UCF Senior Design II

S.M.A.C
Smart Mail Automated Center

Department of Electrical Engineering and Computer Science

University of Central Florida

Dr. Samuel Richie & Dr. Lei Wei

Senior Design II Final Documentation

Group 9

Shane Bramble-Wade	Computer Engineering
Tyler Rothenberg	Computer Engineering
Tyler Guerrero	Computer Engineering
Andre Villaran	Computer Engineering

Table of Contents

Table of Figures

Table of Tables

1. Executive Summary

For our senior design project, we will be developing a smart delivery system named Smart Mail Automated Center otherwise known as S.M.A.C. We wanted to create such a device because within today modern-day world no one yet has truly develop a viable delivery system yet that is automated and allows for any tenant or homeowner to receive a package in a safe manner. Today most packages are just left in front of your door or within a delivery room within your apartment complex which has the risk of the package being stolen, damaged by a person, taken by the wrong person, and even by bad weather if you're in a state like Florida and storm hits. With our system we want to ensure such situations are prevent and allow for customers to never have to be present to receive a package if they are not there to receive. Most people have a normal job that they work 5 days out the week, leading to risks that can occur when they are not present at home.

The simplified general idea is to have to secured box that has the following functionality:

- Ability to lock or unlock the box from a mobile application.
- Ability to take a picture from the delivery system to ensure who opened the box.
- Detect whenever the box is open or closed and display to the client it is in an open or closed state within the mobile application.
- Ability to scan in barcodes to ensure the correct package is being delivered.
- As well be able to kill to germs through a UV light within the box.
- Have a full functioning app to accompany the box to support functions they can use.
- Use a fingerprint scanner to use for authentication purposes to lock and unlock the box.

Within this report we will detail and outline exactly how we will design such a system to meet the requirements and criteria that will allow for such a system to be made. We will detail all the hardware and software requirement necessary.

The paper is divided mainly into two sections, mainly research and then the actual implementation of the design. The first half of the paper is where we did our research section. In the research section we outlined the project into sections. We outlined the first half of the paper into two sections Project Description and Research. Within the project Description we outline what exactly was the motivation of the project and as well the criteria and specifications to it. Then in the research section we dive deep into all the of the possible hardware components we could potentially use for implementation and the software side to our design.

The second half of the paper is the dedicated to the formal implementation of the Smart Mail Automated Center. We divided the second half of the paper into these main sections, Hardware Design, Microcontroller, Software Design and the mobile application design. In these sections we go over software and hardware component choices but as well we detail exactly how we implement those choices we chose from our research such as software language choice, hardware component choice and schematic design.

2. Project Description

The Smart Mail Automated Center (S.M.A.C) makes use of modern technology in order to solve a modern issue. The project includes a system for accepting deliveries in a secure box at a residence without the requirement of physically being there. The device is primarily made up of a lockbox with multiple unlocking options, and a mobile application.

2.1 Motivation

In today's world, ordering packages and food from the internet is a way of life. As convenient as it is, online delivery has a downside. It is not all that uncommon to find out that the package has gone missing before the homeowner has been able to retrieve it. Here in America, it turns out, 36% of people have experienced package theft. 45% of these people have experienced theft more than once [1]. There is no question that online delivery has made our life easier, but our goal with S.M.A.C is preventing package thieves from taking a delivery off the front porch.

S.M.A.C is a rugged lock box which may be unlocked by:

- Smartphone application
- Stored Fingerprint
- NFC / RFID card
- Scanned barcode which matches a known tracking number

When the delivery is made, the delivery driver simply scans the item and places the item in the box, then closes the lid. The box will automatically lock itself and notify the homeowner that their package was delivered. At this point, the package receiver has peace of mind that when they can retrieve their package, it will be there waiting for them.

During the pandemic, keeping your family safe is a top priority and we understand that delivered packages are also a foreign bacteria vehicle. For this reason, a priority feature of S.M.A.C will be to disinfect your packages every time a new one is delivered in a contactless manner using a UV light [2]. From the mobile app, the user may enable or disable the cleansing of packages so when they retrieve it, they have peace of mind they have taken necessary precautions to protect their family.

S.M.A.C is designed to provide peace of mind for homeowners from package thieves. As designers, we hope to create such a product at a reasonable cost to the consumer.

2.2 Function

S.M.A.C. is going to revolutionize package delivery systems. It will be easy to install, use, and accommodate in the home. Unless you live in a gated community or residential building odds are you will not have a secure package location resulting in carriers leaving packages unattended in a nonsecure location like one’s front door or porch. This lack of infrastructure is a problem for most people resulting in an opportunity to improve things in the sector. Technology has always allowed people to adapt to problems that arise in our daily lives and security has been enormously propelled by technology in the recent years.

The main function for this project will be to create a secure smart box that can autonomously receive and protect its contents from unauthorized entities. The box will have a barcode scanner that will allow it to identify authorized deliveries and open its door for deliveries to be ingested into its holding compartment. Once done accepting packages it will lock the door and remained locked until an authorize user comes and gives it an unlock command either from its accompanying smartphone application, using an NFC tag, or authorized fingerprints.

The project will integrate a lot of security and smart features to add value to the box. It will be able to communicate over Wi-Fi which will add lots of functionality to the box. Its accompanying smartphone application will be a portal to unlocking all the features that make this box smart. The application will be able to log authorized packages into S.M.A.C.'s database, unlock it from a remote location, and be able to see a snapshot through S.M.A.C.'s fisheye camera. The most important feature in the app will more than likely be its disinfectant feature. Which you will be able to control and set timers for the UV light inside the box.

The box will be equipped with a top of the line Ultraviolet light strip which will operate in the UVC range which is able to kill germs and viruses that may linger on boxes once they are received. With everything going on due to current events its imperative more than ever that we take all precautions when dealing with other people and foreign items to the home. Home deliveries have surged due to the pandemic which has been casted as a safer alternative to leaving the house in search of items. However, home delivers do not eliminate risk as viruses and germs are still prevalent on carboard boxes. This feature will add peace of mind to that last step for a package's journey to ensure that it carries nothing unwanted into the safe environments of our homes.

This box will revolutionize the way we receive packages with its innovative features but design of it does not stop there. Just like not wanting germs and viruses in our houses having a big unflattering box in the front door or porch is an eye sore and unwanted in many homes. Therefore, the design must go further than just a regular box. It must serve an aesthetic purpose which is why we have designed it in such a way to integrate itself into the environment harmoniously. The box will double either as a flower box or entry way bench for visitors. This will also add security as it will go unnoticed to the common eye.

This product will be something we want every house to be able to have as a necessary function for safety and wellbeing of packages and its residents.

2.3 Objectives and Goals

A set of objectives and goals were created initially to meet the engineering standards we created as a group. Throughout this section, we will discuss why we started with our design idea and the solutions we constructed to solve the problem. The main objective we set forth for our senior design project, was to create a user-friendly smart box that was responsive and secure for any homeowner. In today's society, many individuals make purchases online and have them delivered to their doorstep. If there is a case that a user is not home, this makes them vulnerable to package theft. Because of the increasing number of package

thefts, we created a set of goals to solve the package theft problems in the United States. (Schoolov, 2020)

2.3.1 Objectives

Our objective for this Senior Design project is to create a smart box for users, who are at risk of getting their packages stolen, or to store a residential user's package securely. The idea for this project came about from one of our group members, who's friend had experienced package theft within their area. Package theft has been a significant issue in a lot of cities in the United States. Typical victims of package theft usually lose out on about 300.00 dollars' worth of merchandise. According to many respondents, nearly 44% of individuals receive there purchased packages every week. (Research, 2019) These crimes are very costly since many respondents or users spend close to $222.00 per month on online orders. Porch pirates are the names of people who secretly steal residential user's packages from there top porch or front door. (Research, 2019) These porch pirates have been able to consistently take a user's package, which resulted in them stealing millions of dollars' worth of items each year. After seeing the figures of package theft within the United States, we conjured up making a secure box, both weatherproof and waterproof. This smart box will benefit the average user who purchases online, which causes a delivery driver to drop off their packages. Additionally, our smart box design will cater to merchants and postal services looking for a way to protect incoming mail for a user or the company securely.

2.3.2 Goals

The goal for our smart box is to provide the utmost security for a user's package. A user will be at ease to know that their packaging will be safe if they are not at home. Our smart box design will offer many features to inform users about the state or status of their package. It is effortless to make purchases online in today's society and have them delivered to your doorstep. On average, the waiting time for a package to be delivered is roughly two to four days. Unfortunately, there has been an increasing number of package theft cases from individuals who spy on the homeowner. (Security, 2019) The delivery of a package can be unpredictable as there can be cases of delays due to holidays or events. These events can sometimes be out of the control of a user. Because of this unpredictability, a residential user may not be home when their package gets delivered. This uncertain case makes a user vulnerable for a package thief to steal their package from their front door. The SMAC idea was created to safely and securely store a residential user's package from any thieves nearby. The SMAC protection system will allow a delivery driver to scan the barcode of a user's package, which will unlock the box, for the driver to place the item inside. Once the package is placed inside, it will keep the package secured and protected from any attackers. Additionally, after the package is secured in place, a notification will be sent to a user, letting them know that their package has been delivered.

Our senior design motive is to construct a product from the knowledge and technical skills learned through our 4-year program at the University of Central Florida. Our technical expertise and decision-making skills will be tested for the next few months when building our design. The main driving force for this idea, came from prior package theft experience. Leaving your package out in the open makes a homeowner vulnerable for their packages

to be taken away. Our smart box design possesses a mid-size container that is big enough to hold the contents of a user's package. A barcode scanner will be used to unlock the box once a verified package has been scanned. Mechanical locks will be implemented to lock a user's contents inside the box securely. The compartment will be immovably fixed, keeping up any deluge from getting into it that could not merely harm packages anyway the electronic sections in it as well. Since the case might be put sooner than yards, it must be drilled into the floor appropriately to shield the carton itself from being taken.

2.4 Requirements Specification

Within this requirement section, we will discuss the hardware and software requirements needed to accomplish all the features and needs for our SMAC project. As a group, we thoroughly examined the different uses of our SMAC design and was able to conclude some essential standards and requirements. These requirement specifications will aid us as a club to construct our SMAC design both efficiently and securely. Project requirements are conditions or assignments that must be finished to guarantee the undertaking's achievement or culmination. They give away from the work that should be finished. They're intended to adjust the venture's assets to the targets of the association. The advantages of successfully assembling venture necessities incorporate cost decrease, higher task achievement rates, increasingly powerful change of executives, and improved correspondence among partners.

2.4.1 Hardware/Key Requirements and Specifications

A hardware requirements list is also referred to as a hardware compatibility list. This list below will showcase the different compatible components needed to construct our design. Our hardware requirements list will explain the different tested, consistent, and incompatible hardware devices, we considered throughout our write-up. The following tables discusses the various aspects of hardware requirements. These equipment necessities aids in the plan of study for building the executives that centers around how to configure, incorporate, and oversee complex frameworks over their life cycles. At its center, frameworks building uses structures thinking standards to compose this assortment of information. The individual result of such endeavors is designed framework characterized as a mix of segments that collaborate to all things considered play out a valuable capacity. A prerequisite detail is a reported necessity, or set of recorded necessities, to be fulfilled by a given material, structure, item, administration, and many more It is a typical early piece of building plan and item improvement forms, in numerous fields. Equipment details and imperfection records are required by firmware engineers so as to carry out their responsibilities. In Table 1, we highlight the main specifications we will be demonstrating for our design.

Table 1 Hardware Requirements, Key Specifications

Hardware/Key	Specifications
1	Ability to store ordered packages into its database.
2	Barcode Reader which determines if a package is in the database.
3	Unlock with approved barcodes/tracking numbers.
4	Ability to protect your digital media and sensitive documents.
5	Box will auto close after being open for 30 seconds.
6	Camera will take a picture upon a failed unlock attempt.
7	Camera will take a picture anytime a package is scanned.
8	UV light will be implemented to kill bacteria on a package.
9	**Store fingerprints within scanner which allow the lock to unlock.**
10	An alternative feature to unlock a user box.
11	A user will be able to program their own unique passcode.
12	**A user will be able to unlock box using RFID card.**
13	After 5 attempts, the smart lock will transition into lock-down mood.
14	At least 2 UART communication peripherals
15	At least 1 SPI and 1 I2C communication peripherals
16	The power supply will aid in applying a voltage towards our main MCU to provide functionality.
17	**Unlock using mobile application**

2.4.2 Software Requirements

Programming necessities manage characterizing programming asset prerequisites and essentials that should be introduced on a PC to give the ideal working of an application. These necessities or requirements are commonly excluded from the product establishment bundle and should be introduced independently before the product is introduced. Software engineering is the deliberate utilization of designing ways to deal with the advancement of programming. These programming tools will aid in solving our complex algorithm, which

tries to implement a mobile application. Software engineering is a major section from the Computer Science field.

Table 2 Software Requirements

Software	Specifications
1	Ability to transmit data to a mobile device via Bluetooth or Wi-fi.
2	Can unlock the box through your mobile device.
3	A mobile app will inform a user when their packages have been delivered.
4	After three failed attempts to scan a package, a message will be sent to a client to inform them.
5	The focus for the app will include app response time, load performance, data security, user access and authentication.
6	User-friendly graphical interface
7	Ability to transmit data to a remote rest API through Wi-fi.
8	Ability to receive data from a remote database
9	Ability to read from and write to all sensors and peripherals
10	Smooth integration with hardware
11	Able to store data from a user’s profile
12	Store packages and pictures recorded from the SMAC box.
13	A database that hold sensitive information about packages tracking number.
14	○ Store packages and pictures recorded from the SMAC box.

