     | 81 |
| 3.3.10 Motor Driver .....                        | 81 |
| 3.5 Parts Selection .....                        | 82 |
| 4. Standards & Design Constraints .....          | 86 |
| 4.1 Standards .....                              | 86 |
| 4.1.1 Power Standards .....                      | 86 |
| 4.1.2 PCB Standards .....                        | 86 |
| 4.1.3 Communication Standards .....              | 87 |
| 4.1.4 Electrical Locking Devices Standards ..... | 88 |
| 4.1.5 Insulation Standards .....                 | 88 |
| 4.1.6 Electric Strikes Standards .....           | 89 |
| 4.1.7 Keypad Standards .....                     | 90 |
| 4.1.8 Legal Standards .....                      | 90 |
| 4.2 Design Constraints .....                     | 91 |
| 4.2.1 Time Constraint .....                      | 91 |
| 4.2.2 Health & Safety Constraints .....          | 91 |
| 4.2.3 Ethical Constraints .....                  | 91 |
| 4.2.4 Economic Constraint .....                  | 92 |
| 4.2.5 Energy Constraint .....                    | 92 |

|   |     |
|---|-----|
| 4.2.6 Social Constraints .....                                | 92  |
| 4.2.7 Political Constraints .....                             | 92  |
| 4.2.8 Manufacturability Constraint.....                       | 93  |
| 4.2.9 Sustainability Constraints .....                        | 93  |
| 5. Project Hardware and Software Design Details.....          | 94  |
| 5.1 Hardware Design .....                                     | 94  |
| 5.2 Software Design .....                                     | 101 |
| 5.2.1 MSP430FR6989 Software Development Language Choice:..... | 101 |
| 5.2.2 Raspberry Pi Software Development Language Choice:..... | 102 |
| 5.2.3 Software Overview and Status .....                      | 102 |
| 5.3 Potential Hardware Issues .....                           | 106 |
| 5.4 Potential Software Issues.....                            | 107 |
| 6. Project Prototype.....                                     | 110 |
| 6.1 Integrated Schematics .....                               | 110 |
| 6.2 PCB Vendor and Assembly .....                             | 110 |
| 6.3 Updated PCB Design .....                                  | 110 |
| 6.3.1 Design of Buck converters .....                         | 111 |
| 6.3.3 PCB Layout.....   | 113 |
| 6.3.3 Testing and Troubleshooting.....                        | 114 |
| 6.4 Final Coding Plan .....                                   | 116 |
| 7. Project Prototype Testing Plan .....                       | 119 |
| 7.1 Hardware Test Environment.....                            | 119 |
| 7.2 Hardware Specific Testing .....                           | 120 |
| 7.3 Software Test Environment .....                           | 121 |
| 7.4 Software Specific Testing .....                           | 121 |
| 7.5 Test Results .....  | 128 |
| 8. Administrative Content .....                               | 129 |
| 8.1 Budget and Finance Discussion .....                       | 129 |
| 9. Project Summary .....                                      | 132 |
| 10 Appendices .....   | 133 |
| 10.0 Appendix A: Bibliography.....                            | 133 |
| 10.1 Appendix A: Copyright Permissions .....                  | 135 |
| 10.2 Appendix B: Data Sheets .....                            | 136 |

|  |            |
|--|------------|
| <b>10.3 Appendix C: References .....</b> | <b>137</b> |
|--|------------|

# 1. Executive Summary

## 1.1 Motivation

As people continue to incorporate advanced technology in their homes, smart locks become one of the many things people desire to have. With all the valuable products we have in our homes, we surely want to protect them. Our motivation is to enhance the traditional way of entering homes by integrating advanced as well as secure technology into a door locking system, while still maintaining easy availability for users.

A smart door with a built-in security system, gives us the ability to lock or unlock a house door without a physical key. For any door with a traditional lock, this technology will be helpful. Using the Smart Door system, a new form of authentication will be used, and the standard door key will no longer be required. This system will be much more convenient than searching for keys or juggling objects in your hands to unlock your door.

For a more in-depth notion as to the motivation behind our project, we will explain in greater detail the reasoning and thought process behind the idea of the Smart Door Security System. If you think about the typical threat model for the average home, with regards to the front door, you might think that a standard deadbolt that uses a standard metal key would be sufficient to keeping the deadbolt locked and unwanted visitors out. But unfortunately, this is not true. With a typical key and deadbolt, a potential burglar or intruder very well may just use a lock pick set and unlock your dead bolt without even needing access to the key. That potentially means *anyone* could enter your home without even having to steal or copy your house key. As we have also just mentioned, a potential burglar or intruder may just steal or copy your house key and then gain immediate access to your home. And you might not think this could happen to you, but the statistics behind the number of burglaries, specifically non-forced-entry burglaries, where the intruder doesn't have to break or force their way into the home, is actually quite high. With the advancement in technology over the last 20 years, our group sees vast potential for very important improvement and change that needs to be made with respect to securing our homes. But how can we get rid of the typical lock and key, so as to tremendously increase security and unwanted access to your home, while still maintaining easy accessibility that allows you to come home and have no trouble getting inside of your house when you have groceries in your hands or are carrying your children inside after they have fallen asleep on a long car ride? The solution is to use advanced biometric security solutions, such as facial recognition and voice recognition.

When you think about the problems that we face in terms of vulnerabilities of your current home's security, you can easily see that it is not difficult for anyone else with the correct key or even a lock pick set to gain access. This can be thwarted by using advanced implementations of facial and voice recognition. By using these

two advanced methods of authentication and validation, you can quickly realize that it would not be possible for anyone to gain access to a home, if they were to have these features incorporated in their security system.

Our group's project blossomed from first starting with the idea to create a home security system that uses facial recognition to allow the homeowner to enter. This was a great start, but we soon realized that this had a flaw. What if someone were to print out an image of the homeowner and just walk up to the security system holding that? In this instance any person could print an image of the homeowner to gain access to the home. Unfortunately, most facial recognition implementations are not advanced enough yet or have a high enough rate of success to be able to tell the difference between an image or a real human. To the security system, the camera image being processed of outside the door is just a 2-dimensional representation of pixels, no matter if it's the actual homeowner standing outside or if it's just a picture that has been printed out of the homeowner. Either way, the camera will see the same 2D image.

From analyzing this flaw, we thought about ways to better the security and make sure that this sort of situation was not left unaccounted for as it created the same vulnerability as copying the homeowner's standard deadbolt key. That is when we thought about adding a second method of authentication. From here, the options we were debating were not clear as to what would provide the best addition to our system to boost security while maintaining the same level of accessibility. We eventually decided to stick with the advanced biometric authentication methods and came up with the idea to use Voice Recognition in tandem with the Facial Recognition. Using both of these together as the two main methods of authentication create a much more thorough and secure environment, that would be very, very difficult to "break" in to. Many companies and software in use today have implemented different versions of this "Two-Factor Authentication", not necessarily with facial or voice recognition, but for example: requiring users to enter a password as well as a pin number that was sent as a text message to their phone. That way if a hacker were to possibly steal or hack the user's password, they would not be granted access unless the hacker also had the second method of authentication, which is the user's phone for the pin number that would be sent to it. This is basically what prompted our idea of using a "Two-Factor Authentication" system for our project's design.

Therefore, we finally decided to go with the Facial and Voice Recognition as the two main methods of authentication, where both need to be verified for the person at the door to be allowed access. But to make the system even more user friendly, we thought about situations that might cause the correct homeowner to be locked out of their house. One of our group members questioned what if the Facial or Voice Recognition did not work correctly for some reason? Maybe the homeowner was sick, and their voice did not sound the same, and that caused the system's Voice Recognition to think it was someone else. That would be a very bad situation. The system needs to be reliable and make sure that whenever the true

homeowner, or anyone else who has been added to the system as having access, was at the door, it must open for them. This led us to our final shift in the design, to create a third method of authentication. Here, we decided to use a standard pin number on a 10-digit keypad as the third possible method of authentication. But to not over complicate things, we did not want to require the user to have to enter all three methods of authentication to be granted access. Hence where we got the idea to have these three possible methods of authentication, either Facial Recognition, Voice Recognition or the correct pin number, but we are only going to require two of the three possible methods of authentication to be granted access to the home. This allows the homeowner a backup method, in case of facial or voice recognition not correctly identifying them. If the homeowner is sick and has a raspy voice, for example, the Facial Recognition and pin number entry would be sufficient to be authenticated and granted access to the home. It gives the homeowner some peace of mind knowing if one of the two advanced recognition systems were to fail, they would still be allowed to enter their home.

Finally, if for some catastrophic reason, two or even three authentication methods were to fail such as both biometric recognition systems, we decided to give our system one final backup method to enter the home no matter what. That would be a 6-digit pin number – not to be confused with the 4-digit pin number that is one of three standard authentication modes. But this 6-digit pin number would act as a “Master Pin” that would grant access at any time, no matter if two factors of authentication have been validated or not. This would give the homeowner total peace of mind to know that no matter what happens, they always have their backup “Master Pin” number to enter to be able to get into their home. Because at the end of the day, our system would be useless if the homeowner were to ever be locked out of their own home.

Hopefully now you have a good idea of the overall motivation of our system’s design. Our Smart Door Security System must be reliable and accessible, almost as easily accessible as a standard lock and key, if not more accessible. All the homeowner must do is walk up to the door, speak out loud and be seen by the system’s camera, and the door will automatically unlock. Also, the Smart Door Security System must be very secure. We are designing this system to be much more secure than a standard deadbolt lock and key. There should be no easy way for an attacker to be able to pick our lock or hack their way into the system. The system must be so secure that only the true homeowner, or anyone else who has been given access to the home, will be authenticated, and granted access to the home.

## **1.2 Goals & Objectives**

The fundamental goal of this project is to design, build, and test a fully functioning, cutting edge smart door lock, that will use advanced authentication such as facial and voice recognition to authorize and unlock your home’s door. The Smart Door system will use two-factor authentication to unlock a home’s door, whereas just



the facial or voice recognition alone will not be sufficient to be allowed to enter. As a backup, the user will be allowed to enter a pin on the Smart Door's 10-Digit keypad, as the second form of authentication, in tandem with either the facial or voice recognition. This feature is included in case of situations where the homeowner's face or voice might not be recognized at that time, for example if the homeowner is sick and cannot speak with the same biometric qualities as they normally do when they are healthy. The Smart Door Security System will use an external sonar sensor to ensure the system is not running the high powered computations when it is not necessary. The Smart Door Security System will be practical, and have the same accessibility, if not better accessibility as a standard deadbolt and key locking system.

## **1.3 Functionality**

The functionality of this project involves improving standard electronic door locks by leveraging two-factor authentication, including more advanced authentication methods such as voice and facial recognition. This system will be controlled by a Raspberry Pi, which will collect data from the sensors and process it before making the decision to unlock the door. The sensors will include a microphone for voice recognition, a camera for facial recognition, and a touchscreen keypad for entering a pin number as well as for sending other prompts and messages to the user. The authentication will include any two of the following options: voice recognition, facial recognition, or pin entry on the keypad.

This system will reduce hassle for the user, while also enhancing security. The project functionality will also include multiple colors of LED lights to display various states that the system may be in, such as when the system detects a person that it will attempt to authorize, or when the system has authorized and granted a user access. While the system is idle or no one is near the camera, the lights will be off. Then the LEDs will turn white when the system wakes up from "sleep mode" and will remain white while waiting for user input. The system will be interrupted from sleep mode when someone is detected within range of the sonar sensor. A red light will be displayed if authorization is denied, then it will turn yellow if one factor is authenticated. When two factors are authenticated, the light will turn green, and the door will unlock. The system will also include a speaker that will be used to welcome the user home once the authorization has been successfully completed. Upon closing, the door will automatically lock itself.

## 2. Project Description

### 2.1 Requirements and Specifications

The system will have to meet a variety of requirements and specifications which will ultimately provide quality assurance as well as guaranteed functionality. At a high level the door must open when the requirements are satisfied. Table 1 breaks down the requirements and specifications for the Smart Door Security System (S.D.S.S.).

| ID | Specification      | Metric   | Showcased in Expo   |
|----|--------------------|--|---------------------|
| 01 | Facial Recognition | Shall recognize authorized users 95% of the time. A user will approach the system 10 times, the system shall recognize the users face at least 9 times.  | Yes                 |
| 02 | Voice Recognition  | Shall recognize authorized users 95% of the time. A user will vocally command the system 10 times, the system shall recognize the user voice at least 9 times.                                   | Yes                 |
| 03 | PIN entry          | Shall recognize PIN 95% of the time. A user will enter a PIN 10 times, the system shall recognize the PIN at least 9 times.  | Yes                 |
| 04 | Unlock             | The door shall unlock 95% of the time if and only if two out of the three authentication methods are validated. If the system is tested 100 times, then the door shall unlock at least 95 times. | Yes                 |
| 05 | LED Status lights  | Shall indicate correctly 95% of time. If the system is tested 100 times, then the LEDs shall display the correct sequence of lights at least 95 times.   | Yes                 |
| 06 | Notifications      | Shall be sent upon sensing an unauthorized person 95% of time. If the system is tested 100 times, then notifications shall be sent to the user at least 95 times.                                | <i>Stretch Goal</i> |
| 07 | Sonar Sensor       | Shall signal the S.D.S.S. to start running the advanced facial and voice recognition upon sensing movement 95% of time. If the system is tested 100 times, then the Sonar sensor shall           | Yes                 |

|           |                   |   |                     |
|-----------|-------------------|---|---------------------|
|           |                   | initiate the start of the system's advanced computations at least 95 times.   |                     |
| <b>08</b> | External Speaker  | Speakers shall announce a greeting message 95% of the time. If the system is tested 100 times, then the speaker system shall announce a greeting message at least 95 times. | <i>Stretch Goal</i> |
| <b>09</b> | LCD Screen        | The S.D.S.S. shall implement an LCD Screen to print messages and prompts to users.  | <i>Stretch Goal</i> |
| <b>10</b> | Auto relock       | The door shall relock upon closing 95% of the time. If the system is tested 100 times, then the door shall relock automatically at least 95 times.                          | <b>Yes</b>          |
| <b>11</b> | Input Power       | The S.D.S.S. shall meet all requirements specified in this table when supplied with 12V Input Power.  | <b>N/A</b>          |
| <b>12</b> | Input Power Usage | The S.D.S.S. shall not consume more than 120W.  | <b>N/A</b>          |

Table 1: Design Specifications

## 2.2 Demonstration

For the demonstration we will be performing a series of 10 tests. The system will recognize two of the three following authentication methods at least 9 out of 10 times; facial recognition of the user, voice recognition of the user, or the user's PIN entry on the keypad. Since the other features are integrated into the system, they will be running sequentially. Such as LED status lights changing and the door unlocking when two out of the three authentication methods are validated.

## 2.3 House of Quality Diagram

Figure 1 shows our House of Quality chart, with a more detailed explanation below the image.

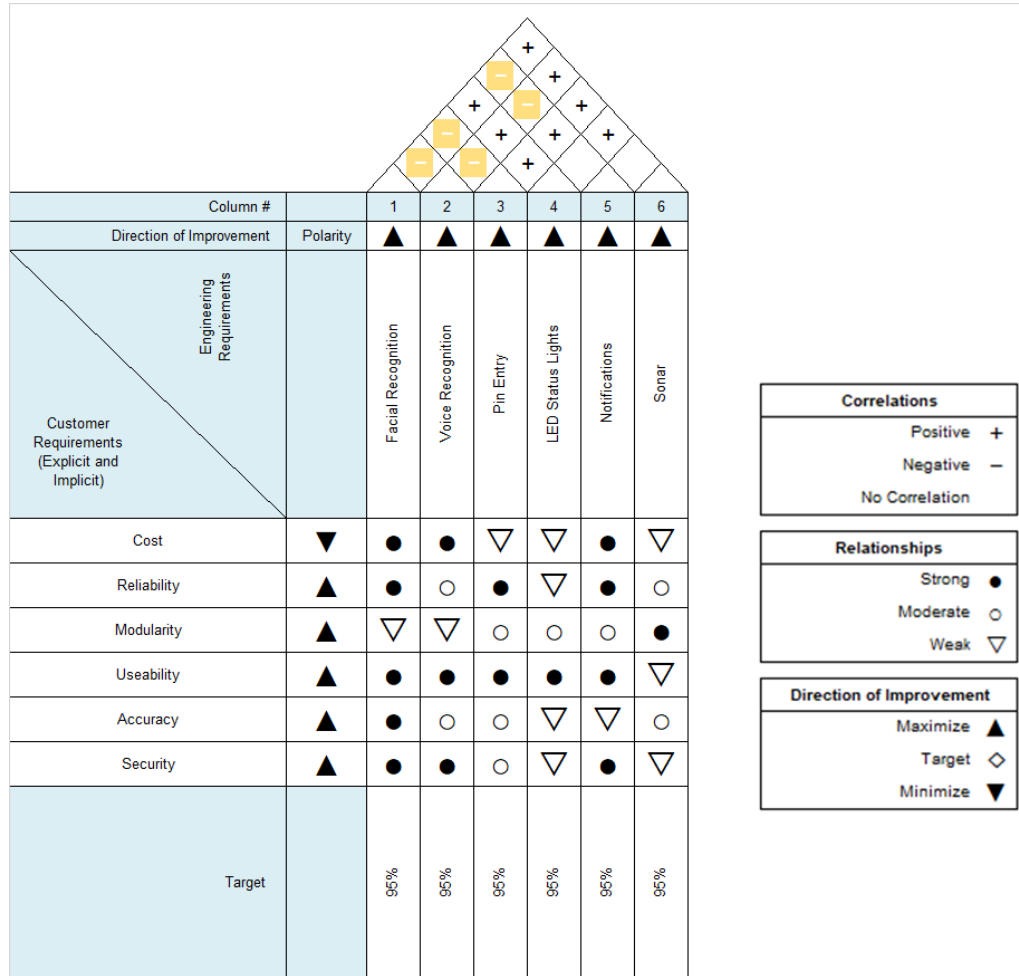


Figure 1: House of Quality Chart

### House of Quality Roof Explanation:

The relationship between the three authentication methods (voice, face, and pin) is negative, because although it provides security, since they are being multi-threaded all of them running at the same time will take up a large amount of the CPUs resources. Therefore, if for instance facial recognition capabilities increase, it will have a negative impact to the voice recognition capability. A positive correlation between the LED lights and the authentication methods is provided because they will give us a visual representation of when something is validated, and they require little power to operate.

The relationship between notifications, voice recognition, and facial recognition is negative because although they will send a notification to the homeowner when an unauthorized user is attempting to gain access, if the system wrongfully identifies an authorized user, then it will send a notification to the homeowner, thus wasting resources. There are positive relationships between sonar and the other features because it will be saving power by waking up the CPU, instead of requiring the

system to constantly be running and wasting battery power. Therefore, as we increase the sonar capability, the rest of the system benefits positively.

## 2.4 Smart Door Security System: Diagrams

### 2.4.1 Diagram Status:

The blocks are currently still being researched since we are in the heart of the planning phase. Ideas, concepts, and responsibilities in this section may change over time. None of the blocks are purchased yet, we are still designing and researching components that will best fit our System. Until we are comfortable with a design, we will not be purchasing anything. Also, none of the blocks are being prototyped yet. The development process is still very early, so we are not currently at a prototyping phase.

### 2.4.2 System Block Diagram:

Figure 2 shows the System block diagram for our project. This diagram has gone through several revisions and is updated with our current progress in this project. It may need to be adjusted in Senior Design 2, depending on our status LEDs implementation and if we develop a need for a magnetic reed switch. This is dependent on the model lock that we are able to procure in a short lead time.

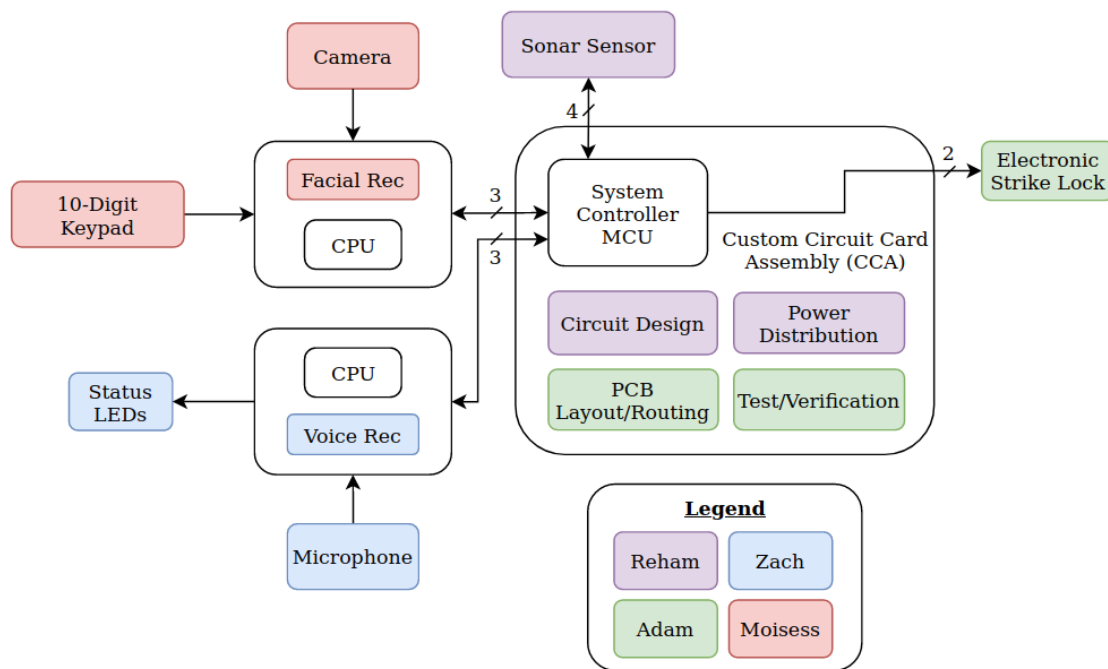


Figure 2: System Block Diagram

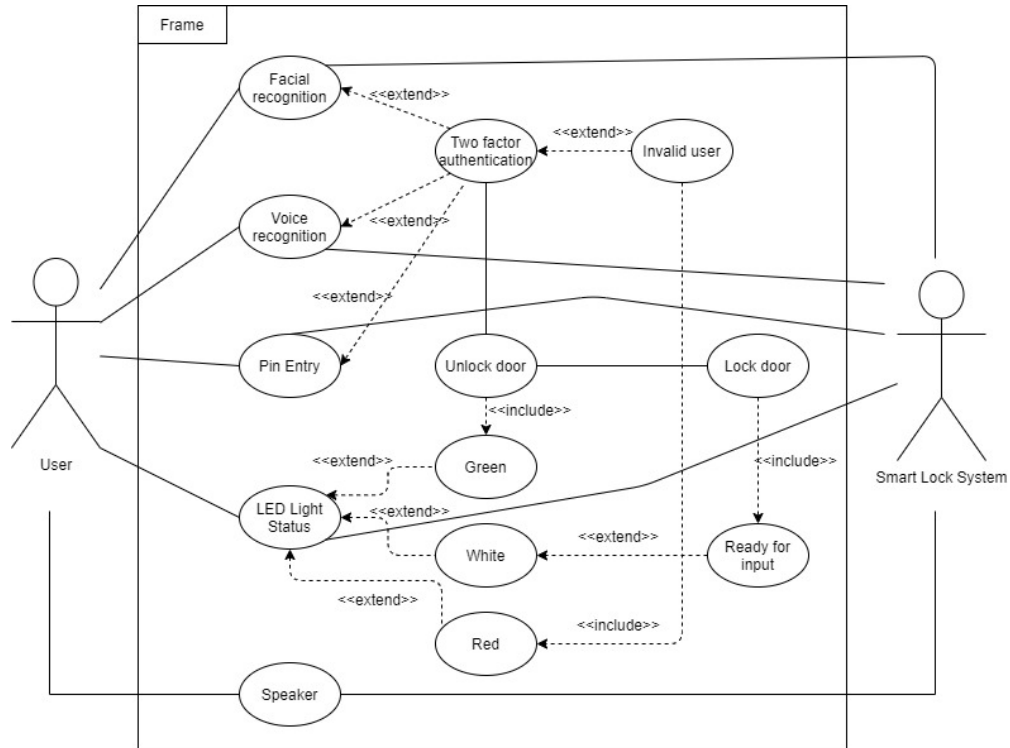
In the Smart Door Security System block diagram, we show a high-level representation of the system's different components, and their relationship to each other. Essentially there will be a microcontroller (MCU) and two computers (CPUs) that will control all of our various functions. In the block diagram shown, there will be two dedicated, high-level CPUs that are each in charge of running the facial and voice recognition software that will be loaded onto it, as well as the many other small peripherals that we need to connect with including the Camera, Microphone, Keypad, and LEDs. The external 10-digit keypad will be connected to one of the CPUs and will optionally take the user's pin number as input for one of the three authentication methods, besides facial or voice recognition. The camera and microphone will be connected to the respective computer boards via a USB connection, both of which are used as the vehicle for the facial and voice recognition inputs. The two high-level CPUs will be connected to the system controller, or central MCU via a wired connection of GPIO pins.

Additionally, the external sonar sensor and the electronic strike lock will be controlled by the "System Controller" microcontroller. This System Controller microcontroller will have complete control of the lock mechanism, so although the high-level computers will be doing all the heavy computations like Facial and Voice Recognition, this microcontroller is in charge of the entire system's control logic and ultimately the control of the door lock itself.

Features that are not shown on the System's Block Diagram but are stretch goals for our group include the use of an LCD Screen that will be used to display prompts to the user throughout the authentication process as well as displaying a "Welcome Home" message after a user is authorized. We are also going to try to incorporate a speaker, as our second stretch goal, that can give the user at the door verbal prompts as well as a verbal "Welcome Home" message. Lastly, the most difficult of the stretch goals we are currently planning for would be some sort of real time update or email notification being sent to the homeowner's phone. This would be much more difficult as it would require either a wireless internet or Bluetooth connection, which would drastically increase the overall design complexity.

### **2.4.3 Software Use Case Diagram:**

Figure 3 shows the software use case diagram that was designed for our system. It shows the user input and the approximate functions that the software will perform based on user input.



**Figure 3: Software Use Case Diagram**

A software use case diagram is provided which displays multiple options for a user while interacting with the Smart Door Security System. The user will be able to use facial recognition, voice recognition, and a keypad pin number as a third method of validation to unlock the door. The pin number will be user specific and will be associated with that user when their profile is created. The LED status lights will be visible and display different colors based on the state that the system is in. If the two-factor authentication protocol is validated, the door will unlock, and the LED lights will turn green. If it is not validated then the door will not unlock, the homeowner will be sent an email notification to alert them of the situation, and the system's LEDs will turn red. A speaker will be used to greet the homeowner upon entry if they are authenticated. Once the door is unlocked the user will open the door, close it, the door will lock, and the LEDs will revert back to white, symbolizing that it is ready to begin detection of another user.

## 2.5 Software Design Overview

### 2.5.1 Software Flow (version 1)

Our project will consist of interacting with pins on a micro controller, multi-threading, reading data from files, and potentially network programming. Since we are in the early stages of research and developing, our design may change. Many factors can cause this. Such as an issue arising during development, we discover a more efficient way to accomplish something, or even if we determine some

redundancy in our program. As we begin developing the complete system, we will be able to get a clear understanding of how the system will operate. In the pipe and filter diagram shown below, we can see a high-level representation of how the software will work.

Figure 4 shows our software flow, and the various states that our software will be in. This will help us divide the code into manageable blocks, and it will also aid in troubleshooting.

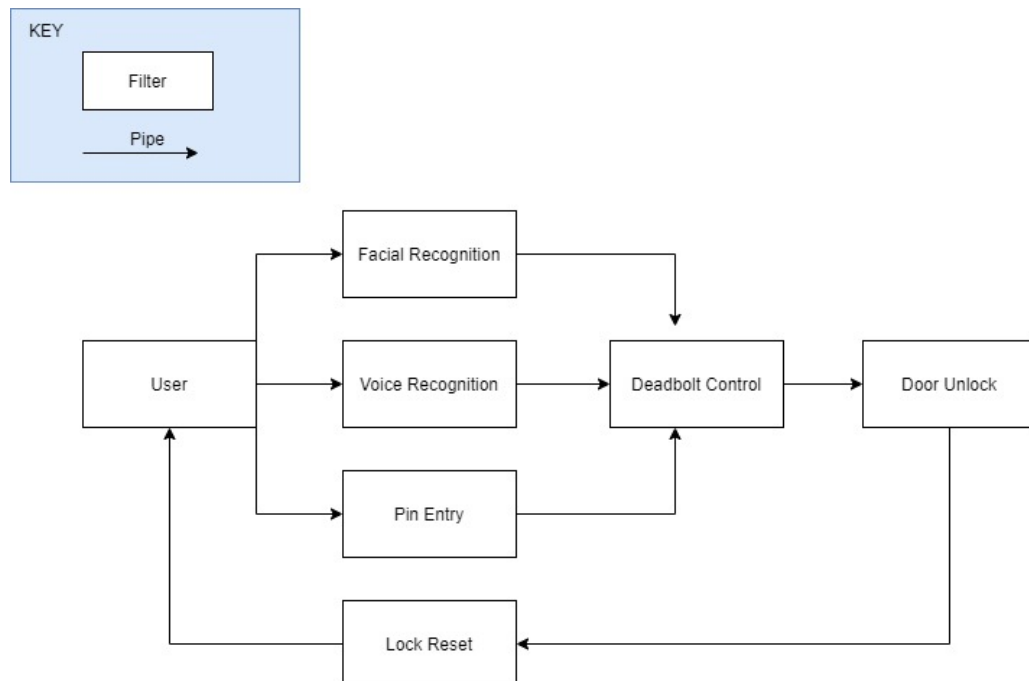


Figure 4: Pipe and Filter Diagram

The user will be the person wanting to be granted access to the door. This is a high-level representation because realistically, these authentication devices will be connected to microcontrollers, which will be updated by sensors and other devices that are connected to the microcontrollers. However, using this diagram we can see that there will be three authentication algorithms running at the same time. Which are facial recognition, voice recognition, and the pin entry. It is possible to achieve this by implementing a method called multi-threading.

Multi-threading from a software perspective is a process where essentially individual methods or actions are able to execute simultaneously. This is extremely powerful when it comes to real-time systems or any other system that would require data to be received from multiple sources at the same time. In our case we will most likely have four threads running. The threads that will be operating are facial recognition, voice recognition, pin entry, and some sort of validation thread. The validation thread will act like the main thread which will be waiting for acknowledgment messages from each of the other threads. This thread will be



waiting until it is being used, and this can be accomplished by utilizing semaphores.

Once the requirements are satisfied then a signal will be sent to the microcontroller that will control the deadbolt lock, as well as other components. All of these messages will have to be obtainable. A way to achieve this would be to create a class for each thread, and within each class, have certain variables that pertain to that specific class. An example of this would be in the facial recognition class, there can be a variable with the user's name, and another variable which is called "authenticated". The authenticated variable can be a boolean variable which is set to true if the person's face is recognized. Each class can have an "authenticated" variable. Then inside of the validation thread, it can check each class for this variable. This can be done by utilizing getters and setters. A getter is a method which retrieves a class variable, and a setter is a method which sets a class variable. Initializing the authenticated variable to false in the class constructor is good practice. This also allows us to know the starting value of the variable.

Also, in the validated thread we can implement a method which looks for intruders. If all three validation methods are false, then the user is an intruder. If so then the system should send some sort of notification to the homeowner, or system owner. This notification can be an email, or even a message through a speaker. The most important thing here is that the door does not unlock for unauthorized users.

On another hand, a possible programming language that can be implemented with this system could be python. Python is an extremely efficient and powerful language to use, and it also has plenty of machine learning (ML) libraries that we can utilize. More specifically there are an abundance of facial recognition libraries with python. This is covered more in detail in section 3.3.2.

Another benefit of using python is that it supports threading, as well being easy to use on certain microcontrollers. We would definitely need a language that supports threading. Certain microcontrollers like the raspberry pi come with IDE's which support python. It is also extremely easy to access the pins of the microcontroller by using python as well. Thus, making this language very appropriate for our application. Other possible languages that support multi-threading would be Java and C++. Although these languages do support threading, utilizing pins may prove to be difficult, mainly because these languages tend to be solely object oriented related, rather than hardware related.

Since multiple people will be working on this system at the same time, practicing proper coding standards is crucial. By doing so we can maintain code organization, improve run time, allow for easier debugging, and many other things. A way we can do this is by creating clear and concise comments above functions. Also, by adding comments at certain parts of the code which can be confusing for someone who is unfamiliar with the particular functionality. As we are programming it is very easy to forget to write comments, but this is very important when working in a group

setting. Our code should be written to where a complete stranger with some coding experience can analyze our application's and understand how it works.

Run time is the time it takes for a code to execute. This is very important because if an application is written poorly then it can take up unnecessary resources, as well as take a long time to execute. This would be a major problem with our system in particular. In order for our smart lock system to be practical then the process of unlocking a door should be as quick, if not quicker than taking keys out of your pocket, inserting into the lock, turning the lock, then opening the door. There are many ways to improve a program's run time. You can optimize your code by removing redundancy. More specifically if there are certain blocks of logic that are being repeated multiple times then you should create a method which can be called to perform the operation. This process will also reduce the number of lines in your application as well.

Furthermore, by inserting known data, analyzing where the program is spending most of the time, fixing it, and repeating this process, the developer can improve the execution time. When programming, the developer can determine where a program is currently executing by using specific print statements. If the known print statement is displaying, then the developer can easily go to that specific part of the code and determine ways to fix the functionality. There are also ways to time the execution of an application, and even insert the data into an excel sheet. So potentially someone can time their program, make a change, time it again, and determine if there are any improvements. This process may take a long time, but it is extremely effective and will aid in optimization.

Proper coding standards also include using proper naming convention. Using naming convention can allow for easier code readability, settle arguments for naming variables or other potential syntax debates, and allows developers to primarily focus on bugs when performing code reviews. There are many different variations of naming conventions, a couple of them are camel case and upper camel case. When using the camel case naming convention, the first word should be under cased, the next word should be uppercase, no numbers with words or special characters, and no letters should be sequentially uppercase.

An example of camel case would be "smartLockSystem", an incorrect example would be "SmartLockSystem". Pascal case is similar, however with this form the first letter would be capitalized, and every other word should be capitalized as well. An example would be "SmartLockSystem", and an incorrect system would be "smartLockSystem". Generally, software companies will have a specific naming convention that they follow. Some companies would also be stricter than another with implementing this standard. A reason for this is that when a company develops software, they are creating it to sell to another company. Therefore, they would want their code to follow their specific standards. If they do not implement this then it can damage their credentials, which would financially affect their company. So, before a software product is delivered to a customer the company

will perform many code reviews as well as doing debugging sessions to determine any potential problems with the application.

We will be utilizing GitHub with our application development which will allow our developers to work on code without interfering with others work. By creating a local git repository on the developer's machine, they are able to essentially create a copy of the code onto their devices. Any changes that are made they can be commented on and pushed to the GitHub repository. Then other developers can read the changes and continue to work on the system. This will help with reducing time reviewing the complete source code or having to continuously communicate with group members to determine changes. Also, GitHub makes it extremely convenient to have the application hosted on a website rather than a local machine. This makes it possible to basically work on the code on any machine. Utilizing databases can also be helpful when dealing with multiple user profiles. In databases you can store many different characteristics or variables. An example of this could be a user named John Smith. Within the database you can store John's name, his age, weight, height, etc. This is especially helpful when developing web applications. The developer can also make a system that would allow for the user to create a profile dynamically. This is similar to what social media platforms do to generate profiles for their new users. Typically, these applications are done through programming languages like JavaScript. However, it is also possible to implement databases in other object-oriented programming languages like Java.

There are many different databases such as MySQL, MongoDB, Microsoft Access, Altibase, and many more. Each database has their own specific features and functionality. Generally which database a developer decides to implement comes down to personal preference as well as experience. MongoDB allows the database manager to log into their website and view the data that is stored in the database. While MySQL stores the database data locally, and you can still view the data by opening the application and selecting on the desired database. If we were to implement a database for our system, the most feasible option would be MySQL. This is because we would not be creating new user profiles on the spot, and we do not need to be connected to the internet to connect to the database. So, the information being stored locally on the machine would work perfectly.

For our specific application we may not need to use a database. This is because we do not have to store much information. The information we would be storing and retrieving would most likely be the user's name, and a pin. When we train the facial recognition algorithm, we will be storing multiple images into user-specific folders. Each folder will be named after the person that the algorithm is training for. So, if we are training the model to recognize John Smith, then we will direct the program to John Smith's folder and train the model from his images. Then we can have another class which associates each person's name with a specific pin number. There are some pros and cons to this design choice. The pros being, it may be quicker to store the images locally, and would remove the internet dependency in terms of hosting a database server. The cons being since we are

storing the images locally then it will take up space on our device. This can cause the machine to run slower since more resources are being occupied. Once we begin the development process, we may find it better to incorporate a data base, but at the moment the most optimal solution would be to store the data locally.

For our system we may need to implement network programming. Network programming in our case would consist of connecting to another device on a local network. The objective of this would be to send messages to the other device. Some of the messages would include information like unlock the door or lock the door.

Below we can see a client-server diagram in Figure 5 for which our potential application would implement. This will assist us in designing the various blocks and which software functions will take precedent over others.

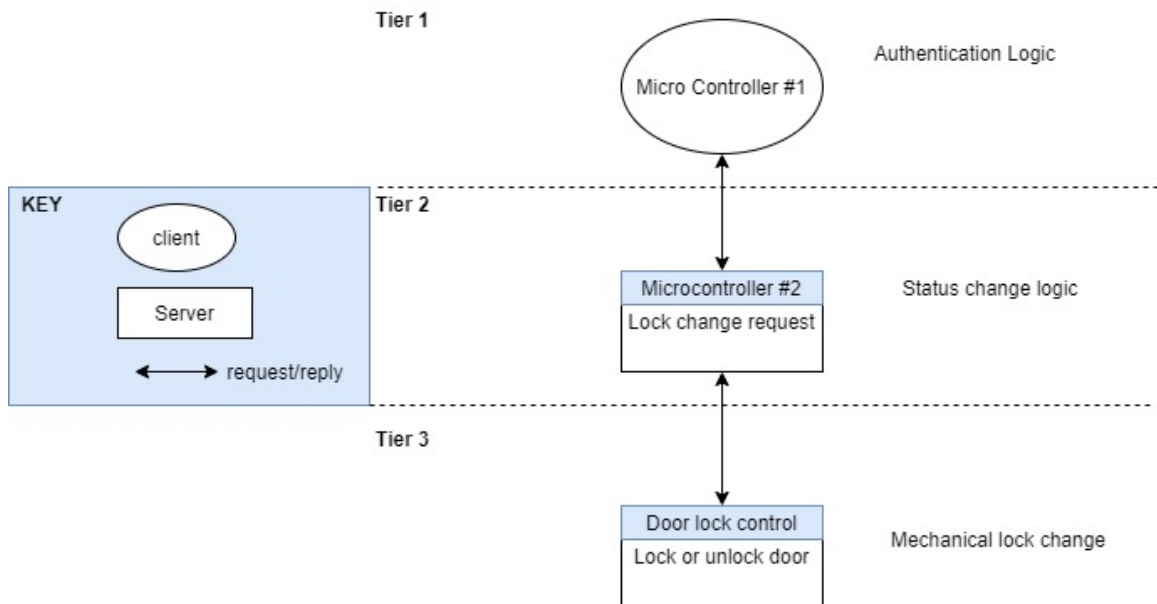


Figure 5: Client-Server Diagram

Analyzing the diagram from top down we can see that there are three microcontrollers. The first being the controller which handles the authentication logic. The second being the controller which handles the lock change request, as well as other components. And the third controller which will mechanically change the door lock. There are also bidirectional arrows which represent requests and replies between the client and servers.

Microcontroller number one is represented as the client because this is where the initial authentication status will change. Once the user is validated then a message will be sent to microcontroller number two. From there microcontroller number two will send a request to the door lock controller asking to unlock the door. From here the door can reply to microcontroller number two. Then microcontroller number two

can send an update to the door stating that the door has been unlocked, and locked once it closes. Once the first microcontroller receives the door unlock and lock message then it reset the system, put it to sleep, and wait until another user approaches the door to begin the process again.

There are many ways to connect to different devices to set up a client-server relationship. Some ways are through ethernet, i2c, UART, SPC, Blue-tooth, Wi-Fi, etc. Generally, as long as the IP address is obtainable of the desired device, and you are able to ping that device then you can begin sending messages. For our system, the most convenient connection would be through Bluetooth. This would require us to have microcontrollers that have Bluetooth transmitters and receivers.

Furthermore, these versions of microcontrollers tend to be more costly, and physically larger in size. If we used a wired connection like ethernet or UART, then we would have to use wires to connect the devices together. This can cause many issues, some being visually unappealing, potential safety hazards, and potential hardware issues. If there are wires running outside of the door then this can look sloppy, and make our system look poorly developed. Wires can be a safety hazard because if placed improperly then the user can trip on them, or even cause potential fire hazards. Hardware issues could occur if the wires damaged or get unplugged. These hardware issues could cause software issues which can cause the entire system to fail.

So again, if we decided to implement this type of door control then the most logical connection would be a wireless connection. Bluetooth would be better than Wi-Fi because it would not require internet connection. Having a device with this sort of dependency can lead to many faults. Such as if the internet goes off then the user would not be able to enter their house while using this system.

However, using wireless communication protocols can lead to cyber security implications. Someone can hack into the network and manipulate the data being sent between devices. So, a hacker could get into the client-server connection and send a signal to the door requesting for the door to be opened. This would then lead to the users, home being at risk of an invader.

These sort of security issues can be resolved by using a wired connection, which has been described to have poor design possibilities. Or by using network encryption, or other cyber security methods. Typically, within certain industries which are safety critical, such as defense industries, wired connections are preferred because this would eliminate the possibilities of cyber-attacks. There are also ways to hack a device through hardware. But there are also hardware-security protocols which could prevent this from happening.

If we plan to use a different locking method, such as a magnetic lock. Then we would not have to use an additional microcontroller. Thus, removing the need for a client-server connection. This method would be preferred due to its simple and

effective nature. Small wires may need to be connected to this lock to send signal through, but these wires would be easy to conceal.

We would also be required to understand how to connect to microcontrollers through GPIO pins. This generally is a simple process, as long as the developer is able to read the microcontrollers documentation. When beginning to develop the system generally the first step would be connecting the devices through the pins and understanding how to control them.

Once the user is able to control the basic functionality then they can begin developing more complex logic which checks certain conditions before an action is performed. There are also microcontrollers like the Raspberry Pi which have USB ports. These ports can be used to connect external devices like a web camera, microphone, mouse, and keyboard. Being able to connect a webcam directly through a USB port is extremely convenient. The other alternative would be to use a camera device that is compatible with the GPIO pins on the device, or a specific camera port if the MCU supports that feature.

It is important to determine compatibility when connecting devices to a microcontroller. If the connected device requires more voltage than the device can support, then the microcontroller can be damaged. This is very bad, especially if the developer is not using a code repository like GitHub and is storing the program locally. If this is the case, then the entire project will most likely be lost. Typically, microcontrollers will have a GPIO pin which provides 5V. Some will have another pin that supports 3.3V as well. Fortunately, there are many external devices which use this voltage. There are also ways to acquire additional voltage if a device needs more power. This can be achieved by using batteries. If using batteries, then the user should take extreme caution to not fry the microcontroller.

So, to reiterate, there will be two microcontrollers, potentially three depending upon the door lock we decide to incorporate. As of now we are anticipating there to be only two microcontrollers. The microcontroller with the recognition software's will have sensory devices, such as a camera, microphone, and a speaker. There will also be devices such as an ultrasonic sensor which will detect objects. These objects ideally will be users, and once detected, the system will wake up from sleep or low power mode. This mode will allow the system to power by only being used when a user approaches the door.

The microcontroller will send signals to the other MCU which controls the door lock requesting the door to unlock, and lock. Once the door is locked then the system should reset and be ready to repeat the process. As mentioned, previously this software design may change throughout the development process. Fundamentally this process should work, and improvements will most likely be made once significant progress is made. Documentation will be done stating any changes that are made to the device that was different than originally planned.

## 2.5.2 Software Flow (version 2)

After consideration we will be implementing the i2c communication protocol. This is due to the fact that we will be utilizing the MSP430 microcontroller to interface with our Raspberry Pi CPU. These devices will be connected to each other by using simple wires. However, we will have to analyze the data sheet for both devices in order to set up the communication protocol.

By using i2c we can send data between devices efficiently. Since the MSP430 microcontroller that we will be using does not support ethernet, or Bluetooth this is the best alternative. At a high level, the microcontroller will constantly be checking register values to preform actions. Once a certain register value is obtained then other logic will be performed.

For instance, while running the application on the raspberry pi, the user will go through the validation process. Once two out of the three authentication methods are satisfied then each thread will set their validation member variable. Then the validation thread will acknowledge each of the variables and set the validation variable for that thread. During this process, a message will be sent to the microcontroller. The message is more of an action. This is because the raspberry pi will be setting the value of a register in the raspberry pi.

In order to do this the raspberry pi will send an instruction to the raspberry pi in the form of "set register t5 to 0x0001". Assuming that the registers are 16 bits, then that specific register value will now be set to decimal value 1. We will achieve this within the microcontroller by utilizing the compiler specific to that MCU, as well as by using bit masking.

The MSP430 microcontroller uses code composer studio, and the language used to program the device is C. We can set any register to any desired value by using bit masking. Bit masking allows us to implement mathematical operations to binary values. This is extremely powerful because we can set known values that our microcontroller will compare to in order to complete an action. If we would like to set the register value of some random register (let's say t5) to 1. Then we would first "&" that register with "0x0000", then "|" it with "0x0001".

When we "&" any binary value with 0 then that binary value will become zero. This essentially clears that value and allows us to set it to whatever we desire. We can set a value by "|", so after we cleared the register, then we set it to 1. In binary operations, any alternative value (when preforming the "or" operation) to zero is that value. Since there are only two possibilities being zero or one, the result will be a one. Thus, makes it good practice to first clear the register, then set the value.

More specifically once our raspberry pi sets a register value when a user is authenticated. Then the microcontroller will recognize the new value and then activate the door lock. A sonar sensor will also be connected to the MSP430 MCU

in addition to the controller for the locking device. The sonar sensor will sense when a user approaches the door. Then a message will be sent to the raspberry pi CPU, the LED lights will illuminate, and the application will begin. The benefit of using the sonar sensor will be to prevent wasted resources. If each of the validation method protocols are running constantly, even when unneeded. Then the CPU will be utilizing power and processor resources for no reason. If we send a message to the CPU to essentially wake up and go to sleep when appropriate, then our system will much more efficient.

From a cyber security aspect, hardwiring the CPU to the MCU will reduce vulnerabilities. The most vulnerable communication protocols would be Wi-Fi and Bluetooth. If the devices were communicating through Wi-Fi, then a hacker could get into the network and manipulate the system, which could be in the form of unlocking the door and gaining access to the user's home.

Similarly, a hacker could compromise the system if utilizing Bluetooth. Although Bluetooth does not require internet access, a hacker could get into the system by scanning the local area for Bluetooth devices, locate the CPU and MCU, and then gain access to whatever component of the system that they would like. Both of these can be prevented by taking complex defensive protocols, such as encryption. However, since hacking has become much more sophisticated as technology has improved. This method is not one hundred percent reliable or safe. Considering the nature of our system, we would need to implement the safest and most efficient communication protocol possible.

Although by hardwiring our devices we can prevent cyber security attacks, we could still be vulnerable to hardware security attacks. This could be done by splicing the i2c wires, and then connecting a malicious third-party device. Once this device is connected then the hacker could gain access to our system.

We could prevent this by taking mechanical or physical safety approaches. Such as by putting our device along with wires in some sort of locked box which could only be accessed with a physical key. Or by making the wires hard to access, such as by drilling into the wall and feeding the wires through the other side. Again, this does not completely eliminate the threat of a hacker getting into the device. But by hard wiring the communication protocol, we can prevent cyber security threats from occurring between the CPU and MCU.

On another hand, after some consideration we have been thinking about adding another raspberry pi CPU into the system. By doing so we would be able to reduce the workload on a singular CPU. The logic of the application will perform exactly the same as before, but the only difference would be that the raspberry pi's will be communicating with each other. Once the user is authenticated then only one pi will send a message to the MCU.

Essentially by implementing two raspberry pi's, one of the devices will have two authentication methods running, and the other pi would only have one



authentication method running. Ideally, we would have the more powerful device running the methods that consume more resources. This is because that device would be able to handle the workload.

Also, one of the raspberry pi's will be hardwired to the i2c communication protocol of the microcontroller. This would make it easier rather than connecting all three devices to each other. There are many ways to communicate between raspberry pi's since they are very powerful devices. Such as through SPI, i2c, UART, ethernet, Wi-Fi, or even Bluetooth.

If we plan to focus on eliminating potential cyber security vulnerabilities, then we should focus on implementing the hardware communication protocols. Which would be SPI, i2c, UART, or ethernet. If we decide to use SPI, then we would have to connect multiple wires between the appropriate connections and configure the device to enable the SPI protocol for both raspberry pi's. Since we will be enabling the i2c protocol to speak with the MSP430 device, this can be problematic. Both i2c and UART would take similar approaches to establishing connectivity between the CPU's.

A simpler method of achieving hardwire connectivity between two raspberry pi CPUs would be through ethernet. An ethernet cord may be a little bit more expensive compared to the wires required to connect the other communication protocols, but you can generally find them for a few dollars. The benefit of using ethernet connection between the pi's is the fact that you just simply plug the cord into both devices and then they are connected. This eliminates the need for referencing any documentation for pins relating to communication protocols, or how to enable them. Also, by using ethernet we would be able to reduce the number of wires used between devices. By having less wires running between devices we can maintain system quality.

The threads will generally remain the same if we are implementing two raspberry pi's rather than just one. On the more powerful CPU two authentication methods will be running, or whatever process is more demanding. So facial recognition and keypad entry will be running on pi number one, and voice recognition will be running on pi two. If it turns out that a single process is more demanding than two processes running together than we will run that application on the stronger CPU.

We will send data from one pi to the other through a communication protocol. Which most likely be through ethernet. Once the second pi validates their portion of the application, then the first pi will be updated. It may be beneficial to run the validation thread on the more powerful pi in addition to the more intensive recognition algorithms as well. This would be better because there would be less of a delay from sending data from the validation thread or class to the raspberry pi.

Alternatively, if the method that is responsible to deliver data is located on raspberry pi two, and the MCU is connected to raspberry pi one. Then raspberry pi two would have to send the data through pi one then pi one would have to send it to the MCU. The delay may be small, but it could cause some potential issues. Also, to maintain maximum efficiency, we should take the route which would make our system the fastest and most secure.

To deliver information from one Pi to another we could use a different method than we took to communicate with the MSP430 microcontroller. We will send data from the Raspberry Pi CPU to the MSP430 MCU by setting register values. Then within the microcontroller, there will be some logic which is scanning the register and checking if the current value is set to the desired amount. It may be easier and more efficient to use another method to send data between CPUs, rather than from CPU to MCU.

A potential way we could send information from one pi to the other is by connecting a ethernet cable, and then establishing a send/ receive relationship. This is also known as a client server relationship. This can be visualized from figure number five. However, instead of having one pi, and then a microcontroller that handles the locking mechanism. Raspberry pi number one would be connected to raspberry pi number two, and then the MSP430 microcontroller will be connected to raspberry pi number one. The client in this configuration could be the second raspberry pi, and the server being raspberry pi number one. This configuration would be appropriate because the device where the server is being held will essentially be the main CPU. Although there will be a client and server, both devices will be able to send information to each other.

If we do implement this communication protocol to communicate between two pi's. Then the data being set will be in the form of strings. The first step we would need to take to establish this protocol from a software perspective would be to disable any firewalls that could prevent data to be sent from one device to another. If the firewall port is enabled or closed then we would not be able to send information, and this could lead to confusion when attempting to trouble shoot the issue.

So, once we determine that the firewall is open for both devices, and the ethernet connection is established. Then we will have to determine the IP address of each device. We can determine the IP address by opening up the terminal. Once the terminal is open then we can type in "ipconfig eth0". This command will display device specific information including the IP address which will be in the form of "inet addr: 192.xxx.x.xxx" the "x's" will be populated with unique numerical values for each device. After we have successfully determined or set each IP address, then within our program we have to set a port number. The port number can be any arbitrary value that is unused.

In network programming there are a few different types of protocols that can be used to send packets, or data to different computers. For our system, the most

feasible options would be either TCP or UDP. TCP stands for Transmission Control Protocol, and UDP stand for User Datagram Protocol. Generally, TCP is considered more reliable than UDP, and this is due to the way each protocol delivers packets.

TCP or Transmission Control Protocol ensures that the data transmitted is delivered accurately. This communication protocol also provides reliable packet transportation. For instance, when downloading information, or accessing a web page, we would want the information to be in the correct orientation and without any data missing. Therefore, while using TCP, the data is received in the correct order which it is sent. This will prevent lost packets as well as data being displayed incorrectly.

Since TCP is a connection-oriented protocol, a session between each computer must first be established and validated. This is done first before any further data is sent or received. A session between each computer is acknowledged by using a “three-way handshake”. In our case, if we are using two raspberry pi’s, let us say raspberry pi number one (the main CPU), and raspberry pi number two. The first CPU will begin establishing a session by sending a “SYN” message. Once the second CPU receives this message it will react by sending a “SYN ACK” message back to the first CPU. Finally, the first CPU will respond by sending “ACK RECEIVED”. Once this process is completed, then a secure session is established, and packets are ready to be sent and received.

Another benefit of TCP is that the delivery of a packet is ensured. If the data is not sent or received, then TCP will resend it until it is properly received. Data will also be resent if some data is missing from a packet as well. Since, this communication protocol is secure, speed is sacrificed to achieve minimum packet loss.

UDP or User Datagram Protocol is faster than TCP, but there are less methods which ensure proper packet delivery or retrieval. UDP has many similarities to TCP, where both protocols are used to send and receive data. However, UDP is a connectionless oriented protocol, where TCP is a connection-oriented protocol. Therefore, a session is not established prior to sending packets, and packet delivery is not validated.

UDP is also known as a “Fire and forget protocol”. This is because when sending data with UDP the sending device does not know if the computer on the receiving end obtained the data. Data will be sent regardless of how it is received, and the protocol does not ensure the quality of the packets. So due to the fact that there are less regulations with this protocol it is much faster than TCP. UDP can be very effective if the data being transmitted does not have to be in any particular order or if you need information very quickly.

So, in relation to our project the best way out of these two communication protocols to transfer data between the two raspberry pi’s would be by using TCP. Although

TCP is slower compared to UDP. TCP will ensure that the information being delivered is done properly, and efficiently. Since our system deals with opening and closing doors to our clients' homes or business. We would not want any information to be delivered incorrectly. Also, the difference of time it would take to send data from TCP and UDP would be unnoticeably minuscule.

Currently, it still makes sense to store user data locally rather than connecting to a database. This is because the only information we would have to store would be the users' images that will train the system. Since this is the only data that our application would have to retrieve then it would not make much sense to incorporate a database. A benefit of not using a database would be that we do not have to be connected to the internet. Having an internet dependency would force the user to have internet service. By having a system that works completely independent from the internet, and solely on power generated from an outlet, or some other power source. We can implement our smart lock system into remote areas.

There are some benefits to incorporating a database, such as reducing space on the device. When using a database, we can store the data into a web application, then reference that application when needed. This would also force our system to be internet dependent. If our system utilized internet service, then we could implement other functions within our system.

As a recap, our current software flow will still function the same as in the previous section. However, the raspberry pi CPU will communicate to the MSP430 microcontroller via i2c. This will be done by configuring the i2c protocol on each device, and then connecting them by wires. Also, we most likely will be implementing another raspberry pi CPU. This will reduce the used resources being used on a single CPU and will also reduce complexity of our application. If we do decide to use two pi's, then we will connect them together via ethernet. If any changes are made to this design, then it will be documented.

### **2.5.3 Software Flow (version 3)**

Due to the significant amount of research and consideration, we have discovered another way to optimize our system. Essentially the components of the system will remain the same as in software flow version two. Where there are two raspberry pi's, and one microcontroller. These are the main devices which will control subcomponents that are connected to them.

Furthermore, the subcomponents of the system will remain the same, where the main features are facial and voice recognition, and the ability to read a pin entry. In order to incorporate these features, we will need a camera, microphone, and a keypad. These devices will be connected to a raspberry pi microprocessor. Some of these subcomponents will require additional power due to the limited power supply of the microprocessor or CPU.

One of our raspberry pi microprocessors will notify our systems microcontroller. Notifications will be sent when a user is authorized. Once this happens then the microcontroller will unlock the door, and ideally send a signal back to the pi. The return message will inform the application that the door was unlocked, and relocked. Once this happens then the system will reset and begin waiting for further instructions.

Initially we were planning to transfer data or send messages from the raspberry pi microprocessor to the MSP430 microcontroller by utilizing the i2c communication protocol. More specifically we were planning on setting a specific register to a known value. So, the CPU would potentially send a message to the MCU. The message will instruct the MSP430 microcontroller to set one of its register to a known value. This known value will be one that the developer uses to compare, and if the value matches, then the door will unlock. Once the register is set then another message will be sent back to the pi indicating that this operation has been done. After the door has relocked then the system will reset, and the registers will clear their values.

Although this approach would work in theory, our group has come up with another alternative to sending data between devices. Instead of utilizing the i2c communication protocol, we could just set a specific pin of the microprocessor, or microcontroller to high. Then on the alternative device it would read the voltage level of the pin. If it is high, then the user is authenticated, else the user is not authenticated. So, the microcontroller would only have to react once the pin is read as high.

The benefit of using this approach opposed to the i2c method is the simplicity. If we were going to use i2c then we would have to configure each device to set up the communication protocol. Once the devices are set up then we would have to program the devices to begin sending messages. The alternative to this is to simply connect a wire between both devices, and then just read the voltage levels between them. Ideally to achieve this we would just need one wire between each device. However, if we would like to read the voltage between each device then we may need to incorporate another wire. This is still not a deal breaker because the wires are extremely affordable, and each device has an abundance of pins.

There are potential risks that could arise from this method. A potential problem that we could run into would be if we accidentally fry a pin by overusing it. If we are constantly setting a pin to high, then over time this pin can become fried. Another issue we could run into would be that the receiving device misreads the signal. This could be problematic especially considering the nature of our system. For instance, in our system the device that will be controlling the lock is the MSP430 microcontroller. If the microcontroller misreads the signal of a pin, an unlocks the door than this can cause safety issues for the user.

This new method of sending data is still more secure than implementing a wireless connection. However, it could be easily manipulated with voltage. Theoretically if

the intruder has some sort of device that can produce the required amount of voltage needed to trigger the events from the MSP430. Then they would be able to open the door and bypass the authentication methods.

Alternatively, if using the i2c communication protocol then this can be prevented. The invader can manage to hack the system still if they are sophisticated enough, but it would definitely be harder than simply putting voltage onto a pin. We will have to do additional research and determine if there are some protocols that we can take to avoid this from happening. Some protocols could include locking the CPU casing with a physical lock that can only be accessed with a key. Or even placing the CPU in a place that cannot be accessed easily. Such as in a very high location, or behind the wall.

Besides for this new way of transferring data between the CPU and MCU, everything else in the previous software flow version will remain the same. We are still planning on using two raspberry pi microprocessors. This method is still very feasible and somewhat easy to implement. They will be connected with an ethernet cord. Once the physical connection between the raspberry pi's are enabled then we will have to configure each device. The first step should be to open the terminal then enter "ifconfig". This will allow us to see what devices are connected. Once the device is found then you can try to ping it. If the ping is successful, then the devices are able to send messages to each other.

In the previous section we mentioned that the most efficient and appropriate communication protocol to utilize between the CPUs would be Transmission Control Protocol (TCP). This method is still very valuable to our system. This is because it will guarantee that the message is sent to the other device. However, this will only happen if the "three-way handshake" is established. User Datagram Protocol (UDP) is very fast, but it can lead to many errors. This is due to the fact that once a message is sent, there are not any procedures that ensure that the message is delivered from the receiving device. Our system will be capable of opening the homes or other doors of our users. Therefore, we will need to take the most secure and reliable approach we can with all aspects of our system.

In terms of databases, we still plan to store the data locally. This decision has been consistent throughout the planning phase so it will most likely become a permanent software design decision. By storing data locally, as opposed to referencing an online data base, we can keep our device completely offline. This is nice because the only thing that would be required to operate our system would be power. If our device, we internet dependent then our user will have to have an internet service provider (ISP).

Also, by having our system offline it makes the device more secure and reliable. This is because, if our device was internet dependent and the user's internet connection is compromised, then they would not be able to open their door.

Although, this can be prevented if we take some sort of procedure for these sort of corner cases.

So, our systems software logic will still function the same as in the previous section. The main difference is that now we are planning on incorporating a new communication method between the raspberry pi microprocessor and the MSP430 microcontroller. We are evaluating the possibility of setting a certain pin to high, and then reading that voltage from the receiving device. If the pin is read as high, then that will cause the device to begin to trigger events. We are still planning on implementing another raspberry pi CPU. If any changes are made to this design, then it will be documented. Also, the finalized coding plan can be seen in section 6.3.

## **3. Research and Investigation**

The following sections provide documentation of the detailed research that was conducted in order to develop this project to the best of our abilities. First, existing products on the market with similar features were studied to successfully design a working system. Then, projects that have been developed and presented in past semesters of the Senior Design course at UCF were studied. Finally, we delve into advanced technical research followed by specific component research to help determine the best technology and components to use, as well as optimize a successful design and project overall. This research is pivotal to being able to create a cost-effective, yet high level project.

### **3.1 Existing Products**

In spite of the fact that smart locks have already been produced by many companies and have been available on the market for a couple of years, our team believes that trying to design our own lock is a great learning opportunity, as well as an opportunity to advance the current products available. Investigating some existing products can help us learn about the technology involved and shine light on existing system's flaws. This could inspire us to generate great solutions and lead us to creating a better design. For the purpose of this project research, multiple products were reviewed. Many products exist in this space, offering features from facial recognition to pin entry to WiFi entry from an application. We are focusing this section on products that are similar to our design.

#### **3.1.1 US:E Camera Smart Lock by Elecpro Group Inc.**

The US:E Camera Smart Lock allows the user to access the door in multiple ways, smartphone, password, figure print, key fob, physical key and facial recognition. The company created two version for this lock. The user can choose either the password or the facial recognition option. For the facial recognition version, the lock comes with a built-in infrared 3D recognition technology and 4 AI recognition levels, which guaranties face recognition in daylight or night and allow the user to unlock the door with a look to the camera. US:E has the capacity to store up to 100 faces and cannot be tricked by photos or videos. The facial recognition version supports palmprint recognition as well. As in many other smart locks, US:E lock uses a mobile app which allows remote control of the lock, and also helps monitor the surroundings.

The password version has a wonderful security feature, which gives the user the freedom to type any random combination before or after the correct code and the door lock would still recognize the password and open the door as long as it was typed in the right order. This feature is great to use when the user have company and doesn't want to reveal the passcode to others. This lock also includes a



fingerprint scanner and a double verification mode where the door can only be unlocked when two unlocking methods are verified.

The release of this lock is coming up in May 2021 which will take smart locks to a different level of security.

### **3.1.2 Camera Smart Lock by Gate labs**

The Camera Smart Lock is an all-in-one security system. It replaces smart doorbell, lock, and security cameras. The camera has a motion sensor for activation, which only start videotaping whenever the sensor detects a movement around the door. The system also includes a two-way audio which allows the owner to interact with someone at the door. The lock has a built in Wi-Fi which can be connected to the house network for allowing automatically updating the system and also upload the recorded videos.

The system also includes a phone application which allow remote access to the door to lock/unlock as needed while the owner is away. It also could be used to create individual passcodes to grant authorized people access to the house. In addition, the lock sends notifications through an app whenever a movement around the door is detected. Also, the provided app enables users to press an emergency button and immediately dispatch police to their home.

The Camera Smart Lock has an easy installation process, it can easily replace any traditional lock and it fits all standard exterior and interior doors. After installation, the user only has to download a mobile app and connect the mobile with the smart lock.

### **3.1.3 FL1000 by ZKT ECO**

FL1000 was the world's first smart lock to use facial recognition technology. It has four unlocking mechanisms face, password, card, and a traditional key which can be used to override the door in case of emergencies. The lock comes with a 3-inch capacitive touch screen, camera, home button, power button, key cap, reset button and emergency power connector.

The touch screen provides higher security than a regular keypad since a touchscreen prevents fingerprint code detection which is possible on a button keypad. The system is capable of recognizing up to 100 faces, storing 100 password and RFID cards and has a log capacity of 30000. This lock has a smart alarm feature which notifies the owner in case of unauthorized activities or if the battery is running low. The lock also supports time zone management which allow visitors to enter only at specific times chosen by the owner. FL1000 smart lock is powered by the building, but also have an emergency power connector which can draw power from 9V battery in case of losing power.

ZKT ECO also produces ZM100 which is a smart lock with hybrid biometric recognition technology. Which provides two unlocking mechanisms. Face and fingerprint. It comes with a 2.8 inches capacitive touch screen and SilkID fingerprint sensor that can perform a live fingerprint detection. Equipped with a double HD camera that uses ZK face algorithm, the lock performs a 3D face scan for high-speed verification, and an accurate recognition at the dark. ZM100 uses a rechargeable lithium battery which can last up to a year when fully charged, but also has an external terminal to draw power from 9 volts battery.

### **3.1.4 August Wi-Fi Smart Lock**

The August Smart lock is one of the leading smart lock designs. It is integrated with Apple Homekit, Google Assistant, IFTTT, and Amazon Alexa. The August lock is installed on the inside of the door against the deadbolt, so it is not visible from the outside. An emergency battery is not necessary since the key portion of the deadbolt is still operation after installation. It is able to send push notifications and email notifications. The August lock has location services, voice activation, and the ability to provide guest access, but it does not have a tamper alarm or touch pad. Voice control is possible through Amazon Alexa or Google Home Assistant. August automatically unlocks when you arrive and locks when you leave.

From the August app, you can lock and unlock your door, track who comes and goes and when, and grant keyless access. In the event that your phone is lost or broken, the app and all guest keys can be disabled from the August website, which enhances security. The lock has a sensor to guarantee that your door is securely closed and locked. The August lock uses Bluetooth Low Energy (BLE) encryption in addition to Transport Layer Security (TLS) encryption. The door can be controlled from anywhere by using a Connect WiFi Bridge.

Certain models have built-in WiFi and require no additional bridge to connect. The system is advertised to run off of 4 AA batteries for a period of three to six months without needing to recharge or change them. The August Smart keypad can also be purchased and used for PIN entry. A smart watch such as an Apple watch can be used to lock or unlock the door. The app has been updated to be compatible with biometric facial recognition or fingerprint scanning from a smartphone for further security.

## **3.2 Similar Designs**

To advance our team knowledge, research for similar designs on UCF senior design website was conducted. Two similar projects were found and studied in order to develop a better understanding of the performance for these types of smart locks. Below are some of these previous designs. By analyzing these projects, we can make ours different from these older designs, and have additional features.

### **3.2.1 Smart Lock, Fall 2019**

From the graduating class of fall 2019, group 6 designed a smart lock system. Their goal was to come up with a system that replaces security cameras and help provide better protection to homes and business with a reasonable price. The smart lock was designed to grant the user access by facial recognition, mobile app, backlit LCD touchscreen, RFID and Fingerprint identification.

The facial recognition feature was only enabled when all the other authentication methods fail. A camera connected to the system would capture the person attempting to gain entry and compare it with the authorized people pictures uploaded to a database. If access permission is denied, the camera will take a photo of the person and send it to a mobile app. Once a picture is received on the app; the user can then choose to unlock the door or keep it locked.

A backlit touch screen was used to allow users to key in their password, the screen lights up at night to allow clear vision of the digits. Passive RFID and a fingerprint reader were used to allow fast access by swiping a card or simply placing the thumb on the scanner. A temporary password can be created through the mobile app and can be used by any authorized user. The team suggested to keep other ways of entry specified to the household members. The system is capable of storing the data of 100 users, so multiple people access does not overwhelm the system.

Smart lock is user friendly; installation time is estimated under 2 hours. The lock is also power efficient, where all authentication methods were set to low power mode for low power consumption purposes.

### **3.1.2 Keyless Entry, Fall 2019**

In the same graduating year of Fall 2019, another group worked on the same idea of smart locks. Group 11 goals were to provide a competitive design for a smart lock that increases security and accessibility to homes while keeping the cost to a minimum.

Keyless entry smart lock provided different access methods. RFID, fingerprint, keypad, and a mobile app. These were all methods that does not require the user to use a traditional key, which was the aim of this project. The used RFID sensor operates at 13.56MHz which is a secure frequency that is used for financial transactions. The capacitive fingerprint reader provides another secure way of entry since it is immune to photo impersonation of fingerprints. A keypad and mobile app were also used as in the discussed previous products.

An accelerometer was also used to serve as another way to gain entry by using a sequence of door knocks, but also provides a safety feature by notifying the

primary user in case of forced entry. Because of size constraints the team chose four AAA batteries to power the lock, these batteries are recyclable and easy to replace. A record of the door lock activities is available on the mobile app and can be used to better monitor the lock.

## **3.3 Technical Research**

In the following sections, we discuss the research into different technologies related to our project and justify the reason to pick one technology over the other. The technology research and comparisons conducted throughout the following sections will help us determine the best low speed communication interfaces, technology options available, algorithms, frameworks, and much more information. All of this will all be used to guide our design process and allow us to create the most efficient and effective design possible.

### **3.1.1 Wireless Communication Technology**

Wireless communication is the method of transferring data between two or more devices without the need of using a physical channel, however, still permit unguided transmission of an electromagnetic wave signal. Common wireless technologies use radio waves to transmit and receive the message signal. Radio waves are electromagnetic waves that float freely in the space in all directions, and therefore the receiving antenna does not have to be aligned with the transmitting antenna to be able exchange information. The most common uses of wireless communications are GPS, remote controls, Bluetooth, Wi-Fi along with other wireless technologies.

Wireless technologies have three main types that help distinguish the applications for each type. The first type is Simplex, which is a one directional communication system such as TV's and radio broadcast. The second type is Half-duplex, which is a bi-directional that has no delays such as walkie-talkie. Last, the Full-duplex system where both parties can communicate with each other at the same time such as mobile phones. There is a variety of wireless communication technologies, we are to discuss the two most relevant to our project.

#### **3.3.1.1 Bluetooth Vs Wi-Fi**

Bluetooth technology was developed later in 1990s, the main goal for the innovation of Bluetooth technology was to allow short range wireless communication between electronic devices, such as a computer and a smart phone. Bluetooth uses short range radio waves to connect close range devices instead of the infrared spectrum that is used by traditional remote controls. This technology did not only remove the need for using wire connections but also eliminated the need to maintain a clear line of sight in order for two devices to

communicate. Bluetooth is a connection-oriented technology, meaning that Bluetooth devices must become connected prior to data being transferred. Wi-Fi technology, which stands for Wireless Fidelity, also uses radio waves to connect two devices in close proximity to each other, but Wi-Fi breaks the signal into pieces and then transmit those over a range of frequencies. This strategy allows the signal to be transmitted at a lower power per frequency rate, and also permits different devices to access the same Wi-Fi transmitter.

Although both technologies allow for wireless communication and sometimes serve similar functions, but they differ by many other aspects like their purpose, capabilities and more. The main difference between the two technologies is that Wi-Fi requires an internet access while Bluetooth does not. Bluetooth is useful for application with short range information exchange, for example Bluetooth is commonly used in headsets for mobile phones. On the other hand, Wi-Fi provide internet access to phones and other electronic devices.

Wi-Fi consistently evolve to handle the growing demands for a faster data transfer rate. It also had to expand to bargain with the fast increase of devices that need to be connected to Wi-Fi network across the world. Wi-Fi uses 802.11 networking standards for any device connected on a Wi-Fi network. 802.11 standards have 5 different types. 802.11a uses 5GHz frequency band on the radio wave spectrum, which allows for data exchange up to 54 Mbps. It also uses Orthogonal Frequency-Division Multiplexing which divide radio signal into multiple sub signals to reduce interference. 802.11b is the most popular Wi-Fi but also the slowest, it uses 2.4GHz frequency band and allow for 11 Mbps data exchange rate. 802.11g combine the two previous discussed networking standards where it works on 2.4GHz frequency band but with a better data exchange rate of 54 Mbps. 802.11n is the most widely used networking standards because its fully compatible with the three previous standards, but also can transmit 4 streams of data simultaneously. Lastly 802.11ac is the newest standard which is still in review process by IEEE. This standard is fully compatible with all the previous standards and work on both 5 and 2.5 GHz bands, and data exchange rate is about 450Mbps.