2.5 House of Quality

The purpose of House of Quality is to create a product planning matrix that will display how the customer requirements correlate with the functions and specifications our group will use to assist in the development process. For the design we wanted our house of quality to reflect the customer's needs and desires to end with a product that was designed with the customer's needs and specifications in mind. When identifying the customer's needs and specifications we come to a net positive outcome that provides better function and utilization of the product designed. Building up target esteems for each plan prerequisite is important. This activity builds up solid objectives for the structure designs and further characterizes client necessities. As a team we agreed on the target values for the customer. In Figure 1 we entered them in the House of Quality diagram shown below.

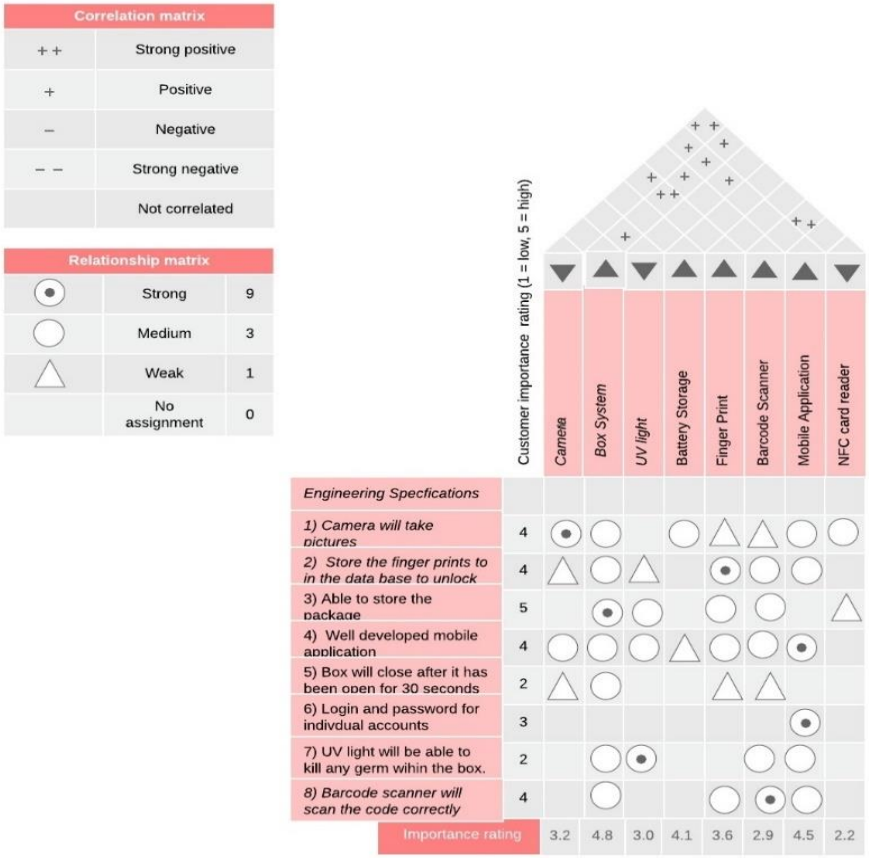


Figure 1 House of Quality Diagram

The benefits of having our house of quality include:

- The clients need and prerequisites are built up which makes us organize and work as per the item's determinations.
- Recognize client needs and prerequisites from the beginning to guarantee that they are accomplished.
- The execution of the House of Quality prompts consumer loyalty and devotion.
- Improving an organization's comprehension of its clients, which prompts better client use, in general deals, and portion of wallet.

House of Quality graphs depend vigorously on recognizing and keeping up the voice of the client. A client's voice is a method of getting and executing the client's expressed needs and prerequisites for their administrations or items.

2.7 Block Diagram

Each person has a key part which forms the foundation to our project. The group members and their responsibilities are given in the table below. We have colored coded each group members responsibilities for the project design.

Group Member	Main Responsibility
Tyler Guerrero	Mobile Application
Andre Villaran	Housing and WIFI
Shane Bramble-Wade	PCB and Lock
Tyler Rothenberg	Microcontroller

Table 3 Group members and their responsibilities

Once the sections above have been implemented the base of the project will have basic function. Then, we will be able to add the other features which complete our project:

- UV Light
- Barcode Scanner
- Camera
- NFC reader
- Fingerprint Scanner.

Although each one of us is responsible for researching specific parts, we must all have a good understanding of the entire project so that when we build S.M.A.C. we may help each other out. The block diagram for the project is included below.

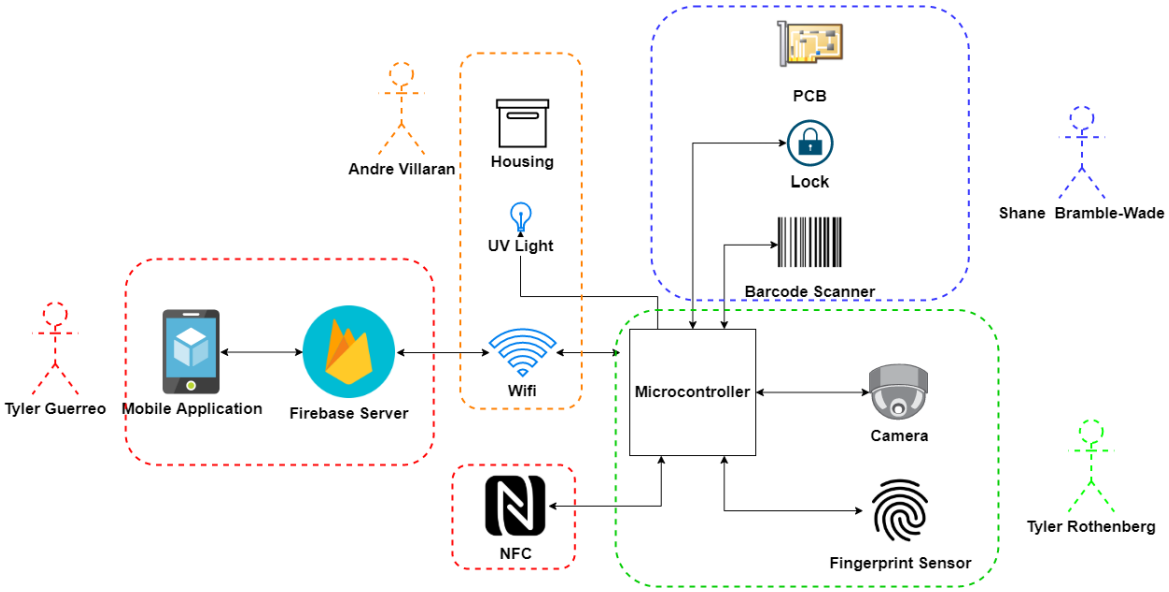


Figure 2 S.M.A.C Block Diagram

3. Design Constraints and Standards

For all systems we require ardent design constraints to effectively execute the design we are trying implement for our project. For the paper the research for the constrains section will take into considerations all the possible constraints and possible dilemmas with the implementation and designs of the smart mail automated constraint. With our project we will try to carefully exam each possible constraint individually and examine them.

Once done when that we will then we will use the constraints on all aspects of our design. The constraints must be in the realm of reason so we can complete a proper analysis of all aspects of the design. As well when analyzing we must not forget to take into considerations when analyzing a constraint does this constraint affect any other constraints within the design and project.

Some of the constraints we will go over will be listed below

- Time Constraints
- Economic Design Constraints
- Manufacturing Constraints
- Location Constraints
- Java coding Standards
- C coding Standards
- Environmental, Social and Political constraints
- Ethical, Health and Safety constraints.

3.1 Economic Design Constraints

When considering the economic constraints of our project we must be limited and take into consideration how much each part will cost and select which ones will make our final design. When we take into consideration the components that existed in our original write up, we will inevitably have to shave down some components due to our budget limit since it will exceed it. Our proposed monetary budget for the smart mail automated center was $500.00. When we are considering each individual component from our NFC modules, to the Wi-Fi modules down to the Arduino we will have to access whether it make it into the final bill as some of the components can cause our limit to exceed. Some of our far-reaching goals for our component may not be able to fit into our budget so we will have to take that into consideration. With these economic constrains we as well have to as well keep in mind when any of our components or our boards get short circuited.

During our development we have short circuited many of our components and test board’s and with this in mind we had to be careful with our development process because the more we accidentally would short circuit components the more we would run up our budget. With the development process we managed to short circuit our first PCB design and our camera, barcodes scanner and RFID. Doing this added a lot of to our actual budget due to the fact that these components were quite expensive and as well since we had time constrictions, we always had to pay an extra premium to order expedited delivery.

Another cause of inflation in our budget was because we did not have many of the tools required to work on certain portions of the project. Many of the expenses that made our budget balloon were due to one of buys of tools we required. Final price of box later on in production should be way less due to having the tools necessary. The inability to have access to these tools due to COVID and the university being close led us to more desperately buy the tools needed even if it was for a one-time use.

3.2 Time Constraints

Now after considering all our potential economic constraints we will investigate the time constraints that are set in place for our project. With the time constraints, they will be set to create a timely plan that would set each component and each facet of our project to completed in an efficient manner to maximize performance and execution. The main way we created these time constraints for us was to develop a thorough Gantt chart that would set hard time constraints on each member of the group to keep every member accountable for the project.

With the design of the Gantt chart we divided each topic and idea to a group member and a time period for them to complete the task to hold them accountable to the given task. The usual time frame for each of us is to complete 5 pages per week. When doing that we can accomplished the paper with spare time to revise and edit it when the major deadlines come.

The major time constraints are the final due date for the final revision of our senior design 1 paper and the final due date for our official project. The final revision due date for us personally is July 21, 2020. We set this due date for us personally to give us an ample amount of time to ensure we can revise and edit it. The other major time constraint is our due date for our final project. As the final date approached, we had to make sure our new PCB would not fail due to the limited time remaining. The final PCB thankfully was a success and allowed for a successful demo review and video.

3.3 Manufacturing and Location Constraints

Another major constraint we have been mindful of is the manufacturing and production of the products we need to buy. The manufacturing can potentially be limited so we may not be able to use the full array of tools that are available to me. Since the advent of Covid-19 manufacturing from china has been put on a major halt. When buying, a lot of the products we need they are being manufactured in china right now so we may have some orders get backed up due to the uncertainty of the situation right now.

When we originally signed up for senior design one and two, we did not imagine our whole senior design experience would be shifted to be online. Since the outbreak of the global virus we had to shift our learning and communication all virtually. This was a skill we had to develop. Moving from in person has been a constraint on the project. When this was put into play, we had to come up with solutions for how to carry out the daily task for our

project and finish everything in a timely manner even though we cannot be in a group physically next to each other.

To combat this, we have made protocols for our group on how to work throughout this process and how will we assemble the project even though we are far distance apart. When working we had to choose which team member would get certain components and parts of the project to work on so they can test and implement their task. For example, in the beginning of the semester we had to choose which team member was going to receive the portable oscilloscope, we had to come to a decision to give it to the team member that would need the most use from it.

Backups had to be also put in place in case a certain team member would not be able to meet or would be unable to bring a certain part. This was extremely crucial as we had limited times and places to meet due to the location of all the team members. It was essentially a logistical nightmare but after all challenges we were able to pull it off. Even if a team member would misplace, forget, or a piece would malfunction we always had a backup to work from.

3.4 Realistic Design Constraints

Most designs need as real of a set design constraint as possible for an effective and efficient path to implementation. When we are designing, we have to take into consideration every possible outcome that can come about when implementing so it's always a smart idea to keep your design constraints realistic and don’t overshoot a design and go for the most complex design because you can run into unforeseen issues because your design constraints probably did not take into account the complexity for your design. For example, in certain situations you can run into communications issues between your components, in our cases one of the constraints we ran into that was an unforeseen communication issue was our communication between the ESP32 and having our camera send the bytes of each pixel over to the ESP32 then to our API. When doing this we ran into design constraints that we unforeseen because we simply could not tell that the REST API post methods could not handle the number of pixels we wanted to send over. With this example this demonstrates the realistic design constraints that we ran into that simply we unforeseen because we had to test it first.

Another pain point for the project was designing an efficient and managed power supply system for all the components. Heat, power deliver limitations, and voltage steadiness were top concerns which had to be address as we tested our design. Some modifications and space concerns had to be worked around to have the system perform as expected. There is no substitution for testing as no matter how perfect something looks on paper it might always not come true in application.

When dealing with our design constraints we all had to come to an agreement on what type of engineering solutions we would come up with and implement for our design. Whether it be a component redesign, a PCB redesign or a power system redesign we all had to ready

for it because during the development process we will run into unforeseen issues that will inevitably arise when trying to do the actual design.

3.5 Environmental, Social, Political constraint

For the Smart Mail Automated Center Social and political constraints do not really apply to our project or design such domains of. Since political constraints are a part of geopolitical and political domains our project really does not have much to do with such areas. As well for the Smart Mail Automated Center it doesn't deal much with Social constraints, usually one trying to make a statement is society deals in the domain of social constraints, but our project simply does not deal with that. Now for the environmental constraints set on the design it does affect how decided to design the box. With a friendly environmental constraint set in mind we try to use materials that were not a negative towards the environment when disposed. Wood can be biodegradable and has less impact on the environment than a plastic or synthetic material composition would. For power source specifications we tried to optimize all components to draw the least power possible to not be a huge burden to the user and their pocket.

3.5.1 Soldering Constraints

When it comes to soldering there are inevitable physical constraints that we will run into when designing our PCB. When we were soldering our components we always ran into the issue of meeting and setting a proper time to use the lab on a given day, none of us had the proper equipment to do it on our own so we always had to use the senior design lab when the lab itself had restrictions due to the pandemic happening currently. As well we were quite new to the craft of soldering so we had to learn all the proper practices associated with it when it comes to the safety practices of soldering and when it comes to how to properly hand the tools that are associated with it.