Like all radio wave signals, Wi-Fi is very sensitive to interference from other electronic devices using the same frequency band, which can include either smart or Bluetooth devices that rely on 802.11b and g networking standards. To solve such a problem, Wi-Fi has multiple channels that allow devices to switch between channels to avoid possible interference. As a conclusion, we now know that we could adjust the channel our system is connected to if the connection keeps slowing down or dropping. A comparison between the two technologies is illustrated in the Table 2 below to help identify the differences between Wi-Fi and Bluetooth, as if one is needed in the project, a quick reference will be handy to help us determine which is the better option in our specific application.

| Criteria      | Bluetooth                | Wi-Fi                  |
|---------------|--------------------------|------------------------|
| Frequency     | Operates at 2.5GHz       | Operates at 2.5 & 5GHz |
| Data Transfer | Up to 25 Mbps            | Up to 250 Mbps         |
| Range         | 30 meters                | Beyond 100 meters      |
| Power         | About 3milliamps         | Uses a lot of power    |
| Connections   | Connects up to 7 devices | No limitations         |

Table 2: Comparison Between Bluetooth & Wi-Fi main Criteria

Using both discussed wireless communication methods above is essential for our project. Upon discussing the advantages and disadvantages of both methods, our group decided to use Bluetooth technology to establish a connection between the smart lock and the microcontroller (MSP430) since the lock is very close to the control unit. Wi-Fi technology will also be used to allow the system to send email notifications to the owner, once an opening attempt is made. Raspberry Pi comes with an on-board dual-band 802.11ac Wi-Fi, which will be used to send notification to the owner. This will be discussed later on in the Raspberry Pi section.

Bluetooth is a wireless technology that permit the transfer of information between devices within short range of each other's. Bluetooth waves only travels for short distances and change their frequencies constantly. This technology only available when the distance between the devices is approximately 30 feet with no obstacles present during transmission. Connecting devices via Bluetooth is a safe way against hacking since Bluetooth uses Frequency Hopping Spread Spectrum (FHSS), where the device can operate on different frequencies and hop between them hundreds of times per second.

A version of Bluetooth with reduced power consumption is called Bluetooth Low Energy (BLE). It operates at 2.4GHz and forms Personal Area Network (PAN). BLE operates on the same band as Bluetooth but uses different FHSS schemes. The major difference between BLE and traditional Bluetooth is the BLE remains in sleep mode unless a connection is initiated. BLE is recommended for longer battery life in connections where there will not be a large transfer of data, such as IoT applications. In this section we are going to discuss some of the commonly used Bluetooth modules in project designs.

**HC-06(\$9):** HC series is one of the most popular series of ICs with hobbyists and engineers. HC-06 can be simply setup with MCU's and is compatible with smart android phones. HC-06 can only be used as a receiver since it works as a slave.

The operational mode cannot be change to a master mode. Table 3 shows some specifications of the HC-06 module.

|                                |                                  |
|--------------------------------|----------------------------------|
| <b>PCB Size</b>                | <b>39.5mm x 20.5mm x 1.6 mm</b>  |
| <b>Bluetooth protocol</b>      | Bluetooth V2.0 protocol standard |
| <b>Operating voltage range</b> | (+3.3 to +6) VDC                 |
| <b>Operating Current</b>       | 40mA                             |
| <b>USB protocol</b>            | USB v1.1/2.0                     |
| <b>Range</b>                   | <100 m                           |
| <b>Operation mode</b>          | Slave                            |

Table 3: HC-06 Specifications

**HC-05(\$8):** Very similar to HC-06, the main difference that HC-05 can be operated as a master and as a slave, therefore it can be used as a transmitter or a receiver. Below are some specifications of HC-05 module. It communicates with the microcontroller or computer through serial communication. The HC-05 is designed for transparent wireless serial connection setup. Table 4 shows the specifications for this module.

|                                |                                  |
|--------------------------------|----------------------------------|
| <b>PCB Size</b>                | <b>28 mm x 15mm x 2.5 mm</b>     |
| <b>Bluetooth protocol</b>      | Bluetooth V2.0 protocol standard |
| <b>Operating voltage range</b> | (+4 to +6) VDC                   |
| <b>Operating Current</b>       | 30 mA                            |
| <b>USB protocol</b>            | USB v1.1/2.0                     |
| <b>Range</b>                   | <100 m                           |
| <b>Operation mode</b>          | Slave - Master                   |

Table 4: HC-05 Specifications

**RioRand® Bluetooth 4.0 BLE Low Energy/Power RF SOC Transceiver Smart (13\$):** This model can be controlled using ASCII AT commands over UART. The range of this module is larger than HC-06/05 and can reach up to 70m. Similar to HC-05,

RioRand can be operated as a master and as a slave. In Table 5 some specifications of RioRand module are shown.

|                                |                                  |
|--------------------------------|----------------------------------|
| <b>PCB Size</b>                | <b>26.9 mm x 13mm x 2.2 mm</b>   |
| <b>Bluetooth protocol</b>      | Bluetooth V4.0 protocol standard |
| <b>Operating voltage range</b> | +3.3 VDC                         |
| <b>Operating Current</b>       | 50 mA                            |
| <b>USB protocol</b>            | USB v1.1/2.0                     |
| <b>Range</b>                   | <70m                             |
| <b>Operation mode</b>          | Slave -Master                    |

Table 5: RioRand Specifications

**XS3868 Bluetooth Stereo Audio Module (\$5):** XS3868 is one of the cheapest modules available. Can be easily connected to iPhones and Android devices. Below are some specifications of XS3868 module. The effective transmission distance is 20 meters, which is suitable for this project. Tabel.5 shows the specifications of this module.

|                                |                                  |
|--------------------------------|----------------------------------|
| <b>PCB Size</b>                | <b>49 mm x 22mm x2.2 mm</b>      |
| <b>Bluetooth protocol</b>      | Bluetooth V4.0 protocol standard |
| <b>Operating voltage range</b> | (+3.3 to +4.2)VDC                |
| <b>Operating Current</b>       | 400 mA                           |
| <b>USB protocol</b>            | USB v1.1/2.0                     |
| <b>Range</b>                   | <70m                             |
| <b>Operation mode</b>          | Slave -Master                    |

Table 6: XS3868 Specifications

**Bluefruit Ez-link: (\$38):** This module is considered one of the best modules available. With a range up to 32feet and can be communicated with wirelessly without additional hardware's or software's. In Table 7 some specifications of the module are shown.

|                                |                                  |
|--------------------------------|----------------------------------|
| <b>PCB Size</b>                | <b>41 mm x 20.5mm x 4mm</b>      |
| <b>Bluetooth protocol</b>      | Bluetooth V4.0 protocol standard |
| <b>Operating voltage range</b> | +3.3 VDC                         |
| <b>Operating Current</b>       | 50 mA                            |
| <b>USB protocol</b>            | USB v1.1/2.0                     |
| <b>Range</b>                   | <10m                             |
| <b>Operation mode</b>          | Slave -Master                    |

Table 7: Bluefruit Ez-link Specifications

### 3.1.2 Machine Learning

Machine Learning is a subcategory of artificial intelligence. It is a specific category or study which develops algorithms which can be trained from data. Once the algorithm is trained then it can predict future values. In our project we will be implementing facial recognition and voice recognition which falls in the realm of machine learning.

Machine Learning comes down to a mathematical process where there are multiple layers, and nodes. These nodes are interconnected, where each connection can have a different weight. Each column of nodes is referred to as a layer, and generally the more layers there are then the more complex but accurate the system is. This will also cause the algorithm to take up more resources. We can see an example of this in the figure below. Figure 6 shows the fundamental topology of a machine learning algorithm, such as the ones that we will need to implement for the facial recognition and voice recognition authentication methods.



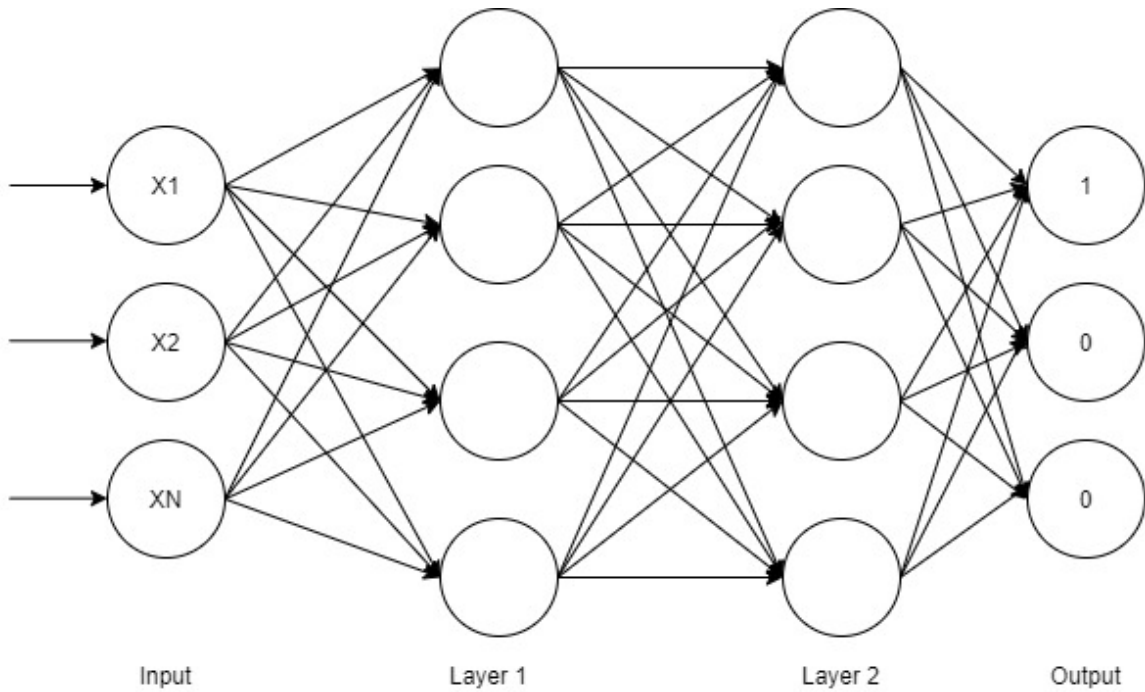


Figure 6: Machine Learning Diagram

In this figure we can see that there are three input nodes, eight inner nodes, and three output nodes. This figure has two layers, and between these layers back propagation will occur. Back propagation is an algorithm which generates the most effective weights and biases given one input. This process will give the changes of biases and weights that would provide the highest accuracy.

A first-order optimization algorithm can be used to determine a local minimum in relation to a differential function. This algorithm is known as gradient descent, and it can be used to improve the parameters of the machine learning algorithm. A gradient descent vector contains different components of the error function. These components are partial derivatives of the weights and biases.

$$\frac{\delta E}{\delta \omega^{(L)}} = \frac{1}{n} \sum_{k=1}^n \frac{\delta E_k}{\delta \omega^{(L)}}, \quad \nabla E = \begin{bmatrix} \frac{\delta E}{\delta \omega^{(1)}} \\ \frac{\delta E}{\delta b^{(L)}} \\ \dots \\ \frac{\delta E}{\delta \omega^{(L)}} \\ \frac{\delta E}{\delta b^{(L)}} \end{bmatrix}, \quad w \leftarrow w - \alpha \frac{\delta E}{\delta w}$$

By using these equations, we can utilize the error function which will give us the optimal parameters for our machine learning model. Also,  $\nabla E$  in this equation represents the gradient descent vector.

### 3.3.3 Facial Recognition

Within our project we will be using facial recognition to identify authorized Users. We will upload multiple images of them to a folder, our code will then open the directory, select the name of the person we are authorizing, and then begin the training process. If the user is not authorized, then the system should raise a flag. As mentioned previously, we plan to make this feature be at least ninety percent accurate.

Facial Recognition allows a program or application to recognize a human's face. Commonly the facial recognition algorithm is trained with multiple images, and by having more images the algorithm tends to be more accurate. However, it will take longer for the program to run, thus taking up more resources on the CPU. Training the algorithm is a process where multiple images are scanned pixel by pixel and compared to each other. Once the facial recognition algorithm is trained then it will be able to predict or recognize future data in real-time or with new images. If the system is not accurate enough then additional images can be added to the training data set, which will increase the accuracy.

So, there must be a balance of having enough data to train the algorithm, but not too much where the processor is being unnecessarily over utilized, and the application is running inefficiently. By testing the programs accuracy after adding and removing training data, the developer may be able to determine the required amount of training data. Having an efficient application should always be the developer's objective because it allows the processor to free up space. As well as allow for easier maintenance when individually working on code, or in a group setting.

In addition, to get the most accurate facial recognition identification the individual should be directly in front of the camera, the lighting should be bright, and the camera quality should be decent. This will make it easier for the system to train and become more accurate. There are libraries that allow for simpler implementation of facial recognition algorithms. These libraries have been around since the 1990s and have significantly improved. The advancement of them allows for some imperfections when taking pictures to upload for training.

Also, it will allow the system to recognize users when they are in different angles. Once the users face is identified, then the algorithm will scan the image and analyze the structure of the individuals face. Next the image will be translated into data. By using some equations, the unique characteristics of the individuals face will be turned into numbers that will be used to develop a face print. Since everyone has different facial features, everyone's face print will be different. When the facial recognition software develops a face print for a specific person, it will compare it to other users and determine a match. Thus, at a very high level is the process for facial recognition.

As mentioned previously, there are different libraries which allow for various ways to implement facial recognition software. These libraries tend to rely on mathematical operations like matrices, and vectorization. Due to the increased demand of this software from different companies and organizations, there has been significant improvements for these libraries. The improvements include but are not limited to increased accuracy, easier implementation, an expansion of supported programming languages, and much more. Initially certain libraries would only support one language, like python, but now they support other languages like C++ or Java.

These different languages can be more beneficial than another depending on the application. For instance, C++ may be better or more reliable than Java when dealing with flight software. This is due to the different way that each language manages memory. For resource management, Java uses garbage collection which reclaims memory. Java's garbage collection is completed through the virtual machine (VM). C++ operated directly on the machine where the code is running. So, the memory is managed via operating system (OS) services.

Python is another powerful language due to the nature of it. Python compared to another language such as C, is somewhat forgiving when it comes to syntax. When programming in C it is very important not to forget any semicolons, and you have to consider memory allocation as well as deallocating memory. In python you do not use semicolons after every line, but it depends heavily on indentation, spaces, and lines. This is not to say that python is better than C, because each language can have a benefit over the other depending on the application. However, since python supports the basic features of Object-Oriented Programming (OOP), but is not considered an OOP language. There are many facial recognition libraries for Python compared to C.

Furthermore, some of the many libraries that support facial recognition are Kairos Face Recognition, Microsoft Computer Vision, Lambda Labs, OpenCV, Amazon Rekognition, and many more. Similar to how the effectiveness of each language depends on the application. Each of these libraries can be more appropriate than the other depending upon the developer's requirements. For example, Kairos is optimal for features detection, and locating faces. While Microsoft Computer Vision excels in processing material from images.

OpenCV began with Intel in 1998, and the official release was in 2000. It is a package or library which primarily focuses on real-time computer vision. It is an also open source, which has allowed it to become very popular. Since the library is cross-platform, many different programming languages can implement it. Some of the different supported libraries are python, C++, Java, and many more. Intel still financially supports OpenCV which allows the library to improve. This is amazing because the package is twenty-three years old.

Azure or Microsoft's Computer Vision is another powerful API which can be used to detect or analyze many things. It is possible to detect objects, images, brands, faces, or even explicit content within an image. This library utilizes powerful algorithms which examine images and returns data related to the desired visual references. This library also has a large database of celebrities, and landmarks, so if an image of one is presented, then they can be detected. There are many tutorials available online which can help guide a developer to begin implementing this package into their project. Since this is an open-sourced Microsoft product there are many references online which is very helpful when creating an application.

Amazon developed Amazon Rekognition in 2016 as a subscription-based service. This software is cloud-based and provides an easy computer vision solution to its clients. It is an easy approach because the user is not required to have any prior machine learning (ML) experience prior to using this service. This software can detect people, objects, texts, hardware components, scenes, and much more. A grocery store could use this software to do an inventory of the products on their shelves. Or it can be used within an assembly line to quickly scan the items on the belt. This software can also be used as a security feature to monitor someone's home for any burglars or monitor a street for vehicle collisions. The client is able to train the model by uploading images that they would like the software to recognize. Once the images are uploaded, then the software will begin to detect whatever the client desires. This software is very easy to use; however, it is subscription based, and the software does all of the work.

As we can see there are many different libraries that are available to implement facial recognition. This makes sense because there are countless amounts of ways to implement this software. On most modern smart phones, there are facial recognition software's which run in real time to detect faces when taking pictures with the camera application. Or on social media websites like Facebook when an image of a person is uploaded the application will make suggestions linking the person in the picture to their social media page. There are also articles online stating that in China there are cameras in the street which are able to detect citizens faces. So, there is a wide spectrum of possible applications for facial recognition. For our application we will be leaning towards an API similar to OpenCV or Kairos, as opposed to a software like Amazon Rekognition. These libraries will allow us to implement facial recognition and send required data to other fields of our program to trigger different events.

For our application we will be using a camera which will be running the facial recognition software in real-time. The system will have to recognize the authorized user and be able to send some sort of validation message. This message may contain the name of the person, or their profile. If the user is not in the database, then they are not authorized, and a signal should be sent as well. To achieve this sort of application we may have to implement message threading. Where on one thread the facial recognition software is running, and once it is ready to send a message then another thread will receive the message and utilize the data. An

example of this would be “door unlock”, this could be running in some validation thread which is waiting to receive all of the required data before unlocking the door. So, once the users face is recognized, that thread will send a message to the validation thread stating that that authentication requirement is met. If all of the conditions are satisfied, then the door should unlock.

### 3.3.4 Voice Recognition

Another main feature of our project is the implementation and use of voice recognition. Voice recognition is a highly sophisticated process. Before getting into the vast details of voice recognition, we would like to make clear the difference between ‘Voice recognition’ and ‘Speech Recognition’. The difference between voice recognition and speech recognition might seem arbitrary, and many people might actually think that they are one in the same. But in reality, they have a very particular distinction that makes them different. Although there are some similarities in terms of the overall process and algorithms used to accomplish each, the difference between the end goal of voice vs speech recognition is key to consider. Although voice and speech recognition are many times used interchangeably, and you can find examples of this all over the internet for those who do not understand or know the difference, they actually mean completely opposite things.

Voice recognition can be defined as the process that allows Artificial Intelligence to recognize and decode *whose* voice is being spoken. It allows the computer to detect specifically who is speaking based on differences in tone, frequency, accent, and other unique features of each human being’s voice. Everyone’s voice is unique and different, especially when broken down to the fundamental levels of frequency, tone and pace of the words spoken. This is a result of their physiology (shape and size of the mouth and throat) and behavioral patterns (their voice’s pitch, accent, their speaking style, etc.). Voice recognition is also (correctly) referred to as “Speaker Recognition” or “Voice Biometrics”. Essentially, the fundamental idea of voice recognition is to identify the person who is speaking, not what words are spoken by that person.

Speech recognition on the other hand, can be defined as the process that allows Artificial Intelligence to recognize and decode specific words that are being spoken. It allows the computer or device to detect the words or text that is being spoken, no matter who is speaking those words. Devices such as smart phones, computers, and even automobiles use audio hardware such as built-in microphones to capture the user’s spoken words and translate it to text. After the spoken words are translated to text, the computer might use that text to understand and execute commands or respond back to the user. This is the case if you consider Apple’s Siri, Amazon’s Alexa, Google’s Home, and many other voice-controlled systems that capture speech from a user and translate that speech to text in order to execute some command or function, such as turning on a light in your home (assuming that the light is connected to your system’s network) or using

its connection to the internet to look up and respond with the latest weather forecast. Speech recognition is also (correctly) referred to as “Voice-to-Text”, or “Speech-to-Text”.

Because voice recognition is used to capture and identify or verify a specific person that is speaking, it is highly related to authentication and cybersecurity, which is perfect for our application. Using a speech recognition implementation to secure your home would not be very beneficial as we learned that speech recognition is used to translate anyone’s voice or spoken language into text to execute commands. This means that anyone could walk up to your home, say the magic word(s) or phrase, and the system would allow it, because it is only identifying the words and not the person saying them. This is why Voice Recognition specifically is of interest to us with regards to our project. Because our project’s goal and main focus is to secure the user’s home in a more advanced and secure way than a typical lock and key, we would benefit greatly from the correct implementation of voice recognition. Using voice recognition, we can use the power of Artificial Intelligence to authenticate users who would like to enter the home. In this case, only specific users that have been pre-authorized will be recognized and granted access. This will essentially keep anyone not authorized out and allow specific users that have been added to the system to be allowed to enter the home.

Now that we have discussed the key difference between voice recognition and speech recognition and the benefits therein, we will examine more details of voice recognition and how it is implemented specifically, including existing frameworks, libraries and more.

To understand how voice recognition works we need to understand the world of Artificial Intelligence. AI, Machine Learning, and Deep Learning are all buzz words commonly used in tech talk, but as Artem Oppermann says in his online article, “There is a big misconception among many people about the meaning of these terms”<sup>[41]</sup>. At the very worst, the average individual might think that these are the same things which is simply false. To give a good overview of the broad difference between AI/Machine Learning/Deep Learning we included the Figure 7, below.

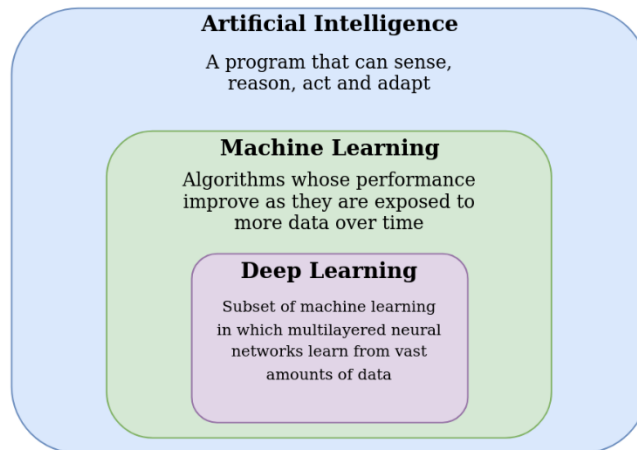


Figure 7: Artificial Intelligence vs. Machine Learning vs. Deep Learning

As you can see, these tech buzz words are not one in the same, or interchangeable. For example, Deep Learning is a subset of Artificial Intelligence, but Artificial Intelligence is not always Deep Learning. This is a very important concept to grasp as the voice recognition algorithm uses deep learning to accomplish its tasks.

The voice recognition model is built using a Deep Neural Network, which can be pictured by a visual representation like the example shown in the previous section 3.3.2 Machine Learning. This is a visual representation of a deep neural network, that was invented to mimic how the human brain actually works. The deep neural network is a network of “neurons” that are connected together through many hidden layers. The neural network takes an input, calculates some specific mathematical functions on that data that was inputted, and is able to use back propagation to learn from the mistakes the neural network model makes over time. This allows the model to be trained to a very high level of accuracy which is essential for creating a model that can be deployed in the field for the specific applications it was trained for. In this case here, the Voice Recognition neural network model will leverage its ability to be trained to recognize specific people’s voices, namely the users that we want to be the homeowners who should be authorized at the door.

By taking advantage of this deep learning neural network model for voice recognition we can train a model to be highly accurate at recognizing specific user’s voices. That is perfect for our application, where we will train the model on the user’s we want to be included as those who are authorized to enter the home.

Once the model is trained, we will be able to deploy the deep learning model for real time identification of users who approach our system’s door. The deployed model will take input from users who are at the door, and their voices will be run through the trained model. Our Voice Recognition implementation will then be able to very accurately determine who is speaking at the door, and if the user that is detected is someone that we have trained the model to recognize, we will be able to open the door for that user [providing the user also meets one other method of authentication, either facial recognition or a pin number].

### **3.3.5 Power Technology**

In this section we are going to discuss the different manners of providing power to our smart door security system. Researching different methods helps our team get the required knowledge to pick the right technique for powering the different parts contained in our design. It is important to be able to supply all components with steady DC source any damages. In order to do that, the use of transformers, rectifiers, buck converters. We will need three discrete voltage levels for our project, so we need to understand how to convert between them. We think that the optimal solution is to bring power in on the highest voltage level required, and then drop it down as needed for the other components in our system.

### **3.3.5.1 Transformers**

Transformers are electromagnetic devices that transfer electrical energy from one electrical circuit to another while keeping the same frequency. It consists of three main pieces, a core made of a ferromagnetic material, and two sets of wire coils referred to as the primary and secondary windings. Applying an AC current to the transformer's primary winding, creates a pulsing magnetic field.

Applying an input voltage to the primary winding, induces an alternating current flow in the primary winding. This current flow produces a changing magnetic field in the transformer core. To eliminate wasted energy, the core of the transformer directs the magnetic field path between the two windings. Without the core, half of the magnetic field generated at the primary would be beyond reach of the secondary coil. As the magnetic field reach the secondary winding, an alternating voltage is produced in the secondary winding.

Transformers primary function as a step up to increase a low AC voltage at a high current, or a step down to decrease a high AC voltage at a low current. Output and input voltages are determined by the ratio of the number of turns between the two windings. If the secondary winding has more turns than the primary winding, the transformer output voltage will be greater than the input voltage which considered a step-up transformer. On the other hand, if the secondary winding has less turns than the primary winding, the output voltage is lower than the input voltage which is considered a step-down transformer.

After gaining good understanding for the working mechanism for transformer and led by the decision of our team to use the wall outlet to power our system. A research for different transformers was conducted. Wall outlet supplies a 120V AC power, which is a lot higher than our design needs. For that we need a step-down transformer to reduce the outlet AC voltage to the right amount of AC input voltage needed to be supplied to the rectifier. Rectifiers are discussed in the following section.

### **3.3.5.2 Rectifiers**

Rectifiers are electrical devices that turn an alternating current (AC) source that reverse direction periodically to a direct current (DC) source that flow in one direction only. Physically rectifiers come in different forms, for this project a simple design will serve the purpose. In this section we are to compare three types of rectifiers, half wave, full wave rectifiers.

#### **Half-Wave Rectifier**



In rectifiers, diodes work as switches. If the diode is connected in forward biased it serves as a closed switch, while a diode connected in reverse biased serves as an open switch. When AC signal, transformer and a diode are connected we get the circuit of half rectifier, where the output is observed across a resistor that acts as a load.

For half-wave rectification of a single-phase supply, half of the AC wave passes, while the other gets blocked. From a mathematical point of view, it is a step function (pass positive half, block negative half). When an AC waveform passes through a half-wave rectifier, it allows only one half-cycle (positive or negative half-cycle) of the AC voltage through while other half-cycle gets blocked on the DC side.

Since only half of the input waveform appears at the output, voltage is lowered. Half-wave rectification requires a single diode in a single-phase supply, or three in a three-phase supply. The good side of rectifiers is yielding a unidirectional current, but it comes with a downside. Fluctuating direct current; half-wave rectifiers produce far more ripple than full-wave rectifiers, which means additional smoothing is needed to eliminate harmonics of the AC frequency from the output and produce a uniform steady voltage.

### **Full -Wave Rectifier**

To reduce the voltage variation or ripple, smoothing capacitor can be connected across the load resistor. While this method may work for application with low power consumption, it's not the best method to use with application that need a steady power supply. This circuit that allows using every half-cycle of the input voltage instead of using every other half-cycle is called Full wave rectifier.

The very first advantage of a full wave rectifier is having a higher DC output voltage than the half wave rectifier. It also has less ripple which produce smoother output waveform.

Full Wave Rectifier circuit uses two diodes, one for each half-cycle. The used transformer has its secondary winding split equally into two halves with a common center connection. This design allows the two diodes to conduct in turns when its anode terminal is positive with respect to the transformer center point which makes the full wave rectifier 100% efficient.

### **Full Wave Bridge Rectifier**

Full wave bridge rectifier produces the same output waveform as the full wave rectifier circuit. This type uses four rectifying diodes connected in a closed loop called "bridge" to produce the desired output.

Full wave bridge circuit does not require a special center tapped transformer, which considered the main advantage for the bridge rectifier as it reduces the cost and size.

Although four diodes can be connected in bridge configuration to produce a full wave bridge rectification circuit, there is premade rectifiers made by different electronics companies. Such designs can be considered to simplify soldering the required components to a PCB.

### 3.3.5.3 Buck converters

Buck converters are necessary to convert the DC output voltage of the rectifier to the DC voltage level needed for the different component in our system. In this section we are to discuss the basic buck converter operates.

A buck converter is a DC-to-DC power converter, that steps down voltage from its supply (input) to its (load) output, while drawing less average current. Typical buck converters contains at least two semiconductors (a diode and a transistor). Modern buck converters frequently replace the diode with a second transistor used for (synchronous rectification) with at least one energy storage element. To reduce voltage ripple, filters are normally added to the converter output (load-side filter) and input (supply-side filter).

Buck converters provides greater power efficiency as DC-to-DC converters than linear regulators. Linear regulators are much simpler circuits that lower voltages by dissipating power as heat, without increasing the output current.

When designing a buck converter, logical steps need to be followed in order to construct a usable part. These steps are:

- **First:** The first step is to determine the voltage at the input side of the converter, the required output voltage, and the load current. The duty cycle is another factor to be considered to realize voltage or current feedback control. The duty cycle is the ration of the turn on time to the complete cycle length. A step-down converter duty cycle is given by:

$$D = \frac{V_{out}}{V_{in}}$$

- **Second:** the second step is determining the output power delivered by the converter.

$$P = VI$$

By conservation of energy, and assuming the buck converter is 100% efficient, the power delivered to the buck converter should be equal to the output power of the converter.

$$P_{in} = I_{in}V_{in} = P_{out} = I_{out}V_{out}$$

If we consider a buck with efficiency ( $\eta$ ).

$$P_{out} = \eta P_{in}$$

- **Third:** Power transferred per pulse

By dividing the output power by the selected switching frequency, we get the power transferred per pulse to the connected load.

For simplicity, we can assume that the output power is the output energy per second. For example, if the output of our converter is 30 Watts, then it's safe to say that the output energy is thirty Joules every second.

- **Last:** Calculating the needed inductance inductance can be found using the relation.

$$L = \frac{2E}{I^2}$$

Where E is the energy transferred per pulse and I is the input current. Using the values of the inductance, frequency, and duty cycle, we can now design a simple converter.

### 3.3.5.4 Power Circuit Design

Comparing the specs for different transformers, rectifiers, and buck converters, the final schematic of the power circuit is shown in Figure 8. The factor in determining which components to use was led by multiple factors like cost, efficiency, size, and availability of the parts. Our power requirements at various voltage levels are shown in Table 8. The camera and speaker requirements power requirements are not provided by the manufacturer, but the requirement for USB devices is that they are not to use more than 2.5W.

|        | Component            | Voltage (V) | Current (amps) | Power (Watts) |
|--------|----------------------|-------------|----------------|---------------|
| Supply | Adapter              | 12          | 10             | 120           |
|        | Electric Strike Lock | 12          | 0.5            | 6             |

|        |                      |      |          |          |
|--------|----------------------|------|----------|----------|
|        | Magnetic Reed Switch | <=12 | marginal | marginal |
| Supply | 5V Buck Converter    | 5    | 5        | 25       |
|        | Raspberry Pi         | 5    | 1.8      | 9        |
|        | Raspberry Pi         | 5    | 1.8      | 9        |
|        | HC-SR04              | 5    | 0.025    | 0.125    |
|        | Camera               | 5    | 0.5      | 2.5      |
|        | Speaker              | 5    | 0.5      | 2.5      |
| Supply | 3.3V Buck Converter  | 3.3  | 0.2      | 0.66     |
|        | MSP430FR6989         | 3.3  | 0.001    | 0.0033   |

Table 8: Power Supply Requirements

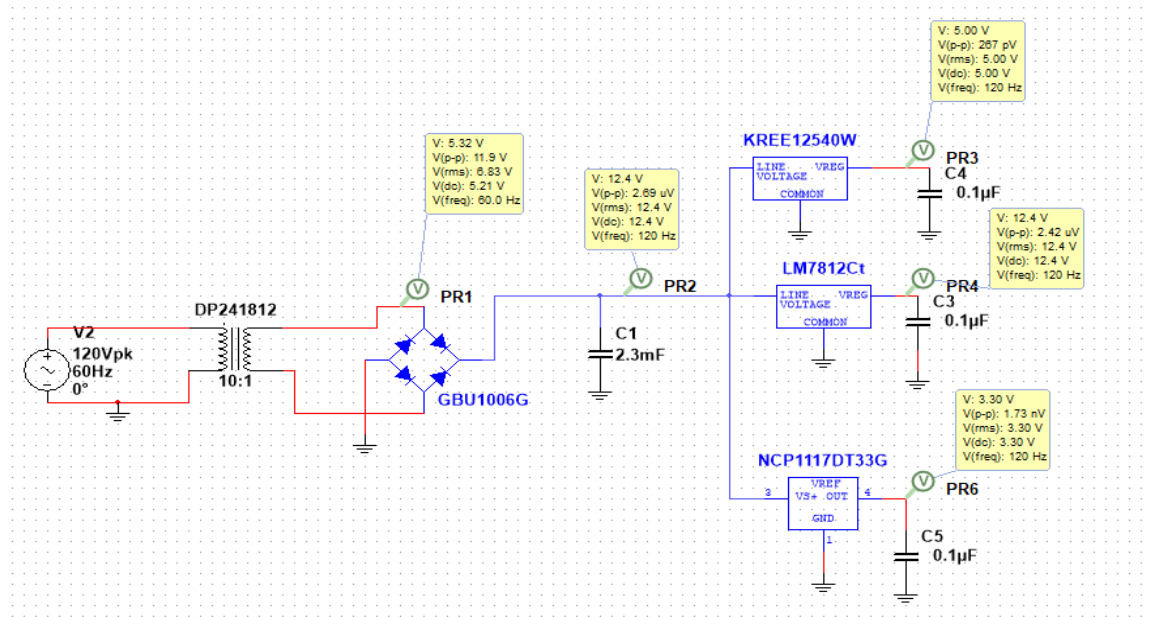


Figure 8: Schematic of the Power Circuit

We have started by designing the transformer, by determining the turns ratio between the two windings by using the equation below:

$$\frac{V_p}{V_s} = \frac{N_p}{N_s}$$

The ratio between the voltage of the primary winding to its secondary winding equals the number of the turns of the primary winding to the secondary winding. In order to step 120VAC to a 12VAC we need a transformer with a turn ratio of 10. Which means that the number of coils of wire at the primary is 10 times greater than the secondary. Table 9 Shows the specifications of the chosen transformer.

| <b>DP-241-8-12</b> |                   |                   |         |
|--------------------|-------------------|-------------------|---------|
| Primary Voltage    | Secondary Voltage | Rectified Current | Cost    |
| 115/230 V          | 12.6 V            | 8A                | \$23.14 |

Table 9: Transformer Specifications

Then a diode was designed to allow us to convert an AC power source to a DC power source. For our design, a Bridge Full wave rectifier was used because it provides an output with less ripple.

In choosing a Bridge rectifier some parameters are important to consider. The voltage peak reverse is the first parameter to consider, which is the maximum voltage that a diode can withstand in the reverse direction without breaking down. The rectified current determines the maximum current each diode can take and handle without breaking down. The forward voltage drop is the voltage lost to thermal heat, the lower the voltage the more efficient and this voltage must be subtracted from the AC voltage to determine the final DC voltage. Taking all these parameters into account, GBU1006-G was chosen. Table 10 Shows the specifications.

| <b>GBU1006-G</b>     |                   |                 |                         |        |
|----------------------|-------------------|-----------------|-------------------------|--------|
| Voltage peak reverse | Current rectified | Forward voltage | Reverse leakage current | Cost   |
| 600V                 | 10A               | 1V              | 10uA                    | \$1.73 |

Table 10: Rectifier Specifications

To convert the output of the rectifier into a nicer/ steady output a capacitive filter was introduced. The capacitor reduces the voltage fluctuation at the output of the bridge rectifier. By increasing the capacitor value, the ripple voltage decreases, the capacitor value was tuned to get the best performance.

Buck converters were then added to produce 12, 5, and 3.3-volt nodes to power our different components. LM7812Ct is used to provide a 12V voltage at 1.5A which will be used to power the Electric Strike Lock, and the Reed Switch. KREE-

12-5-40W was needed to provide 5V to the two Raspberry Pi's and the Ultrasonic sensor, while supporting enough current to power the Pi's and the components connected to them. Lastly, NCP1117DT33G was needed to reduce the voltage to the microcontroller (MSP430FR6989) needed level of 3.3 Volts. The chosen buck converters specifications are listed in Table 11.

| <b>Buck Converters</b> | <b>Conversion</b> | <b>Output Voltage Min/Max</b> | <b>Current</b> | <b>Cost</b> |
|------------------------|-------------------|-------------------------------|----------------|-------------|
| KREE-12-5-40W          | 12V-5V            | 4.9-5.1V                      | 8 A            | \$8.99      |
| LM7812CV               | 12V-12V           | 11.75-12.25V                  | 1.5A           | \$0.38      |
| NCP1117DT33G           | 12V-3.3V          | 3.2-3.4V                      | 1A             | \$0.52      |

Table 11: Specifications of the Buck Converters

After doing the analysis, we realized that a premade power supply adapter of 12V is a cheaper and less complicated choice. The 12V DC supply coming from the adapter will be supplied to our PCB, and we're to design 2 buck converters using Webench.com to step down the voltage from 12V to 5V, and then another step down from 5V to 3.3V. Due to the large amount of power required by the Raspberry Pis for computation, we settled on a 10A model. This will also enhance the modularity of the project by providing additional room for add-ons, as we are able to supply more current than we think will be necessary at this point in time.