ALLOY COMPOSITION	MELTING RANGE	TENSILE STRENGTH (ksi)	CREEP RESISTANCE
Sn63/Pb37	183°C	4.92	Moderate
CASTIN	217°C	5.73	High
Sn96.5/Ag3.5	221°C	8.90	High
Au80/Sn20	280 °C	40.00	Excellent
Sn95/Ag5	221 - 240°C	8.09	High
Sn95/Sb5	232 - 240°C	8.15	High
Sn5/Pb85/Sb10	245 - 255°C	5.57	Excellent
Sn5/Pb93.5/Ag1.5	296 - 301°C	4.30	High
Sn5/Pb92.5/Ag2.5	299 - 304°C	4.20	High

*Additional alloys available upon request

Compatible Fluxes No-Clean, Water Soluble, Rosin

Above is a table that shows the melting ranges depending on the solder type used. Usually solder come with a stock preset of an equal mix of tin and lead for the heat to properly disperse evenly throughout the actual tool. The temperatures of the tool as you can see get quite hot which can lead to unintended consequences of accidentally harm by burning oneself when one does not take the proper safety precautions. Some of the most common ways to cause safety malfunctions is by touching the iron, the iron itself is extraordinarily hot and even accidentally getting into contact with it can cause extreme burns, another common safety malfunction is not keeping the sponge wet when not using the solder anymore, It useful to keep it wet to cool down the solder tip once done using and last another common one is simply keeping it on for lengthy periods of time. When using the solder tools, you should always turn It off and unplug it once you are done with it.

3.6 Ethical, Health, and Safety constraints

With the Smart Mail Automated Center, we try to provide the user with the safest experience possible. We try our best with the design constraints to mitigate unethical behavior. For our design, the ethical constraint is to provide the user a design that prevents unethical behavior such as stealing, theft, or unnecessary damage to a package through our design. As we with our safety constraint we simply just want our customer to feel that their package is safe with the design, that is the main safety constraint we have in place or the health constraint Smart Mail Automated Center doesn't have much to do with the domain of Health. There is no Health constraint existing within our design.

Some of the safety constraints that were present were when it came to testing was knowing which electronics and which components could handle certain amounts of current or voltage. When we were in the testing phase, we ran into multiple safety issues that were a result of just not knowing that a capacitor can take a certain amount of voltage. When doing this testing we burnt out many components that result in a couple of safety concerns like for example we input to much voltage into a capacitor that caused it to explode. We for that reason started to use safety goggles and safety gloves for when we are soldering component’s on to the board. As well we in the middle of a pandemic so every time we met, we had to make sure to keep a safe distance of six feet from each other and wear mask while we were in each other presence to not get each other infected with the virus.

3.7 Java Coding Standard’s

For the Java coding standards, we look towards the way we write comments, you see that we can create web pages that will spawn in the method and classes. When you look at the Java API documentation that exactly what you end up seeing. That sort of style of coding paradigm is originated in the Javadoc which is short Java documentation. The java documentation is the industry standard to code in. It used in all the way to the industry level down to the academic level.

When we use the java documentation within our programs, we will be able generate webpages from our program that are close the java API. The Java Coding Standard was

developed by oracle the creators of Java, they outlined exactly how java programs should be written and formulated.

The main purpose of this section is to explain the proper formatting for how to code in java to get the most optimal results.

3.7.1 The Essential Coding Rules to follow

- Use extremely descriptive and suitable name for any identifier you have (methods, classes, variables, constants and thing of that nature.
- Comments for every 3-7 lines of code to explain what is going.
- Be as neat as possible.
- Create readable durable Robust code that is able to run bug free.
- Always keep in mind when edge cases occur within your code.

3.7.2 Identifier Naming and Capitalization Guidelines

Within this section we go over the java coding standards within the Javadoc that refers to the formulation of identifiers and how to correctly create them because there is copious amount of forms and styles, they come in so we must detail them.

- You have to use proper adequate names for all you function, variables, identifiers and constants.
- We can only use single character identifiers for looping purposes, when do so you make it more readable in your code. Having such a coding paradigm as well makes your code readable.
- Classes name will always start with an upper-case character to keep camel casing I all your coding for class naming.
- All conventional common variable names must start with a lower-case letter. Variables are usually defined to them being parameters, any sort field that is allocated for data and locally defined variables.
- Method names aka functions in java much of all start with lower cases, to keep consistency within your program.
- If and when your you have a multi-word identifier, they must follow came casing meaning first word Is lower case but following after that they will all be upper case.
- The java documentation makes it a rule to refrain from using dashes or hyphens to separate multi worded identifiers.

3.7.3 Comments: Classes

Within this section he java documentation outlines exactly how we are supposed to formulate a comment to explain exactly how we are about to formulate a class, it goes like:

- All classes in java should be preceded with a comment using the java documentation notational convention o explain exactly what the object was designed to do and how it is supposed to function.;
- The comment should be well descriptive to explain the classes purpose.
- All classes names will start with an Upper Case.

3.7.4 Program Modules

In this section the java documentation goes over how we are supposed to formulate a proper module of code. Within the lines of code there are rules and constrains we must adhere to in order to make our code proper and readable.

- All our lines of code should never exceed more than 100 character, they should generally stay in the range of 80 – 100 character each line to make our code compact and readable.
- All public classes should be restrained to existing in one file to make your file structure more readable and logical. You want to be able to tell what object is on that file by its name, if you do not follow this convention it could potentially make collaborating on a code base unnecessarily difficult.
- The file name should contain the public class you defined within.
- Try and avoid the use of default package statements.

3.7.5 Indentation

For this section we will talk about what are the standards that java has when it pertains to indentation within your code.

- Whenever we begin a new block of code, we must indent by two spaces.
- We never start open braces on a new line
- We do indeed start a closed brace on a new line that are indented with the block of code they were closed with.
- All comments must align with the block of code they correspond to.

With all this we will have to keep all these constraints and standard in mind when coding our mobile application to have an effective outcome for our project.

3.8 The C language Standard

C11 is the unformal way the community has decided to name the ISO/IEC 9899:2011 manual. The ISO/IEC 9899:2011 is the current standard set by the ISO for the C language. C11 was brought about to update the language but as well to standardize any features that were already present in common multiple implementations within the C community. As well they set out to define memory models there were better fit for multithreading programming. Essentially C11 was built to ratify and update the implementation of C that was set before.

C99 the previous C language C standard had a whole host of issues they had to deal with. The issues it had were some of the required features to use the programming language have been so far hard to implement on multiple platforms. Some of the new features that were brought about by C11 were as followed:

- They no had designated initializers
- Type-generic math library's
- New introduced data types
- Variable length arrays
- Restricted pointers
- Inline functions
- One-line comments now
- Collections of declaration of variables and data types

C11 was used to bring light to the C security concern within the past because some of their string manipulation functions were proven to be unsecured because they never created C to check bounds or have File I/O functions validate the arguments. When they didn't do this, it led to many malicious software attacks.

It was once this was addressed C11 began to provide Unicode support, complied with floating point arithmetic, memory alignment, structures and unions then now built in support for multithreading programming.

C11 was used to bring light to the C security concern within the past because some of their string manipulation functions were proven to be unsecured because they never created C to check bounds or have File I/O functions validate the arguments. When they didn't do this, it led to many malicious software attacks. It was once this was addressed C11 began to provide Unicode support, complied with floating point arithmetic, memory alignment, structures and unions then now built in support for multithreading programming

When using the language C for our hardware components we will have to keep these standards and principals in mind to have an effective outcome for the embedded systems portion of the project.

4. Research

In this section, we investigate previous projects, available parts, and the different technologies available to meet the needs of our project. Significant research must be done so that when we build out the project many of the potential roadblocks will have already been seen and worked through. A large portion of the total time for this paper time was devoted to researching all aspects of this project.

4.1 Similar Projects

The idea of a lockbox to protect your mail and packages is not a new one. Different solutions have been explored for solving the same issue. Our goal is to explore the designs that have already been created to find out what did and did not work. Investigating similar projects provides a quick reality check as we begin our research. In this section we will explore three projects, two of which were designed here at UCF.

4.1.1 SOPBOX

A similar project, SOPBOX (Secure Outside Package Box) was developed here at the University of Central Florida between fall 2018 and spring 2019. Their team was mixed between two electrical engineers and two computer engineers. Their intention was to design a box which could act as a smart lockbox outside the user's home for incoming packages.

Their design included an RFID reader, a camera, and a locking mechanism which communicated with a mobile application. The box would unlock through an RFID system which the delivery driver would have a card for, then automatically lock back up upon closing. The mobile application notified the user upon open, close, and picture.

The group went with the ATmega328P microcontroller primarily because of its low cost, host interface ports, power consumption, and general input/output ports. They chose a MFRC522 for the RFID component for its price, support, and documentation. They chose an ESP8266 for their Wi-Fi module and a Pixy2 for the camera.

Their mobile application was coded using React Native so that the code will work on both IOS and android devices. Their database was chosen to be Firebase Realtime Database because a lot of the groundwork is already laid out and ready to go. (08, 2019)

4.1.2 Black Box

Figure 3 Black Box housing. Permission requested from designers.

Another similar project, Black Box, was developed here at the University of Central Florida between spring 2019 and summer 2019. Their team was mixed between two electrical engineers and two computer engineers. Their intention was to design a rechargeable outside box which would receive the users' personal mail. The black box project was the highest quality project researched. Their design included a barcode scanner, a

fingerprint reader, and a locking mechanism which connects with a mobile application. The delivery driver would unlock the box by scanning the package. The barcode reader result would be compared with the tracking numbers in the database. If a match is found, unlock the box. The group went with the ATMega2560 microcontroller primarily due to the amount of program memory size, and digital communication peripherals. (03, 2019)

4.1.3 uCella

This is a concept that never seemed to make it to the light of day. Nonetheless, it is a great project to research as it has many of the features we plan to implement into our box. UCella included a camera & scanning so that it could scan packages for bar codes and take picture or videos as proof of delivery.

Figure 4 uCella housing design. Permission requested from Coming Soon-Tech

The package tracking numbers may be uploaded from the user's linked e-mail account and synced to the user's uCella device as an unlocking code. The physical design is creative and works like a murphy bed. The box attaches to the wall and expands out for larger packages. The mobile application was designed for Android devices to track all online purchases in one place. Once a package is delivered, the device notifies the user about the delivery with photos. One of the reasons this device was unsuccessful was that the device was battery powered and needed to be recharged. It also included a 4" IPS touchscreen which drained its battery every forty-eight hours. When offline, the device still acted as a mailbox for the USPS through its mail slot. This slot could be accessed by any delivery person. (Tech, 2016)

4.2 Microcontroller Chips

The microcontroller will be the brains of this project therefore we must thoroughly research the available options. Throughout this section, we will investigate two microcontrollers: MSP430FR6989 and ATMega2560. The main categories for discussion are I/O pins, cost, support, memory, programming language, architecture, and speed.

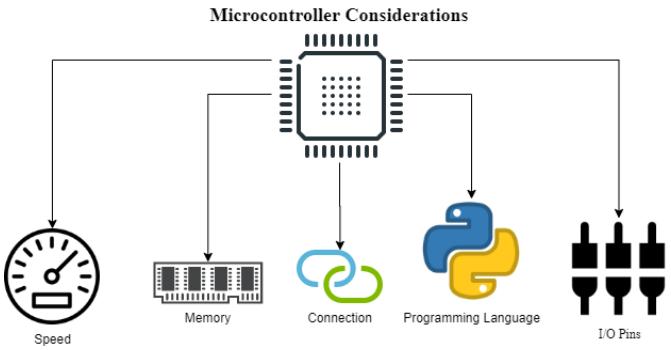


Figure 5 MCU Considerations

I/O pins will be the primary focus of the research as we must carefully consider the use of each pin in the design. We must also make sure the microcontroller is user friendly for fast prototyping. Careful consideration must be made such that all sensors and peripherals will be able to connect and communicate with the microcontroller through the standard communication protocols.

4.2.1 Digital Communication Protocols

It is important to understand the three communication protocols as each sensor / peripheral will need to communicate with the microcontroller. We will look at the main characteristics of the three protocols that are available to us: UART, I2C, and SPI.

4.2.1.1 UART

The **UART** (universal asynchronous receiver transmitter) protocol is an asynchronous communication protocol which means it requires no shared clock. Each device generates their own clock.

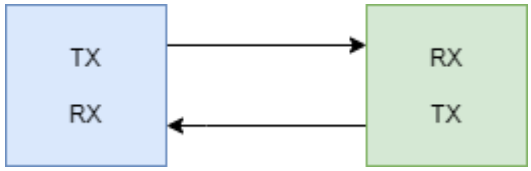


Figure 6 UART connection setup as full duplex

UART is simple and may be setup as either half-duplex or full duplex. The important parameters we must know from the sensor are given in the table below.

Parameter	Meaning
Baud Rate	Transmission Speed
Data Size	Number of bits
First bit	MSB or LSB
Parity	Bits to detect errors
Stop Bit	Signals the end of the transmission
Flow Control	A mechanism to pace the transmission

Table 4 UART Important Parameters

UART is the easiest protocol to setup, however its downsides are that each UART port can only connect one device, and it is slower than both I2C and SPI.

4.2.1.2 I2C

The **I2C** (Inter Integrated Circuit) protocol is a synchronous communication protocol which means the sensors share the same clock signal. I2C became popular in 2006 and usually consists of one master and multiple devices / sensors. The master device always initiates the communication (read / write) and is responsible for generating the clock. I2C is a half-duplex connection so data is one directional.