### 3.3.5.5 Redesign of the Power Supply Circuit

After deciding to use a premade power supply, we needed to redesign our power supply circuit. A power supply adapter of choice converts the wall outlet AC voltage to a 12 DC voltage. The output jack of the adapter will be connected to the printed circuit board and two buck converters will be used to step down the 12V output to the needed two voltage levels (5 and 3 Volts). Table 12 Shows the specifications of the power supply adapter. Figure 9 outlines the needed modules for our new design.

| <b>ALITOVE AC 100-240V to DC 12V 10A Power Supply Adapter</b> |                |         |                        |
|---|----------------|---------|------------------------|
| Input Voltage   | Output Voltage | Current | Output Jack dimensions |
| 100-240VAC  | 12VDC          | 10A     | 5.5mm x 2.5mm 2.1mm    |

Table 12: Specifications of the Power Supply Adapter

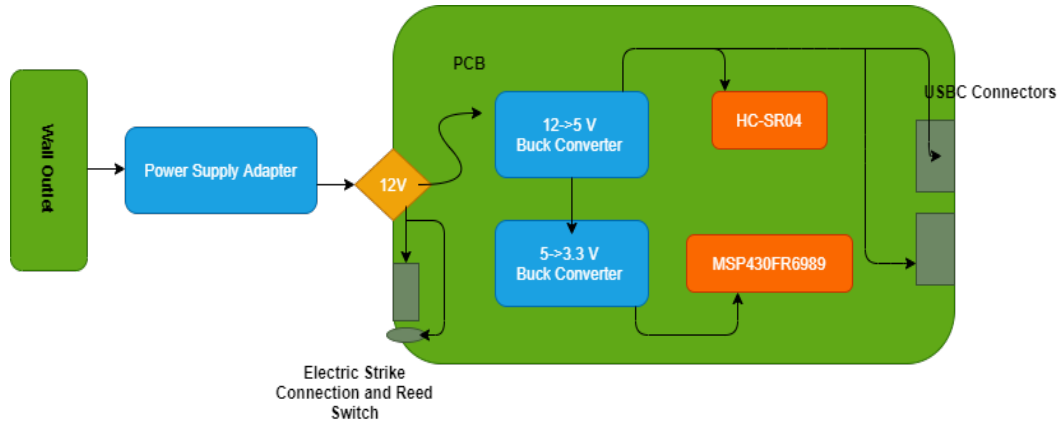


Figure 9: Power Supply Redesign

### 3.3.5.6 Designing using Webench Tool

We used the Webench tool to design our 3.3V and 5V power supplies. We needed to be able to supply enough current at 5V to run both Raspberry Pis as well as our HC-SR04 sensor, camera, speaker, and microphone. We picked a 12V, 10A power supply adapter so that we have a safety factor/margin in the event we expand our project and add more sensors.

For the 5V supply, we used the constraints of 12V input, 5V output, with a minimum load of 5A. We filtered down the designs by efficiency and BOM cost before arriving at a design using the LMZ22005 buck module. The LMZ22005 has an integrated inductor, so we are able to reduce the area needed on our PCB. The output on Webench is shown in Figure 10 and Figure 11.

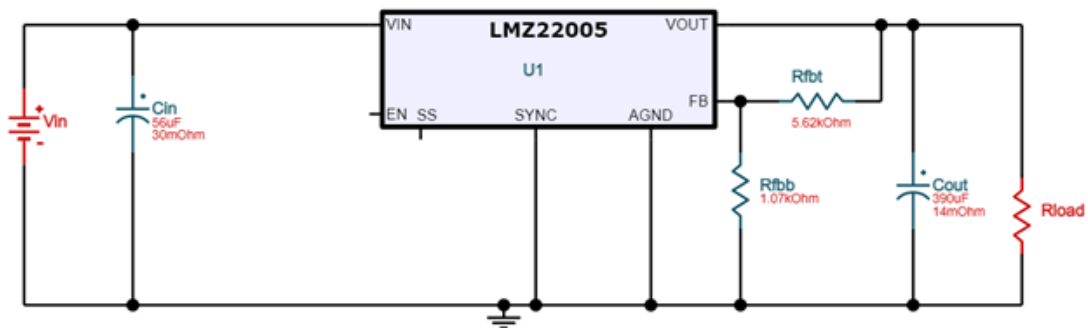


Figure 10: 12V to 5V Buck Converter

We also need a small amount of 3.3V power to operate the MSP430FR6989. The MSP430FR6989 is a low-power device, but we wanted to make sure that we had room to expand the project if necessary. Our design uses the TPS6206063DSG chip to drop the 5V down to 3.3V, with up to 1A output at 91.8% efficiency.

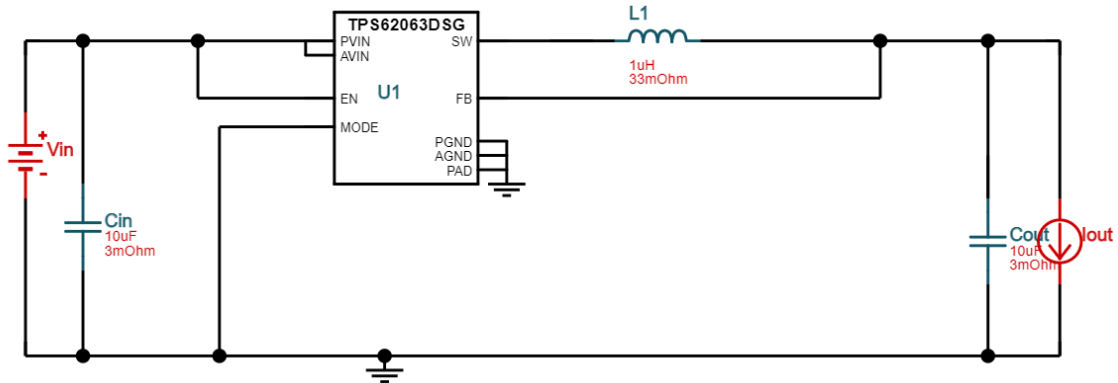


Figure 11: 5V to 3.3V Converter

### 3.3.5.7 Power Distribution

To distribute power to the various sensors and modules, we use a variety of methods. The power supply adapter is plugged directly into the wall, where it reaches the PCB through a 2.1mm jack plug. The 12V is also transferred to pads on the PCB, where the electric strike lock will be soldered on.

The voltage level is dropped to 5V, and then transferred to the Raspberry Pis using a USB-A to USB-C cable. We chose USB-A due to simplicity and availability of female sockets on Eagle. The camera and speaker plug directly into the USB-A ports on the Raspberry Pis, so no design is needed for that. We will undergo extensive testing to ensure that the revision 1 USB-A ports on the PCB are able to withstand the power draw from the Raspberry Pis.

The 5V is dropped down to 3.3V for the MSP430FR6989, but that voltage level is not required by any other components and therefore is used only on the PCB, so no external connections are required at the 3.3V level. The only other place that 3.3V is required is GPIO communication between the MSP430 and the Raspberry Pis, but they do not require an additional supply voltage to communicate.

### 3.3.6 Printed Circuit Board

A printed circuit board (PCB) is a plastic board that electrically connects electronic components using conductive tracks. All components should be soldered/mounted onto the PCB to ensure the electrical connection and to physically secure them to it.

Printed circuit boards are used in all electronic products to replace bread boards and wire connections. The necessity of the PCB for any project is to coordinate power distribution to all parts.



PCBs comes in different types, it can be single-sided (one copper layer), double-sided (two copper layers on both sides of one substrate layer), or multi-layer (outer and inner layers of copper, alternating with layers of substrate).

Multi-layer PCBs usually used with component that have high density. To allow some spaces between the components, the traces are connected on a different layer. Otherwise, the traces will take up the surface space between the components.

### 3.3.6.1 Terminology

Basic understanding of printed circuit board terminology would make working with a PCB much easier. This section will discuss some of the most common words in the industry when referring to a PCB.

- **Via:** Multi-Layers PCBs are connected through a number of holes called Via. It helps preventing traces overlap and very beneficial when different components are connected through different layers. To make soldering easier, Vias are usually kept uncovered.
- **V-Source:** when a design required a smaller PCB size, V-source is used to make an incomplete cut through the panel to help break apart panels of PCBs into single units.
- **DRC:** is a design technique used to make sure that traces are not overlapping. it also makes sure that the size of the drill hole is size appropriate to ensure the other components fit.
- **Pad:** to make electrical connection of the components, a small portion of metal is left on the surface of the PCB. This metal section is called Pad.
- **Surface Mount:** is a technique used for proper soldering on any printed circuit board. It's an easier way to place components. In compare of other technologies, Surface mount provides more compact designs.
- **Trace:** copper tracks that replace the old wire connections. Traces are printed on the PCB itself rather than requiring a jumper wires to connect the nodes of the components.

### 3.3.6.2 Powering PCB

PCB board have multiple layers of the copper which can be used depending upon the complexity of the design. Our design uses a two-layer board which has copper on both the top and bottom sides. Using 2-Layers PCBs allows mounting

conductive copper and components on both sides of the PCB, in which case the traces can cross over each other.

Our PCB is powered by a 12V, 10A adapter. After calculating the current draw for all of our devices at the various voltages, we found that this was the best fit for us. The 12V power is received from a 2.1mm jack plug, for which we have a female connector on the PCB. This means that our power is limited by 120W, so we need to take into account the power requirements for all of the voltage levels required in this project. This 120W requirement is detailed in our design specifications and it should help us choose low-power components when possible. It will also limit the amount of heat generated and hopefully prevent issues associated with cooling the project.

### **3.3.6.3 PCB Design & Fabrication**

For our project, we are hiring a board house to fabricate our PCB. A simplification of the manufacturing steps is explained in this section. Initially, the board design is printed onto the bare board. Images are sketched or flashed onto the PCB. Alternating layers of fiberglass and conductive materials are laminated. The inner layers are printed first, then the outside layers. After all layers are printed, the board is pressed together. The vias are drilled and then pads are placed on the PCB.

Due to the time constraints of this project and the fact that we do not have a pick and place machine, we narrowed our selection of board houses to ones that have low lead time and place the components themselves. We found that OSH Park and RushPCB were potentially a good fit for us due to low cost and quick lead time. Also, they offer assembly, which is essential due to the small package size of many of our components in addition to our lack of access to a pick and place machine.

## **3.3.6 Low-Speed Communication Interfaces**

There are several low-speed communication protocols commonly used for electrical communication between different devices and integrated circuits. A very important note about these protocols; as these are low-speed communication protocols, there is no need to worry about impedance control or transmission line behavior as the data is not sent at a high enough rate to need to factor these things in. Using these protocols is easy to implement and extremely common, as most Integrated Circuit designers have built their devices to be compatible with any or all three of them. Here we will discuss what each of the communication interfaces are and decide what is appropriate to use with our MCUs and peripheral devices.

### **3.3.6.1 Universal Asynchronous Receiver-Transmitter (UART)**

UART is one of the most commonly used device-to-device communication protocols. It is a very simple low speed communication protocol that uses only two wires to enable one device to communicate with another device. The two lines or wires that are used are labelled as: Tx and Rx, which stands for the Transfer and Receiver wires, respectively. UART uses bidirectional, asynchronous, and serial data transmission. Bidirectional transmission of data means that data can be transmitted as well as received by either device. But that is not to get confused with bidirectional transmission of data on a single wire. Each wire is going to only send data in one direction, namely from the Transfer pin of one device to the Receiver pin of the cooperating device. So, since each of the 2 wires involved in the UART communication protocol only send data in one direction, the wires themselves are unidirectional. In other words, the Transfer (Tx) pin on one device or integrated circuit will only ever send data from that pin to the Receiver (Rx) pin on the other device.

Therefore, the wires are not bidirectional, but the communication protocol is. That means that UART is capable of working as a full-duplex communication protocol because data can be sent in both directions simultaneously, (although it is important to note that in order for the UART to be full-duplex the hardware must have a dedicated transfer and receiver buffer, otherwise if the hardware uses a shared buffer for transfer and receiver data then the system would be considered half-duplex). Serial data transmission means that data is sent one bit after another, from one pin of the transmitting device, as opposed to parallel data transmission that sends many bits of data (for example the 8 bits of one byte) at one time, in parallel, using 8 wires between devices.

This protocol is useful because synchronization management is not necessary between devices as there are no clocks involved with sending or receiving this data. This is where UART gets the “asynchronous” part of its name. Asynchronous means the devices are not synchronized, in other words they are on different clock domains. When sending data, the Transmitter will just send the data packet and as long as the Receiver is set up to look for and receive data at that same baud rate, the receiver will receive the data and process it at that expected rate. The technical term for this would be the receiver “sampling” the data at the same baud rate that the transmitter should be set to send data. This is how the devices can be “synchronized” even though they are running on different frequencies based on their internal clocks, but still transmit data effectively. Baud rate effectively means bits per second, so the baud rate determines how fast your data is being transmitted.

There are Start and Stop bits are included in each packet of data so that the processor can determine where the data starts and stops during transmission. There are many standard baud rates for UART, but the most common baud rates are 9600 and 115200 baud. Even though UART is asynchronous, both devices must operate at the same baud rate, otherwise the receiver will not know how fast

the data is being sent to it, and the data will just be corrupted garbage values. The negative thing about the UART communication protocol is that it can only be used between two specific devices that are wired together. One device's Tx pin must be wired to the other device's Rx pin, and vice versa.

Therefore, in situations where you have many peripherals that you need to communicate with, UART is not the protocol to use. But when you need two devices to talk to each other, this is a great choice for a communication protocol as it is very easy to implement. This is most likely the protocol that will be used for each MCU to communicate to its respective Bluetooth module, where each MCU will send data and receive data between itself and its respective Bluetooth module. This is necessary for the System Controller MCU to be able to send and receive information to the MCU that is incorporated in the deadbolt of the door lock. Here the System Controller will send the lock and unlock signal based on if the user that is trying to enter has been authorized, or if the door is closed and needs to be locked behind the user. Figure 12 below shows the typical interface between two integrated circuits, where the devices are connected using two wires, one goes from Tx of one device to Rx of the other, and the Rx of that same device is connected to the Tx of the other.

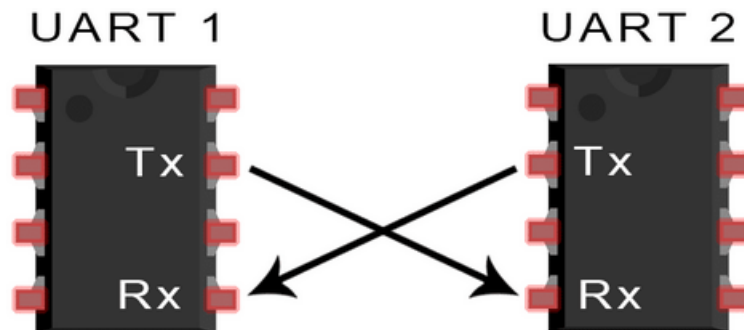


Figure 12: UART Interface

### **3.3.6.2 Inter-Integrated Circuit (I2C)**

Another very popular and powerful low speed protocol used for communication between devices is I2C, which stands for Inter-Integrated Circuit. I2C has many benefits but a major one to point out is that this interface only requires two wires for communication. This is important to note, because although there can be many devices on the I2C bus, an engineer or designer only needs to use two total wires to connect them all together. The two wires shared between all of the devices are "SCL", short for serial clock, and "SDA", short for serial data. Both of the active signal wires on the I2C bus are bidirectional, meaning data can be sent in either direction. I2C is a half-duplex, serial, synchronous communication protocol. A synchronous interface, as opposed to an asynchronous interface like the one we

introduced in the previous section for UART, means that all devices on the bus are driven by the same clock signal, which we introduced as SCL. This is a half-duplex communication protocol because data cannot be sent in both directions simultaneously.

As we will explain later, a master can send a request to a slave and then it must wait for the slave's response of data back. Since this data transfer happens in only one direction at a time it is considered a half-duplex protocol. The other wire that I2C uses, SDA, is the serial data line which is used to send and receive data on the bus. As you might have noticed, I2C is also a serial communication protocol, similar to UART, which means that data is streamed bit by bit but in this case for I2C, the data is sent over the SDA line. The I2C interface is also very commonly called a bus, as there is generally a master device that is connected to many different slave devices that all share this bus, (the SCL and SDA wires). I2C is capable of having multiple masters that can control and write or read data from slaves on the bus, but generally there is one Master device.

Also, there can be up to 128 slaves on the I2C bus, which is another major benefit of I2C. Using UART, only two devices can be directly connected together to communicate with each other but using I2C this absolutely not the case. Using I2C you can have a master microcontroller (and possibly multiple masters) connected to effectively an unlimited number of slave devices, which are usually many different kinds of sensors that are incorporated in your system. This is where the real benefit of I2C comes into play. An example set up for the I2C bus could include a master microcontroller, connected to an Analog to Digital Converter device as one slave, a Digital to Analog Converter device as another slave, as well as another microcontroller as a third slave. This is just an example, but the actual implementation can be extended to incorporate up to 128 different devices and sensors acting as slaves, all with the need to only use two wires to connect them together. In Figure 13 below, we show one possible example of a configuration of devices on the I2C bus, as we have just previously described.

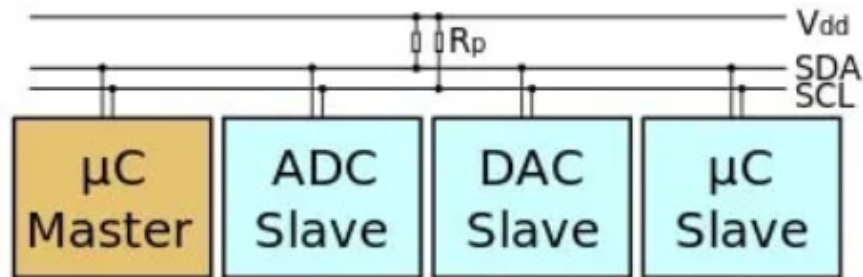


Figure 13: I2C physical connection of many devices, only using 2 wires SCL and SDA

The I2C protocol itself uses a specified data frame for data transfer, which allows the master to write data, or read data from any slave on the I2C bus. By using this

data frame as a consistent way to communicate, we can expect to not have any data get corrupted, incorrectly transferred or have collision on the SDA line. Every read or write request is initiated from the master device by issuing what's called a START condition. A START condition happens when SDA gets pulled low, as SCL is high. This means that during any clock pulse, the master needs to pull the data line low, and this will signal the START condition to all the slaves on the bus, signaling to them to be ready for a request. This streamlines the protocol so that slaves are always ready to receive the request that is coming from the master after the START condition has been initiated. After the master issues the START condition, the first byte that is sent by the master is the "address byte". This address byte includes the 7-bit address of the slave that the master wants to read from or write to, (hence the 7-bit slave address is why there can be a total of 128 slaves on the bus), followed by the 8<sup>th</sup> bit which determines if the master is requesting to read from the slave at the specified address, or if the master is requesting to write to the slave at the specified address. Once the master has issued the START condition, sent the slave address and whether it's a read or write request, the slave that has been requested will send an ACK bit, which is short for acknowledge, so that the master can get confirmation that the correct slave got the request.

If the master is requesting to read from the slave, the slave will immediately send the first byte after it has sent the ACK bit. The master will read the byte and if it wants to read another byte it will respond with an ACK itself, signaling to the slave to send another byte of data, or it will send a NACK bit, short for Not-Acknowledge, signaling to the slave to stop sending data. In the case of a write request, the master will receive the ACK from the slave, confirming that the request was correctly received, then the master has to send a register address byte corresponding to which of the slave's registers the master would like to write to. Following the register address being sent by the master, the slave will again send an ACK signal, and the master then sends the byte of data that it would like to write to the slave's register. Just as the START condition signals when data transmission is going to begin, a STOP condition signals when the data transmission is done. The STOP condition is always asserted by the master and is signaled when SDA goes high as SCL is also high. This is the opposite that we observed for the START condition. With the protocol using the START and STOP conditions along with the handshake method of sending ACK bits to confirm whenever data is being requested or when data is actually being sent, it makes I2C a simple and powerful communication interface that is very common. This sums up the I2C data frames that are used for communication between integrated circuits and devices on the I2C bus.

Physically, the I2C bus consists of the SCL and SDA lines that are both pulled up to Vcc using pull-up resistors. This prevents the signal on the lines to not float, as

they are going to be pulled high to Vcc when the bus lines are unused. This is called active low, since the lines must be actively driven low. I2C also has limited speed due to the open-drain design, but there are many different standard I2C speeds that devices can support, depending on the manufacturer of the IC or sensor. The standard speeds are “Standard Mode”, “Fast Mode”, “Fast Mode +” and “High Speed Mode”, which have effective data rates of 100Kbps, 400Kbps, 1Mbps, and 3.4 Mbps respectively. Figure 14 below, show a visual of the standard configuration between two I2C devices on a typical I2C bus.

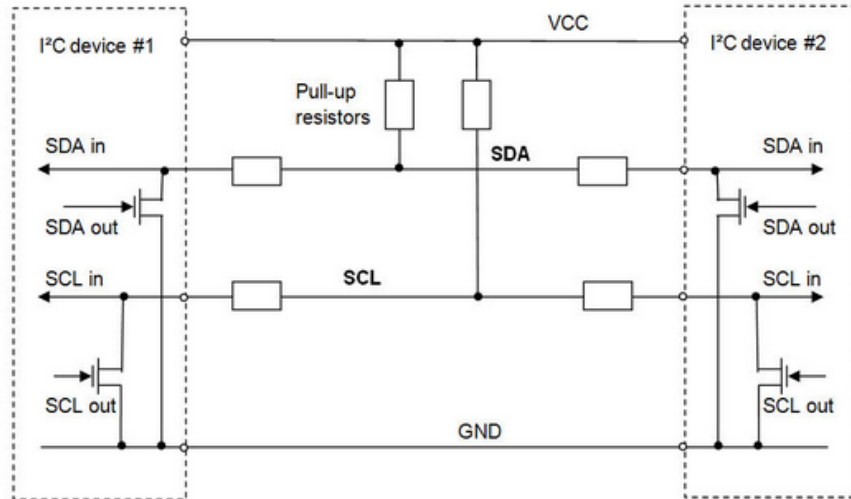


Figure 14: I2C Bus with 2 devices connected. SCL and SDA are connected to Vcc through pull-up resistors. Each device controls the bus lines outputs with open drain buffers.

### 3.3.6.3 Serial Peripheral Interface (SPI)

The Serial Peripheral Interface, most commonly known as SPI, is the last of the low-speed interfaces that we will highlight and compare in our paper. Just like UART and I2C that we detailed previously, SPI has its own advantages and disadvantages. SPI is a synchronous, serial, full-duplex protocol and is a very easy interface to implement in hardware. SPI is a single master protocol and is connected with 4 main wires between master and slave. The wires are SCLK, MOSI, MISO, and SS, standing for Serial Clock, Master-Out-Slave-In, Master-In-Slave-Out, and Slave Select respectively. For each slave that is added to this configuration there only needs to be another SS, slave select, line added to the interface. The clock signal, SCLK, is sent from the bus master to all slaves on the bus, therefore this is also a synchronous interface.

As we have mentioned previously, a serial communication protocol sends data in a stream of bits sent one after another, and in this case for SPI, from master to slave using MOSI line, or from slave to master using MISO line. The fourth wire is used as an active low slave select line, so that whenever the master would like to read or write from a specific slave device, it will pull that corresponding slave select

line low. For example, if there were 3 slaves on the SPI bus, then there would be a slave select for each of the 3 slaves, and the master would pull the corresponding slave select line low to determine which slave device it wants to communicate with. Figure 15 below shows two very common topologies for SPI, the top shows a single Master-Slave configuration, and the bottom shows a typical multiple slave configuration.

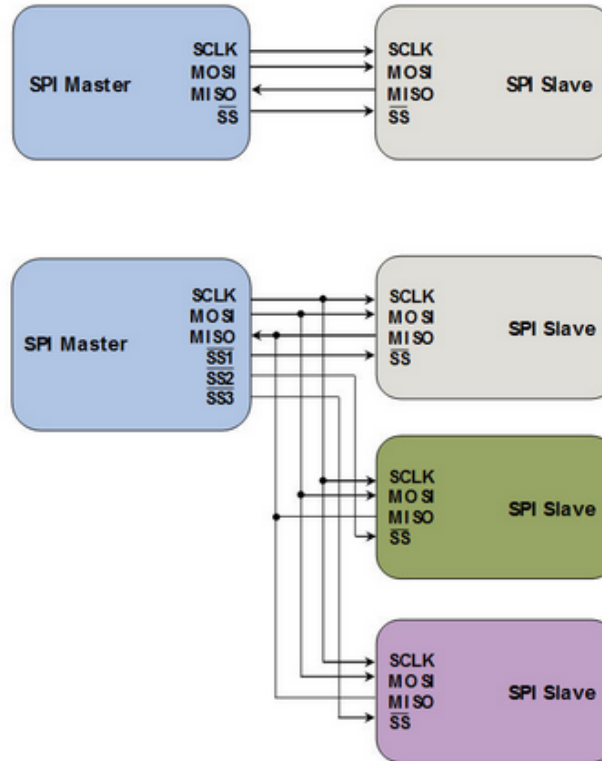


Figure 15: Example SPI configurations. The top shows a single Master-Slave interface and the bottom shows a common multi slave interface configuration. As you can see, there is a slave select line for each of the 3 slave devices.

The SPI protocol is simpler as there are no stop and start bits that need to be transmitted, nor are there issues with slave addressing like I2C. But SPI can be operated in four different communication modes that are available. The four modes essentially define which edge of the SCLK clock signal data is sampled and which edge data is toggled. The modes are defined with the parameters “CPOL” and “CPHA” which stand for clock polarity and clock phase, respectively. For example, mode 0 toggles data on the rising edge of SCLK and samples the data on the falling edge of SCLK, and when SCLK is not active (otherwise called the idle level) SCLK is low or 0.



The master device and each and every slave device must use the same set of parameters in order for the protocol to work correctly. If a slave device is not using the same parameters, the data will be corrupted and not transmitted correctly. SPI does not define any maximum data rate, nor any particular addressing scheme. The data rate that is reachable using the SPI interface is determined by the frequency of which SCLK runs in the master. Overall, SPI is easy to understand and easy to implement. It does not get as complex as I2C, and it can transmit data at a much faster rate than I2C and UART both.

### **3.3.7 Electrical Relay**

A relay is switch operated by electrical waveform. It includes a set of input terminals, and a set of operating contact terminals. The number of contacts is not an issue, a switch can have any number of contacts for one or multiple input control signals. They are usually used when a circuit is being controlled by an independent low-power signal. They were first used for applications as long-distance telegraph circuits as a repeater. Relays are a useful circuit component that helps refreshing the signal coming from one circuit by transmitting it to another.

Circuits that have a relay can easily control another circuit by opening the relay the connected circuit can be physically detached. In relay structures, there is two available options to choose from. The first option is the normally open structure, in this structure there will be no connection between the two circuits as long as the relay is energized. The other option is the normally closed where the two circuits are connected by a short, which will be broken once the relay is energized. The principle of both types is basically the same where the state of the connection will be changed based on whether or not voltage is applied.

In today's market, different types of relays exist to help engineers with their designs, the most common types of relays are electromechanical, reed, and solid-state relay. The simplest type of relays is electromagnetic relay. As the name suggests, this relay uses magnetic force to open or close the contacts.

The reed relay is basically a reed switch enclosed in a solenoid. The switch has a set of contacts inside an evacuated tube that helps protecting the contacts against environmental damage. These contacts are made of magnetic material, so they can be easily moved under the influence of the field of the enclosing solenoid or if an external magnet is close by. One of the reed relays benefit is there fast switching ability, and they also require minimal power for the control circuit. The disadvantage of the reed relay is having a very low switching current and voltage rating.

Last, a solid-state relay (SSR) is a solid-state electronic component which performs the same function as an electromechanical relay. An optocoupler is used internally to electrically isolate the input and output. One of the benefits of using an SSR is that there are no moving components, which increases reliability.

## 3.4 Components Research

In this section we are going to discuss the potential components that may be used in our project and the detailed research that was done in order to pick the right parts. The work was divided between the team members and each member conducted their own research.

### 3.4.1 Occupancy Sensor

Different technologies are used to detect the presence or absence of an object in a space. The two main types of motion sensors are active and passive sensors. The active sensors are a radar-based motion sensor. They radiate a radio wave or microwave across the area and wait for the signal to be detected upon reflecting after the wave hits an object. The sensor mechanism is basically to detect the frequency shift in the returning wave which indicates that the wave did hit a moving object. After that the sensor sends an electrical signal to the operating device such as a light or an alarm system. On the other hand, passive sensors, simply detect the infrared energy emitted by a living being. These sensors can be set to detect the motion of an object within some level of emitted heat to avoid triggering the alarm when an animal is moving within the detection area.

**Passive infrared (PIR)**, ultrasonic and many other sensors are all methods to spot a moving object within a specified range. Acoustic detection is also used in some sensors. A comparison between the different technologies was conducted in order to select the right sensor for our project needs.

**Microwave sensors** use Doppler radar to detect motion. The sensor generates and electromagnetic field between the transmitter and receiver creating a hidden volumetric detection zone. A continuous wave of microwave radiation and phase shift is emitted in the reflected microwave due to the motion of an object which results in a change in the signal at low audio frequency.

**Tomographic Motion sensors** detect the change in radio waves as they travel from node to another in a mesh network. This sensor emits radio waves at a frequency able to pass through walls, which makes it very valuable in application that needs to scan large areas since its able to sense the presence of a person behind walls or large objects.

**Passive infrared sensor (PIR)** works by sensing the difference between the heat emitted by a moving person with respect to the background heat. It's called passive

because the sensor doesn't emit energy. For PIR sensor, line of sight between the sensor and the object is essential which can be problematic for some applications. This sensor is suitable for applications in enclosed places, where low level of motion is needed.

While **PIR sensors** are good in detecting a general movement, they cannot provide any other information about the object. To gain more information active infrared sensors are used. Active infrared sensor works by using a dual beam transmission as structure. Where the transmitter is responsible for shooting an infrared Ray (light beam) to an in-line receiver, the receiver sees the IR beam and send a motion detection signal if the beam gets interrupted. Some active IR sensors uses a transmitter and receiver facing the same direction, and very close to each other which allow the receiver to detect a reflection of the object in the monitored area.

Video Camera software is another method used for motion detection purposes. This method is highly valuable when the application purpose is to record a video after being triggered by a movement in the monitored area. the output of the camera can be used to detect motion using software. Video camera software can be used with infrared illumination to detect movements in the dark.

Dual technology motion sensors, combining multiple technologies into one sensor can help reducing false triggers but it comes at the cost of increasing vulnerability. Motion is not detected unless both sensors are triggered simultaneously. This means that if an intruder is able to defeat one of the sensors, then the system will not alert, so it is potentially less secure, but it depends on application.

**Ultrasonic sensor** is a good example for the active sensors, it sends out an ultrasonic wave into a medium and measure the speed needed for the wave to return in order to detect the presence of a person. This sensor can cover the entire space and doesn't need a line of sight which allows it to detect people behind an object. Ultrasonic sensor is very sensitive to motions, which makes it suitable for applications in open spaces since the ultrasonic sensor doesn't require a line of sight.

Our project incorporates an ultrasonic distance sensor to wake the system up from sleep mode. The way that this type of sensor works is by sending out a frequency and seeing when it arrives back. The formula used to calculate the distance of an object is  $\text{distance} = (\text{time elapsed} * \text{velocity of sound}) / 2$ . In this section we will compare small, commercially available sensors to decide which is suitable for our project.

#### **HC-SR04**

The HC-SR04 is an ultrasonic sensor with range of 2cm-400cm with a resolution of roughly 0.3cm. It runs off of 5V at 15mA. When the sensor is triggered by

sending a pulse to the TRIG pin, the transmitter sends out a frequency, which bounces off of an object. The ECHO pin records when the frequency returns to the receiver. Figure 16 shows the size and shape of an HC-SR04 sensor that is placed into a solderless breadboard.

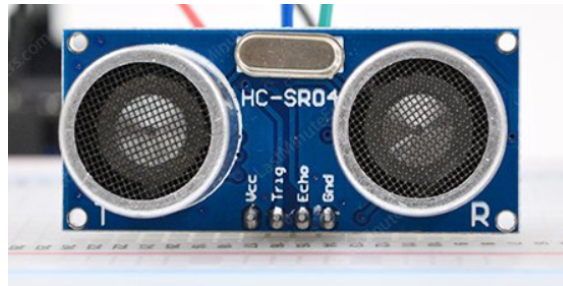


Figure 16: HC-SR04 Sensor on Breadboard Showing the Pins

### **Grove Ultrasonic Sensor**

The Grove Ultrasonic Sensor is similar to the HC-SR04 but has only 3 pins. It is 3.3V and 5V compatible and has a measurement range from 3cm-350cm with a resolution of 1cm. Like the other models, the Grove Ultrasonic sensor uses a 40kHz ultrasonic wave. The operating current drawn is 8mA, which is half that of the HC-SR04. Common applications are distance measurement, ultrasonic detection, and proximity alarms.

### **Parallax Ping**

The Parallax Ping))) runs off of 5V and has a measurement range from 3cm-300cm with a resolution of 0.5cm. The operating current is 35mA. The benefit of using this sensor rather than the others listed is that it computes the distance for you via a variable-width pulse. The microcontroller does not have to do any computation, it just receives a signal with the object distance. The Ping))) uses only 3 pins, so the signal pin both receives the signal and sends the distance. The issue with this sensor is that the onboard computing capabilities drive up the cost.

## **3.4.2 Microcontrollers**

A microcontroller (MCU) is a single metal oxide semiconductor integrated circuit chip. It contains one or more CPUs along with a memory and input/output ports all integrated in one integrated circuit chip. Microcontrollers are used in automatically controlled product devices, in all fields. MCU's are commonly used because of their reduced cost. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers are a better economical choice. Also, because microcontrollers are generally very low powered integrated circuits, they are great for mobile or Internet of Things (IoT) end devices that do not have the ability to be powered by your home's outlet power source. This is especially the case when the MCU might need to run on a battery

and last for many years. In this section we are to discuss different microcontrollers in order to determine which is the best choice for our design.

Microcontrollers are characterized by bus-width, memory structure and instruction set. Microcontrollers of the same family could have wide forms with different sources. Below we are to discuss some different types of microcontrollers.

### **3.4.2.1 Types of microcontrollers:**

There is a wide range of microcontroller topographies to work with, however this section is to discuss the most popular microcontrollers used in embedded systems.

- **STM32F103C8T6:**

This product is very famous among the STM32F103 family. It features a high-performance ARM® Cortex®-M3 that process 32-bit data and operate at 72 MHz frequency. The development board based on MCU is the Blue pill.

- **ATmega328:**

Is one of the most famous microcontrollers in the world. Atmega328p is the obvious choice any designer trying to avoid the bulkiness of the Arduino board, while still easy to program. It is an 8-bit AVR microcontroller based on an advanced RISC architecture and combines 32KB ISP flash memory with read-while-write capabilities.

- **PIC16F877A:**

The PIC16F877A is the most popular 8-bit microcontroller in the PIC family of MCUs which is still commonly used around the world. It is the first choice of microcontroller for beginners working with embedded development with PIC and it ends up as the microcontroller of choice for them when they become experts. Unfortunately, this board is not commonplace with higher end needs because of its limitations and stature.

- **Attiny85:**

This series of microcontrollers are considered as the most desired microcontroller for projects where a small form factor is desired, and the number of GPIOs required is low. Of all the microcontrollers in this series, the Attiny85 is regarded as the most popular, presumably because it seems to have more I/O pins compared to others. Attiny85 has a total of 8 pins. The CPU processes 8-bits of data within its architecture, with 8Kbytes of programable memory. It communicates through UART, I2C, as well as SPI. This microcontroller is ideal for applications with low power consumption.

- **ATMEGA32U4:**

This microcontroller is an option for application with low-power consumption. The microchip runs on an 8-bit AVR® RISC-based, featuring 32 KB self-programming

flash program memory, 2.5 KB SRAM, 1 KB EEPROM, USB 2.0 full-speed/low-speed device, 12-channel 10-bit A/D-converter, and JTAG interface for on-chip-debug. ATMEGA32U4 can execute powerful instructions in a single clock cycle and can achieve up to 16 MIPS throughput at 16 MHz, which give designers the ability to optimize power consumption versus processing speed.

• **STM8S103F3:**

STM8 family offers a high-performance 8-bit core and a state-of-the-art set of peripherals in a tiny form factor, similar to the Attiny series of MCUs. The family is made up of 4 series including the STM8S, the STM8L, the STM8AF, and the STM8AL. Among all, the STM8S series is considered the mainstream MCU and the STM8S103F3 is considered one of the most popular MCUs in the Series.

The 8-bit microcontroller offers 8 Kbyte Flash program memory, with an integrated true data EEPROM, advanced core and peripherals, a 16 MHz clock frequency, robust I/Os, independent watchdogs with separate clock source, and a clock security system, all of which ensures its high performance and overall system robustness.

Comparing all the above microcontrollers, our group decided to use MSP430 microcontroller from Texas Instruments since we are more familiar with it as we have used it in other classes. Below, we discuss more about the MSP430 microcontroller in detail.

• **MSP430:**

MSP430 chip is made by Texas instruments and is one of the most widespread mixed-signal microcontrollers. The MSP430 microchip has a very low power consumption level since it operates using less than 1mA of current. The reason why MSP430 can sufficiently work with low power consumption is its wake-up functionality, that allows the microcontroller to go to sleep mode when it is not being used. MSP430 can use full peripherals such as internal oscillator, timer, watchdog, USART, and I2C. It does not have an external memory bus, so it is limited to its on-chip memory.

The MSP430 is a 16-bit microcontroller that has a number of special features not commonly available with other microcontrollers such as:

- Complete system on-a-chip — includes LCD control, ADC, I/O ports, ROM, RAM, basic timer, watchdog timer, UART, etc.
- Extremely low power consumption — only 4.2 nW per instruction, typical
- High speed — 300 ns per instruction @ 3.3 MHz clock, in register and register addressing mode.
- RISC structure — 27 core instructions Orthogonal architecture (any instruction with any addressing mode)
- Seven addressing modes for the source operand

- Four addressing modes for the destination operand Constant generator for the most often used constants (-1, 0, 1, 2, 4, 8)
- Only one external crystal required — a frequency locked loop (FLL) oscillator derives all internal clocks.
- Full real-time capability — stable, nominal system clock frequency is available after only six clocks when the MSP430 is restored from low-power.

MSP430 is used in almost every embedded system such as industrial sensing and communications to control and monitor sensors within a digital communication system, it's also used in energy harvesting (renewable energy) to reduce average current consumption due to its low power needs. Figure below the block diagram shows the architecture of a basic MSP430. There are many types of MSP430 microcontrollers with varying pin counts and capabilities. For this project we decided to use the MSP430FR6989 for our system controller. Figure 17 shows functional block diagram of MSP430FR6989.

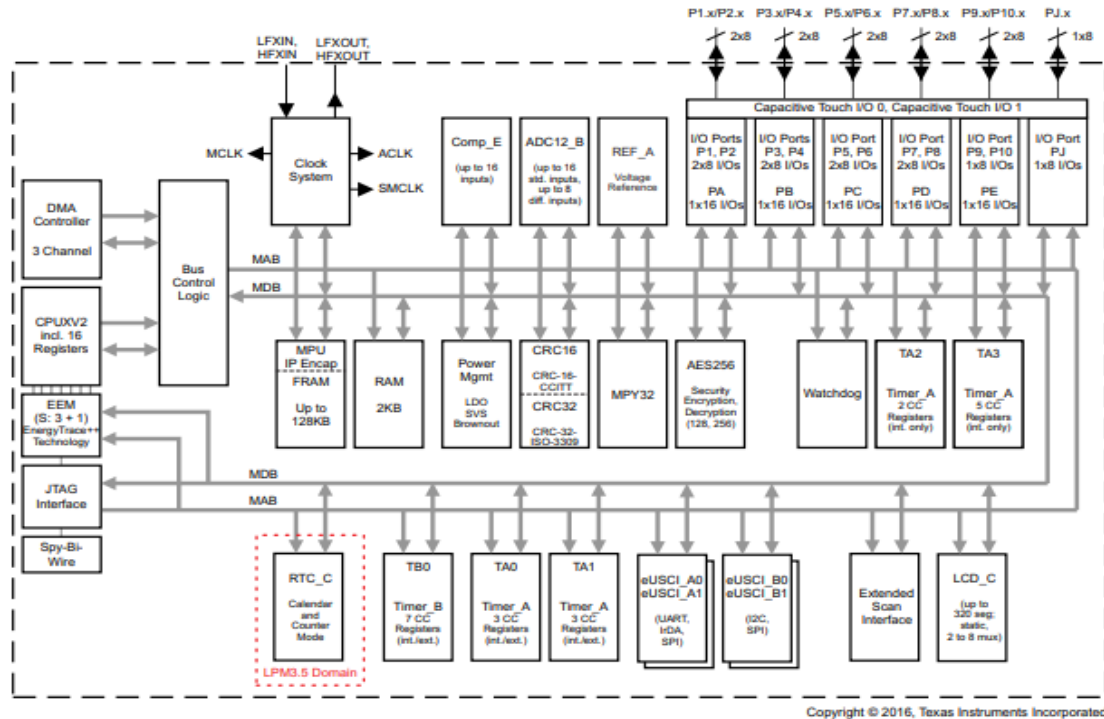


Figure 17: Functional Block Diagram

### 3.4.3 Other Integrated Circuits:

This section details the various other chips that we will need to use on our PCB design and the constraints associated with them.

### 3.4.3.1 H-Bridge:

The motor used in the original August lock is a Mitsumi M22E-13. The specifications for this motor are taken from the datasheet and are shown below. These motor specifications determine what type of H Bridge we will need to drive the motor. Our design requires bidirectional control and a small voltage, which narrows down the list of possible integrated circuits. Table 13 describes the DC motor specifications for the motor inside of the August lock, which we may be reusing if we decide to go the cannibalization route in choosing a lock.

| Item            | Specifications |
|-----------------|----------------|
| Rated Voltage   | 3.5 V          |
| Voltage Range   | 3.0 ~ 4.0 V    |
| Rated Load      | 9.3mN.m        |
| No Load Speed   | 6100rpm        |
| No Load Current | 120mA          |
| Starting Torque | 27mN.n or more |
| Rotation        | CW             |

Table 13: DC Motor Specifications

#### **DRV8833:**

The DRV8833 is our first choice, as this is the H bridge that was used to drive the motor in the August lock design. It is a current-controlled motor driver with low MOSFET On resistance. Its benefits include a wide power supply voltage range, which is important because the batteries in our lock system will have time-varying voltages as they drain. It is able to output 1.5A at 5V input. At room temperature and 3.5V operating voltage, it consumes only 1.5mA current.

#### **DRV8836:**

The DRV8836 is in the same family as the DRV8833 so it has similar characteristics as a dual-output motor driver. It runs on a power supply voltage from 2V to 7V. It also supplies up to 1.5A. A benefit to this H bridge is that it has a low power sleep mode and a sleep input pin. The package size is also very small at 2x3mm. It has overcurrent protection, short-circuit protection, undervoltage lockout, and overtemperature protection.

#### **TB6612:**

TB6612 is a driver IC for DC motors that has 4 modes and low output on resistance. The output current is 1.2A average with peaks up to 3.2A. This model has low output resistance, thermal shutdown, and low voltage detection. Another useful feature for our design is the standby system that is designed to save power. The problem with this H bridge is that the minimum supply voltage is 4.5V, which may not be suitable for our AA-battery system.



## **L9110:**

The L9110 is an H bridge with low quiescent current that can run off of a wide power supply range: 2.5V to 12V. This is suitable for our design because we are using batteries for our power supply. It is capable of supplying 0.8A continuous current and 1.5A peak current, which is on the low end of what we require. It has a built-in clamp diode to reverse the impact of the motor's inductive load.

### **3.4.4 Central Processing Unit (CPU):**

A CPU at the basic level is an electronic circuit capable of executing instructions. There are a few small single board computers about the size of a credit card that are capable of doing the processing for image and voice recognition and we will discuss them in this section. The objective for selecting the most appropriate CPU for our project comes down to affordability, power, connectivity, memory, and functionality.

For instance, we would not want our CPU to be extremely expensive, and too powerful for our application. This will cause the system to be expensive to replicate, as well as resources being wasted. However, we also do not want to purchase a CPU that is very cheap and not powerful enough to perform our required tasks. So, to find the most suitable CPU we will need to determine the middle ground.

In order to determine the CPU that will be most appropriate for our system we will have to take in account for our desired functionalities. For instance, our system will incorporate Facial recognition, Voice recognition, and will take in a pin entry from a user. Also, our CPU will have to be able to connect to a microcontroller that will be able to control a door lock. So, the CPU would also be required to receive data back from the microcontroller (MCU).

From a hardware perspective, we would need a device with multiple GPIO pins. This will allow us to connect different components to our CPU. Such as a keypad, LCD screen, wires for communication protocols (i2c, UART, or SPI), and other components. A device which supports at least 40 standard GPIO pins should be suitable, but we would not need a device with over 70 pins.

In addition, a device which has a simple but efficient operating system (OS) would be very beneficial. This will allow our developers to program directly on the device, as well as create local repositories to websites like GitHub. By doing so our team can view our programs code remotely and perform code reviews easier. Also, by uploading the application to an online repository (repo) we can essentially save our code. So, if there is a hardware malfunction, or if we accidentally destroy our CPU, our program will not be lost.

A CPU which supports at least four USB input would also be favorable, anything over four would be excessive. With four USB inputs we can insert a mouse, keyboard, and a web camera. This will leave use with a free USB slot for some other input if needed. Most modern web cameras come with a microphone attached to it, which is convenient because we can use this for our system will use both facial and voice recognition.

Furthermore, a device which has a Wi-Fi module, Bluetooth module, and Ethernet port would be desired. Having a device that can connect to the internet will allow for many opportunities. Such as creating an internet of things (IoT) application, researching directly on the device, uploading code to an online repo, downloading libraries directly from the internet and onto our device. Also, if we can connect our CPU to the internet then we can implement a data base, send notifications to a user (email or text), or even connect to another device which is connected to the same network.

Having a device with an ethernet port would also allow us to connect two devices together with an ethernet cord. This connection is both secure and simple. It is secure because it can reduce the risk of cyber security attacks, compared to if we connect two CPUs together via some sort of wireless connection. The connection is also simple because to link the two devices you just insert an ethernet cord into the ethernet port of both devices, and with some configurations, then the developer is able to transfer data between both CPUs.

Due to the complexity of our project, our device would also require a decent amount of processor power. Our CPU will be responsible for multi-threading, running machine learning algorithms in real time, and other complex functionalities. However, our application will not be extremely complex, so our CPU would not have to be highly advanced.

Having a CPU that is able to provide 5V from a GPIO pin, would also allow us to power additional devices off of. This would simplify the process of having to include external power into our system for simple components. Some devices that could be powered off of the pi would be, an LCD screen, LED lights, motors, sensors, or even a keypad. So, this would also be a necessary requirement for our CPU.

Based off of these requirements the most realistic choice for our system would be the Raspberry pi 4. The specs for this device is shown in the next section, but this device contains all of the required components that our system will need. It has a simple operating system, sufficient GPIO pins, an ethernet port, internet capabilities, and everything else that was mentioned previously.

Also, this device is extremely affordable, so purchasing another one to reduce the workload on a singular device is very realistic. Since this device is very popular, there are a lot of online resources that our developers can reference. This will be

very helpful when developing the application, as well as configuring certain external components.

In addition to the raspberry pi having a plethora of online resources. This companies' documentation for their product is very simple to read and reference from. Some of these documentations include the raspberry pi data sheet, schematic sheet, and the basic setup/ tutorial sheet that comes with the device. Other complex computers can have extremely long data sheets. This is due to the additional functions of the device. The Raspberry pi 4 can handle all of our systems needs, but it is not overly complex. For instance, the data sheet for the raspberry pi 4 is only 13 pages.

Within this document you can find many of the device specific information. Such as information pertaining to Hardware, Software, Mechanical Specs, Electrical Specs, Power Requirements. Also, inside of the data sheet you can find information relating to the GPIO interface, pin assignments, GPIO alternative functions, camera and display interfaces, USB, HDMI, and even temperature range and thermals. This information can prove to be helpful when setting up external devices, or when needing to determine power requirements. While developing the system the data sheet will be referenced frequently, so having a simple document directly from the device developer is extremely beneficial.

In the next section we also listed some possible CPUs that may work for our system. Based off of our research we noticed that each CPU have similar specs, in terms of USB ports, memory, and SD card support. However, each device varies with power and price, after researching, the most feasible CPU is still the raspberry pi 4.

#### 3.4.4.1 Raspberry Pi:

The Raspberry Pi is a small, low-cost, open-source single board computer. The current generation, the Raspberry Pi 4, uses a 1.5 GHz 64-bit quad-core ARM Cortex-A72. The Raspberry Pi 4 is available with 2GB, 4GB, and 8GB of RAM. This system uses the ARMv8-A instruction set. Access to peripherals includes 40 GPIO pins and 4 USB ports. This generation Raspberry Pi includes Bluetooth 5.0 connectivity as well as GPIO, UART, SPI, and I2C. Also, there is support for 2.4 and 5GHz 802.11 as well as BLE. Table 14 shows the core functionality of the Raspberry Pi 4 system on a chip that we are investigating for use in this project.

| <b>Raspberry Pi 4</b> |   |
|-----------------------|---|
| <b>Processor</b>      | Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz               |
| <b>Memory</b>         | Depends on the device purchased, the options are:<br>1GB, 2GB, 4GB, or 8GB LPDDR4 |

|                                     |  |
|-------------------------------------|--|
| <b>Connectivity</b>                 | 2.4GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE Gigabit Ethernet   |
| <b>GPIO</b>                         | Standard 40-pin GPIO headers   |
| <b>Video &amp; Sound Multimedia</b> | 2 micro-HDMI ports (up to 4Kp60 supported)<br>2 lane MIPI DSI display port.<br>2 lane MIPI CSI camera port<br>4 pole stereo audio and composite video port |
| <b>SD card support</b>              | Supports insertion of a micro-SD card  |
| <b>Input Power</b>                  | 5V DC via USB-C connector (minimum 3A1)<br>5V DC via GPIO header (minimum 3A1)<br>Power over Ethernet (PoE)–enabled (requires separate PoE HAT)            |
| <b>Environment</b>                  | Operating temperature 0–50°C   |
| <b>OS Support</b>                   | Debian Based (Raspbian OS)   |
| <b>USB</b>                          | 2 USB 3.0 ports<br>2 USB 2.0 ports   |
| <b>Price</b>                        | 1GB ROM: \$30<br>2GB ROM: \$35<br>4GB ROM: \$55<br>8GB ROM: \$75   |

Table 14: Raspberry Pi Specifications

Above we can see a spec table for the Raspberry Pi 4. There are many different versions of the Raspberry pi, however this version is the most suitable for our project. Using this CPU, we are able to add another raspberry pi, to reduce the workload of resource intensive applications. We can also connect the raspberry pi 4 to a microcontroller, by using one of the communication protocols mentioned above (UART, SPI, and I2C). In addition to this, since this is such a popular device many companies make external components that can be connected directly to this device. There are also many tutorials online for connecting components, and how to implement basic applications.

#### **3.4.4.2 Jetson Nano:**

The Jetson Nano is another single board computer that was considered for our processor. It uses a Quad-core ARM Cortex-A57 MPCore processor and has 4GB of LPDDR4 memory. The Jetson Nano also has 4 USB ports and GPIO, I2c, SPI, and UART connectivity. It is less appropriate for this project than the Raspberry Pi as the Jetson has a strong graphics architecture that makes it more expensive. However, the Jetson Nano is also more geared toward machine learning/AI. Table

15 details the main features and functionality of the Jetson Nano that we are using to determine if it is a good fit for our needs in this project.

| <b>Jetson Nano</b>                  |   |
|-------------------------------------|---|
| <b>Processor</b>                    | CPU: Quad-core ARM® A57<br>GPU: 128-core NVIDIA Maxwell™ architecture-based GPU |
| <b>Memory</b>                       | 4GB 64-bit LPDDR4; 25.6 gigabytes/second  |
| <b>Connectivity</b>                 | Gigabit Ethernet  |
| <b>GPIO</b>                         | Standard 40-pin GPIO headers  |
| <b>Video &amp; Sound Multimedia</b> | 4k @ 30fps [H.264/H.265]/ 4k @ 60fps [H.264/H.265] encode and decode            |
| <b>SD card support</b>              | Supports insertion of a micro-SD card   |
| <b>Input Power</b>                  | 5V DC via USB-C connector (minimum 3A1)<br>5V DC via GPIO header (minimum 3A1)  |
| <b>Environment</b>                  | Operating temperature 0–60°C  |
| <b>OS Support</b>                   | Linux for Tegra®  |
| <b>USB</b>                          | 4 USB 3.0 ports   |
| <b>Price</b>                        | \$99  |

Table 15: Jetson Nano Specifications

Above we can see a table displaying the specifications of the Jetson Nano. This version of the Jetson board was developed as a low-cost alternative to the other Jetson boards. Some of their other products can cost hundreds of dollars, so for this device coming in at \$99 is a reasonable price. However, it is still more expensive than the top version of the raspberry pi 4. Also, this device may be too powerful than what we need for our system. Thus, making this CPU not a feasible option for our application.

#### **3.4.4.3 Beaglebone Black:**

The Beaglebone black is a single board computer that runs off of the ARM Cortex A8 processor. It only has 512MB of DDR3 RAM, so it is less powerful in that regard than the Jetson Nano and Raspberry Pi 4. The Beaglebone has a Wi-Fi variant available that provides 802.11 connectivity. This board also has two PRU 32bit microcontrollers to support further IoT functionality. This board is also not our best option because it is not quite powerful enough to run facial and voice recognition.

Table 16, shown below, details the primary features and functionality that we are using to decide if the Beaglebone Black is suitable for our needs on this project.

| <b>Beaglebone Black</b>             |  |
|-------------------------------------|--|
| <b>Processor</b>                    | AM3358 ARM Cortex-A8   |
| <b>Memory</b>                       | 512MB DDR3 (800MHz x 16), 4GB on-board storage using eMMC                                |
| <b>Connectivity</b>                 | 2.4GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE Gigabit Ethernet |
| <b>GPIO</b>                         | Standard 2 x 46-pin GPIO headers   |
| <b>Video &amp; Sound Multimedia</b> | microHDMI, cape add-ons  |
| <b>SD card support</b>              | Supports insertion of a micro-SD card  |
| <b>Input Power</b>                  | 5V DC via USB-C connector (minimum 3A1)<br>5V DC via GPIO header (minimum 3A1)           |
| <b>Environment</b>                  | Operating temperature 0–50°C   |
| <b>OS Support</b>                   | Debian   |
| <b>USB</b>                          | 4 USB 2.0 ports  |
| <b>Price</b>                        | \$45   |

Table 16: Beaglebone Black Specifications

Above we can see a specification table for the Beaglebone Black CPU. There are many different versions of the Beaglebone. The Beaglebone Black is a step above the cheapest Beaglebone (PocketBeagle), and a step below the (BeagleBone Blue). The most expensive, and more intricate Beagle Board device is the BeagleBone AI, which comes in at a price of \$99. The BeagleBone AI is a comparable price to the Jetson Nano and is still more expensive than the Raspberry Pi 4. This board comes in at an affordable price of forty-five US dollars. As mentioned, although this the CPU is affordable, it is not very powerful, at least compared to other CPUs like the Raspberry pi. This CPU would be more suitable for simple tasks. Since our system will be dealing with complex algorithms, and multithreading, we would need a more powerful CPU that would be able to process information in real time.

#### **3.4.4.4 Arduino Uno**

The Arduino Uno uses the Atmel 8-bit ATmega328P processor. This device was another possibility for our system. The flash memory for this device is 32KB, and

0.5Kb is utilized by the bootloader. SRAM and EEPROM is 2KB and 1KB, respectively. This compared to the other devices listed is very low. It does support 16 standard GPIO pins, which 6 of them provide pulse width modulation (PWM) output. This device is not a good option for our application, because it will not be able to handle the different required functions. Table 17 shows the functionality of the Arduino Uno that we are using to compare to the other systems for this project.

| <b>Arduino Uno</b>                      |  |
|---|--|
| <b>Processor</b>                        | Atmel 8-bit ATmega328P   |
| <b>Memory</b>                           | Flash Memory: 32KB (0.5KB is used by the bootloader)<br>SRAM: 2KB<br>EEPROM: 1KB |
| <b>Connectivity</b>                     | Does not come with built in Wi-Fi, but can be added with external components     |
| <b>GPIO</b>                             | Standard 16 GPIO headers   |
| <b>Video &amp; Sound<br/>Multimedia</b> | N/A  |
| <b>SD card support</b>                  | Does not support SD card input.  |
| <b>Input Power</b>                      | 7-12V  |
| <b>Environment</b>                      | Operating temperature 0–50°C   |
| <b>OS Support</b>                       | N/A  |
| <b>USB</b>                              | 1 input for USB 2.0 type A/B   |
| <b>Price</b>                            | \$22.46  |

Table 17: Arduino Uno Specifications

From the table above we can see that at a price of \$22.46 you can get a somewhat powerful device. This device is more of a microcontroller rather than a microprocessor like the raspberry pi4. Therefore, this device would not be able to support the complex functions that we would need. Such as facial recognition, voice recognition, multi-threading, connecting to external MCUs or CPUs, or even running complex algorithms in real time. The Arduino Uno would be better suited for a simpler system, and our application would require a device that can support complex operations.

### 3.4.5 LED Lights

A Light Emitting Diode or LED is a semiconductor device that emits light due to electroluminescence effect. LED is basically a PN Junction Diode, which emits light when forward biased. LEDs are commonly used in a wide range of applications, it can be found in cars, streetlights, home, and office Lighting, televisions and many more. The reason for the wide range of implementation of LEDs is its advantages over traditional incandescent bulbs and the recent compact fluorescent lamps (CFL). LEDs are low power consumption, smaller in size, long lasting, and can be produced to emit all the visible spectrum wavelengths. This makes LED lighting our ultimate choice to fulfil our project design needs.

### **Types of LEDs:**

LEDs are available in variant options, they come in different shapes and sizes. Some of the commonly used types are discussed below.

#### **Through-hole LEDs**

Through-hole LEDs are available in different sizes and shapes. The most common ones are 3mm, 5mm and 8mm LEDs. They come in different colors like Green, Blue, White, Red, Yellow, etc.

#### **SMD LEDs (Surface Mount Light Emitting Diodes)**

SMD or Surface Mount LEDs are a special package that can be easily surface mount on a PCB. They can be differentiated based on their physical dimensions. For example, the most common SMD LEDs are 3528 and 5050.

#### **Bi-color LEDs**

The next type of LEDs is Bi-color LEDs, they can emit two colors. Bi-color LEDs have three leads, that is usually two anodes and a common cathode. Colors to be activated based on the configuration of the leads.

#### **RGB LED (Red – Blue – Green LED)**

RGB LEDs are the most popular LEDs among designers. They are very popular for implementing RGB LEDs in Computer Cases, Motherboards, RAMs, etc.

#### **High – Power LEDs**

A High-Power LED is a LED with power rating greater than or equal to 1 Watt. That's because the regular LEDs have a power dissipation of few mill watts. High – Power LEDs are very bright and are often used in Flashlights, Automobile Headlamps, Spotlights, etc.

Since the power dissipation of High – Power LEDs is high, proper cooling and usage of heat sinks is required. Also, the input power requirement for these LEDs will be usually very high.

#### **LED Strip lights**



LED strip lights are becoming more popular in various applications. They are more efficient, brighter, long lasting, and don't produce much heat. They can be easily controlled and programmed. LED strip come in extremely compact packages that are durable and resistant to shock, which make them usable in a variety of applications.

LED Strips comes in a wide range and include different types of LED strip lights. Depending on the application, one type may be picked over the other. For example, LED flex strips are better choice for outdoor applications, because they have a protective covering that make them waterproof. There are different types of LED strip lights available, below we are to discuss each type.

- **AC LED Flex Strips**

AC LED Flex Strips dimensions are a 0.5-inch for width and 0.25-inch for length, which make it wider than DC LED Flex Strips. They are perfect for applications that are in distant since multiple power supplies or dimmers are no needed. They can be directly attached to a 110-120 AC line voltage.

- **DC LED Flex Strips**

DC LED Flex Strip Light is commonly used for DIY projects, because of their simplicity to work with. They come in small size comparing to the AC LED flex strips, with a width of 10mm and a length of 3mm, and also offered in waterproof option.

DC LED Flex Strips generate minimal heat, which make them a good choice for applications that are in small areas. It requires an input of 12VDC, which mean the lights need a transformer that provides 12VDC.

- **LED Rope Light**

The LED Rope light is encased in a round rubbery plastic providing a rope resemblance. It has a waterproof case and can be bent in different directions. LED Rope light is compatible with the standard wall outlet that runs from 120VAC. The only downside of the LED Rope Light Strips is its brightness level, it would be a choice for application that require a dim lighting.

- **High Output LED Strips**

The High Output LED Strip Light is perfect for application that needs high brightness. It is ideal to use a 24VDC power supply on High Output LED Strips, with the ability to set up a power and dimming control on the strip. The control is a low-profile board-to-board connector connected at the end of the strip, which help focusing the lightening on the desired area.

When it comes to connecting an LED to a circuit, there are a few characteristics of LED that are important. These specifications are usually written in the datasheet of the component and can be referred to for more information about physical dimensions, absolute maximum ratings, and electrical characteristics such as the polarity, forward voltage and current.

The polarity is an indication of symmetry of an electronic component. In an LED, the positive terminal is called “Anode” and the negative terminal is called “Cathode”. The current in LED flows from Anode to Cathode, which mean that the Anode terminal should be at a higher potential than the Cathode to ensure that the LED is working properly.

The amount of current flowing through an LED is very important, since it determine the brightness of an LED. Allowing current more than the rated current to pass through an LED will burn it. For that reason, every LED is rated with a maximum forward current that can safely pass through it. Most used 5mm LEDs have a current rating of (20 to 30) mA and the 8mm LEDs have a current rating of 150mA (refer to the datasheet for exact values). In order to control the current flowing through an LED, we make use of current limiting series resistors.

LEDs are also rated for forward voltage, which is the amount of voltage required for the LED to conduct electricity. For example, all 5mm LEDs have a current rating of 20mA but the forward voltage varies one LED to another.

### **3.4.6 Accelerometers**

There are multiple areas in this project where an accelerometer may be useful in determining if components have moved, been tampered with, or what state they are in. In this section we discuss the basic premise behind accelerometers as well as the different types that are available.

Accelerometers measure proper acceleration, which is the acceleration relative to gravity/freefall, meaning that the sensor would feel changes in acceleration not caused by gravity. These mechanical sensors are built by placing a small mass on a spring along with the required circuitry to measure changes in the motion of the mass. Accelerometers are usually attached to the component whose acceleration is being measured, so printed circuit boards containing them are often built with mounting holes.

Compression mode accelerometers are composed of a sensing crystal which is bonded to a mass and emits a charge when it is under compression. Shear mode accelerometers are attached between a center post and seismic mass. These shear mode sensors measure the proportional electric output to stress applied to the crystals. Shear mode accelerometers are better at rejecting thermal transient and base banding affects and they have a higher frequency response. Capacitive

accelerometers measure changes in capacitance that are produced by a crystal undergoing movement. Detection circuits capture the peak voltage.

Both digital and analog accelerometers exist. Analog accelerometers produce a continuous voltage that is proportional to acceleration. Generally, the bandwidth of an accelerometer can be selected by picking the output capacitors. Digital versions use pulse-width modulation (PWM). Pulse-width modulation works by adjusting the duty cycle, or the percentage of time the signal is high versus low. The average voltage is computed by taking the duty cycle times the high voltage level. For example, a 50% duty cycle is a square wave, and a 100% duty cycle is a DC voltage. In the case of accelerometers, the duty cycle of the signal is proportional to the acceleration.

An important specification in picking an accelerometer is the number of axes. Modern sensors are mostly 3-axis, but 2-axis variants still exist. The maximum swing is also important to pay attention to, and for projects with low expected acceleration, you would want to select an accelerometer that measures  $\pm 2g$  to  $\pm 5g$ . The unit that acceleration is measured in is g, which is equal to the acceleration of Earth's gravity,  $9.81\text{m/s}^2$ . A higher sensitivity is better, but that can increase the package size, cost, and computing power required, so the sensitivity is a specification that should be balanced.

### **ADXL337**

The ADXL337 is a small, low-power accelerometer with a 3mm x 3mm x 1.45mm package size. Normal operation with this accelerometer pulls 0.3mA, and this accelerometer has great temperature stability. Single supply operation uses 1.8 to 3.6V. It is 10,000g shock survivable. This accelerometer is a complete 3-axis sensor and can measure acceleration with a minimum full-scale range of  $\pm 3g$ . Bandwidth can be selected to suit the application by adjusting capacitors.

### **ADXL377**

The ADXL377 is from the same product line as the ADXL337 and it has similar features and specs. It is also highly orthogonal with little cross-axis sensitivity. This sensor has the same package size as the ADXL337. The main difference is that it has a much larger range and is capable of measuring accelerations with a minimum full-scale range of  $\pm 200g$ . Its primary application is high-force event detection.

### **MMA8450Q**

The MMA8450Q is a low-power, 12-bit, 3-axis accelerometer. The package size is 3mm x 3mm x 1mm. In normal operating mode, the current can be as high as 250 microamps, but in low power mode the current draw stays between 27 and 150 microamps. There is a shutdown mode with current draw less than 1 microamp. The dynamic range has 2g/4g/8g options. This model only has I2C digital connectivity. This accelerometer can be set to a wide range of sampling

schemes from 1.5 to 400 samples per second, with operating current proportional to sampling rate.

### **BMA400**

The Bosch BMA400 is a tiny, ultra-low power accelerometer that is commonly used in wearables. It has full-scale acceleration ranges from  $\pm 2g$  to  $\pm 16g$ . The voltage range is from 1.2V to 3.6V, which is wider than that of the ADXL line. Also, the package size is smaller, as the BMA400 is only 2mm x 2mm x 0.95mm. It has orientation detection as well as auto low-power and auto-wakeup. The digital interface includes 3 and 4-wire SPI and I2C along with two interrupt pins. Typical current draw is less than 14.5 microamps even with highest performance.

## **3.4.7 Electronic Locks**

Many different configurations of electronic door locks exist; in this section we will discuss the ways to electronically control the ability of a door to physically lock and unlock. There are fail safe and fail secure variants. Fail safe locks disengage when power is removed, while fail secure locks remain locked until power is reapplied. The hold open feature refers to the ability of the lock to unlock and remain that way until it is used, when it locks itself.

### **Electromagnetic Locks**

Magnetic locks consist of an electromagnet and an armature plate. They provide positive instantaneous release and are one of the most robust types of electric locks as there are no moving parts to wear out. Electromagnetic locks are generally fail safe. The primary disadvantage of this type of lock is that they require a constant power source to remain engaged. In the event of a power outage, they will not be secure. Also, magnetic locks do not immediately release as there is a delay caused by the breakdown of the magnetic field. Commercially available electromagnetic locks range from holding force of 600 to 1200lbs.

### **Electronic Strike Locks**

Electronic strike locks use currents to engage or disengage a latch bar. The standard fixed mechanical strike is replaced by an electrically controlled one. This type of lock is available in both fail safe and fail secure versions. The disadvantage is that the lock is visible to the outside and susceptible to tampering. Also, electronic strike locks must be properly matched to the door or they will not operate correctly.

### **Internal 'Smart' Locks**

Smart locks have no visible component on the outside of the door. They attach to a traditional deadbolt with a custom baseplate on the inside of the door and are normally battery operated. The utility of a standard mechanical deadbolt is

retained as a spare key can be used in case of an emergency. Smart locks work with various authentication methods. Smart locks lock and unlock by sliding the existing deadbolt. We made the decision to reuse the electromechanical and mechanical components of a smart lock because these devices do not require additional drilling or running wires through the door.

### **3.4.8 Webcam**

Our project uses a webcam for the facial recognition feature. In this section we discuss budget webcams and their specifications so that we are able to make an informed decision regarding which to use.

#### **Wansview 101JD**

The Wansview is a 1080p CMOS image sensor. This webcam automatically adjusts to low-light conditions by adjusting the white balance of the picture. This feature is done automatically without the need for a driver, which avoids a potential issue in interfacing with the Raspberry Pi. It supports H.264 and H.265 image compression standards, which provide good image quality at lower bit rates than other standards. This camera can reach 30 frames per second, which makes for a better viewing quality.

Also, this webcam has a built-in microphone, so using it would prevent us having to buy a discrete microphone for the image and voice recognition aspect of our project. A potential issue with using this webcam is that the focus has to be set manually, but we should be able to calibrate it once to the distance that the facial recognition would be taking place and leave it set. This camera weighs only 4.6 ounces and has a 90-degree viewing angle. The 101JD will work when the humidity is between 10% and 90%, which should be sufficient for outdoor use.

#### **Aukey FHD PC-LM1E Webcam**

The Aukey PC-LM1E webcam is similar to the 101JD in that it is a 1080p, 2MP camera. It is a 1 / 2.9", 30 frames per second image sensor. It has a built-in noise canceling microphone, which again would eliminate our need for a discrete one. The microphone is actual a dual microphone which is how it eliminates noise. A benefit to using this webcam is that it weighs only 2 ounces, so it may be easier to keep mounted than the slightly heavier 101JD.

#### **Logitech C922x Pro Stream Webcam**

The C922x is a very good 1080p webcam. One of the major advantages of this webcam is the autofocus. It can deliver 1080p at 30 frames per second or 720p at 60 frames per second, which makes it good for detecting movement. This camera also has two microphones, which helps reduce noise in the audio. This camera has a 78-degree field of view and is optimized for streaming and recording.

## **Logitech BRIO Ultra HD Webcam**

The Logitech BRIO is a 1MP webcam that has 4K capabilities. It has a high dynamic range and autofocus capabilities. Also, it has the ability to digitally zoom up to 5x. This camera uses RightLight 3 and high dynamic range to correct lighting automatically in the video feed. The BRIO has three different field of view options: 65, 78, or 90-degrees. Also, like most of the other webcams listed, it has dual microphones.

### **3.4.9 Speaker/Intercom**

Our team is exploring the use of some sort of a speaker/intercom system implementation to communicate back and forth through the door. This is so that the homeowner can interact with and potentially grant access after talking to the person at the door. In this section we discuss the various commercially available intercom systems and their merits. There are some nice benefits to incorporating some kind of intercom system, but it might not fit the scope of our project, and it might just add unnecessary complexity that we may be better off not dealing with. If we determine that we are going to use this in our system, we can further explain the implementation details and how this would be incorporated in the system. We can include this portion of the system in our dialogue of possible stretch goals for the Senior Design 2 semester.

After careful consideration, our group has decided to not incorporate an intercom system in our design. We believe our current design without the intercom is sufficient and trying to incorporate this would add unnecessary complexity and effort. The design will continue as previously discussed without the use of an intercom system. In the future, we may reincorporate the intercom as a stretch goal.

### **3.3.10 Motor Driver**

Our project includes a motor and gears to manipulate the deadbolt. While we will be reusing these mechanical components from the lock that we cannibalize, we will need to control the motor from a microcontroller by using an H-Bridge.

An H-Bridge is a type of circuit that uses four switches to control the flow of current to a motor by switching the polarity of voltage to the load. Diodes can be included next to the switches to prevent damage from the inductive spikes due to back-emf. Resistors are normally used by attaching to the base (or gate) or the transistors to limit the base (or gate current). Adjusting the value of these resistors adjusts the current that drives the motor. H-Bridges can be built with discrete components, but integrated circuit versions are available from a variety of chip manufacturers. The Figure 18 below shows a standard schematic for an H-Bridge.

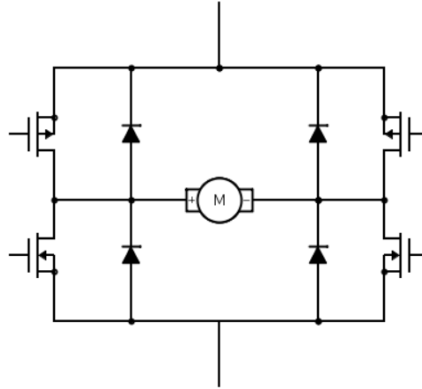


Figure 18: Basic H-Bridge Schematic

### 3.5 Parts Selection

After detailed investigation, all of the parts included below are the components that we will incorporate in our project to give us the best and most successful design. All the parts were carefully selected based on performance, capability, power consumption and cost. This section is to conclude our parts research and explain the reason behind our team's decision regarding the specific parts and components to be used.

#### Microcontroller(s):

After reviewing multiple options (refer to section 3.4.2). Our team agreed that the MSP430 family by Texas Instruments is the best microcontroller for our project's design needs. MSP430FR6989 fits our needs because it is an ultra-low-power microcontroller and can help to reduce the standby current. It has plenty of online support, and also cheaper than other high-end controllers. All of the group members are very familiar with this microcontroller and have had experience working with it in our previous coursework. This will allow us easier implementation for our custom application, instead of requiring us to familiarize and learn about another brand or type of microcontroller.