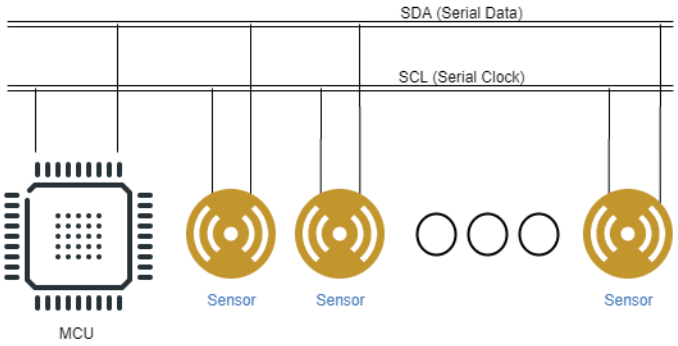


Figure 7 I2C Diagram

The important parameters we must know from our sensors and peripherals for I2C are given in the table below.

Parameter	Meaning
Mode	Known max clock frequency
Test Register Address	Known Address to read from the sensor
Config Register Address	Address to update the pointer
Result Register Address	Address to read data

Table 5 I2C Important Parameters

I2C is a bit harder to setup as the MCU and Sensors share a clock. We must make sure that all devices which will connect through I2C are compatible and able to share the same clock signal.

4.2.1.3 SPI

SPI (Serial Peripheral Interface) is also a synchronous communication protocol where a master communicates with one or more devices. This protocol is full duplex therefore there are two data wires between the master and the device. The master is responsible for generating the clock signal and it is sent to the device.

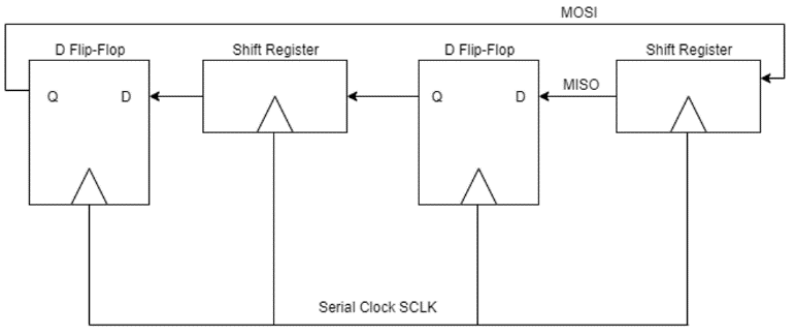


Figure 8 SPI Diagram

The important parameters we must know from our sensors and peripherals for SPI are given in the table below.

Parameter	Meaning
Serial Data In	Pin which data is sent to sensor
Serial Data out	Pin which data is sent from sensor
Serial Clock	Pin which accepts the master's clock
Chip Select	Register which lets the sensors know to be active
Mode	Clock idle low/high

Table 6 SPI Important Parameters

SPI is the fastest and most simple protocol.

4.2.2 Options for Microcontrollers

In this section, the features of the MSP430FR6989 and the ATmega2560 are explored in detail. There are many microcontroller chips available to purchase, but we must make a best effort to choose the right one for our project.

4.2.2.1 MSP430FR6989

Figure 9 MSP430FR6989. Permission to reproduce requested.

The MSP430FR6989 is an old friend to all of us at the University of Central Florida as it is used throughout our Embedded Systems course on a development board. It is a mixed-signal microcontroller developed by Texas Instruments. It is a particularly straightforward 16-bit processor with a von Neumann architecture, designed for low-power applications.

Since our device will be plugged into a wall outlet, the low-power features of the MSP430 are not as enticing, but it remains an important microcontroller to research due to all of us having experience with the device. The MSP430 includes a generous number of GPIO pins. The large number of GPIO pins allows for the following digital communication peripherals. 2-I2C connections, 4-SPI connections, and 2-UART connections. The program memory size is 128 KB with 2KB of SRAM. The MSP430FR6989 may be programmed in Assembly, C, or C++. (Texas Instruments, 2014)

On the chip there are five 16-bit timers. The default clock source is an internal RC low-frequency oscillator or a digitally controlled oscillator. These sources may be calibrated by the user, but for higher accuracy, a 32-kHz watch crystal oscillator may be connected.

The size of the chip varies depending on the package of the chip. In the table below are the available options for the MSP430FR6989.

Part Number	Package	Body Size
MSP430FR6989IPZ	LQFP (100)	14 mm × 14 mm
MSP430FR6989IPN	LQFP (80)	12 mm × 12 mm

Table 7 MSP430FR6989 Chip Sizes

The larger package size would fit our design better as we will have to solder the chip onto the PCB ourselves. Putting everything together, the specifications of the MSP430FR6989 are shown in the table below.

MSP430FR6989 Specifications	
Supply Voltage	1.8-3.6V
Max Current	2mA
Communication Interfaces	2-UART, 2-I2C, 4-SPI
Program Memory Size	128KB
Ram Size	2KB
Manufacturer	Texas Instruments
Cost	$3.61

Table 8 MSP430FR6989 Specifications

4.2.2.2 MSP430FR6989 Prototyping

Figure 10 MSP430FR6989 Development Board. Permission to reproduce requested.

An important aspect of building our design is getting a quick prototype. Texas Instruments sells a launchpad development board which uses the MSP430FR6989. This board is an easy to use module which contains everything we would need to start developing on the chip. This device would only be used in the prototype / testing stage as it includes many features not necessary for our project.

The development board allows us to program the MSP430FR6989 using a USB cable from our computer. Then, we may debug our code through the same USB interface. The development board also includes large pins for direct access to the pins on the microcontroller. Some other features are buttons, LEDs, a crystal oscillator, a reset button, and an LCD.

4.2.2.3 ATmega2560

Figure 11 ATmega2560. Permission to reproduce requested.

The ATmega2560 is both a high-performance and low-power microcontroller. Most UCF senior design projects we researched use some version of the ATMega, and for good reason. It is a multipurpose chip which has a good mix of low-level functions and libraries. The ATmega2560 is a RISC-based microcontroller with 86 general purpose I/O pins and 32 general purpose registers. The program memory size comes in at 256KB with 8KB of SRAM. The ATmega2560 may be programmed in Assembly, C, or C++. (Arduino, 2010)

The ATMega2560 provides many digital communication peripherals. 4-UART, 5-SPI, and 1-I2C will allow us to connect many peripherals and sensors. On the chip there are 2 8-bit timers, and 4 16-bit timers. The default clock source is an internal RC oscillator at 8.0MHZ. This can be calibrated by the user, but for higher accuracy, a crystal oscillator may be connected. There are different ATMega versions as shown in the table below.

Device	Flash	RAM	I/O	UART	ADC
ATmega640	64KB	8KB	86	4	16
ATmega1280	128KB	8KB	86	4	16
ATmega1281	128KB	8KB	54	2	8
ATmega2560	256KB	8KB	86	4	16
ATmega2561	256KB	8KB	54	2	8

Table 9 ATMega model options

The 2560 would best suit our projects needs due to the larger flash size and UART communication peripherals. Going a little deeper, the ATmega2560 comes in two versions. The difference between the two are its speed and power supply requirements.

Device	Speed MHz	Power Supply
ATmega2560V-8AU	8	1.8V - 5.5V
ATmega2560-16AU	16	4.5V - 5.5V

Table 10 ATmega2560 specific options

The ATmega2560-16AU would be best for our project as time is more important than price. Putting everything together, the specifications of the ATmega2560 are shown in the table below.

ATmega2560-16AU Specifications	
Operating Voltage	1.8-5.5V
Max Current	20-50mA
Communication Interfaces	4 UART, 1 I2C, 5 SPI
Program Memory Size	256KB
Ram Size	8KB
Chip Size	16 mm x 16 mm
Manufacturer	Microchip
Price	$11.85

Table 11 Arduino Mega 2560 Specifications

4.2.2.4 ATmega2560 Prototyping

Figure 12 Arduino Mega. Permission to reproduce requested.

As seen with the MSP430FR6989, we will need to quickly prototype our microcontroller to make sure our code and project work properly before integrating everything into a custom PCB. Arduino manufactures a development board called the Arduino Mega which contains the ATMega2560 chip. This device would only be used in the prototype / testing stage as it includes many features not necessary for our project.

The development board allows us to program the ATmega2560 through USB. Then, we may debug our code through the same USB interface. The development board includes headers for direct access to the pins on the microcontroller, buttons, a crystal oscillator, a reset button, and LEDs.

4.2.3 Comparing Microcontrollers

In this section we will first compare the choices in a table, then select a chip for the project. The table below shows compares the main differences between each of the microcontrollers researched.

	MSP430FR6989	ATmega2560
I/O Pins	83	54
UART	2	4
I2C	2	1
SPI	4	5
Program Memory (KB)	128	256
SRAM (KB)	2	8
Architecture	RISC	RISC
Speed	16 MIPS	16 MIPS
Timers	5 16-bit	4 16-bit, 2 8-bit
Size	14 mm × 14 mm	16 mm x 16 mm
Manufacturer	Texas Instruments	Microchip
Cost	$3.61	$11.85

Table 12 Microcontroller comparisons

The ATmega2560 was chosen for three main reasons.

1. **Prototyping -** Prototyping for the ATmega2560 will be seamless as we will have the Arduino mega for testing of code. Speaking of code, many of the components we have selected have open source libraries which work with the ATmega2560. These libraries are discussed in more detail in chapter 5.

2. **Digital Communication Peripherals -** The ATmega2560 has 4-UART, 1-I2C, and 5-SPI peripherals. UART is our most important communication peripheral as we will have many devices which communicate with the microcontroller through UART.

3. **Program Memory -** The devices attached to our microcontroller such as the camera will be sending large files to be saved to the disk. Having a larger program memory and ram provides a more of a cushion for these devices.

4.3 Housing Design

The housing design will be one of the most important parts of this project. It will determine the practicality, durability, and cost effectiveness of our project. The housing design of the project will have to be able to fit most common package sizes from most carriers. For the application of this project we will be taking into consideration amazon, FedEx, UPS, and USPS. In this section we will be focusing on the dimensions of the box as it will impact greatly on materials we can use and overall design of the box.

4.3.1 Dimensions

For the dimensions we must be careful to include most package sizes but not all as there exist outliers in package sizes. If we were to consider all package sizes instead of the most common ones it would greatly impact our practicality and cost effectiveness of the project. Amazon for example has a plethora of package sizes they use however their most common boxes are like the flat rate boxes of USPS, the express box from UPS, and the One Rate boxes from FedEx. These boxes tend to be the most common sizes that are used by consumers. They provide a wide range of sizes and can act as a baseline for our dimensions. Taking into consideration the average of the largest measurements of the box sizes can help us to determine the right size for our box. Notice how I did not mention the average size of the box used as we were not able to find shipping volume of these boxes by size, so we must try and consider the largest flat rate boxes possible. The next few charts will show the data collected for all the boxes. The only one with incomplete data will be amazon as they do not have a flat rate system to use. Instead, the pooled sizes will be those closest to the flat rate boxes as those tend to be the most used regardless. Largest dimensions will be highlighted and recorded. The largest dimensions will be recorded to try and simulate all the biggest sizes possible for a theoretical box size that will be able to fit all boxes in their respective carrier.

USPS

Name	Length (in)	Width (in)	Height (in)	Volume (in^3)
Small	8-11/16”	5-7/16”	1-3/4”	83
Medium-1	11-1/4”	8-3/4”	6”	591
Medium-2	14-1/8”	12”	3-1/2”	593
Large	12-1/4”	12-1/4”	6”	900
Large Game Box	24-1/16”	11-7/8”	3-1/4”	929
Largest dimension	24-1/16”	12-1/4”	6”	1769

Table 13 USPS

UPS

Name	Volume (in^3)
XS	100
S	250
M	650
L	1050
XL	1728

Table 14 UPS

*UPS does not have defined dimensions they allow all box sizes and go by max volume. Will try an achieve max volume with dimensions from other services.

FedEx

Name	Length (in)	Width (in)	Height (in)	Volume (in^3)
Small-1	12-3/8”	10-7/8"	1-1/2"	202
Small-2	11-1/4"	8-3/4"	2-5/8"	258
Medium-1	13-1/4"	11-1/2"	2-3/8"	362
Medium-2	11-1/4"	8-3/4"	4-3/8"	431
Large-1	17-1/2"	12-3/8"	3"	650
Large-2	11-1/4"	8-3/4"	7-3/4"	763
Extra Large-1	11-7/8"	11”	10-3/4"	1404
Extra Large-2	15-3/4"	14-1/8"	6"	1334
Largest dimensions	15-3/4"	14-1/8"	10-3/4"	2392

Table 15 FedEx

Amazon

Amazon was a complicated vendor to get box information from. They do not release their box sizes but thanks to their ample community of vendors they have come up with a list of boxes amazon uses in their inventory. This list helps the vendors recycle and reuse the boxes they receive from amazon for their own products. This list currently has a staggering 89 boxes which created a problem in coming up with what dimensions to justify for our box but thanks to the list I was able to create graph all dimensions and come up with the largest dimensions of about 75% of the boxes in each category, L x W x H. To sort through the data a pareto graph was used. A pareto graph divides the data into interval bar graphs and displays a curve of percentage of boxes in that dimension. This allowed me to see at which point the number of boxes reached a 75% threshold to and used that limit as our largest dimension in that class. The data is as follows:

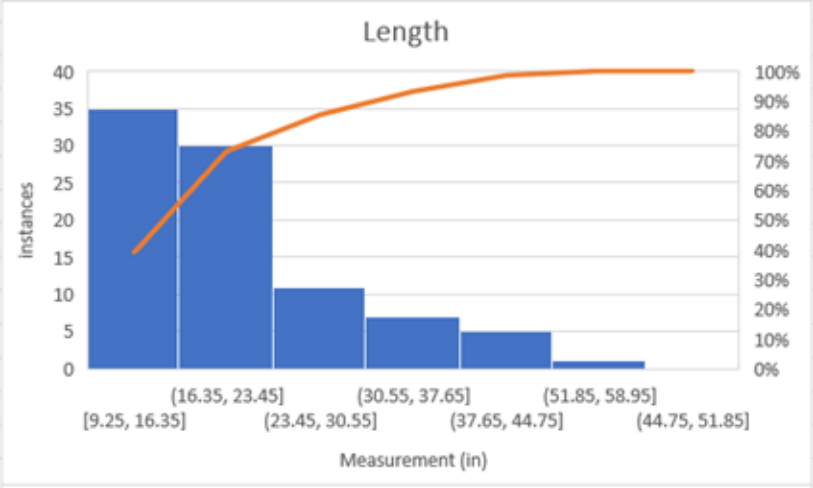


Figure 13 Length

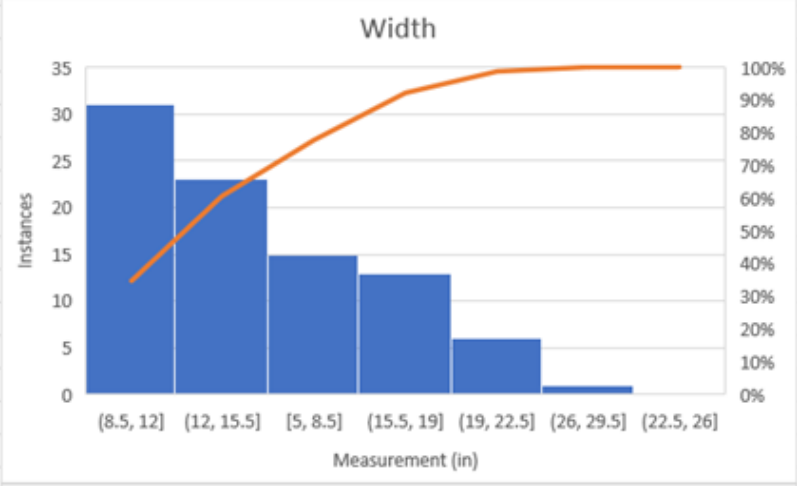


Figure 14 Width

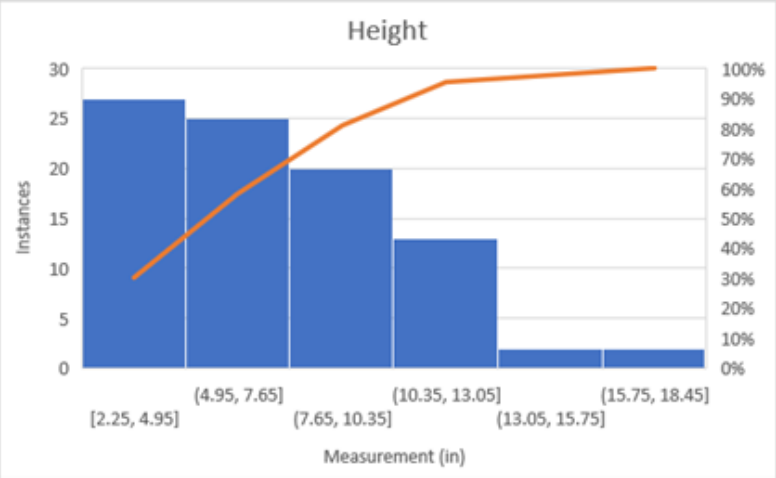


Figure 15 Height

As we can see in the pareto graph we can take the largest dimension in our 75% threshold for each of our measurements. This eliminates most outliers in the data. From the graph we deduct that for length 23.45 inches is the biggest dimension for that set. For width we end up with 12 inches the largest set of data after passing the 75 percent threshold. Lastly, for height we end up with a modest 7.65 inches. This estimation creates a box with the following dimensions:

Table 16 Amazon Dimensions

Name	Length (in)	Width (in)	Height (in)	Volume (in^3)
Largest dimensions	23.45	12	7.65	2153

These values end up close to our results for the other carrier.

Conclusions

With data from all major carriers we can now begin to get the results for our final dimensions. Collectively this is the information we have gathered for each carrier:

Table 17 Overall Dimensions

Carrier	Length (in)	Width (in)	Height (in)	Volume (in^3)
USPS	24-1/16”	12-1/4”	6”	1769
UPS	N/A	N/A	N/A	1728
FedEx	15-3/4"	14-1/8"	10-3/4"	2392
Amazon	23.45	12	7.65	2153
Largest dimensions	24-1/16”	14-1/8”	10-¾"	3696

Using the largest dimensions allows for the introduction of the largest most common boxes to fit inside S.M.A.C. comfortably. However, the dimensions of the box itself cannot be the exact as the box to be put in. There needs to be room for components and materials used for the box itself. After research of dimensions of the material used for the box and leaving some room for components adding about 4 inches to each dimension should create a box with enough space to fit either a couple of small boxes or the largest common size box comfortably with leaving space for components and box materials to fit. This creates the following dimensions for our box to be:

Table 18 Final Dimensions

Name	Length (in)	Width (in)	Height (in)	Volume (in^3)
S.M.A.C	28”	18”	15"	7560

After careful considerations, these dimensions fit perfectly into our design as it allows creative freedom on how to add the box into its surroundings. One of the goals for our project is that we want the box to fit harmoniously with the environment it is set in. One consideration is to make S.M.A.C. double as an outdoor bench or flower box acting as it was part of the environment instead as a secure box calling to be pried open.

Figure 16 Design examples. Permission to reproduce requested.

4.3.2 Materials composition

The material used on the box will be crucial for the performance of its functions but more importantly for the security and longevity of the product. The materials composition will take into consideration our prototype phase and ease of workability with the material which might skew our results a little. We will also be looking into what blends best with the environment and keeps our cost down. Wood, aluminum, and plastic were the chosen materials to consider as they are the most common materials used for production and are widely available.

Wood

Wood offers a lot of strengths yet still provides some glaring drawbacks. Wood's strength relies on its ability to aesthetically blend with almost any environment. This material can also give users huge ability to customize their S.M.A.C box any color they please. Wood can be painted over its surface easily in almost a million different ways or left untouched and still look great. This material also gives us huge amounts of malleability. Malleability will be a huge part in our design testing process allowing us to make changes on the fly without any heavy or special machinery. Wood is also durable in the long term however it is susceptible to decay if not taken care of correctly. Wood is also more expensive in comparison with our other competing materials in a mass production environment.

Aluminum

Aluminum is the strongest and most secure material of the list. However, aluminum poses a lot of challenges in its protype phase due to its malleability. To be able to make changes to the design or incorporate the components to its hull you need special machinery to be able to cut and shape it in the desired way. In addition, its high price compares to that of wood. Still it is the most secure option and the one that would essentially last the longest due to its rugged strength of the material. Nevertheless, ruggedness is also a drawback aesthetically as it offers very limited design choices for users.

Plastic

Plastic is by far the cheapest option and easiest to mass produce. Plastic however offers no security value as it is not as strong as wood or aluminum. Malleability is a hinderance here too because the material must be produced with all the intended designs from the get-go offering almost no room for change after it has been produced. In the aesthetic front its malleability in the initial production process allows us to give the plastic some style in design to better blend into the environment than aluminum does. Plastic also allows the manufacturer to offer more color options and designs without breaking the bank. Nevertheless, it takes complete power from the consumer to change colors or design once they own in contrast with wood.

Conclusion

Wood is the clear winner from a consumer and manufacturer standpoint. Even though it would raise the cost of our unit. The malleability and aesthetic incorporation into the environment make it a perfect choice for this project. The material fits more in line with our design goals and vision for the product. We do not want S.M.A.C. to be an eyesore or disturb the peace of the environment it is in. Considering current events accessing certain machinery is harder than usual making malleability a huge design choice for the prototyping stage.

4.4 UV Light

Ultraviolet (UV) light is a form of electromagnetic radiation which is shorter than visible light but longer than X-rays. UV radiation is mostly present on sunlight but can also be produced by electric arcs and specialized lights, such as mercury vapor lamps, tanning lamps, and black lights. UV does not have the power to be considered an ionizing radiation however, it can cause chemical reactions and cause many substances to fluoresce. Consequently, the chemical and biological effects are more than just simple heating effects and many of its uses derive from its interaction with organic molecules like germs and viruses. Short wave ultraviolet light damages DNA and sterilizes surfaces. Ultraviolet light scrambles and damages the nuclear material of germs and viruses causing their DNA to mutate and prevent them from reproducing properly. However, not all ultraviolet light kills germs. The electromagnetic spectrum of ultraviolet radiation (UVR) defined most broadly as 10–400 nanometers, can be subdivided into a number of ranges. Ultraviolet A or UVA is ultraviolet radiation in the wavelength of 400-315 nanometers. UVA is mostly emitted

by black lights and is the sort of radiation not absorbed by the atmosphere. UVA is known as soft UV. Ultraviolet B or UVB sits in the wavelength range of 315-280 nanometers. UVB is mostly absorbed by the ozone layer and is considered intermediate UV. Ultraviolet C or UVC sits in the 280-100 nanometer range and is completely absorbed by the ozone layer upon entry to earth. This short-wave UV is known as hard UV as it is the germicidal type. At this frequency most organic organisms are affected by its radiation and tend to mutate and cease to reproduce are result.

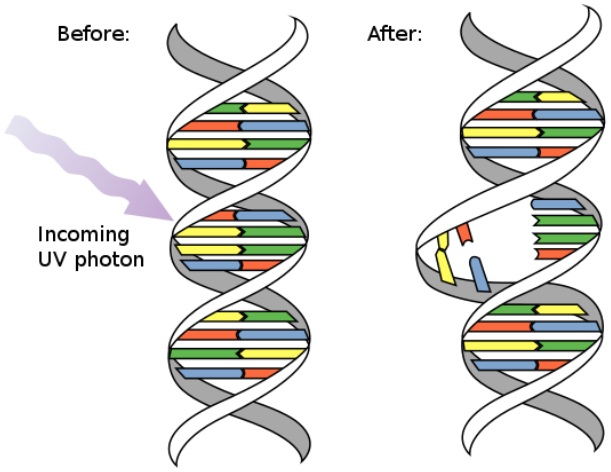


Figure 17 UV and DNA. Permission to reproduce requested.

With the established types of ultraviolet radiation, we can move on to explore the artificial sources of UV light. UVA lights are the most common and are found to be reproduced by black lights, ultraviolet LED's, and incandescent lamps. All these produce light in the 400-300 nanometer range which are not beneficial for our cause. UVB light is not commonly used for much so not many artificial light sources are made for that range. On the other hand, the other widely used type of artificial light for UV light is UVC which is mostly used for industrial purposes. These are short-wave ultraviolet lamps. Short-wave ultraviolet lamps are operating with an efficiency of 30-40 percent which make them inefficient but highly needed in certain applications where an area must be under constant sterilization. Ultraviolet LED's are trying to fix this issue as they can achieve a 50 percent efficiency however, they have not been verified their effectiveness in disinfecting. UVC LED's are quickly advancing and it will not be long before we see them develop further.

Using UV light as a means of disinfectant is nothing new. This application has been used in operation rooms, butcher shops, and water treatment facilities as it provides a versatile method of combating germs. There are certain areas were hard chemicals are not an option and this the only alternative. There are several devices that have been produced to kill a range of dangerous bacteria and viruses from MRSA to E. coli. One of them is a UV-light-emitting robot that quite literally zaps operating rooms clean of all pathogens. Even though it has not been proven or tested to kill the coronavirus scientist believe it to react no different than other viruses. Boeing has also developed a prototype for self-cleaning airplane bathrooms that blast the room with UV light after each use to keep it clean for the next person and minimize the risk of infections. This is where SMAC takes the spotlight

and uses these methods to minimize the cross contamination that occurs in the delivery of packages. Coronavirus is believed to be able to survive in cardboard for about 24 hours. Now a days with prime same day and one day shipping that cross contamination could happen anywhere and infect the user. Adding UVC to SMAC will minimize that cross contamination and prevent other viruses and germs from affecting the user giving peace of mind.

4.4.1 UV Light Hardware Option 1

Figure 18 Cold Cathode. Permission to reproduce requested.

The first hardware option available looks to be a hot cathode bulb that comes with in a complete set that includes the lamp tube, lamp holder, and power cord. This combo would help us keep cost down. The brand is Shellett and the model is ZW8S15Y-Z287. It produces a wavelength range of 185 – 254 nm which is the desired range and outputs 8W of power which is a good sign as it does not consume a large amount of energy and should be plenty for our application. The tube diameter is 15mm and the lamp length is 287mm not counting the lamp pins. The overall lamps length is 300mm. The most notable part of this compact Shellet offering is the features it packs in such a small design as being perfect for outdoors due to its ceramic lamp holder which is resistant to oxidation and radiation. The bulb has an average 8000 hours lifespan and without the need to turn it on and off constantly it might last longer. This straight double-ended quartz ultraviolet germicidal lamp is a good candidate for our project with a price tag of about $18.40 per unit.

4.4.2 UV Light Hardware Option 2

Figure 19 Hot Cathode. Permission to reproduce requested.

The second option is a cold cathode option that tends to be a little less reliable in its wavelength output but is able to turn on and off without preheating. The brands name is Lcamaw and model name is 8W-USDISL. It’s an a 8W 310mm lamp that comes in its own lamp enclosure which is a plus. The lamp is said to operate in the 254nm range, but as aforementioned cold cathode lamps tend to be unreliable in it production of UVC light. The good part of this lamp is that it has a rating of 10000 hours of use making it a good option to last longer than its hot cathode counterpart. The enclosure is made of aluminum which is not optimal on the long run but still is able to keep cost down in a mass production level. The cost per unit of this item is $8.99 which makes it much cheaper than its hot cathode counterpart.

4.4.3 UV Light Hardware Comparison

When looking for hardware options for the UVC light it gets tricky when researching the correct UVC lamp needed. There is a plethora of fake and incorrectly labeled hardware options in the market due to the current world events. A market has sprung for any item that can minimize the impact that COVID-19 has brought upon our daily life. This market of unqualified UV lamps ranges from misleading LED UV light knock offs to incorrect cheap wavelength lamps. It requires a keen eye to check that one is picking the correct wavelength light solution that can achieve a 180-254 nm wavelength. The spectrum between 180-254 nm is the prime spot were the nucleus of cells absorb the most radiation which is what makes them so effective against germs and viruses.