#### Single Board Computer(s):

The Raspberry Pi SBCs are chosen due to their high degree of connectivity from the USB ports and GPIO pines. Our system uses two Raspberry Pis: one for facial recognition and one for voice recognition. Separating these two computationally intensive tasks increases the speed of our system and reduces likelihood of error due to state mismatch when communicating to the MSP430FR6989. The Raspberry Pis will also communicate with and power the keypad and, if we are able to include it, the LCD screen. The Raspberry Pis have been chosen to interface with the Keypad and LCD screen because there are already libraries in place and we are avoiding reinventing the wheel as much as possible. In the end,

the primary use of the Raspberry Pis is to accept and process the sensor data to determine if a user is authenticated or not. After the user is authenticated, the Raspberry Pi will communicate with the microcontroller and alert it with which method was authenticated. At that point, the microcontroller can make the decision to unlock the door if two factors have been identified as authenticated. Physically, the Raspberry Pi SBCs will be in a 3D-printed enclosure with the system controller PCB. A slit will be in this case for cooling and to allow the ultrasonic sensor on the system controller to check the range of the user inside.

**Occupancy sensor:**

Discussing all types of occupancy sensors, our group decided to use the Ultrasonic sensor for multiple reasons. The sensor is easy to use and can easily interface with our microcontroller of choice. It is highly immune to weather factors like dust and high moisture environment. It can be used in dark environment, and it is not affected by color or transparency of the object. Ultrasonic sensor is a low-cost option for our design, and we have had some experience working with it from previous class.

The ultrasonic sensor that we chose is the HC-SR04. Our primary motivation for using this particular sensor is familiarity, as we used the same one in junior design and we already know how to implement it on Eagle and have it interface with the MSP430 series microcontrollers. This component has four pins, and the wire connections direct as follows:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

The specifications of the HC-SR04 Ultrasonic Sensor that drove our decision to choose it over the alternatives are shown in Table 18.

|                      |  |
|----------------------|--|
| Dimensions           | 45*20*15mm   |
| Working voltage      | DC 5 V   |
| Working current      | 15mA   |
| Working Frequency    | 40Hz   |
| Max/ Min range       | 4m – 2cm   |
| Trigger Input Signal | 10uS TTL pulse                                     |
| Echo Output Signal   | Input TTL lever signal and the range in proportion |

Table 18: Ultrasonic Sensor Specification



## LED Lights:

Upon researching different types of LEDs lights, our team decided to use the LED strip lights. This type fits our design needs and have various other advantages. They are more efficient, brighter, longer lasting, they do not produce much heat and due to the linear design, the heat is better dissipated than with other architectures. Additionally, they are easily controlled and programmed, so we should be able to interface it with one of the Raspberry Pis or our MSP430 without great difficulty. LEDs come in extremely compact packages that are durable and resistant to shock, which means that we will be able to mount it to the door frame without it sticking out too far. Table 18 shows the specifications for the LED Strip light used for this project.

|                 |         |
|-----------------|---------|
| Working voltage | DC 12 V |
| Working current | 0.5 A   |
| length          | 16.4 ft |

Table 19: Specification of LED Strip Light

## Bluetooth module:

Our team was originally considering using a Bluetooth module to communicate between our system controller (MSP430FR6989) with the August lock. After conducting some research, it was the team's decision to use an electric strike lock instead of the August lock and therefore a Bluetooth module will not be required on the system controller PCB.

## Physical Door Lock:

For our project, we elected to use an Electric Strike Lock. We chose this option over cannibalizing a smart lock for several reasons. First, it reduces project complexity and increase our likelihood of succeeding. Second, the electric strike locks last longer and have a higher holding force than smart locks, and they are less affected by obsolescence. The final reason that we chose an electric strike lock is that it will be more convenient for the homeowner, as the lock will be plugged in to 12V power instead of relying on batteries, so the system will have a higher uptime.

We decided on the UHPPOTE ANSI Standard Heavy Duty Electric Door Strike Lock. It is relatively inexpensive and is tested for 500,000 cycles. At 2200lbs of holding force, this model lock is stronger than a standard deadbolt. The other motivation for choosing this particular lock is that it runs off of 12V power and only requires 220mA.

This lock is fail-secure, but can be changed as needed to fail-safe. Our PCB is wired such that the relay has NO and NC female pin headers corresponding to fail-safe and fail-secure, so that by changing the software and moving one wire and a screw on the lock itself, the lock can be reconfigured. This enhances our flexibility and opens our project up to more applications.

### **Camera Module:**

\*Most likely going to be the webcam that Moissess and Zach both have. This is a standard USB Webcam that has an integrated microphone, so using this will possibly simplify the process and system as we won't need a dedicated microphone (Although we need to test this with our voice recognition implementation to ensure that this built-in microphone is going to be sufficient enough to detect different user's voice's\*)

### **Microphone Module:**

The Microphone that we decided to use for this project is the MAONO 192Khz/24Bit Podcast PC Computer Condenser Mic. We picked this microphone because it has an integrated gain knob and it is relatively inexpensive. This microphone also has a 192kHz sampling rate, which should gather enough data to operate the voice recognition without issues. The gain knob is important because it allows us to tune the microphone to the particular doorway during installation. We will likely find a default value for gain to start with and then work for there. We can also test how the different gain settings affect the voice recognition accuracy. This microphone has a folding tripod, so it should be easy to mount to the door/doorway. The final reason that we chose this microphone model is that it is plug and play, with no need for a driver on many systems.

### **10 Digit Keypad/Pin Entry Pad:**

The keypad that will be used to take user input for a pin number, either one of the three authentication modes or the Master Pin Number, is the Parallax Inc. 4x4 Matrix Membrane Keypad (Part Number #27899). This simple to use 16 button keypad makes user input of digits easy to use with any microcontroller or computer, especially the Raspberry Pi 4. There is also an adhesive backing that can be used to mount the keypad as needed and if we were to choose to mount the keypad on the wall or system's enclosure.

Key specifications of the 10-digit membrane keypad that we found suitable for our project are listed in the below Table 20.

| Key Specification     | Specification Value       |
|-----------------------|---------------------------|
| Maximum Voltage       | 24V DC                    |
| Maximum Current       | 30 mA                     |
| Operating Temperature | 0°C - 50°C (32°F - 122°F) |

Table 20: Parallax Inc. 4x4 Matrix Membrane Keypad Specifications

## **4. Standards & Design Constraints**

Standards is a set of guidelines to be followed to ensure the high quality of the system. Following standards give the customers a feel of security that the system is safe to use. Being able to measure system features, allows us to discover our limitations and therefore better assist the customers in providing them with some guidance about handling the system and protecting themselves.

### **4.1 Standards**

In regard of electronics devices, standards are often defined by a team of professionals and scientists by conducting research and performing a set of tests to come up with a set of guidelines that limits the applications of a device to ensure customers safety. Since the success of our project is bounded by these standards, those are to be discussed in this section.

#### **4.1.1 Power Standards**

Maintaining power safety standards is essential to prevent electrical shock, injury, or fire.

International Electro technical Commission (IEC) and the Associated International Organization for Standardization (ISO) are two main agencies which deals with electrical safety standards.

Products that meet this standard display the safety mark as identified from the applicable standards organization. This indicates compliance within a specific economic region. The governing agency for the United States and Canada that gives the seal is the CSA mark.

For example, if IEC standard such as 62368-1 standard are met by a product, the product will be identified with the standard number and the economic region for which the standard is maintained. IEC 62368-1 is applicable to the safety and electrical equipment technology with a rated voltage does not exceed 600V. As an example, EN 62368-1, means that the product follows the specified standard in the European Norm.

#### **4.1.2 PCB Standards**

In 1957 a group came together and agreed about some standards for PCB design. IPC, is the institution of printed circuits boards who put standards for the manufacture and assembly requirements to ensure high quality and safety of

PCBs. All commercial made PCBs are required to meet the IPC standard for circuit boards known IPC-2221. This standard has three different classes.

**Class One (General Electronic Products):**

Most commercial made products fall under this class. Class 1 products are mainly concerned with functionality rather than reliability. An example of this class is flashlights, or personal computers.

**Class Two (Dedicated Service Electronics products):**

This class products considers functionality, reliability and durability. This class electronics must maintain prolong functionality, but failures are acceptable sometimes. If environment must not impose a failure on the functionality of the product, then its considered a class 2. Example on this class is home appliances.

**Class Three (High-Performance Electronic Product):**

This class is for high performance electronic products. Class 3 must work exceptionally well under any conditions and equipment downtime are not acceptable. These products have the strictest requirements because their failure would cause human casualty. Class 3 products are generally found in military and medical applications.

IPC helps support industry standards in 5 ways:

**Maintaining High End product:** When IPC standards are implemented correctly, it ensures that a product is to meet the quality generally accepted standards every time.

**Maintaining Consistency:** Maintaining consistency is essential to ensure product quality will be maintained regardless of the manufacturing company.

**Better Communication:** The internationally accepted IPC standard is for all product manuals and technical publication language, to provide customer support regardless of the manufacture country.

**Maintains the Reputation:** The IPC standards are a trusted standard that companies aim to accomplish, as it is a well-recognized standard in the electronics community.

**Cost Reduction:** If IPC standards are implemented correctly, the cost of quality inspection will decrease. Which results in the product cost to be reduced.

### 4.1.3 Communication Standards

Communication Standards measures the ability of two or more devices to communicate properly by successfully sending/receiving information through a physical quantity. IEEE 802 is a set of network standards which covers both the physical and the datalink layer specifications. The IEEE 802 standards help ensuring that the internet services is following some recommended measure so the devices connected on that network can communicate smoothly. It is the root of both IEEE 802.11 and IEEE 802.15.1 standards.

IEEE802.11 contain various standards that covers everything required for networking. they address common topics to all Wi-Fi systems. Security, quality of service, authentication, and they are all important to build a strong environment for the development and use of Wi-Fi technology. Since our system uses Wi-Fi technology to send notifications to the owner, these standards are relevant to our project.

#### **4.1.4 Electrical Locking Devices Standards**

ANSI/BHMA A156.25-2018 covers a large number of devices that falls under the category of electrified locking devices. The addressed guidelines in ANSI/BHMA A156.25-2018 are placed into that document to provide assistant with the safety and proper composition of electrified locking devices.

Electrified locking systems consist of four functions: locking devices, input devices, controlling devices, and power supplies. ANSI/BHMA A156.25 Standard initiate the requirements for locking devices that are described in the applicable BHMA® Product Standards. Also, if the input or controlling device are an integral part of the locking device, they shall also be tested with the locking device covered by this Standard. This Standard also includes testing for device qualifications such as security, strength, and environmental tests. The tests listed under this standard are required to be performed in a lab condition. Note that the actual test results may differ due to environmental conditions.

The electrical requirement of the lock follows a couple of UL-1034 test standards, which specialize of Burglary/Resistant Electrical Locks Mechanism. For a device to pass the tests, it must have a minimum of 500 pound-force of static strength, 33 ft/lbs of dynamic strength, and can withstand 100,000 cycles. All these requirements are satisfied by the electric strike lock since it has a static strength of 1,500 lbs, dynamic strength of 70 ft-lbs, and an endurance of 500,000 cycles.

#### **4.1.5 Insulation Standards**

To avoid having contact between live wires that have high voltages and other components on the circuit insulation standards need to be followed. There are five types of insulation standards with a variant usage depending on the

classifications of the component. For a higher classification of an equipment, a higher order of insulation needs to be followed. Insulation types from least to the highest degree of protection are listed below:

- **Operational and Functional Insulation:**  
This type doesn't meet any specific standard. Its main role is to ensure the proper functionality of a certain equipment.
- **Basic Insulation**  
This type provides minimum protection against electric shocks by requiring a minimum level of insulating live wires.
- **Supplementary Insulation**  
In this type of insulation, an extra layer is added to give an extra protection against electric shocks in the event of failure of basic insulation.
- **Double Insulation**  
This type of insulation provides protection from electrical shocks without the need of an additional grounding. It's usually used with class 2 equipment like hair dryers.
- **Reinforced Insulation**  
This type of insulation is very similar to the double insulation. It's a one-layer protection but has a higher thickness.

Most power supplies require a minimum insulation of type reinforced with a dielectric strength of at least 3000 Vrms for primary to secondary and a basic insulation with dielectric strength of 1500 Vrms or greater for primary to ground power supplies.

#### 4.1.6 Electric Strikes Standards

Since we are using an electric strike for locking mechanism in our project. ANSI does specify the dimensions for the body of a lock, however, there is no standard in how the latch bolt, dead latch and deadbolt are to be arranged on the lock. Which means that manufacturing companies have the freedom to choose how to design their locks with different configurations. Based on this information, we need to carefully choose the lockset manufacturer before choosing the electric strike to ensure the proper functioning of the lock.

Electric strike comes in 12 or 24 volts and it also falls under the ANSI/BHMA A156.31 standards. Since we are using a power adapter to a wall outlet to power the system, providing the needed power to our lock should not be a problem, but we are choosing a lock with 12V input to make our system consume less power.

Also, we are already using 12V power in our project, so we decided that it was best to not introduce a fourth voltage level.

### 4.1.7 Keypad Standards

Keypad flows the ITU-T Standards. (ITU Telecommunication Standardization Sector). This standard assigns the basic 26 Latin letters (A to Z) to a keypad. Based on this standard, whenever letters or digits appear on a key, there is a recommended relationship to be considered. The recommended relationships between letters and numbers are explained in Figure.19

|         |         |         |
|---------|---------|---------|
| 1       | 2 [ABC] | 3[DEF]  |
| 4[GHI]  | 5[JKL]  | 6[MNO]  |
| 7[PQRS] | 8[TUV]  | 9[WXYZ] |
|         | 0       |         |

Fig.19 Relationship between letters and numbers on a Keypad

Based on E.161 standards, the researchers discovered that this arrangement for the keypad result in a shorter entry time and lower error rate comparing to other arrangements.

### 4.1.8 Legal Standards

Federal regulatory agencies are the government departments that have responsibility for the legislation (acts and regulations) for a given sector of the United States government. These agencies are responsible to put standards to ensure that any product or device targeting the US customers is declared safe to be used. Below are the most common agency and their standards.

- **Consumer Product Safety Commission (CPSC):**  
CPSC main responsibility is to reduce the risk of deaths or injuries that can be caused by consumer products. CPSC develops voluntary standards with industry, like labeling hazardous products and stress customer safety by banning products with no standards.
- **The Environmental Protection Agency (EPA):**  
EPA is an independent agency of the United States federal government. Its task is to ensure environment safety. Following EPA regulations offer protections for both human and environment against significant health

issues. It also sponsors and conduct research regarding energy ratings and toxic materials.

## **4.2 Design Constraints**

Every Engineering design comes with its own limitations which should be considered in order to deliver a working design. Different constraint types should be considered, in this section we are to discuss the constrains relative to our project in more details.

### **4.2.1 Time Constraint**

Time is one of our biggest constraints, because we are taking Senior Design 2 over the summer. The summer semester is about 12 weeks which might not be enough time to debug and redesign in case of a system failure. Any delay on the schedule might cause a serious problem. This can be prevented by carefully inspecting every stage in building our design.

### **4.2.2 Health & Safety Constraints**

Safety is another constraint to be considered in our project which means delivering a product that can be constructed without imposing and threat to the customer. Our Smart Door Security System should be designed in a way to keep all electrical components in a waterproof case to prevent any damage that might lead to a safety threats to the end user.

Since our design will be powered by wall outlet, safety measures should be followed to protect both the designers, testers, and the customers.

### **4.2.3 Ethical Constraints**

Ethics is very important constraint as it must be applied when designing any project. Engineers are bounded by a code of ethics and are expected to follow the highest standards of honesty.

Safety of environment, customers, and engineers should be the basic pillar considered for any design. Engineers are responsible for designing machines and devices that control the modern world, therefore they have a big responsibility to safely and ethically develop their projects. Engineers who do not follow a code of ethics may put themselves or others in danger.

For the purpose of our design, we will follow the highest ethical standards to ensure our system is not to be used in bad ways to the best of our ability. For example, using a reed switch at the door will allow us to identify when the door is not locked



and that can help us protect our system user by detecting the state of the door. Using an electric strike lock is also another way to protect our system users since electric lock are hard to be picked by an intruder.

#### **4.2.4 Economic Constraint**

The economic constraint is mainly related to the cost and budget of our system. Our project is funded by the team members and therefore our budget does not have much margin to grow. To deal with this constraint we have tried our best to shop around for the needed components that satisfy the needs of our project with reasonable prices. Our final design is expected to be cheaper than existing products in the security field that perform the same function.

#### **4.2.5 Energy Constraint**

Energy constraints refers to the usage of power for a specific device. Since our system power will be supplied through wall outlet, energy constraints are minimal in our case. The smart door security system keeps all components in a sleep mode until the ultrasonic sensor detects a moving object within a close range. In order to reduce energy demand, some components might be operated in low power or power down mode. We will do our best to make our system as energy efficient as possible.

#### **4.2.6 Social Constraints**

Social constraints refer to the social impact our design has on customers or community. We aim to provide a system that is very user friendly that allow the customers to have easy access to their homes. The fact that our system uses facial and voice recognition to detect authorized users will make the future customers very comfortable to rely on us when it comes to protecting their homes. Updates and notifications will be sent directly to the owner email, which is another safety feature that we aim to provide. We believe our system will be socially successful and our goal is to have all customers happy with the product and its features.

#### **4.2.7 Political Constraints**

Just like the social constraints of our design were considered, a design's political constraint must also be taken into our consideration.

Political constraints can be defined as the impact of a product or an engineering design on any government, political infrastructure, or construct. Our Smart Door Security System is being designed as a part of the educational process needed to satisfy a bachelor's degree requirements. With that being said, our design should not have any political impact on any way and therefore there is no political constraints associated with our design.

## **4.2.8 Manufacturability Constraint**

Since the word constraint is defined as anything may affect the ability to make progress, manufacturability constraints appear as obstacles when it comes to manufacturing or reproducing a product. Manufacturability is the ability of an engineering design to be produced with fewer like parts count, labor, or maintenance. By developing a set of specifications requirements our smart door security system should be designed to allow easy development and produce. The work was explained in this paper and all chosen components were listed with their specifications; these specifications should serve as a guideline when a component needs to be replaced.

Another manufacturability constraint that is addressed is that we used mostly common components and component footprints in our PCB. Also, we are following the guidelines of our chosen board house in limitations such as the routing of the board and size of traces and pads. Finally, by fully documenting our work, it should be easy for a manufacturing company to follow our design and make development to enhance the design before reproducing.

## **4.2.9 Sustainability Constraints**

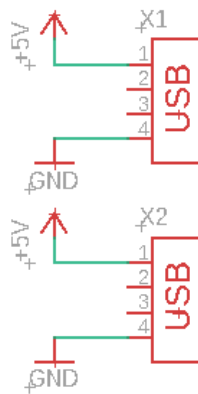
Sustainability is defined as the ability to execute an engineering design in a normal operating condition for a specified period of time. Since most of the smart door security system will be mounted inside the door, the system should be able to fully operate with no issues. To protect the PCB and the wired components, a casing can be used to host the PCB in it which helps protecting it from environmental conditions. We also make sure that our board house uses lead free solder, as this will reduce environmental contamination.

# 5. Project Hardware and Software Design Details

## 5.1 Hardware Design

Our project is controlled by the system controller PCB, which we are designing. This board will handle power distribution and check if voice recognition, facial recognition, or pin entry were successful. The MSP430FR6989 is at the core of the system controller, and it will make the decision on whether or not to open the door. The MSP430 checks if two of three criteria are met, and this microcontroller directly activates the electric door strike.

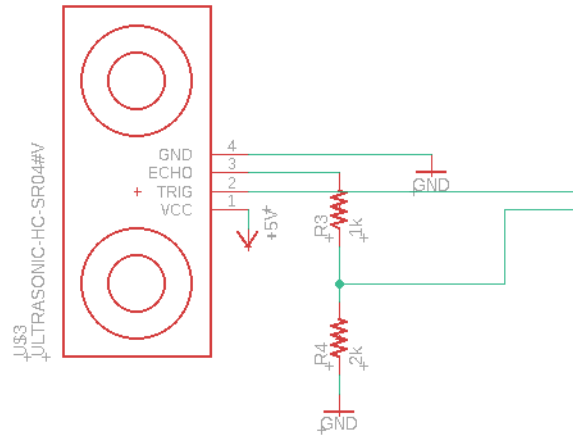
Initially we will discuss power distribution, as this is a major part of the board. A female dc jack plug accepts the 12V in from the power supply adaptor. This 12V is distributed to the electric door strike and the 5V power buck converter. The 5V buck converter was designed using Webench and is shown in the power section of this paper. 5V is used to supply the Raspberry Pis, so the 5V that is produced by the buck converter is routed to the edge of the board, where two USB-A female connectors are. These USB-A connectors are wired such that only power is transferred, so pins 2 and 3 are left unconnected. 5V power is transferred to each Raspberry Pi by using a USB-A to USB-C cable. Last, 5V is converted into 3.3V using a second buck converter. The 3.3V power is only used by the MSP430FR6989 microcontroller, so it stays on the board. The wiring for the USB-A female plugs that will be used for providing power to the Raspberry Pis is shown in Figure[20].



Figure[20]: Power From the PCB to the Raspberry Pis

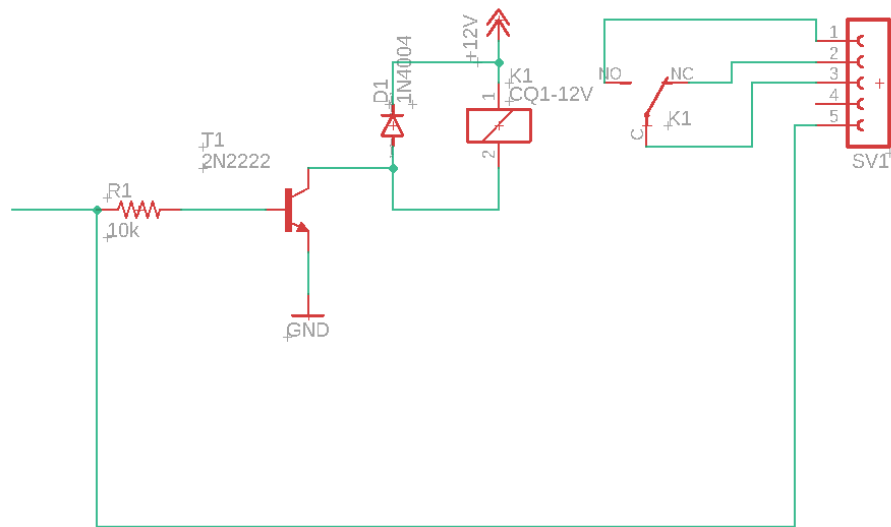
The HC-SR04 is also mounted on the PCB, and it receives 5V power from the PCB. Additional pads or headers are attached to the PCB to connect between the GPIO pins of the MSP430 and the Raspberry Pis. The electric strike lock and led strip connect to the female header pins of the PCB. Fig [21] shows the way that we decided to wire the HC-SR04 the resistors are used to convert the logic level

to 3.3V so that we do not harm the MSP430. The two wires on the right correspond to MSP430FR6989 GPIO pins 2 (P1.4) and 3 (P1.5).



**Figure [21]: Ultrasonic Sensor Wiring and Logic Level Conversion**

The LED strip and electric strike lock run off of 12V, which cannot be provided by the MSP430FR6989. We designed a relay to interface between the MSP430 and the 12V systems so that they can be controlled by the MSP430. The base design that we created is shown in Figure[[]].



**Figure[22]: Relay to Interface Between MSP430 and Electric Door Strike Lock**

The wire on the left of Figure[22] connects to an output pin of the microcontroller. If a signal is applied, the NPN transistor turns on and allows the 12V current to flow, activating the relay. The female pin headers on the righthand side of the schematic are where the electric strike wires attach. We brought out both NO (normally open) and NC (normally closed) so that we have options in picking a lock. We will likely use the NC pin because it corresponds to fail-secure. The

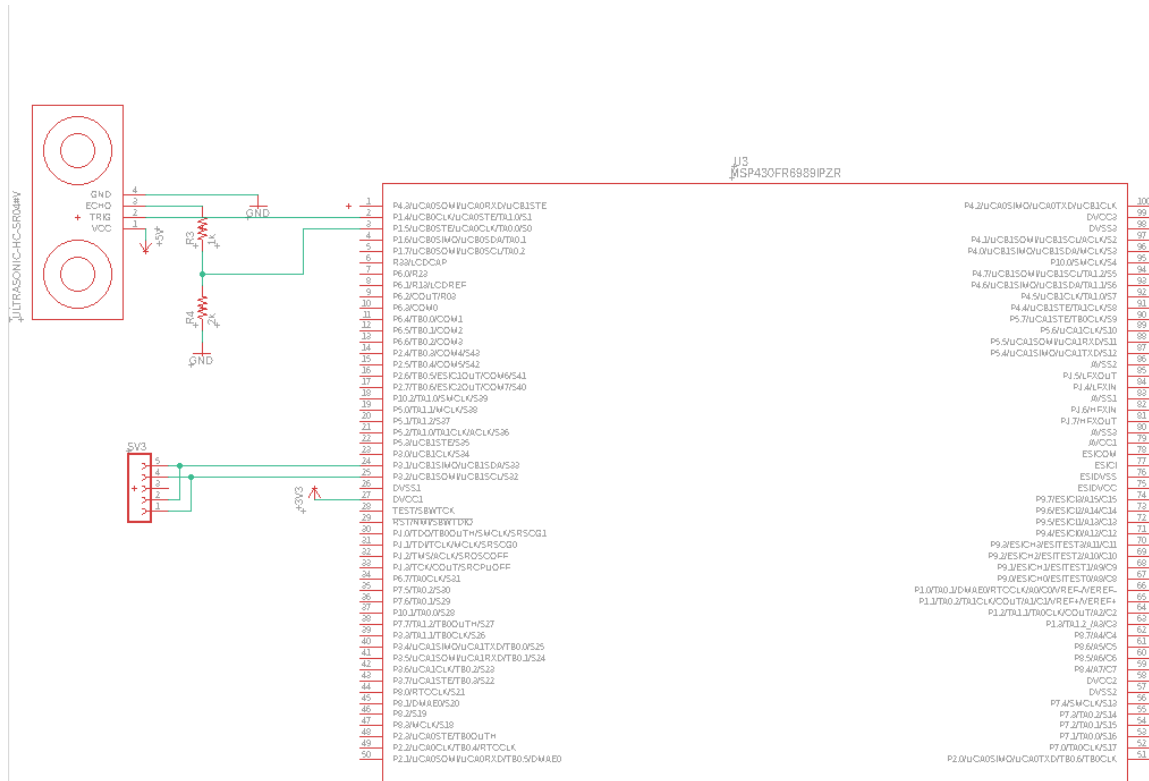
1N4004 diode is used in this circuit to prevent back-EMF from the breakdown of the electric field after the lock is de-activated, as this has the potential to hurt the microcontroller. Header Pin 5 is used for troubleshooting in the event that the lock does not open on the first revision of the board. This will allow us to check to see if the microcontroller is providing the correct signal, in the event that we have other issues with the relay circuit.

A similar relay module may be used for the LED strip, but we are unsure at this point in time if the LED strip will be bare-wires or have a 2.1mm DC jack plug, or other type of connector. We are also unsure of whether or not this will be necessary. We may redesign to include 5V instead of 12V RGB LEDs, and this would enable us to reduce cost and complexity by simply plugging the LED strip into a USB-A port on one of the Raspberry Pis rather than having to design an additional relay circuit to operate the LED from the microcontroller 3.3V circuit. Including an additional relay would drive up the BOM cost and potentially increase lead time if the corresponding components to that particular circuit need to be sourced from somewhere else. We are attempting to use only in-stock, readily available components in our design so that we can rapidly prototype and send out PCB revisions quickly.

The current schematic for the MSP430 showing outside connections is shown in Fig[ ]. The female pin headers connected to pins 24 and 25 are for the SPI communication to the Raspberry Pi GPIO pins. An engineering investigation is currently in progress to determine if SPI communication between the microcontroller and single-board computers is necessary for our project. An alternative was proposed that would set GPIO output pins on the Raspberry Pi and GPIO input on the MSP430. The microcontroller would then check if the GPIO input is at a high voltage level or not, indicating whether or not a user is authenticated. There would be three such pins, corresponding to facial recognition, voice recognition, and keypad entry. This would reduce the number of connections necessary between the PCB and Raspberry Pis by one, as well as providing room for expanding authentication methods equal to the number of GPIO pins available on the MSP430 and Raspberry Pis.

The HC-SR04 ultrasonic sensor is wired directly to the MSP430FR6989 and is also mounted on the PCB. The microcontroller sends a signal to turn the sensor on and marks the time when the ultrasonic sensor reports a signal. We need to be careful when placing the HC-SR04, because it cannot be directly behind other components. We need to ensure that it faces out so that it does not simply record the distance to the next component on the PCB, but rather senses the surrounding room. A hole will be included in the 3D-printed enclosure so that the ultrasonic sensor can read the distance to a person trying to unlock the door. Figure [23] shows the current connections to the MSP430, minus a few pins that are yet to be

determined for communication to and from the Raspberry Pis as well as powering the microcontrollers.



**Figure[23]: Current MSP430 Connections**

The resistors are used with the HC-SR04 in order to scale the voltage level down to the 3.3V that the MSP430FR6989 uses. The ultrasonic sensor uses 5V, so we need to ensure that we do not harm the microcontroller when the echo is sent.

We plan to make a smaller, roughly 4:1 scale door and door frame to showcase our design at the senior design exposition. When the door is created, with the electric strike lock installed, we will include a photo to show how it is wired and attached into the frame. We also may need to include a magnetic reed switch to determine door position (opened or closed), but this is dependent on the lock that we pick. Most electric strike locks commercially available do not include great datasheets or explanations of operation, so we do not know for certain if our design requires a magnetic reed switch. We will attach the necessary MSP430FR6989 pins to female pins headers with a current limiting resistor in the event that we need to include a magnetic reed switch, so that quick software implementation is possible. We are trying to make revision 1 as modular as possible with numerous test points so that we can use the board for testing and potentially add sensors as we find a need.

Communication to and from the Raspberry Pi by the MSP430 will be done by setting the logic level on the Raspberry Pi GPIO pins. If we were using SPI or I2C, the Raspberry Pi would be slave devices and the MSP430FR6989 would be the master device. We decided that the best and most fool-proof method of implementing this idea would be to set GPIO pins to logic high levels on the Raspberry Pi. There would be two pins on one of the Pi, corresponding to keypad and facial recognition, and one pin on the other pi, corresponding to speaker recognition. The microcontroller would make the decision to authenticate the user based on the number of pins that are active high (two for two-factor authentication). This would avoid the complexity of implementing a communications protocol in two different languages, Python and C, which would be an issue because C is a lot lower level than Python. The GPIO pins on the MSP430 will be attached to female pin headers in order to connect to the Raspberry Pi. The specific GPIO pins on the MSP430 have not yet been determined.

The final design detail that needs to be worked out is how to program the MSP430FR6989 after it has been assembled on the circuit card. We have not yet determined a great solution to this problem. Initially we were not aware of the potential options in which we are able to interface to the MSP430FR6989 after it has been populated on a circuit card. All of us only have experience with this microcontroller on a development or evaluation board set up. In that situation, the development board and microcontroller is plugged into a PC via a micro-USB cord, and then programmed from the Code Composer Studio development environment using the built-in USB to UART bridge. But since the MSP430FR6989 will be on our own custom circuit card assembly, we are going to need a dedicated interface to be able to program the microcontroller, and that interface is still yet to be determined. The method for programming MSP430 microcontrollers is detailed in Texas Instruments' slau320: "Programming with the JTAG Interface". We are likely to choose the easiest available option, and considering our application is not extremely sophisticated, we do not believe we will need to create any custom interface or programming solution. Essentially, we do not need to reinvent the wheel to get this microcontroller programmed.

There are three possible options for programming the FRAM device that is built-in to the MSP430FR6989 microcontroller.

- 1) Program the FRAM with JTAG or Spy-by-wire

JTAG is an industry standard interface for verifying designs and testing printed circuit boards after manufacturing that has been used for many years. JTAG was named after the Joint Test Action Group who created it, and it was accepted as the IEEE standard 1149.1 in 1990. The IEEE standard 1149.1 is entitled *Standard Test Access Port and Boundary-Scan*

*Architecture.* It essentially is an industry standard used all over the world to verify and test that your design does what it was designed to do, as well as allows an interface for programming different microcontrollers and configurable logic devices such as Field Programmable Gate Arrays (FPGAs). The JTAG interface uses 4 standard pins which are TDI, TDO, TCK, TMS which stand for Test Data In, Test Data Out, Test Clock, and Test Mode Select, respectively. These 4 pins allow for the serial input of test data, using the TDI pin, which can then be verified against the data that is received out on the TDO pin. This TDO pin allows the user to receive the data that is being outputted by the circuitry and compare it to some calculated values that you are expecting. This allows for easy verification of your circuit's design as you just need to create test vectors to input serially to the TDI pin, and then read the output of the TDO pin, and compare that the data being received from TDO is what you are expecting it to be. Along with the verification aspect that was just explained, JTAG can be used for a dual purpose, which is programming device hardware. Many FPGA's and Microcontrollers in the industry can be programmed using the JTAG interface.

## 2) Program the FRAM with the Bootloader (BSL)

Every MSP430FR6989 device contains a bootloader stored in Read-Only-Memory. The bootloader on an MSP430 device lets users communicate with embedded memory in the MCP430 microcontroller during the prototyping phase, final production, and in service. Both the programmable memory, FRAM, and the data memory, RAM, can be modified as required. The Bootloader enables users to read or program the FRAM or RAM using a UART serial interface or an I2C interface. The user can either use a software specific bootloader invocation or a hardware specific bootloader invocation. The method for programming the MSP430FR6989 microcontroller using the bootloader is very intricate and you must have an exceptional knowledge of the memory architecture, bootloader configuration, and bootloader protocol that is used to execute the bootloader commands and correctly and effectively erase/load/program the memory space. Due to the fact that this is extremely complicated it would not be the best idea to use this method to program the MSP430 device, as there are many ways that this could go wrong. All of the ways that this method could go wrong are also not easy to troubleshoot and might very well be the demise of our project. With that being said, we will most likely just stick with JTAG programming for our device.

## 3) Program the FRAM with a custom solution

This programming solution is the last option, probably the most time consuming, and realistically the second most complicated (behind using the



bootloader). The MSP430FR6989's CPU has the capability to write to its own FRAM device. Because of this capability, it allows for in-system and external custom programming solutions to be developed. This feature allows the user to provide data to the device through any necessary means available, for instance through a SPI or UART interface. Considering this, the user-developed software can then receive data via that UART or SPI interface, and then use that received data to program the FRAM device. Since this programming solution is developed by the user, it can be completely customized to fit the application needs for programming or updating the FRAM however the user sees fit. Because of this level of customization, it turns out to be the most complicated and in-depth solution for programming the FRAM of the MSP430FR6989 microcontroller. It also does not fit out application needs, so because of the complexity and it not being a good fit we will not pursue this method of programming the microcontroller.

After the discussion and research of potential options that we can use to interface and program the MSP430FR6989 microcontroller once it is populated on our custom circuit card, we are going to implement a standard JTAG interface that will be used for both programming as well as potentially debugging if we need to. The JTAG interface will be mainly used to program the FRAM device of the microcontroller with our system's code. After compiling in our development environment, we will be able to flash/bootload the microcontroller.

On each revision of the board, we will break out the MSP430 pins used in JTAG programming and attach leads to pin headers so that we can program the board. We also will attach female pin headers to the MSP430 pins used in Spy-by-wire so that in the event that we are unable to program the microcontroller using JTAG, we have a backup plan already set up. This will prevent additional cost and lead time in ordering a second custom board just for the purpose of programming the microcontroller, as there will likely be other bugs that have not been found yet. For example, if we have an issue with the power supply, and we are unable to program the microcontroller, we will not be able to find the issue with the power supply until the second revision of the board, causing us to have to order a third revision of the board, which may not be possible to receive and test due to the 12-week time constraints of taking Senior Design 2 during the summer semester.

Upon further review of the design, it was discovered that there was not a fail safe way to control the door lock manually. This would be needed in case of an emergency or to grant access manually for strangers. The button will be in series between 12V and the lock terminals, so that when the button is pressed, the door gets unlocked. This feature will also be used when exiting the house. We are currently investigating ways to keep the door unlocked for a set amount of seconds after the button is pressed without involving the microcontroller. We may include

energy storage via a large capacitor or battery pack so that in the case of an emergency our fail secure lock can still be operated to leave the house.