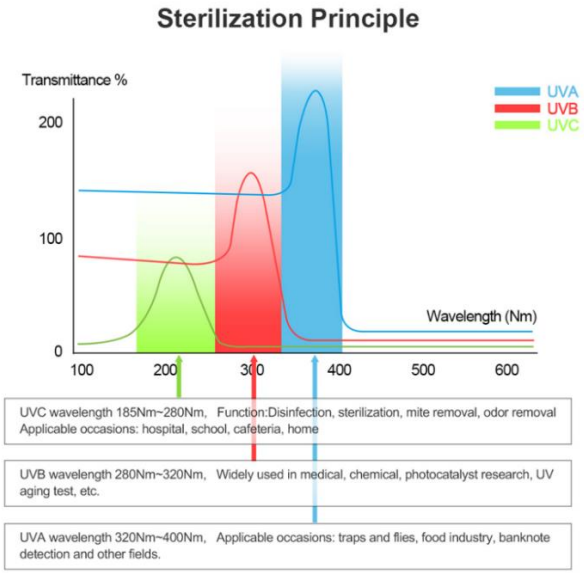


Figure 20 UV Spectrum. Permission to reproduce requested.

Once the correct wavelength light bulb is found one must consider buying either a cold or hot cathode bulb. Both have their own drawbacks and points of strength. Hot cathode bulbs are the most stable and deliver the best range of UVC light however they tend to have a short lifespan and must be preheated to be able to start and perform at the desired wavelength. These hot cathode bulbs are usually used when they do not have to be turned on and off multiple times, they are used in scenarios where they can be left on and untouched. However, these light bulbs achieve much higher power density than their cold counterpart making them the most effective UVC light source. Cold cathode bulbs on the other hand have a less reliable wavelength output and produce less efficient power output. The strong points of the cold cathodes fall under the pretext that they last longer and have no problem in being turned on and off constantly. The lack of preheating makes them better at outputting their desired wavelength and make them last way longer than their hot counterpart.

For this project with the research conducted I think a hot cathode bulb would make the perfect pairing for out project. The accurate wavelength and the lack of strain we would

impose on the bulb by not having to turn it on and off would make it a perfect candidate. This will allow a more effective and consistent disinfectant that a cold cathode would maximizing our project on its safety front.

4.4.4 Automation with MCU

The UV light will have an automated turn on and off function when a package is dropped off. The MCU will be able to read when the lid is opened and closed or manually turn on the UV light when instructed. To not damage the bulb a very strict timing on the bulbs activity and usage will have to be implemented. In addition, using too much UV light could cause damage to humans if exposed constantly to it. Recommended times for activation will be instructed and timers will be set, and the user will be notified when their package is disinfected and ready for pickup.

To achieve this, we will be using a style of smart outlet which will control the flow of electricity supplied. This can be done by using a 5V relay module to act as a switch for the circuit. This relay switch can be attached in any circuit or between any electrical outlet and its power source to control when an item turns on.

The UV light is recommended to be on for 30 – 45 minutes for maximum disinfecting effect. It is also recommended to leave it for another 5-15 minutes after turning it off to let the residual radiation dissipate. Two modes could be created depending on the user's choice one for fast disinfecting and another for maximum disinfection at two different time frames. The first mode, fast disinfecting, could turn the UV light on for 30 minutes and have a cooldown period of 5 minutes. In maximum disinfecting mode it could have a UV exposure time of 45 minutes and a cooldown period of 15 minutes.

Another caveat to look after is to ensure the UV light bulbs lifespan is to restrict the use the bulb with 1-hour intervals to not have it turn on with every delivery. Another solution could be to set a schedule time to turn on instead of with every opening and closing of the lid and implement a schedule right before the user gets home or whatever time works best for them. Warnings and a recommended usage section could be created within the app.

The app and MCU will create a vital role on the effectiveness of this portion of the project. A good management, various options, and recommendations of usage will impact how the user experience will be with this feature of S.M.A.C. A light bulb lifespan and replacement recommendations will also be used to keep track of when the UV bulb might be close to need of replacement. All these little things will allow for a safe and responsible use of the UV light portion of the project.

4.5 Power System

To talk about the power system, we must be able to talk about the design a little bit and its components. The power system is what is going to make all this possible and be able to bring all the components to work in unison. Each component has different voltage ratings specially for the microcontrollers and sensors which normally operate either under 5 V or 3.3 V. Aside from voltage disparities between sensors another issue is the current consumption. All components won't be able to be powered by just the Arduino as we would exceed the max rated current of 500 mA over USB power or 800mA over wall wart power. Sensors would require extra external power to be able to power all of them at the same

time. Not having the correct current requirements or being too close to reach the max current draw can impact sensors ability to work properly. All these power requirements must be compounded into a single power outlet cable and power everything in unison for ease of the user.

4.5.1 Main Power

All our power will be supplied by a single feed of 120 V. The main power will have to be split into different voltage regulators like a wall wart to output more manageable voltages into our components. S.M.A.C essentially will need to connect in series 3 120V power sources to power its components. The first component to power is the UV light enclosure which has its own integrated power converter and regulator. However, between the 120V source and light a 5V relay module as switch and will be used to control it from the Arduino to turn it on and off with commands from it.

The second device that needs to have its own power supply to maximize its efficiency is going to be the Arduino. The Arduino can be powered 2 different ways. The first is by feeding it 5 V through a USB connector which can output a maximum of 500 mA in total to all its components due to the onboard regulator. The second way is connecting a 2.1mm DC plug into its power jack. This configuration is by far the best option as it can supply a maximum of 800 mA and take any voltage in the range of 7 V to 12 V however it is recommended to use 9 V for best results. The Arduino just in case has a regulation circuit that can protect it, to a certain extent, from wrong voltage inputs and current surges. A 2.1mm 9V DC wall plug will be used to give the Arduino power for maximum efficiency.

The Third device that we need to power is the external power supply for the components. As mentioned earlier. This will need to convert voltage to two different voltages, 5V and 3.3V. There is one solution to this and it is to use a breadboard power supply module. The breadboard power supply module typically used in breadboards is a tiny but powerful voltage converted that cleanly converts voltage just like the Arduino from a 2.1mm 9V DC wall plug but into two voltages a 5V and 3.3V. This completes our three main components that need power.

All these components will be connected in series from a single 120V source just like they would in a power strip. The only exception will be the UV light with a 5V power relay for control of the bulb. Current draw should not be a problem in this section as we are powering directly from the wall and using relatively low power components. Current draw will be more prevalent in the next section when we talk about components.

4.5.2 Components Power

The components will be a vital part for this power section. This is where a single number or calculation can impact the project in a huge way. All voltages must be correct for each component and there needs to be enough current to feed all the different components in that load line. The following chart will show the voltages and max current requirements for each component:

Component	Voltage (V)	Maximum Current (mA)
Barcode Scanner	3.3	190

Fingerprint Scanner	3.3	120
NFC Module	3.3	140
Camera	5	140
Wi-Fi Module	3.3	240
Solenoid Lock	12	650
Magnetic Latch	N/A	N/A

Figure 21 Power Requirements

Usually in small projects an Arduino would be able to drive power to all its components however, once we start scaling up like we have for this project the Arduino falls short on its power delivery. The absolute maximum current an ATMega2560 can deliver through its pins is 200 mA which in our case with one device we are getting close to the max. This is where external power comes into play. We need to be able to drive more components than the Arduino can handle by itself. Two breadboard power supplies will be used to deliver the correct voltage and be able to drive up to about 700 mA of current each. This will give us enough room for to drive the components and be protected against any current spikes the devices might have. It is safe practice to give devices a little more current than their theoretical maximums as current is subject to draw spikes. Draw spikes are moments when the device can for a moment draw more current than specified.

On the 3.3 V line we have about 470 mA of combined current draw staying under our rated 700mA that a breadboard power supply can drive. On the 5 V line we have about 360 mA without considering the dynamic current draw that the servo can draw depending on its speed and voltage given. However, the breadboard still has a budget of 340 mA that it can provide the servo which should be more than enough to operate.

Each of the breadboard power supplied will be supplied with a DC 9V 1A power supply to convert the voltage most efficiently. The components will then be connected in series through the output of the breadboard. For simplicity one will be used for exclusively as a 5V line and another as a 3.3V line. Although these board can deliver both voltages simultaneously due to the high currents the components will be drawing, I want to put the least strain possible on the power supplies to ensure reliability.

4.6 Wireless Communication

Most microcontrollers do not include an integrated Wi-Fi module. These modules typically require the use of a system on chip (SoC) module. A SoC is an integrated circuit that integrates almost all components of a PC which almost always include a CPU, memory, and I/O ports. All these items are integrated in a single compact PCB. These chips are mostly used for radio frequency signal processing which is what our application of this SoC will be used for, Wi-Fi. This Wi-Fi module will implement the 802.11 standard protocols into our MCU allowing it to transmit and receive packets, gain access to networks, and process wireless signals. Wi-Fi modules now a days do more than just connect a board to the outside world. With technology getting smaller and more efficient everyday these SoC's have been able to pack more capabilities and processing power allowing them to also use Bluetooth technologies and other connection types.

4.6.3 Wireless Module Options

Wi-Fi capabilities are a necessity for S.M.A.C as it needs to communicate with the outside world. These communications range from approved package list to simple opening commands from a remote device. The search for the correct Wi-Fi module will be based on a specific set benchmarks which include the following:

- Cost effectiveness
- Ease of integration
- Wireless capabilities

ESP8266 ESP-01

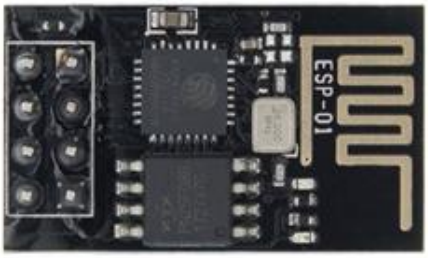


Figure 22 ESP8266. Permission to reproduce requested.

The ESP8266 is the most common Wi-Fi Module and its suitable for most microcontrollers that do not have Wi-Fi capabilities embedded into them. As result of its wide usage and popularity wide support has emerged from an active community. As a result, there exists many open source libraries for many different applications. The ESP8266 comes with a software development kit which provides sample codes for many common applications. This SoC can operate as a standalone application or can serve as a slave to a host MCU. The ESP8266 communicates with the MCU through either SPI or UART protocols.

The ESP8266 conforms to the 802.11 standard and has 802.11 b/g/n support. Within the 802.11n support, it can process up to 72.2 Megabits per second.

The 32-bit Tensilica processor on this Wi-Fi module runs at a typical speed of 80 MHz with a max speed of 160 MHz This is what makes it possible to achieve the 72.2 Megabits per seconds. The processor is an inexpensive alternative as it performs at a decent clock speed, but it is still on the lower end side of modern Wi-Fi module processor. There is only one issue with the ESP8266. The SoC is powered using 3.3V but the ATMega2560 UART pins are powered by 5V. For this reason, a voltage divider will need to be used when connecting the two in order to convert the voltage to a safe range for the ESP8266 to use.

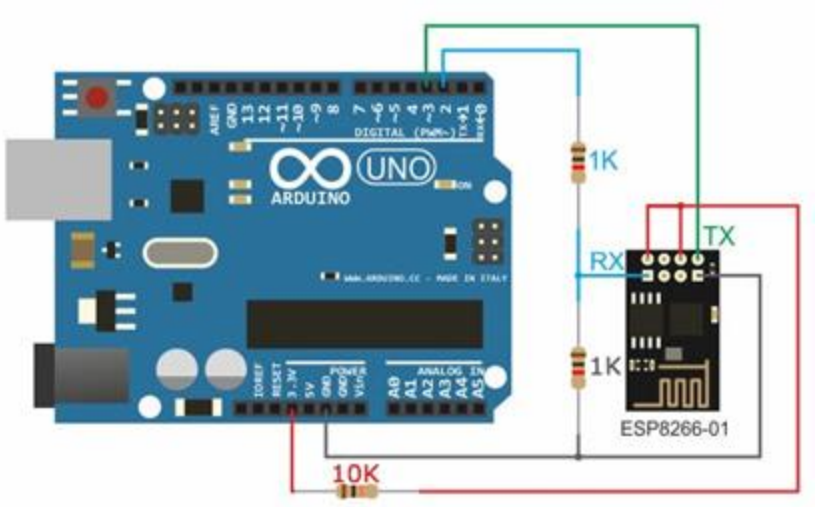


Figure 23 Connection Schematic. Permission to reproduce requested.

Once configured correctly the SoC can be configured into three different Wi-Fi infrastructure modes:

- Station mode
- Soft access point mode
- Station + soft access point mode

These different modes create different operating conditions for the Wi-Fi module. In station mode for example the module connects to an established Wi-Fi network by an access point. Once in this mode many features to manage Wi-Fi connectivity are granted. One of those features is to automatically reconnect to an established access point. This can be achieved by saving the authorization information previously used in its flash memory.

On another hand, the soft access point mode sets the module to allow other Wi-Fi devices to connect. These devices that connect to the module can be described as stations similar as the operating mode. The lack of wired connection that the module has is what give this mode its name. with lack of expansion through ethernet. The module earns its soft access point naming. We will not be using this mode for this project.

For this module to be active through code it would only need one parameter. In this case the parameter needed is the SSID while the other parameters such as password and channel are optional. The final configuration possible for the module is the station and soft access point mode. This setting combines the function of the modes. Essentially the Wi-Fi module can connect to a Wi-Fi access point and provide network access to other devices in a synchronous fashion.

Our main use for the module will be to use it as a means of communication with for the MCU which will result in this module being configured in station mode. It is still a good idea to have a Wi-Fi module with different modes if we want to add more functionality in the future. A rough sketch of how our communication would work would be something like the picture below.

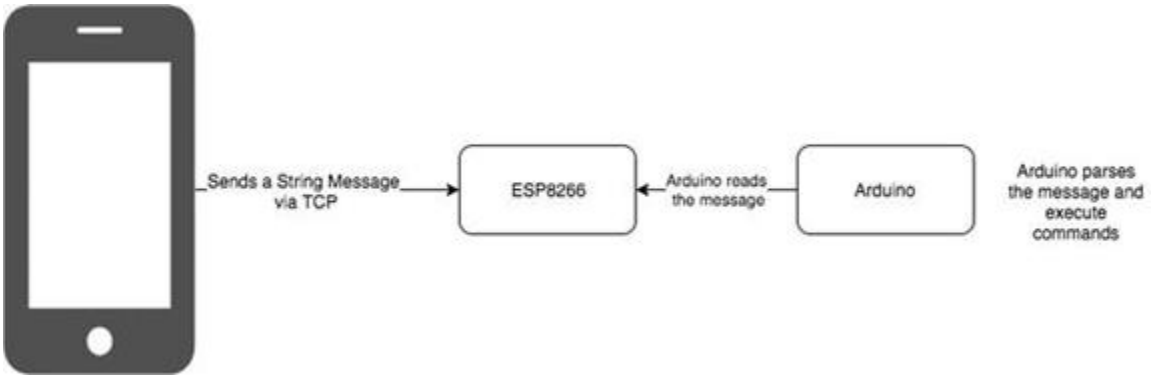


Figure 24 Protocol Schematic

Lastly, this SoC is the most inexpensive out of the bunch coming at just 4-6 dollars per unit. This is largely due to the underpowered processor it has, and its primitive capabilities compared to other modules. This SoC was one of the first Wi-Fi modules made for MCU's to be able to connect to the internet as Wi-Fi became more widely known. The cheap price tag is also a blessing in comparison to other modules as more accessibility is usually a sign of more support and this is no exception. Even though the ESP8266-ESP01 is not a flashy chip packed with features it gets its message across just fine when projects aren't demanding.

In conclusion, the ESP8266-ESP01 provides almost any MCU with wireless capabilities with a dynamic range of options on how to connect to the MCU. The Module is the most inexpensive solution but limited by its lack of extra features and weak processor. It's a compelling chip with simple use and support that is very functional for our project. This Wi-Fi module check 2 of our demands list by being a perfect choice for its ease of use and cost effectiveness.

Texas Instruments CC3200

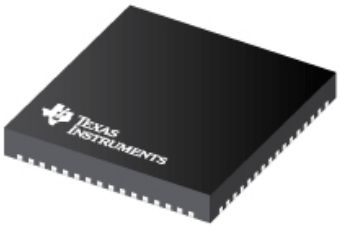


Figure 25 TI CC3200. Permission to reproduce requested.

The CC3200 is a small powerful chip that contains an ARM Cortex-M4 MCU giving the user the choice of fabricating their own applications. This SoC is very different from the ESP8266 ESP-01 as it is not as simple to use and apply to most MCU's. This Wi-Fi module being made by such a big name in semiconductors makes it a little more restrictive in compatibility and kind of ties you down to the Texas Instruments ecosystem. The CC3200 is a small powerful chip that contains an ARM Cortex-M4 MCU giving the user the choice of fabricating their own applications. This SoC is very different from the ESP8266 ESP-01 as it is not as simple to use and apply to most MCU's. This Wi-Fi module being made by such a big name in semiconductors makes it a little more restrictive in compatibility and kind of ties you down to the Texas Instruments ecosystem.

Even though the CC3200 is in a more closed ecosystem it still has powerful features that the ESP8266 does not have. The CC3200 can also transmit data through 802.11 b/g/n baseband but includes stronger encryption technologies for more secure communication. These technologies include a fast-crypto engine that keeps communications safe and secure with 256-bit encryption. Its most impressive features are the fact that this SoC also doubles as an MCU. Code can run natively without the need of a master MCU to do the heavy lifting. Lighter applications or code can be put into this SoC which makes it perfect as a standalone smart home integration tool.

This Wi-Fi module has three operating modes like the ESP8266. They are the following:

- Station mode
- Access point mode
- Wi-Fi direct mode

Wi-Fi Direct certified devices enable Wi-Fi devices to directly connect to each other without a nearby centralized network. For application examples, mobile phones, PCs, and gaming devices can connect to each other directly to transfer any content or similar applications. It is like Bluetooth but Wi-Fi Direct can handle around 10 times the data that Bluetooth can making it a great choice for peer to peer connections that need to transmit large files. One of its biggest advantages to allow devices to connect to it and act as a medium for information sharing when there is no network available.

In comparison with the ESP8266 this Wi-Fi module has less processing power than the latter. Its ARM Cortex M4 chip core clocks at only 80MHz. However, do not judge this chip by its lack of power as it packs a plethora of smart features which make up for its limited clock speed. One of its hidden strengths over the ESP8266 is that it can operate at a wide range of voltages rating from 2.1 to 3.6 volts making the chip more flexible to its MCU. Another hidden but sometimes overlooked functions is its pin multiplexing technology giving the user the freedom to choose what serial communication protocol one needs or has available on their main MCU.

The biggest drawback of the CC3200 is its price coming in at a whopping 40 dollars. Having an integrated MCU into the Wi-Fi module makes things a little more expensive and out of reach for someone that will not take full advantage of the MCU this chip has to offer. With the right application it can be worth the money.

The CC3200 is a versatile small Wi-Fi module packing a punch with features that make the ESP8266 blush at times. Although it is not the fastest most straight forward offering using its wits and hidden features it is able to stand toe to toe with its competition. Its ability to operate without an MCU make it fantastic for creating different uses rather than just acting as a slave to an MCU. The biggest drawback is that versatility and features come at a cost. Since this chip is made by one of the biggest semiconductor companies in the world and pack a separate MCU chip for standalone processes the SoC has a quite high price tag in comparison to its competitors.

ESP32-DevKitC-32D

The ESP32 is a variant of the ESP–8266 Wi-Fi module. The ESP 32 is a generational upgrade of its predecessor the ESP-8266. The ESP-8266 looks primitive compared to the

Figure 26 ESP32. Permission to reproduce requested.

feature set this little guy carries. The ESP-32 has Wi-Fi and Bluetooth capabilities giving us an extra feature to play with. It earned the name of ESP32 due to its 32-bit architecture and clock speed of up to 160 MHz on a dual core processor allowing. This allows the device to pack more features and achieve a higher data transfer speed. This module can achieve 150mbps in an 802.11 n (2.4 GHz) configuration. That is double the ESP8266.

The module can support 802.11 b/g/n/e/i protocols as well as Wi-Fi Protected Access (WPA), WPA2, WPA2-Enterprise, and Wi-Fi Protected Setup (WPS). The biggest benefit of this module aside from its new features is the inclusion of Bluetooth technology. Bluetooth could allow our project to have an alternate way of connecting if Wi-Fi is not available or when S.M.A.C. is setup in a remote location were this could commonly happen with a device of this nature. Most homes in the U.S. have inadequate Wi-Fi setups which could hamper the ability of our project to natively connect to the internet.

The new architecture that the ESP32 sports also comes with some GPIO benefits. They added pin multiplexing which allows you to decide which pins are UART, I2C, SPI – you just need to set that on the code. This might seem like a small improvement but allows for much refined control of your pins specially when dealing with limited interfaces of an MCU. However, the crowning jewel of this Wi-Fi module is its ability to use 5V intake making it a perfect pairing for our ATmega2560.

While still being part of the Express if family and a cousin to the ESP8266 the ESP32 enjoys a similar level of support as the ESP8266 however with the generational changes some libraries may not work and would require some tweaking to make work on the newer platform. The ESP8266 also has had a longer lifespan making it have the largest library in comparison meaning that even though it iss possible to adapt all libraries to the newer ESP32 it would be an extra step and time spend on fixing code from existing libraries. The last drawback of this amazing SoC is the fact that all generational improvements come at a price. Even though it is still an affordable option compared to other SoC's the ESP32 is about on average of 6-12 dollars. That is about double the price of the much simpler and cheaper ESP8266. How much the extra functionalities weight in will be the determining factor of which module this project will use as its communication medium.

Conclusions

After a long thought process and thorough consideration many factors come into choosing the right module for our project. To recap the three Wi-Fi modules, we considered were the ESP8266, the CC3200, and the ESP32.

The ESP8266 had an 80 MHz processor that could boost up to 160 MHz, transfer data at 72.2mbps, operates at 3.3V, and cost around 3-6 dollars. Its major drawbacks fall with how basic the functionalities are. However, it makes up for it with its wide support and libraries available online.

The CC3200 had an integrated MCU, Wi-Fi direct technologies, a stronger encryption engine, and multiplexing pins for more choice with the communication interface. The major drawbacks it has is its closed ecosystem within T.I. and its huge price tag of around 40 dollars per unit.

The ESP32 has a huge library of pros. This small but powerful generational improvement over the ESP8266 packs a 160 MHz dual core processor able to achieve 150mbps over Wi-Fi. This module also has Bluetooth and BLE technology integrated into the same chip giving it more uses. Pin multiplexing was also an addition to the arsenal this MCU packs. Only drawbacks are the increased price tag that it carries over its predecessor and its newer architecture creates some issues with code compatibility with ESP8266 wide library.

The clear winner out of the list that met all our requirements, and more was the ESP32 Wi-Fi module. It met the cost-effective requirement, ease of integration, and had the best wireless capabilities out of the bunch. The fact that it already operates in a 5V range made it extremely easy to integrate in comparison with the ESP8266. Its price increase is minimal in comparison to the features it packs and the extremely useful pin multiplexing a commodity that the CC3200 had but for a luxury price. This module should give a look into the future with Bluetooth integration if we were to include in a future update or use it to enhance features. The ESP32 will easily integrate into our Arduino IDE and pair well with the ATMega2560.

4.7 NFC

NFC – Near-field communication Is a set of communication protocols for communication between two electronic devices 10cm or less in distance. Near field communication provides a low speed connection with an easy set up that can be used to create a self-started process for wireless connections. Most famously NFC are used in the financial sector for its ease of two-way communication. For example, apple pay will allow for users to pay at points of sales by letting the user hold their authenticated apple device to the point of sale NFC reader. The service adds a layer of security, NFC enabled devices act as an electronic identity document that services as an authentication mechanism for both users of the service. We thought it would be a great tool to use for our system = to provide an easy way of opening our system whenever a mail man makes the delivery.

4.7.1 Near-Field Communication – Design

NFC operates at 13.56 MHz on ISO/IEC 18000-3 air interface and at rates ranging from 106kbits/s to 424 Kbits/s. The NFC always need something to initiate the protocol between the person who initiated the protocol and the target. When the near-Field Communication is generated a RF (Radio Frequency) that has the capabilities to power a passive target. The NFC tags contain data and are typically read only but can be changed to write to as well.

The NFC uses a concept known as inductive coupling which just means two conductors are configured such that a change in the current through one wire induces a voltage across

the ends of the other wire through electromagnetic induction. This inductive coupling between the two nearby by loop antennas create an air-core transformer. Since the distance involved that are involved are small with comparison to radio waves the interactions are named to be near field.

With this we only allow for alternating magnetic field so that not much power radiates in radio waves. When doing this, it essentially prevents with accuracy the interference between both devices and any other RF (radio frequency) that communicating on the same frequency or If somehow a device with NFC capabilities trying to communicate when its far out of the range. They work on the widely use radio frequency ISM band of 13.56MHz. Working distance with the standard antennas and the power level that could be potentially up to 20cm.

The communication that takes place between a device that starts the initiation, and the target device could be either Passive or Active.

Passive

The device that begins the initiation has a carrier field and the end device that is the target, it essentially is a transponder (a device that receives a signal sends back a different signal), It ends up modulating the incident field. When the target device is in the passive mode it can get Its operating power from initial magnetic field.

Active

When we have the active mode on both the initial initiator and the end target device, they create their own fields alternatively. The device then will stop transmitting so it can receive data from the other device. All NFC have the capabilities to work in one or more of the three modes available.

NFC card emulation

NFC card will enable NFC capable devices to have like an iPhone to replicate a smart card, it will allow user to make simple payment transactions or even ticketing.

NFC reader/writer

The NFC reader/writer will enable NFC capable devices to have the ability to read stored data on an NFC tag that inside a label, poster or virtually anywhere.

NFC peer-to-peer

If we have two devices that are both NFC capable, we can use both those devices to communicate between each other. They both will be able to exchange data between each other.

4.7.2 NFC Communication – Standards

The NFC standards for the communication protocols and the formal exchange of data formats all cover the communication protocols. These protocols are based on the existing RFID standards that include ISO/IEC 14443 and the FeliCa. NFC is standardized in ECMA-340 and ISO/IEC 18092 The standards are used to elucidate the modulation schemes, the internal coding, the frame formats, the schemes creations, the proper requirements for data-collision control when both the passive and active NFC modes and even the transfer speeds of the communication. The standards also create the transport protocol which encapsulates the activation and the methods of data exchange.

4.7.3 NFC– Hardware Options

The most common hardware the industry uses to transport data is NFC tags. These NFC tags are passive data stores. These data stores can be read and with some circumstances be written to, by an NFC enable device. This device space capacity is between 96 to 8,192 bytes. Some common applications for using an NFC tag include securing personal data storage (debit/credit cards), some sort of loyalty program, or a way to have personal identification. The personal identification would be based on the person PIN. For our project we would like to use an NFC tag as a method to correctly identify whether this is the correct mailman to be at the box. As well even a way to correctly have a method to identify if that's the mailman to deliver then if so, unlock the box and place the package in there, the notify the consumer that their package has been deliver in a safe manner.