Because the wiring for the fail secure lock would be internal to the house, we will be able to still open the door to get outside by flipping the switch or button, even if the system outside is compromised. By choosing a fail secure lock, our system is more secure, and access cannot be granted simply by clipping the power lines from the system controller to the lock.

A stretch goal for our project is to create a physical model for the door to show the system and lock operation at the senior design exposition at the end of Senior Design 2. The idea is to have the system in place in the approximate locations that it would be in a real house, and this would allow simulation of walking up to the door from both sides and then unlocking it to get in or out. We would also be able to showcase the different authentication methods and how they affect the LED lights that would be in the overhang of the doorway. We are still determining where the different components of the system will be placed, and whether they will be inside or outside. Part of our design process in Senior Design 2 is to figure out where the optimal locations of the system modules are, and where we will have to route wires through the doorway. Likely, the wires will be routed in the same manner that they would be for an outside doorbell.

## **5.2 Software Design**

### **5.2.1 MSP430FR6989 Software Development Language**

#### **Choice:**

The MSP430FR6989 is a low power, mixed signal 16-Bit RISC architecture Microcontroller, meaning that it is the perfect microcontroller option to build a low powered system around. But being a low-level RISC architecture device that requires very low power means that it cannot handle the highly complex and computational heavy workload that our system needs in order to be able to successfully implement Facial and Voice Recognition. The MSP430FR6989 microcontroller will be our “System Controller” or central brain of the system. It will interface with two other Raspberry Pi’s that will handle the highly complex and computational heavy workload of Facial and Voice Recognition.

When the Raspberry Pi’s recognize the homeowner, the MSP430FR6989 will be able to see this and then control the door’s lock accordingly. To implement this functionality, we will use C programming that will be compiled and stored on the microcontroller’s non-volatile flash memory. The MSP430 family of microcontrollers supports development of C and C++ as well as assembly language programs. As the Computer Engineering majors in our senior design group are more comfortable with C programming, we will utilize this language to

develop the main system controller software that will run on the MSP430 microcontroller. C programming is a low-level procedural language that is perfect for the development of programs that run on a microcontroller that needs to have easy memory access, easily set register values, use bit masking and more. C programming is high enough level that programming does not take an excessive amount of time, but it is low enough level that we are able to perform the needed embedded system tasks and interface with the sensors.

### **5.2.2 Raspberry Pi Software Development Language Choice:**

Because the Raspberry Pi is a fully functioning computer that runs an operating system – in our case we are using the standard Raspbian OS – any Unix-supported compiler can be installed on the Raspberry Pi, meaning that it supports virtually any programming language for development. We chose to do the development for the software that will be running on the Raspberry Pi in Python, because it is a perfect fit for our application's purposes. Python is a very high-level object-oriented language that has an enormous number of packages and libraries that are either built in or externally developed and can be imported for the programmer's use.

All of these packages and libraries were created to help make development using python even easier. Also, many of them were specifically created for the realm of our project including; Image Processing, Facial Recognition, Speech-to-Text processing, Speaker Recognition, and much more. Doing the development of the software that will implement facial and voice recognition in Python will allow us to take advantage of using these existing packages and libraries, essentially making the overall development process much simpler and easier to do.

By utilizing these existing packages and libraries we will not have to develop from scratch our own algorithms or functions that are needed to perform the complicated mathematics necessary for processing video or audio for our facial recognition and voice recognition authentication methods, but instead will allow us to focus on the integration and implementation of these packages and libraries to suit our system's application goals.

### **5.2.3 Software Overview and Status**

#### Raspberry Pi Software Features:

Facial Recognition:

- Current Status: Running stand alone on the Pi w/ USB webcam
- Future Tasks:
  - o Integrate into the entire system
  - o Implement this process in its own thread using multithreading
- Risks or Issues:
  - o Can't dynamically add user's once the system is set up and running

- o High resource usage – probably best to use Pi w/ 8G of RAM

#### Voice (Speaker) Recognition:

- Current Status: Still under investigation and research
- Future Tasks:
  - o Choose and test a speaker recognition implementation using a Pi
  - o Implement this process in its own thread using multithreading.
- Risks or Issues:
  - o Can't dynamically add user's once the system is set up and running.
  - o High resource usage – probably best to also use Pi w/ 8G of RAM (but 4 might be okay if we don't want to purchase another one, testing will determine this)

#### Keypad [for pin # entry]:

- Current Status: Running stand alone on the Pi using 8 GPIO pins as it requires
- Future Tasks:
  - o Implement storing homeowner's pins that are entered from the keypad
  - o Implement validation checking of what the user enters on the keypad and what are stored as valid pin #'s
  - o [*Possible stretch goal?*] Figure out/implement being able to dynamically add pins after the system has been set up – this would require a certain 'key' to enter the "Add a Pin" mode, for ex: If you enter, \*0000#, at the door; that would go into a mode to allow the user to add a new pin and then store it for future authentication.
- Risks or Issues:
  - o (Not including the stretch goal) We might have to hard code the homeowner's pins, if it's too difficult to dynamically add or remove stored pins after the system is set up and running.

#### I2C Bus:

- Current Status: I2C has been implemented for use with the LCD screen [which is a stretch goal]
- Future Tasks:
  - o Use I2C for inter-chip communication, test with communication between Pi's and between Pi and MSP 430
  - o For our system's final implementation: The Pi's will be SLAVE devices, and the MSP430 will be the MASTER device – keep this in mind and implement accordingly.
  - o Figure out how to set a variable whose value will be stored in a register as a flag for authentication – that register will be continuously read from by the MSP430 to check if the authentication flag was set or not (signaling the homeowner is detected)

- Risks or Issues:
  - It's unclear how we can set a register that we will use to flag if the homeowner has been authenticated or not – Need to be able to do this so when the facial/speaker recognition authenticates a user, the MSP430 will be able to know (and therefore unlock the door)

#### 16x2 LCD Screen [*Stretch Goal*]:

- Current Status: Set up on the Pi using I2C
- Future Tasks:
  - Incorporate in the system to print prompts to the user, and/or just a “Welcome Home!” message when the homeowner is authenticated, and the door unlocks
- Risks or Issues:
  - The LCD screen will be outside and possibly exposed to the elements. We need to ensure that we choose a ruggedized LCD screen for use in our project

#### MSP430FR6989 Software Features:

- We will have one main C program, which will be the brains of the system and control everything that is going on (hence the MSP430 being labeled “System Controller” on the block diagram). This main C program will include the implementation of the following features:

#### Sonar Sensor:

- Current Status: Let's take advantage of possibility of re-use: we currently have the C code from Jr Design using the same mcu/sensor. Have not yet tested the code out myself to see if it works
- Future Tasks:
  - Test the code from Junior Design that corresponds to the MSP430 and the sonar sensor
  - Incorporate this implementation into our own system software
- Risks or Issues:
  - May need to get permission from Dr. Richie to Re-Use the Jr Design code, since this is not our IP

#### Electric Strike Lock:

- Current Status: Need to investigate and research how this can be/will be controlled.
- Future Tasks:
  - Figure out what is needed for the electric strike lock.
  - I am thinking the default should be to hold the lock in the “Locked” state (which I believe should be active low) and send a pulse [aka pull the line low] for a set period of time to unlock the door. This would basically work as: a constant voltage is sent to the lock to keep the

- o door locked, until user is authenticated, and we pull the line low, and, during that time, no voltage is applied hence the door is unlocked
- o Need to determine the amount of time that our pulse should be pulled low, or in other words how long the door is unlocked for after the homeowner is authenticated – \*this will automatically implement “Auto-Relock” feature as well\*
- Risks or Issues:
  - o Figure out/make sure the implementation is possible like this – testing may be the only way this is possible unless documentation for the Electric Strike Lock contains this information

#### I2C Bus:

- Current Status: No code currently written, but another possible re-use opportunity. I2C was implemented in Embedded Systems and Jr. Design for the LCD screen.
- Future Tasks:
  - o I2C needs an initialization function – can be re-used from Embedded or Jr Design
  - o Implement I2C to be able to communicate with a Pi, then with both Pi’s
  - o Figure out the correct implementation to read our “Authentication” flags from the Pi’s – which will store whether the facial recognition (on Pi 1) is authenticated, and whether the speaker recognition (on Pi 2) is authenticated, and whether the keypad pin number that was entered (on either Pi – have decide which one will implement this) is authenticated.
- Risks or Issues:
  - o Being able to read our “Authenticated” flag from the Pi’s – how is this done? We will need to know the specific Pi’s device address as well as the register address that we are going to read from in that Pi

Besides the main features shown above, our System Controller MSP430’s main C code will also need to include:

- Clock configuration
- Timer configuration/implementation
- Interrupt configuration/implementation

These are all things that we have learned and previously implemented in our Embedded Systems course and we will most likely re-use much of that code to make the coding we will need to do for our project quicker, smoother, and easier overall. Code re-use is a staple of software and firmware engineers in the field, and as we are targeting the development of a professional style system, we will hold ourselves to the same professional standards.

## 5.3 Potential Hardware Issues

Any hardware design can potentially have some issues, some of those imperfections might occur due to improper installation of the system, electrical surge, high current or thermal incidents.

During the installation process, we need to ensure every step is reviewed. Detailed step by step instructions need to be followed to ensure the proper working of the system later on. After ensuring that the system is installed, and all components are properly connected. Another issue we might face is an electrical surge. Electrical surge might occur due to multiple reasons such as lousy wiring, lightning strikes, damaged power lines and more. To protect the voltage regulators, Zener Diode Transient Suppression Devices can be used. When the 5V voltage regulator input terminal detects a surge greater than 5V, the diode will open, allowing the leakage current to be released to ground, and this limits the input voltage to 5V. The same technique can be used for the 3.3V regulator.

High current flow is another common reason for issues in electrical devices. If the components receive a higher current than the recommended operational current, these components can get damaged. To protect the elements against high current flow, a fuse is needed at the connection between the board and the main power source. Inserting a fuse protects the device because if the fuse is blown, the connection is broken and only the fuse is damaged. The fuse can be replaced at a much cheaper cost than replacing the entire device. Another way that high current flow is controlled is by using current-limiting resistors at key connections with low resistance.

Thermal events are another reason that can easily damage the components. Each component is supposed to operate under a recommended temperature that is specified in its data sheet. If the circuit is experiencing a high current flow, component may heat up and the circuit might no longer operate properly. Thermal damage can be introduced by multiple factors like high current flow or long exposure to sun. Heat sinks can be used to protect the voltage regulators against thermal damage. We will explore this further to determine if thermal events are going to be an issue in our design after receiving and testing revision 1 of the PCB.

A potential issue for our project is programming the MSP430FR6989 while it is on the PCB. There are a couple methods of doing this, but we need to ensure that we have the correct pins attached to headers so that we can access them. Flashing the board without using the launchpad is possible using the JTAG interface, as explained in the Texas Instruments document SLAU320ai. Also, the MSP430 can be programmed using SPI-by-wire, but this approach is not as well

documented. The potential issue here is that the model MSP430 that we are using is a surface-mounted device, so we cannot simply plug the microcontroller into the Launchpad interface board and program it. We may need to design a special interface between the MSP430 and a personal computer to be able to flash the chip. Use of a USB to UART bridge is also being researched in support of this effort.

Another potential hardware issue is powering the Raspberry Pi's from the PCB that we are designing. While we went through the necessary calculations to ensure that we are not using more current than can be supplied at each voltage level, we are worried that during heavy computation, the Raspberry Pi boards may try to draw more current than can be delivered through the USB-A to USB-C cables. This could cause the cables to heat up, but alternatives are being investigated. USB-A was originally chosen on the PCB side because it is larger and has fewer pins. This will be extensively tested in Revision 1 of the board.

The last major hardware issue is lead time from the board house. We are planning on at least two revisions of the PCB to work out bugs and ensure our product works as intended. Because the summer semester is short, minimizing the lead time from the board house is a top priority. A potential issue is if they are unable to quickly procure components; this would cause us to redesign.

A minor hardware issue is operating the door strike and LED lights. Both of these run off of 12V, but the MSP430 only supplies 3.3V. The use of relays is being investigated to ensure that the electric door strike and LED lights get the proper voltage when they are turned on.

Another potential hardware issue is that we will have a hard time determining if the PCB was printed and assembled correctly. If the board house neglects to solder a pad or two, our project will not work or will not work correctly, even if it is programmed correctly. We are attempting to mitigate this possibility by only picking a reputable board house and staying within their tolerances in our PCB layout.

We need to ensure that we make the traces on the PCB that transfer power large enough and with enough clearance so that they do not excessively heat up or short to other traces. We will likely make any unused space on the PCB into a ground plane to help with heat dissipation. This will be detailed in the DRC and cross-checked by the board house requirements.

## **5.4 Potential Software Issues**

Just as there are many potential hardware issues that could arise, there are also a vast number of software issues that could come up. The follow section will



highlight and try to plan for the potential software issues that may come up during development. It will also include possible solutions to the issues, if necessary. The implementation of our Facial and Voice Recognition algorithms will be the most difficult portion of the software development. Specifically, the integration of those stand-alone algorithms into one large piece of software, using multiple threads, and being used for the purpose that we need it to in our entire system. This can create many issues that we need to be ready for during software development in Senior Design 2. However, this issue can be mitigated by using two Raspberry Pis in our system, as they will be able to run these functions in parallel, rather than one Raspberry Pi trying to run both facial and voice recognition concurrently.

The Facial and Voice Recognition algorithms themselves are both going to be implemented using open-source code that is already written by other developers and is freely available to the public. The open-source code that we try to implement could potentially create some issues as it might have bugs that we are not able to see or find out right away. All of the open-source code, as well as the code we write ourselves, will need to be tested thoroughly before we will be able to determine that it is good for final deployment in our system. The open-source code also might not be compatible with our current system's settings or tool version numbers, for example, so we could possibly have issues trying to use the open-source code and get it to work for our own custom application.

The solution to any issues that come from the open-source code we try to implement in our system will be solved by making sure that our tool versions and systems will be compatible ahead of time. If this is not the case, we can either try to implement with our current environment and test the functionality, or we can look to find other open-source implementations that are compatible with our system and tools. This effort is essential to streamlining our project development and making sure that we are consistent in our approach.

Another possible issue that we are most likely going to face is the integration of the stand-alone pieces of software into one large program that will satisfy our custom application's needs. If you take Facial Recognition, for example, we were able to get that feature's open-source code running and tested successfully on its own Raspberry Pi, as a stand-alone program with nothing else happening concurrently. This was simple enough, and only took about a day, but the problems and issues will most definitely arise when we are going to have to incorporate this code into a higher-level program, that will use this facial recognition implementation to interact with our system and communicate when there is a person recognized. We are going to need to run the Facial Recognition in its own thread of the overall program running on the Raspberry Pi. The main program running on the Raspberry Pi will then incorporate some wrapper logic to check whenever the facial recognition thread recognizes someone at the door. This

integration of existing open-source code into our own system's custom application is likely to cause some issues.

There is nothing that we would be able to do to avoid these specific integration issues, since our development of a custom application must incorporate the Facial and Voice Recognition, and we don't have the time, money, or knowledge to develop and implement our own Facial and Voice Recognition. We will just need to make sure to do extensive debugging and testing before deployment of the full system, to ensure that the wrapper logic around the open-source code is working correctly and does not have any bugs that will cause the system to fail.

The last potential software issue that may present itself during development in Senior Design 2 is the possibility of code compilation issues or compiled binaries not being able to properly run on a specific MCU platform. Luckily, this can easily be avoided by the development environment that is chosen. For the development of our main C program that will run on the MSP430FR6989, we will use the Texas Instruments Code Composer Studio development environment that is the standard development environment used for applications that are going to be run on the MSP430 line of Texas Instruments microcontrollers.

## **6. Project Prototype**

### **6.1 Integrated Schematics**

The schematics are not completed on Eagle, but they are in process. We are in the final stage of the Revision 1 schematic design. The only remaining pieces before we can publish a completed schematic are determining pins and process for JTAG programming of the MSP430 and decided on a suitable method of communication between the MSP430 and Raspberry Pi. Once we know what additional pins are needed on the MSP430, we can paste the completed schematics.

The power section of this paper shows the schematics for the 12V to 5V and 5V to 3.3V buck converters. These designs are straightforward in topology and were produced online using Texas Instrument's Webench tool.

### **6.2 PCB Vendor and Assembly**

We are going to try and get Revision 1.0 of the Schematic finalized soon, that way we can pursue getting a quote from multiple different vendors. In order to do that we must complete the schematic and then get the board layout design complete. The board layout will include all of our specifically chosen part's footprints, as well as all the signal traces routed effectively. When the board layout is complete, we can put together a Bill of Materials that will include all our component manufacturer's part numbers. Once the board layout, schematic, and Bill of Materials are all finalized and complete, we can submit the board layout and schematic documents to different PCB vendors for our quote.

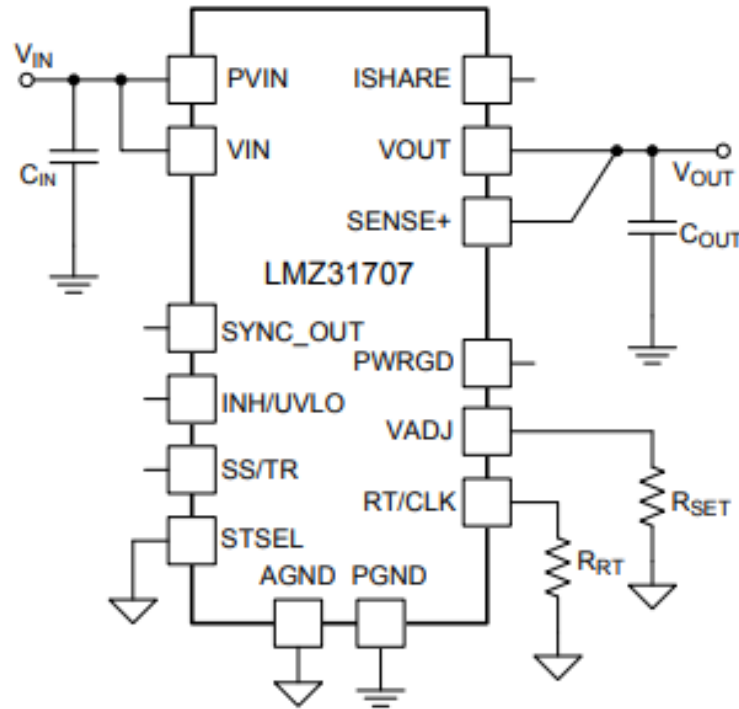
We will choose a vendor based on the quickest available lead time for our circuit card's fabrication and population of components, as long as it is reasonably priced. This will put us in the best situation going forward for Senior Design 2 over the summer. We are aiming to get our quote submitted and have a Printed Circuit Board vendor fabricate and populate our board by the time the summer semester starts, that way we can get our circuit card fully tested for functionality as soon as the semester starts. This will be crucial to the success of building our project. By having our circuit card in hand right as the semester for Senior Design 2 starts, we will have time to troubleshoot and update our circuit card's design, if necessary and then resubmit that updated version to the vendor for re-fabrication.

### **6.3 Updated PCB Design**

Initially we considered the design of buck converters using discrete components, after doing some more studying of the parts we figured that some parts were not



It comes in a size of 10-mm × 10-mm × 4.3-mm, and follows the QFN package which makes it easy to solder onto a printed circuit board and allows a compact point-of-load design. This part achieves greater than 95% efficiency and excellent power dissipation capability with a thermal impedance of 13.3°C/W. The LMZ31707 offers the flexibility and the feature-set of a discrete point-of-load design and is ideal for powering a wide range of ICs and systems. Below is the simplified schematic of the LMZ31707, since LMZ31707 has an adjustable output voltage, the resistor Rset is used to set the output voltage to the desired value.



Figure[25]: LMZ31707 Simplified Schematic  
The figure display the simplified schematic of the buck converter namely LMZ31707

### L78L33ACUTR

The L78L00 series of three-terminal positive regulators employ internal current limiting and thermal shutdown, making them essentially indestructible. If adequate heat-sink is provided, they can deliver up to 100 mA output current. They are intended as fixed voltage regulators in a wide range of applications, also, they can be used with power pass elements to make high-current voltage regulators.

In our design this part is used to step down the 5V to a 3.3V to supply the microcontroller with it's needed power. Since the MSP430FR6989 only needs 100uA of current to operate and needs an input voltage in the range of 1.8-3.6V this part is more than capable of supplying the desired power.

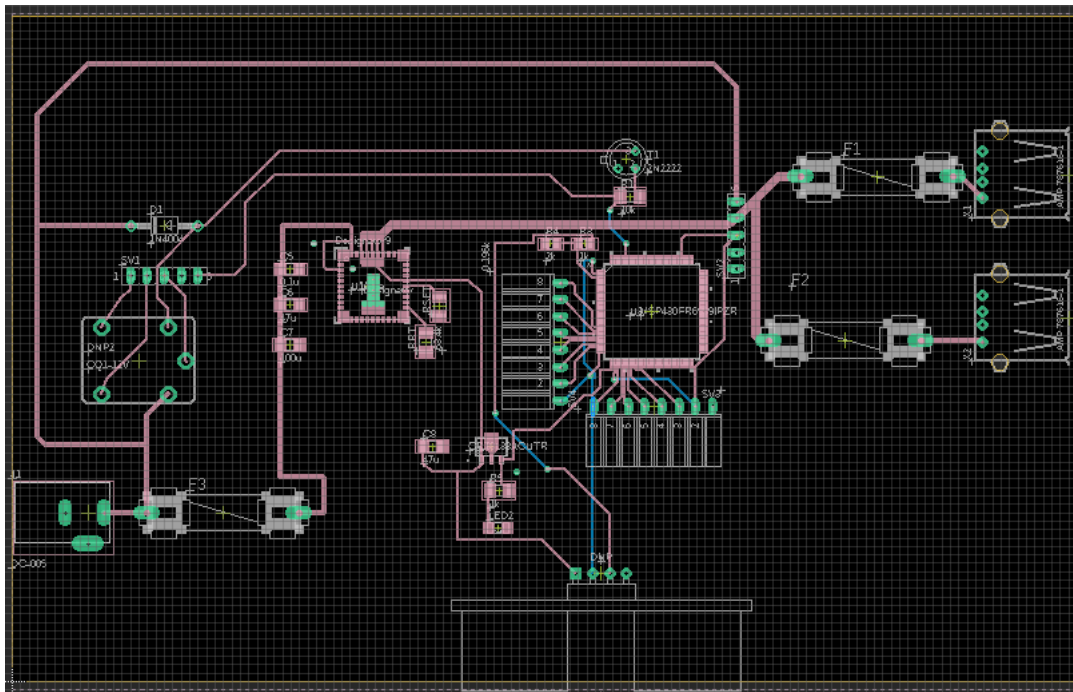
The used buck converters are LMZ31707 and L78L33ACUTR, after reviewing the datasheets of the chosen components, we verified that these parts are capable of supplying our design with the needed power, so we decided to proceed with the new design. Below are the specifications of the buck converters:

| Specifications | LMZ31707 | L78L33ACUTR |
|----------------|----------|-------------|
| Input Voltage  | 12V      | 5V          |
| Output Voltage | 0.6-5.5V | 3.3V        |
| Output Current | 7A       | 100mA       |

Table[21]: Specifications of Buck Converters

### 6.3.3 PCB Layout

The final Schematic of the design was then printed on a board using the footprint of the used components, connected through the traces of different width to accommodate the different currents bring down. This results in the layout of the circuit design. The Figure 26 below shows the layout of our printed circuit board.

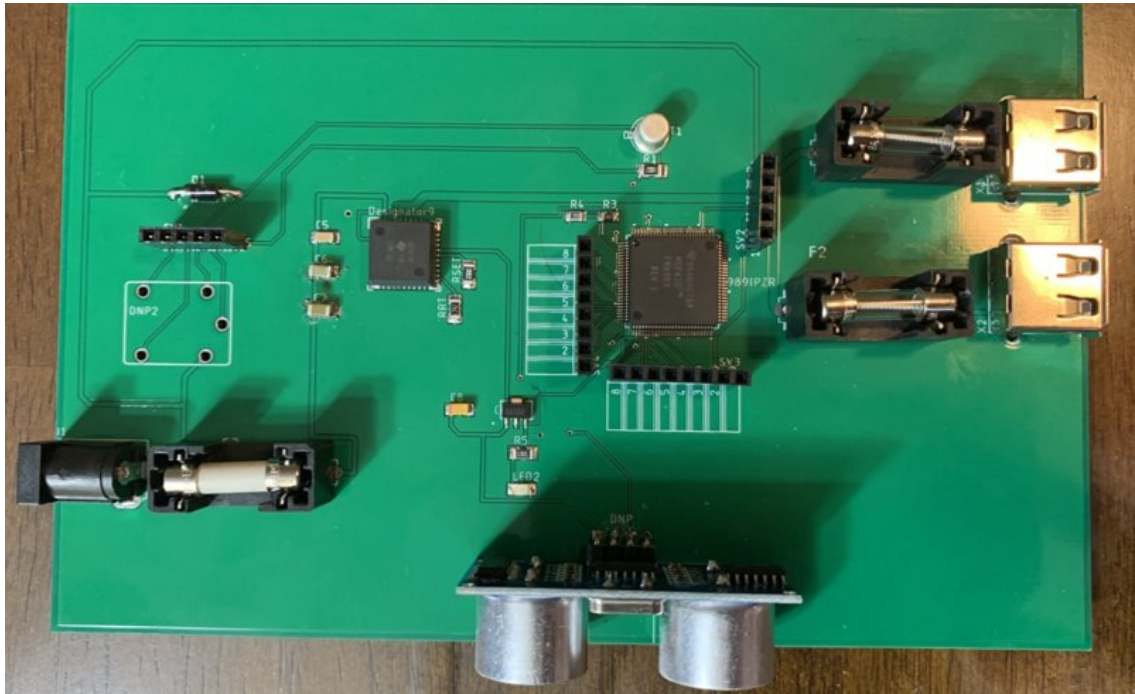


Figure[26]: Printed Circuit Board Layout

The figure shows the layout of our PCB

The physical dimensions and deployment of the copper pads are determined by the footprints of the electrical components.

For better design, the current should flow in a direct path from one section to another. In layout, the power should be supplied to each section separately through the traces of equal length. This solidifies the working of the circuit by providing each section equal voltage level. Attention and precision are required to avoid the voltage drops which can affect the circuit functionalities. A board after the deployment of the layout design is shown in figure 27.



Figure[27]: Populated PCB

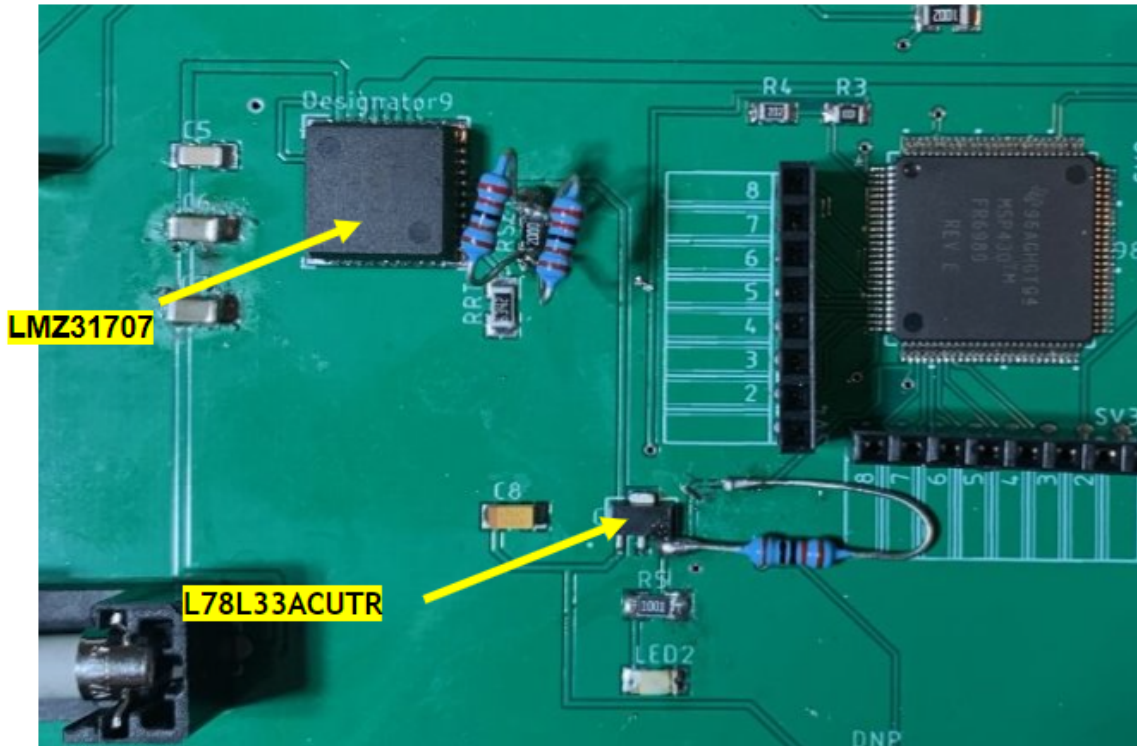
The figure shows the populated printed circuit board for our design.

### 6.3.3 Testing and Troubleshooting

After receiving the PCB we've performed some tests to ensure the proper functionality of the board, we've first started by testing the voltage levels at the different stages. The first problem that we faced was with the output of the L78L33ACUTR being 4.1V instead of 3.3V which was outside the operation range of the MSp430FR6989, namely (1.8-3.6)V.

The measured voltage of 4.1V was lowered by approximately 1 Volt using a 10k $\Omega$  resistor connected in series between the output of the buck converter and the input of the microcontroller. Figure 28 shows the reworked PCB.





Figure[28]: Reworked PCB

The figure shows the reworked PCB to get the correct voltage values.

After connecting the two Raspberry Pi's, we notice a large drop in voltage at the output of the LMZ31707. To account for that, we maximize the output voltage by adjusting the value of the resistor Rset, which controls the output voltage. This was done by soldering a two 2.2K $\Omega$  resistors in parallel to the existent 0.2K $\Omega$  to. This increased the output. Voltage to 5.7 volts in the absence of any load. However, when the PCP was used to power the Raspberry Pis, the drop in the voltage level was still high.

To emulate the loading effect a 100 $\Omega$  resistor was connected as a load to the LMZ. The measured voltage was 3.24 Volts which corresponded to a load current of 30 mA. From these results, we concluded that this buck converter (LML31707) was not operating correctly and is unable to supply the desired current, although it is rated as a 7A component in its datasheet. We could not use the PCB to power the Raspberry Pis and resorted to a breadboard solution.

Additionally, our team ran into an issue with the MSP430/lock interface. Because of lead time of components, we did not have the ability to prototype and troubleshoot all circuits prior to ordering the PCB. What this meant is that when our relay arrived and we tested it with the MSP430, the pinout was different than expected and would always be on. To overcome this issue, we used one of the female pin headers on the PCB/MSP430 and connected the 3.3V, Ground, and Signal line to a 3.3V optocoupled relay daughter board. The 12V supply from our PCB was connected in series with the lock and relay, but the ground was connected to the NO node on the relay daughter board. With this configuration, we were able to control the 12V lock from our MSP430.



## 6.4 Final Coding Plan

Considering the nature of our system, our application must be both safe and reliable. Therefore, we must avoid redundancy in our program, as well as aim to have an appropriate run time. Our system should be able to quickly authenticate authorized users and be able to quickly recognize unauthorized users. As well as being fast, the system should also be accurate. As mentioned in section 2.1. The system's facial recognition, voice recognition, and pin entry should recognize input at least 95% of the time. Ideally the system will be 100% accurate, but realistically there may be some errors.

More specifically the main components of the system will be the two raspberry pi's microprocessors and one MSP430 microcontroller. The raspberry pi's will be the "brains" of the system, and the MSP microcontroller will primarily control the door lock. Each of the devices will be connected to each other, but only one of the CPUs will connect to the MCU. The raspberry pi's will connect to each other through an ethernet cord. The raspberry pi and MSP430 will connect to each other by simple jumper cables.

The main software features of our system will be facial recognition and voice recognition. Being able to read a user's pin entry through a keypad is also important, but realistically will not take very long to program, compared to developing the code for the facial and voice recognition features. Our developers will begin the system by creating the recognition algorithms one at a time. Once both of them are working, then they will begin working on receiving pin entries.

In order to run the authentication methods at the same time, as well as in real-time, we will have to incorporate multi-threading (this is explained in detail in section 2.5.1). This operation will be resource intensive due to the fact that we will be running two machine learning algorithms at the same time. However, by utilizing another CPU, we are able to split the workload in half.

So, in order to make the multi-threading method to work, we would need four different threads. The first thread being for the facial recognition algorithm, the second thread for voice recognition, the third thread for the pin entry, and the fourth thread will act as a validation thread. The simplest way to make this work would be to divide each thread into their own class. Then create a member variable for each class, we can call it validationKey or something similar. This variable can be a boolean variable, and we can initialize it to false. If the user is authenticated in the specific class, then it will set its validation variable to true. Inside of the validation thread, there will some logic checking each of the thread's validation variables. If two out of the three classes are validated, then the validation variable of the validation thread will be set to true.

Once the validation variable in the validation thread or class is set then it is time to unlock the door. This will be done by sending a message from the raspberry pi

CPU to the MSP430 MCU. Initially we were planning on using the i2c communication protocol between each device and setting a specific register value.

After some consideration we are now planning to set a specific pin to “high”, then read the voltage of that pin. On the receiving device, if the voltage is read as high, then events will begin. This transmission method is preferable due to its simplistic nature. If we use i2c then we would have to configure each device to set up the communication protocol. After the devices are set up then we would have to program the devices to begin sending messages.

The new transmission method to deliver information between the CPU and MCU is much simpler. To send data we would just connect a wire between both devices, and then just read the voltage levels between them. Ideally to achieve this we would just need one wire between each device. However, if we would like to read the voltage between each device then we may need to incorporate another wire. These wires are extremely affordable, and each device has an abundance of pins so adding another wire is a feasible task.

In addition, we will have to use subcomponents in order to make our system work. These subcomponents are cameras, microphones, keypads, and even LEDs. They will be connected to the raspberry pi’s with the GPIO pins and USB ports. Since we will be using two raspberry pi’s to split up the workload of the system, then these devices will be connected to different CPUs. This is also beneficial because it reduces the power consumption of a single CPU. A single raspberry pi would not be able to handle all of these devices as well as preform the necessary tasks for our system.

The main portion of our system will be written in the programming language Python. Python is a very powerful language, and there are many libraries that we can use to implement our desired features. The MSP430 microcontroller does not support python, however it does support C. C is very useful for bare metal programming, which is what we will be doing with our microcontroller.

Also, we are able to control external devices that are connected to the raspberry pi with python. Essentially the only thing that would be required is to download the necessary libraries for each device, connect them to the raspberry pi, and then write a script to control them. Generally, it is a good idea to get the basic functions of the device working before writing the logic that will trigger them.

For instance, if the developer is connecting a LED strip to the raspberry pi, then they should first be able to turn on the LED, and understand how to change colors, or use the other built-in functions of the LEDs. Since both python and raspberry pi microprocessors are very popular, there are many references online. This will make it easy for the developers to trouble shoot issues if they become stuck or confused while creating the application.

Our developers will be utilizing GitHub during the development of our system. This will allow our programmers refrain from overlapping code. By creating a local git repository on the developer's machine, they are able to create a copy of the code onto their devices. A benefit of the raspberry pi is that we are able to connect it to the internet, so during development we can create an online repository directly on to the microprocessor.

As mentioned previously any changes that are made, they can be commented on and pushed to the GitHub repository. Then other developers can read the changes and continue to work on the system. This will help with reducing time reviewing the complete source code or having to continuously communicate with group members to determine changes. Also, GitHub makes it extremely convenient to have the application hosted on a website rather than a local machine. This makes it possible to basically work on the code on any machine.

Having the code backed up to an online database also acts as a safety mechanism for our system. If our whole application is stored locally onto the raspberry pi, and the device is fried, then we would lose all of our work. However, if we back our program up to cloud, then we would just have to get another CPU, download the libraries, and then download our program from GitHub. By practicing proper coding standards, we will be able to develop a system that is not only functional, but also professional. Using a naming convention is also a simple but effective method to maintain proper coding standards. By implementing a specific naming convention, our system will have easier code readability. As well as being able to settle arguments for naming variables or other potential syntax debates, and it even allows developers to primarily focus on bugs when performing code reviews. The naming convention we will follow will be camel case. Different naming conventions, as well as examples are shown in section 2.5.1.

Our developers will perform frequent code reviews to ensure that any update to the system are explained. This will allow for additional evaluation of individual developer's code. By having multiple people look at a single task, then alternative solutions may be offered. These alternative solutions may even be better than the original, thus making our system more efficient.

Again, some of our main goals of the system are reliability and efficiency. Since our device will be running on limited resources, such as a microprocessor compared to a standard computer. We have to make sure that the system is not wasting resources that can be utilized for an alternative task. We will split the more resource intensive tasks onto the more powerful devices and put the less resource intensive tasks on the less powerful devices. Since we are still currently in the research phase of our system, any design decisions, or ideas that we have mentioned may change. This is due to the fact that issues are likely to arise while developing our system. These potential issues are unknown to us at the moment and will only be discovered during the developmental phase. Some examples of issues that come up would be with the way that we plan on implementing the authentication methods. Since we will be using another CPU to control part of the

code, then we may run into difficulty with running a multithreaded application as well as using a TCP communication protocol. However, any changes to our initial design will be documented, and the justification will be provided stating why we decided to take a different approach.

## **7. Project Prototype Testing Plan**

### **7.1 Hardware Test Environment**

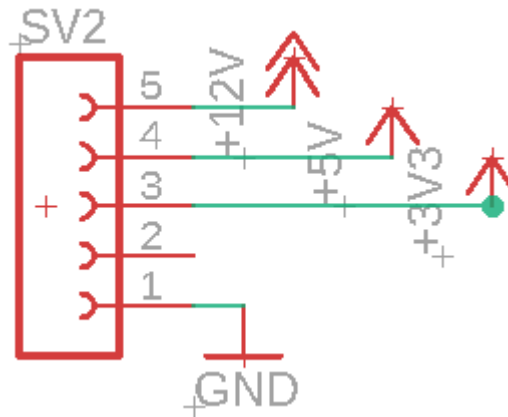
Our hardware will be tested using the Digilent Analog Discovery 2 kit. This includes a multimeter, logic analyzer, and oscilloscope, as well as breadboards and leads. The Digilent Analog Discovery 2 also operates as a waveform generator and power supply. Last, the Analog Discovery 2 has a spectrum, network, and impedance analyzer, which will help us troubleshoot signals. We will be able to use this kit to check voltage levels and signals on our breadboard by using female header pins to plug into. The majority of testing will involve the oscilloscope and multimeter. Future revisions of the board will likely include more test points to enable us to better check for root cause of failures. A key design of our breadboard is the testability, as we use a large number of female header pins as test points.

Because the majority of our components are SMT, we are unable to conduct extensive breadboard testing prior to ordering the PCB. We are attempting to get around this obstacle by ordering a Revision 1 board in the near future, that we can use to verify functionality before ordering our Final Revision of the board.

Due to the high number of SMT components, we elected to include several female pin headers on our PCB. This will ensure that we are able to troubleshoot if necessary, using oscilloscope and multimeter probes.

One of the major concerns in our project is communication between the Raspberry Pis and Microcontroller. If the door fails to unlock, we will probe the terminals of the PCB to check if the Raspberry Pi is behaving appropriately to the test condition. We also have a female pin header with all of the different voltage levels used in our project as a first point of design verification, shown in Figure[24]. We will test

to see if there is significant ripple in our voltages, as well as making sure that they remain within tolerances of the sensors, MSP430, and Raspberry Pi when under load.



Figure[29]: Test Points for Voltage Converters

We provide several test points on our board by using female pin headers, and we will be able to probe the signals and voltages corresponding to these in order to check if the devices are working as intended. Most MSP430FR6989 pins in use will have a corresponding female pin header so that in the event of a testing failure, we can dig deeper into the root cause and determine appropriate corrective action. Most problems in the PCB are anticipated to be found in Revision 1 of the board, but if we will retain the pin headers in future revisions just in case issues arise later in the project.

Due to our device being indoors during standard operations, we will not be performing environmental testing in relation to system operation. Humidity, barometric pressure, and temperature will not be factors in the operation of our system. We are assuming a that the project will be in room temperature under standard humidity for the entire operation. We may revisit this topic if heat dissipation is found to be an issue in our design, as it uses a large amount of power.

## 7.2 Hardware Specific Testing

A potential issue that could cause a redesign is if the camera, speaker, and microphone draw more current than expected. If they draw more than the specified 2.5 watts, we may be unable to provide sufficient current at the 5V level to power everything. We will perform tests to see the power consumption when the Raspberry Pis are running facial and voice recognition. We do not believe that the keypad or small LCD screen will have significant power requirements.

We will also check the power consumption of the electric strike lock when it is activated. Our design provides plenty of 12V power, but again due to the

modularity of our project, we would like to ensure that there is room for additional growth. The majority of electric strike locks are manufactured without detailed datasheets, as they are intended for wiring directly into the grid power. Our electric strike lock is powered by the System Controller/Power Distribution PCB, so we need to ensure that it is capable of providing sufficient power. Most commercially available electric strike locks are made for commercial and not engineering use, so we may discover additional testing requirements at a later time.

Also, we will need to test the relay design to ensure that the 1N4004 diode is capable of protecting against back-emf. We believe that it will be adequate because this particular diode is supplied by the manufacturer of the electric strike lock that we chose. We can accomplish this by turning the transistors on and off and watching the voltage in the oscilloscope. Because we do not have the capability of breadboard testing for this project, we are disadvantaged in that we do not have a quick iterative process for figuring out the ideal way to switch the lock on and off. We will do extensive testing with the oscilloscope and voltmeter to verify that our design is capable of switching the 12V lock on and off without damaging the microcontroller.

Depending on which MSP430 pins we decide to use and what problems we face in design verification, we may decide to test additional signals on the PCB by probing necessary pin headers. One of the design rules for revision one of the PCB was to include as many pin headers as possible to aid in troubleshooting errors. For every sensor or module that we include, we will provide additional female pin headers to probe during the testing and troubleshooting process.

## **7.3 Software Test Environment**

The software will first be tested in an IDE before we program the MSP430 and the Raspberry Pi devices. Since the MSP430FR6989 microcontroller and the two Raspberry Pi's are going to be running separately it is not a problem to test that each of the device's programmed software is working correctly, separately, as well. Once we can verify that all of the specific pieces of software function correctly on their own, we can then move to integrating and interfacing them together to communicate data back and forth with each other.

The MSP430 development environment that will be used is the Code Composer Studio platform. The Raspberry Pi development environment will be a combination of a simple text editor such as Microsoft's Visual Studio Code, as well as the Thony Python IDE that comes standard on the Raspbian OS for the Pi.

## **7.4 Software Specific Testing**

Ideally, if this were a product that was being developed by a company in the field, a Software Test Plan would be generated for coverage of the software portion of development. Whether it was for a Research and Development program effort, or a well-established production program, a Software Test Plan would be a necessary document that would outline and define exactly how and what would be tested at the software level. This section will be a concise version of what you would typically see highlighted in a Software Test Plan document.

Thankfully, the scope of our software testing is not extremely broad. Since our system is not implementing anything from scratch, for example complicated math computations or encryption algorithms, we will not need to supply a huge number of test cases and then check the result against those test vectors. Generally, if you were to develop your own implementation of an encryption algorithm or an advanced mathematical function from scratch, you would need to generate many test vectors as different test cases and supply them as input to the system, then verify the outputs of those test cases match the expected output, hence verifying the computation was correct.

For our project, there are very specific things that we must verify are working correctly, especially the facial and voice recognition portion of the software. Without these two main features of our system's software working correctly, our entire system becomes compromised. The system relies on being able to leverage these two features for the most secure and accessible system design, and if we do not get these two features to be bug-free, then our system will be majorly flawed.

We also must test that the communication between devices is correctly being transmitted and received. Without the correct signal pulses and data being transmitted or received correctly, the system will not work as intended. There must be a good amount of testing done to ensure that all communication between devices is implemented and working correctly, otherwise the main features that were previously mentioned are irrelevant. That will be an easy step to verify though as once we have devices connected together, we can create test cases and make sure all communication is correctly implemented, compared to the known values supplied during the test cases. We also must test that the algorithm that is implemented on the MSP430FR6989 microcontroller is correctly handling the authentication values and the correct logic is implemented so that when there are two out of the three authentication methods that are confirmed, we unlock the door. Finally, there may be a few minor implementation details that will be needed to be tested as our project develops, such as when stretch goals are pursued.

The testing methodology is going to be really straight forward for our group. Essentially, we just need to test the following five features, in the following ways

as

described

below:

1) Facial Recognition Testing Method:

This can be easily tested after we have implemented the python code. All that needs to be done is to add users that we would like to be included as authorized users. To do this we must add an image, or multiple images for better performance, of the user who we want to be added to the directory that will store all images of users to be trained. Each image must be correlated to the user that we are trying to add, so that the model knows during training who that image is for. We can add multiple users or even as many as we would like to that directory. The only drawback to have a very large number of users that we would like to add, is that training time for that set of users is going to be much longer than if we were to train the model with one user.

After training is complete, our facial recognition model is ready to be deployed for testing. To do this we just run the python program on the Raspberry Pi, and assuming we already have the correct peripherals set up, specifically the USB webcam, the program should start. When the program starts a window pops up showing us a visual of what the webcam sees. Whatever is within the field of view of the webcam will be constantly checked for any faces that the model was trained on. If a user's face is recognized by the model, the program will draw a box around the user's face and print that user's name to show that it recognizes that person.

Testing this implementation is going to be very arbitrary then, as we need to train the model with different images of users that we would like to recognize, and then run the program and check that the user is identified correctly. If we can verify that the facial recognition implementation correctly identifies authorized users, does not falsely identify users (meaning the program says one person's face is actually a different person's), and shows unknown for the users who are not added to the group of authorized persons, we can then say that our facial recognition is working correctly and is ready for integration into the system.

2) Voice Recognition Testing Method:

Testing of the voice recognition implementation is going to be very similar to our test method for facial recognition, as it has been described above. Essentially, we will add voice recording clips, (the total size and length of these clips, as well as what it is the users should be saying in the clips, is still under investigation at this time), to a directory that will hold the recordings of each user that we want to train our model to recognize. Each clip must be correlated to the person that we are adding so that the model knows which clips are for which person during training. Just as we



mentioned for facial recognition, we can add as many users as we desire to be trained, but the drawback, again, is that the training time increases tremendously if we were to try to train the model on a huge number of users. Precise mathematical calculations can be done to determine training time based on the number of users the model is trying to train, but there are also many variables to this problem such as the computing environment that is doing the training, so we leave the calculations for training time out of the scope of this paper.

After training is complete for the voice recognition model, we are ready to deploy it for testing. To do this we just run the python program implementation of our voice recognition on the Raspberry Pi, and assuming we already have the correct peripherals set up, specifically the microphone (this is a component we will need to test to make sure that the built-in microphone of the USB webcam is going to be sufficient for the voice recognition implementation. There may be a need to switch to using a stand-alone microphone instead of the built-in mic on the webcam, since the stand-alone microphone might provide the system with a higher quality audio input – which is crucial to the success of the voice recognition implementation. If the audio input is not clear then the recognition will be extremely difficult to get working correctly, so the discussion and testing of using this built-in microphone is still underway and a path forward will be determined soon). Once the microphone is correctly set up, we can deploy and start the program for testing. The actual details of the process and what happens during deployment is still under investigation. We will update this section as necessary once we determine how deployment works and how the voice recognition is run. Getting the voice recognition implemented is our next major milestone task, and the highest priority for our group at this time.

Testing of the voice recognition implementation is going to be straight forward; just as facial recognition testing was. We need to train the model with different voices of users that we would like to recognize, and then run the program and check that the user's voice is identified correctly. If we can verify that the voice recognition implementation correctly identifies authorized users, does not falsely identify users (meaning the program says one person's voice is actually a different person's), and shows unknown for the users who are not added to the group of authorized persons, we can then say that our voice recognition is working correctly and is ready for integration into the rest of the system.

### 3) Keypad Testing Method:

To test the keypad for user pin number entry, we will simply need to make sure that the keypad is correctly connected to the Raspberry Pi through

GPIO pins. Then verify that the 8 GPIO pins that are being used are correctly defined in the python program, because if they are not, it is possible to damage the Raspberry Pi's I/O pads.

The 4 pins that correspond to the rows of the keypad need to be set as output pins in the python program. The 4 pins that correspond to the columns of the keypad need to be set to input pins in the python program. All the GPIO pin numbers need to be defined correctly in the python program or it will not work correctly. For the input pins, we also must make sure to set them up with pull-down resistors in our program, which will take that line and pull it down to ground (or logic Low/0) whenever the line is left floating. That means the line will be pulled to 0, unless the switch that is connected within the keypad is pressed, which in that case would pull the GPIO line high for whatever keypad number is pressed. This cross layered 4x4 structure of rows and columns is easily decoded to be able to determine what keypad number is pressed. For example, if row 1 (from the top) and column 1 (from the left) is sensed as being pressed, we can decode that to be the digit "1" on the keypad.

To fully test the functionality of this keypad implementation in our python program we can simply use the program's standard I/O feature to print the values of any key pressed to the screen. That way we can run the program and simply press each number on the keypad, and check that the correct number was sensed by our code. After verifying that each digit is correctly being read, by comparing what is printed to the screen and what is pressed: we can confirm that our implementation is correct and are ready to begin integrating this keypad into the rest of the system.

#### 4) LCD Screen [stretch goal] Testing Method:

This is just a stretch goal for our team currently, but considering we have high confidence to be able to meet all of our other design requirements, we are including this feature in the test plan. The LCD screen is a very simple feature to implement, especially because we are using the Raspberry Pi and programming the code in python. Python has very useful built-in packages that makes the programming easy.

The LCD screen is connected to the Raspberry Pi through the I2C communication interface. Similar to the keypad, we must first ensure the electrical connection between the LCD screen and the Pi is correct. Vcc and GND for the LCD screen just connect to any 5V and GND pins on the Pi's GPIO header. SDA and SCL are then connected to the GPIO pins 2 and 3, respectively. Once the I2C connection is complete we can test the software implementation.

To test the LCD software is working correctly, we simply wrote some test code to print different values, strings, characters, and special characters like a comma and exclamation point to the screen. We were able to verify that the LCD screen was correctly printing those different characters and strings, as well as clearing the screen as it was programmed to do so in our code.

Even though our team got this LCD screen implementation completed and verified during the Senior Design 1 semester, we are putting this on the bottom of our list of things to do. We are only going to include this feature as a stretch goal, if we are able to complete and verify the functionality of the rest of our more important system features first. Assuming we are able to correctly implement and verify the functionality of our system's main features, we will explore integrating the LCD screen into our project.

#### 5) Sonar Sensor Testing Method:

The sonar sensor is going to be a very important piece of our system as the thought process behind using this sensor is to try to save resources and power whenever there is not someone directly in front of the door waiting to be authorized. Because the facial and the voice recognition algorithms are extremely complex and computationally heavy, we thought it would be best to not run those unnecessarily. That means only when the sonar sensor senses that someone is within one meter from the door, we will then start to run those complex and power intensive functions.

To test that the sonar sensor is correctly implemented in software, we first need to double check that the electrical connection between the sonar sensor and the MSP430FR6989 is correct. We would not want our software to be implemented correctly, but we are led to believe otherwise because we accidentally connected the sensor incorrectly. That would potentially lead to a massive amount of debug time that is completely unnecessary. It is also possible to damage the sensor or the I/O pads of the MSP430FR6989 if the electrical connection is not correct. Specifically, there needs to be a voltage divider circuit implemented on the "Echo" pin of the sonar sensor that is an input to the MSP430FR6989 GPIO pin. If you don't use a voltage divider circuit to step down the 5V to 3.3V, the MSP430FR6989 pad will be ruined. This circuit was initially designed in Junior Design and is detailed in the hardware and schematic explanation section of this document. After checking the electrical connection is correct, we can then test the software is implemented correct.

Finally, for checking the C code implements the sensor correctly and that the values of the sonar sensor are being printed out, the simplest and most trivial thing to do would be to print the values to the terminal using standard

I/O in C. This allows us to not have to set up any communication interfaces to print the values to an LCD screen, for example. All that we must do is run the program and verify that the sonar sensor is printing its values to the screen. If we would like to check relative accuracy of the sensor, even though our system does not require an extremely precise reading from the sonar sensor, its easy to move objects closer and further from the sensor, then verify the readings are moving, relatively, in the correct direction. This test has been run successfully already by our team. We saw that as you move your hand closer to the sonar sensor the values drop to a low number in centimeters, and as you move your hand and body farther away from the sensor, it was correctly printing out a value of meters. This allows us to assume that the sonar sensor was correctly implemented and is ready to be integrated into the rest of the system.

6) Electric Strike Lock Testing Method:

Thankfully, the Electric Strike Lock is a very simple and easy to use piece of our project, that will require very minimal functional testing and only a few lines of code to implement on the MSP430FR6989. The Electric Strike Lock only has 2 wires: Power and Ground. We need to connect these wires correctly to the MSP430FR6989 (although, it is important to note that this connection of the power wire is not directly connected to the MSP430FR6989, but through a special relay circuit that was developed due to the fact that the Electric Strike Lock needs 12V power and the MSP430FR6989 cannot provide 12V output through its GPIO pins).

The lock that we chose has the ability to operate in both normally open and normally closed modes, corresponding to fail safe and fail secure. We designed the PCB such that we have female pin headers for both modes on the relay, so that we can change the mode by moving wires if we discover a problem in our implementation. The electric strike lock mode can be changed from fail safe (NO) to fail secure (NC) by adjusting the position of a screw on the physical lock.

Once the connection is correctly set up, we can write a simple code to test that the Electric Strike Lock works as it is intended to. The Electric Strike Lock needs to be connected in an active low fashion, meaning that the Lock will be receiving power (logic High or 1) constantly, and when we would like to unlock the door, we will pull the power line Low or to 0. This is called active low since the action of driving the line low, or to 0, is what causes something to happen – in our case the door will unlock. We will also include a counter mechanism that will start counting and hold the door unlocked for a set amount of time, for example 30 seconds. That way when the homeowner is identified and authenticated, we can drive the lock signal low which will unlock the door, and then hold the door unlocked for that set

amount of time to allow them to open the door and walk in. We need to develop a similar method to use the button/switch to hold the door unlocked for a set number of seconds.

The testing of this implementation is going to require us to have the lock in hand, which we do not yet have. Once we get the lock, we can hook it up to the MSP430FR6989 and run our code, with a random set of unlock pulses being sent to the lock. Then we can verify against the random sequence, that the software for our lock implementation is correct. Finally, when the lock is functionally verified, we can move forward with integrating into the full system, and attach it to the PCB.

The Software Test Environment has already been described in the previous section (7.3 Software Test Environment) but is important to keep in mind during our testing process, as the environment for testing determines the limits of what we can or cannot test.

Overall, as you can see, we are taking a specifically designed, broken down approach, to testing all of the features that we plan to implement in our system and eventually integrate together. Once we have successfully verified that each piece of our system works on its own, we are going to move forward with combining features and integrating them together. As we move forward with integrate different features together, we will use a continuous integration method to ensure that we are still testing and verifying functionality along the way. This will allow us to be sure that each feature we integrate within the system does not somehow break the rest of the system's functionality. Continuous integration is a very important method of execution and is widely adopted in the field, so we believe that it will be crucial to the assistance of our project's success as well.

## 7.5 Test Results

Once the system was fully integrated, we ran into some issues that we did not anticipate. Most issues were related to the communication method that we chose. Setting specific GPIO pins between devices to signal an event turned out to be an unreliable method. Sometimes the GPIO pins on the raspberry pi would read as high or low when it shouldn't have, which would trigger events at unwanted times Such as the MSP430 waking up the system, or the pi-to-pi communication sending faulty signals.

The facial recognition algorithm would sometimes recognize an unauthorized user as an authorized one. This could be due to the quality of the camera, lighting, as well as the number of images we chose to train the model with. So, by getting a better camera, improving the lighting, and using more images to train the model we can improve the accuracy of the facial recognition algorithm.

However, the accuracy of the facial recognition algorithm after testing came out to be 72%.

The Speech to Text method was very reliable. Although rarely the microphone would detect unwanted words. Which would result in the system comparing each word to stored pins, and once the system determines that the pin is invalid it would request the user to enter another pin. If the user is also saying pins without waiting for the system to catch up, the system may crash or take a while to catch up. To fix this we could create some logic which parsed through the input string of the user and looks for specific integer values.

However, this issue will arise regardless due to the nature of the system. For instance, if someone is approaching the door and talking on the phone or with someone next to them then the system will pick up those words. The accuracy of the system could also be improved by selecting a higher quality microphone. Overall, the accuracy of the Speech to Text Algorithm resulted in 76%.

The keypad entry was very accurate, and the system read the users pin entry correctly input every time. In conclusion for the door being a prototype, the accuracy of the entire system after 100 tests came out to be around 75%. To improve the accuracy of this system we would implement some other form of communication. However, we noticed that by turning off the devices it would reset the GPIO pins to low. All in all, given the circumstances, the system was very accurate. Where the majority of times authorized users are granted access, and unauthorized users were denied access.

## 8. Administrative Content

### 8.1 Budget and Finance Discussion

Currently our group does not have any funding from a sponsor, we will be funding the project out of our own pockets. Below shows a preliminary/rough estimate of our budget that will be fluid throughout the first semester during Senior Design 1. We tried to overestimate costs so that we can come in under budget. Table 21 shows the cost breakdown of our major expenses for this project.

| <b>Equipment</b>                    | <b>Cost</b> |
|-------------------------------------|-------------|
| <b>MCU's (possibly more than 1)</b> | \$200       |
| <b>Camera</b>                       | \$30        |
| <b>Keypad</b>                       | \$20        |

|   |                                 |
|---|---------------------------------|
| <b>Broken “August” Lock</b>                   | \$40                            |
| <b>Custom/Populated Circuit Card Assembly</b> | \$50                            |
| <b>General Jumper Wires for Peripherals</b>   | \$20                            |
| <b>Speaker</b>                                | \$30                            |
| <b>Microphone (x2)</b>                        | \$50                            |
| <b>LCD Screen</b>                             | \$11                            |
| <b>LED Strip Light</b>                        | \$20                            |
| <b>Total:</b>                                 | <b>\$471 -<br/>\$118/member</b> |

Table 22: Cost Breakdown of Major System Components

## 8.2 Project Milestone Discussion

The milestone section is divided into two parts, the first part is to be concluded in senior design 1 (Spring 2021), and the second part is to be continued in senior design 2 (Summer 2021). In order to meet the deadlines for our Smart Door Security System, milestones were set at the beginning of the semester as shown in the two tables below. According to the class syllabus, specific dates were set to ensure our project design gets finished on time. Since each group member is required to write at least 30 pages, setting deadlines can help us keep track of our progress and can be used as a guideline for weekly goals to be met. We will meet regularly as a group to discuss concerns and ensure that adequate progress has been made by all individually. This also ensures that we can discuss design decisions early and often. Table 22 shows the schedule that we will be following in this project.

| Number                 | Task                              | Duration        |
|------------------------|-----------------------------------|-----------------|
| <b>Senior Design 1</b> |                                   |                 |
| 1                      | Creating Groups                   | Jan 11 – Jan 17 |
| 2                      | Brainstorming                     | Jan 18- Jan 22  |
| 3                      | Initial Doc- Divide and Conquer 1 | Jan 23- Jan 29  |
| 4                      | Research & Documentation          | Jan 30- Feb 6   |

|                        |                                   |                 |
|------------------------|-----------------------------------|-----------------|
| 5                      | Initial Doc- Divide and Conquer 2 | Feb 7 -Feb 12   |
| 7                      | 60 pages Draft                    | Due Apr 2       |
| 8                      | 100 pages Draft                   | Due Apr 16      |
| 9                      | Final Document                    | Due Apr 27      |
| <b>Senior Design 2</b> |                                   |                 |
| 10                     | Build Prototype                   | May 21-June 4   |
| 11                     | Test & Redesign                   | July 10-July 24 |
| 12                     | CDR Presentation                  | June 25         |
| 13                     | Final Presentation                | July 25         |
| 14                     | Final Report                      | July 30         |

Table 23: Schedule for our Project

Because we are currently enrolled in Senior Design 1, the exact milestone due dates for Senior Design 2 are unknown at this time. This table will be updated once the deadlines and requirements for Senior Design 2 are announced. We will be updating the table as the assignments are released, and we will aim to complete them early, as we have done during Senior Design 1.

We met project milestones 7 and 8 without major issues, and we are on track to complete milestone 9's 120 pages for the Final Document by the indicated deadline. The next project milestone that we have to worry about is getting the Revision 1 of the System Controller/Power Distribution PCB sent to the board house for fabrication and assembly. Our goal is to reach this milestone within the first week of senior design 2, if not sooner. This is a major milestone because it will allow us to proceed with troubleshooting to finding design errors and potential issues, as well as a functional verification of our design so far. As soon as we receive and perform design verification on Revision 1 of the PCB, we will be poised for success in this project and be much more comfortable moving forward.

The most important milestone remaining after completing the first revision of the PCB is testing & redesigning, as necessary. We will need to be able to quickly and accurately test and troubleshoot the board and fix bugs, since it may have a long lead time. We are not particularly concerned with meeting our software design deadlines, as we can immediately test for and fix bugs, unlike the hardware. The final report in Senior Design 2 should be easily completed by our team, because we laid solid groundwork in the report for Senior Design 1. \



## 9. Project Summary

In conclusion, as of right now our project will include the following components two microprocessors, a custom circuit card assembly (CCA), keypad, camera, speaker, an electronic strike lock as the locking device, LED lights, and a sonar sensor. Specifications are still being determined but will be finalized as we begin to make developmental progress with our system.

The single board computers that we will be using to run our complex facial recognition and voice recognition algorithms will be the raspberry pi 4. This microprocessor has the perfect amount of processor power that our project will need to satisfy our system requirements. Also, the price for this device is extremely reasonable, which is a benefit in case we need to purchase multiple.

The microcontroller that will control our locking mechanism, as well as the sonar sensor will be the Texas Instruments MSP430FR6989. This device is also affordable and can handle complex operations. It will read data from the raspberry pi and perform the decision-making operations, such as determining how many methods have been authenticated and whether the door is open or closed. Such as when the user is authenticated from the raspberry pi, a flag will be raised. The MSP430 will read this flag and unlock the door. The microcontroller will also be able to relay relevant data back to the single-board computers. Using flags is the chosen method for reducing complexity, as we will not have to implement i2c communication in different programming language levels.

The main software features will be the authentication methods which include facial recognition, voice recognition, and the pin number validation. Once two out of the three methods are validated, or if the master pin is entered then the door will unlock. By implementing LED lights, the user can have a visual representation of the current door status. The total price for our project as of right now comes out to \$440, which is \$110 per group member. However, this price may change as details are finalized. All updates to our system will be documented.

## 10 Appendices

The appendices section lists the appendices that reference the media used to support and develop the smart lock project.

### 10.0 Appendix A: Bibliography

This section outlines the members of our team along with a brief profile of their future endeavors.

#### The Engineers



Adam Stefanik will graduate with his Bachelor's degree in Electrical Engineering and a minor in Mathematics. He will be attending the University of Florida for his Master's Degree in Electrical and Computer Engineering starting in the Fall of 2021. He is interested in Analog Circuits and Optics and will be working as an engineer at a defense contractor after his UCF graduation.



Moissess Rodriguez will graduate with his Bachelor's degree in Computer Engineering. He will pursue his Master's degree in Electrical and Computer Engineering at the University of Florida beginning in Fall 2021. His area of study will be in Artificial Intelligence and will be working full time as a software engineer for Northrop Grumman after graduating from UCF.



Reham Hammad is a senior graduating in August 2021 with a bachelor's degree in Electrical Engineering. She got accepted in the Electrical Engineering master program at the university of Central Florida and she is looking to pursue a carrier in the RF field.



Zach Jackovich will graduate with his Bachelor's Degree in Computer Engineering and is interested in Hardware & Systems Security. He will begin his Masters Degree in Electrical and Computer Engineering from the University of Florida in the Fall of 2021. He has also accepted an offer to work for Lockheed Martin as an ASIC & FPGA Design Engineer after graduation from UCF.



## 10.2 Appendix B: Data Sheets

[1] DC Mini-Motors M22E-13 Series,

[https://www.mitsumi.co.jp/latest/Catalog/pdf/motorav\\_m22e\\_13\\_e.pdf](https://www.mitsumi.co.jp/latest/Catalog/pdf/motorav_m22e_13_e.pdf)

[2] DRV8833 Dual H-Bridge Motor Driver

<https://www.ti.com/lit/ds/symlink/drv8833.pdf?ts=1616747032491>

[3] Toshiba Bi-CD Integrated Circuit Silicon Monolithic T B 6 6 1 2 F N G

<https://www.sparkfun.com/datasheets/Robotics/TB6612FNG.pdf>

[4] DRV8836 Dual Low-Voltage H-Bridge IC

<https://www.ti.com/lit/ds/symlink/drv8836.pdf?ts=1616770492964>

[5] Motor control driver chip

<https://www.elecrow.com/download/datasheet-l9110.pdf>

[6] Accelerometer ADXL337

<https://www.analog.com/media/en/technical-documentation/datasheets/ADXL337.pdf>

[7] Accelerometer ADXL377

<https://www.mouser.com/datasheet/2/783/BST-BMA400-DS000-1509606.pdf>

[8] BMA400 Accelerometer

<https://www.mouser.com/datasheet/2/783/BST-BMA400-DS000-1509606.pdf>

[9] MMA8450Q Accelerometer

<https://www.nxp.com/docs/en/fact-sheet/MMA8450QFS.pdf>

[10] LMZ31707

[https://www.ti.com/lit/ds/symlink/lmz31707.pdf?ts=1627611992208&ref\\_url=https%253A%252F%252Fwww.google.com%252F](https://www.ti.com/lit/ds/symlink/lmz31707.pdf?ts=1627611992208&ref_url=https%253A%252F%252Fwww.google.com%252F)

[11] L78L33ACUTR

<https://pdf1.alldatasheet.com/datasheet-pdf/view/100655/STMICROELECTRONICS/L78L33ACUTR.html>

### 10.3 Appendix C: References

- [1] [Online], US:E - Camera Equipped Smart Lock with Facial Recognition, <https://www.kickstarter.com/projects/1624790698/us-e-camera-equipped-smart-lock-with-facial-recogn>
- [2] [Online], Best Smart Lock with Camera, <https://smartlocksquide.com/best-smart-lock-with-camera/>
- [3] [Online], FL1000, <https://www.zkteco.me/product-details/fl1000>
- [4] [Online], All About Motion Sensors, <https://www.thomasnet.com/articles/instruments-controls/all-about-motion-sensors/>
- [5] [Online], ZM100 FACE AND FINGERPRINT BIOMETRIC SMARTLOCK – COPPER, <https://www.ojismart.com/product/zm100-face-fingerprint-lock-copper/>
- [6] UCF ECE Senior Design Website, <http://www.ece.ucf.edu/seniordesign/projects.php> [6] [Online], UCF ECE Senior Design Website, <http://www.ece.ucf.edu/seniordesign/projects.php>
- [7] All About Motion Sensors, <https://www.thomasnet.com/articles/instruments-controls/all-about-motion-sensors/>
- [8] What is the Best Motion Sensor for Home Security?, <https://www.frontpointsecurity.com/blog/best-motion-sensor-for-home-security#:~:text=Unlike%20active%20sensors%2C%20which%20detect,signals%20radiated%20from%20living%20beings.>
- [9] [Online], WiFi vs. Bluetooth: Wireless Electronics Basics, <https://www.autodesk.com/products/eagle/blog/wifi-vs-bluetooth-wireless-electronics-basics/>
- [10] [Online], What's the Difference between Bluetooth and Wi-Fi, <https://www.britannica.com/story/whats-the-difference-between-bluetooth-and-wi-fi>

[11][Online], What is the Best Motion Sensor for Home Security?, <https://www.frontpointsecurity.com/blog/best-motion-sensor-for-home-security#:~:text=Unlike%20active%20sensors%2C%20which%20detect,signals%20radiated%20from%20living%20beings.>

[12][Online], types of microcontrollers , <https://www.electronics-lab.com/top-10-popular-microcontrollers-among-makers/>

[13][Online], Raspberry Pi, [https://en.wikipedia.org/wiki/Raspberry\\_Pi#Processor](https://en.wikipedia.org/wiki/Raspberry_Pi#Processor)

[14][Online], Understanding Relays and Wiring Diagrams, [https://www.swe-check.com.au/editorials/understanding\\_relays.php#:~:text=A%20relay%20is%20an%20electrically,when%20the%20coil%20is%20activated.](https://www.swe-check.com.au/editorials/understanding_relays.php#:~:text=A%20relay%20is%20an%20electrically,when%20the%20coil%20is%20activated.)

[15][Online], Webench Power Designer, <https://webench.ti.com/power-designer/switching-regulator?powerSupply=0>

[16][Online], MSP430 Programming with the JTAG Interface, <https://www.ti.com/lit/ug/slau320ai/slau320ai.pdf?ts=1618507167039>

[17][Online], NSPE Cde of Ethics for Engineers, [https://www.nspe.org/resources/ethics/code-ethics?gclid=EAlaIQobChMliIC\\_5oX-7wIVAx6tBh3PTq80EAAYASAAEqJWDvD\\_BwE](https://www.nspe.org/resources/ethics/code-ethics?gclid=EAlaIQobChMliIC_5oX-7wIVAx6tBh3PTq80EAAYASAAEqJWDvD_BwE)

[18][Online], IPC Standards, <https://amp.blog.shops-net.com/10463746/1/ipc-electronics.html>

Printed Circuit Board Terminology, <https://www.pcbcart.com/article/content/glossary-of-terms.html>

[19][Online] , Understanding the UART, <https://www.embedded.com/understanding-the-uart/>

[20][Online] , I2C bus protocol interface, <https://aticleworld.com/i2c-bus-protocol-and-interface/>

[21][Online] , HC-SR04 Ultrasonic Sensor, <https://lastminuteengineers.com/arduino-sr04-ultrasonic-sensor-tutorial/>

[22][Online], Introduction to I<sup>2</sup>C and SPI protocols, <https://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/>

[23][Online] , Face recognition using Artificial Intelligence, <https://www.geeksforgoeks.org/face-recognition-using-artificial-intelligence/>

[24][Online] Available: <https://opencv.org/intel/>

[25][Online], What is Computer Vision?

<https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/overview>

[26][Online], Amazon Rekognition, <https://aws.amazon.com/rekognition/?blog-cards.sort-by=item.additionalFields.createdDate&blog-cards.sort-order=desc>

[27][Online] , August Wi-Fi Smart Lock, <https://august.com/products/august-wifi-smart-lock>

[28][Online] , H bridge motor driver circuit,

<https://www.circuitstoday.com/h-bridge-motor-driver-circuit>

[29][Online] , Design of a buck converter, <https://components101.com/articles/buck-converter-basics-working-design-and-operation>

[30][Online] , Difference Between Voice Recognition and Speech Recognition

<https://www.totalvoicetech.com/difference-between-voice-recognition-and-speech-recognition/>

[31][Online] , Difference Between Speech Recognition and Voice Recognition, <https://aurayasystems.com/2019/07/26/difference-between-speech-recognition-and-voice-recognition/>

[32][Online] , Why don't flyback diodes in H-bridge damage power supply? ,

<https://electronics.stackexchange.com/questions/386798/why-dont-flyback-diodes-in-h-bridge-damage-power-supply>

[33][Online],Key Difference Between Speech and Voice recognition

[https://www.streetdirectory.com/travel\\_guide/139545/technology/key\\_differences\\_between\\_speech\\_recognition\\_and\\_voice\\_recognition.html](https://www.streetdirectory.com/travel_guide/139545/technology/key_differences_between_speech_recognition_and_voice_recognition.html)

[34][Online], Key Differences Between Speech Recognition and Voice Recognition

[https://www.streetdirectory.com/travel\\_guide/139545/technology/key\\_differences\\_between\\_speech\\_recognition\\_and\\_voice\\_recognition.html](https://www.streetdirectory.com/travel_guide/139545/technology/key_differences_between_speech_recognition_and_voice_recognition.html)

[35] Artificial Intelligence vs. Machine Learning vs. Deep Learning,

<https://towardsdatascience.com/artificial-intelligence-vs-machine-learning-vs-deep-learning-2210ba8cc4ac>

[36] LED – Light Emitting Diode, <https://www.electronicshub.org/led-light-emitting-diode/>

[37] LED Strip Guide, <https://www.ledsupply.com/blog/ultimate-guide-on-buying-led-strip-lights/>

[38] What are the different types of LED strip light? ,

<https://www.elstarled.com/different-types-of-led-strip-lights/>



[39] Explaining Different Types of Transformers, <https://www.galco.com/comp/prod/trnsfmrs.htm#:~:text=There%20are%20four%20primary%20parts,is%20connected%20at%20this%20point.>

[40] Full Wave Rectifiers, [https://www.electronics-tutorials.ws/diode/diode\\_6.html](https://www.electronics-tutorials.ws/diode/diode_6.html)

Rectifiers, <https://en.wikipedia.org/wiki/Rectifier>

[41] Design of a buck converter, <https://components101.com/articles/buck-converter-basics-working-design-and-operation>