4.7.4 NFC – Possible Implementation's

With our project we had two ideas on how to potentially use NFC tags as a potential way to have an authentication method or just simply a tool to direct the delivery driver to the app by waving his phone over the delivery system through a NFC tag. Our first Idea was to first to use the NFC reader/writer model that lets you send and receive data from a mobile device to an NFC tag, we would have that NFC tag immediately direct the delivery driver right to the login in screen to the app so he could quickly login and being to scan in the items to unlock the box and store the potential packages to deliver. Our second idea was to maybe implement a possible authentication method where to NFC tag can recognizes a specific id the mobile device would have to potentially signal that this is the correct delivery driver and not the wrong person but we came to conclude that It may be to convoluted to implement.

4.7.4.1 PN532 NFC RFID Module Option 1

One of the options we could use for the NFC's hardware is the module named PN532 NFC RFID, with this module it is one of the most used ones and affordable. With the module it

is created around the NXP PN532. With the NXP532 chip all the I/O pins exist on the module so it compact and small. The module is Arduino compatible which is super useful because we can now fully utilize all the supported libraries the Arduino has to offer for the module. With this module it has the following features:

- This module supports SPI, HSU (High Speed UART), and I2C.
- The module has a typical 5 volts reserved for the I2C and the UART and reserves 3.3 - 5.5 volts for the SPI.
- The module is capable of near field communication with android enables devices.
- The module as well has a PCB antenna that was printed on to the board.
- RFID reader/writer mode support for multiple smart cards.

Possible Implementation:

When implementing the module Arduino provides open source libraries for the PN532 module. To get the module to work you just need an Arduino that can work with it. On the module the I2C and the HSU both share the same exact pins with the HSU mode being set on default mode. After you implement the HSU interface onto an Arduino. Now we can use the ISO1443A smart card as a form of an authentication method for the delivery man only allowing him to have access to that card. When the mailman waves the card over the NFC tag the box will open and only then will they be allowed to store the clients package within the box, so they feel assured it was a safe transaction.

Cost: $9.66

Manufacturer: KNARCO

4.7.4.2 NFC Hardware Option 2

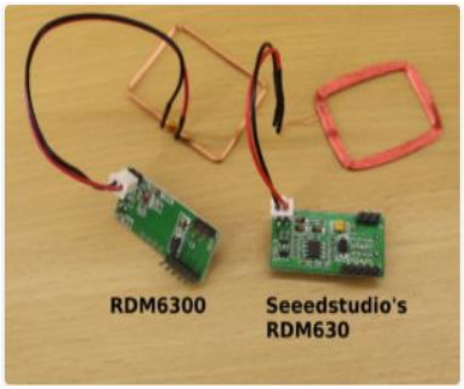


Figure 27 Seeedstudio's RDM630 and RDM6300 NFC Reader. Permission to reproduce requested.

For our second hardware option for the NFC we came across Seeedstudio's RDM630 and the RDM6300 RDID readers. These readers are similar, but one is far better in terms of functionality and overall performance that one being the RDM630 RFID reader. With these to readers they use the same protocol for them via serial connection. Since It does that, we should be to communicate date in bits now. How both work both RFID readers is first they are equipped with these strong mini antennae to help them identify the EM4100 compatible tags that are within near proximity to the reader. When a user would waive the tag across the antenna it would trigger the reader to be switched on to enabled now.

Cost: $12.50 per unit

Manufacturer: SeeedStudio

4.7.4.3 NFC Hardware Option Comparison

When comparing these two NFC hardware options I think we must be mindful to what features would serve us the best to correctly solve our problem in an efficient way that will derive a good product and result. Between our hardware option 1 or 2 we felt that the RFID module on that correctly served us in our goals and intention. With that option we could use their RFID readers to read in the smart cards and would use that as a potential way to open the box and correctly identify whether that was the correct person that should have opened the box or not. As well it was an overall more versatile piece of hardware in comparison it seems to be more universally used compared to the latter. For us it was the better option because it fit our specific needs overall.

4.7.4.3 NFC Hardware Option Comparison Table

In this table we detailed the important specifications we want to focus on for our implementation. For our design one of the most important specifications is the supply voltage. We have to keep this in mind cause of our other components supply voltages their needs.

Table 19 Specifications for NFC RFID Module and Seeedstudio's RDM630

Specifications	PN532 NFC RFID Module	Seeedstudio's RDM630
Supply Voltage	**5V**	**5V**
Supply Current	**< 50mA**	**< 50mA**
Dimensions	**2.54mm X 2.54mm**	**38.5mm X 19mm X 9mm**
Baud Rate	**115200**	**9600**

4.7.4.4 NFC Hardware Choice

With our analysis of the potential NFC hardware choices we researched, we concluded that the PN532 NFC RFID Module will be our best option. One of the reasons we decided to go with the PN532 NFC Module is because of some of its functionality features that come with the component. With our Smart delivery system, we want the mailman to have a smart card that could open up the box with ease. With this module it allows for such functionality. When the module reads the smart card, we can develop a simple protocol to unlock the lock whenever our PN532 NFC Module correctly reads the card when someone waves it over the module or the tag.

4.7.4.4 Android API Options

The android API allows for you to share tiny payloads of data between an NFA tag and a device that operates on android or with two NFC enabled android device. The data that is stored in these tags can also be written in a multitude of formats, but the android framework API adheres to the NFC Forum standards as discussed above the standards of the NFC Data Exchange Format.

The two main use cases when we work with NDEF data and Android powered devices is:

- When we want to read in the instructions (NDEF tag) from the NFC tag.
- When we want to beam NDEF messages from one android device to another android device.

The android API's handle reading in NDEF data from the NFC tag. When the android device picks up the signals it gets handled by the built-in tag dispatch system. The tag dispatch system repeatedly when the screen is unlocked runs so it can discover any NFC tags in the area and then proceeds to parse in the data wanted. When the application open It will take care of the read in NFC tag and it will create an intent filter then request the data. Androids tag dispatch system scans in the NFC tags, parses it then tries to locate the applications that is of interest to the user by doing series of protocols. It first parses the NFC tag trying to figure out what is the MIME type and then tries to correctly label the data payload on the tag. Secondly it will then Encapsulate the MIME type and the payload, and it will create an intent. Lastly it will then start the activity that was based on the intent created. When the tag dispatch system is finished cresting the intent that will ultimately encapsulate the NFC tag and all the necessary identification information it will send it off to an application that is of interest for the user to choose the activity.

4.8 Open / Close Sensor

For this section we dive into the uses of an open and close sensor and detail which options we have for our project. In this we detail which open/close sensors we will be analyzing for potential use for our project. We analyzed both a reed switch and then a hall effect sensor to determine which component we will use for the delivering system.

4.8.1 Open / Close Sensor: Room Entry Sensor

Figure 28 NTE Electronics Reed Switch. Permission to reproduce requested.

This is the NTE Electronics Inc Reed Switch which gives the user an actual real time detection of when some type of door opens. Most of the time how most engineers use it is to mount the wired Room entry sensor to the intersection where the door meets the frame where the door shuts. When the door opens the two components separate and then the senor will trip. When the two components get separated, they get triggered when they reach a certain point cause the components use a magnetic field that trigger the sensor to detects when a door is open. These types of sensors are used in many security systems to detect when a door is opened or even a window is opened. When they two components get separated, they trigger the security system set in place. How this sensor works is uses a reed switch that open when there is no magnetic field near hence it is open and when a magnet is near the switch, the switch closes hence the door is closed**.**

Features & Specification for the Reed Switch:

- Constant Rating: 10 W/VA
- Contact Resistance: 150miliohms
- Switching Voltage 200 VDC Max
- Switching Current 500mA max
- Terminal Type: Wire Leads
- Life Expectancy: 20,000,000 cycles

Implementation for our project:

For our project, we have a delivery system that would use these contact sensors to determine when the box is open or not. The sensor would trigger a small beep sound on the box to notify the mailman that the box is now open and send a signal to the client apps that the box is now open. Most developers implement the sensor by using an Arduino and a breadboard with a couple of components added to have it functionally. The magnets have wires attached to it so we would connect the wires to the Arduino to have c code to give it instructions to do what we want the system to do but as well join it to a breadboard to have a component on there that would set off a buzzer to sound an alarm to notify the mailman that the box is opening.

Cost: $2.30 per unit.

Manufacturer: NTE Electronics

4.8.2 Open / Close Sensor: Hall effect sensor - US1881 (Latching)

This is an integrated Hall-Effect latched sensor. When you hold a magnet close to the sensor it will result in the output being toggled. When doing this it creates a strong sensor. When comparing the hall effect sensor and a reed sensor the hall effect sensor is smaller and handles less current then the reed switch. With the Hall effect sensor, it has a hall voltage generator for magnetic sensing, a comparator that will amplify the hall voltage, an open collector output and a Schmitt trigger to switch hysteresis to reject noise. How it works is the magnetic flux density becomes larger than the threshold allowing for the DO to be turned on, then to have the component turn on to the output state is when the magnetic influx density falls under the Brp leading to the DO it be turn off to the off state.

FEATURES:

- 3.5V to 24V DC operation voltage
- Low current consumption
- Temperature compensation
- Wide operating voltage range
- Open-Collector pre-driver
- 50mA maximum sinking output current
- Reverse polarity protection
- Lead Free Package: TO-92

Important Specifications

In this table we detail all the important specifications we thought we relevant to make a decision on which component to use for our design.

Table 20 Specifications for the component.

Parameter	Symbol	Test Conditions	Min	Type	Max	Units
Supply Voltage	V_{dd}	**Operating**	**3.5**		**24**	**V**
Supply Current	I_{dd}	$B < B_{RP}$			**5**	**mA**
Output Saturation Voltage	$V_{ds}ON$	I_{out} **= 20 mA**			**0.5**	**V**
Output Leakage Current	I_{off}	$B < B_{RP}$ $V_{out} = 24\,V$		**0.3**	**10**	μA

Cost: $ 0.95 per unit

Manufacturer: Allied Electronics and Automation

Implementation:

When implementing the hall effect sensor, the most common way developer did it was by using an Arduino and a magnet. To set it up we connected the hall effect sensor to the Arduino and to the breadboard with a resistor. Essentially after we implement it works like this, once the Hall effect sensor detects the magnet on the edge of the door when it closes, and it outputs a high 5V voltage to the Vout pin. The Vout pin is connected to the Arduino interrupt pin that senses the rise in voltage.

4.8.3 Open/Close Sensor Component Comparison

For this section we made a table to compare important specifications from the datasheets from the parts we are going to potentially used. We need to know these important

specifications for design and implementation purpose to know how much voltage and current will be use in our system. These specifications are as well important to determine which component we will choose for our final implementation.

Table 21 Specifications for Hall effect sensor and NTE Electronics Reed Switch

Specifications	Hall effect sensor: US1881	NTE Electronics Inc: Reed Switch
Supply Voltage	**3.5 - 24 V**	**N/A**
Supply Current	**5mA**	**N/A**
Switching Current Max	**N/A**	**500 mA**
Switching Voltage Max	**N/A**	**200 V**

4.8.3 Open / Close Sensor: Component Choice

After a detailed analysis of the reed switch and the Hall effect sensor we decided to pick the reed switch because we felt that it fit in better for our specific needs. With the reed switch it has a very useful Arduino library that makes the implementation of the component efficient. With the library it has predefined states to tell us whether the door is open or not, we determined having those predefined states is crucial to the implementation. As well with the reed switch, we felt on a mechanical level the component fit well with the door of the box which will make the implementation efficient. With the component it has an ease of implementing the component on the door that the Hall effect sensor does not have.

4.9 Locks

A smart lock mechanism will be used to lock and secure a user's packages upon delivery. Using different unlocking features implemented, the box will unlock only by a unique and verified user. The box can either be locked or opened via a mobile device, fingerprint scanner, or barcode reader. The structure of the smart lock is a limit switch that unlocks, or locks based on positive feedback. This feedback will be monitored and controlled by a microcontroller. The power supply will aid in applying a voltage towards the locking mechanism that will both lock and unlock the box. A smart lock is a Wi-Fi or Bluetooth-empowered keen home gadget that permits clients to bolt and open an entryway by imparting secure signs from a portable application on their cell phone, or any mobile device. Wi-Fi connectivity allows a user to view entry and exit logs in real-time so that a user will know when their box has been open or closed. This will aid a user to know when the box was last opened. Get alarms on your cell phone on the off chance that somebody attempts to break in or alters the shrewd lock. The images showcased below illustrate the different locks or lock devices we will be considering. Most of the images shown below use a locking mechanism called a lock actuator.

4.9.1 Smartphone-controlled, Deadbolt Actuator

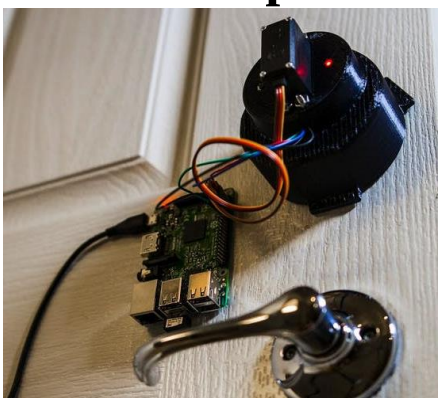


Figure 30 Deadbolt Actuator. Permission to reproduce requested.

This deadbolt actuator is an internet-connected lock that is controlled from a mobile device. The lock can let a user know when a door is locked or unlocked. (Shack, 2017) Through this notification feature, a user will be able to understand when someone locks or unlocks the door. Within the deadbolt's actuator, a pure LED is located inside its infrastructure. A green LED light will indicate to a user when the door is locked. A high torque metal servo will be used to operate the locking mechanism for the actuator. We found that a 3D printed enclosure would be needed to hold the servo in place through our research. The 3D printed housing mounts on the front side of the box, which will activate upon a push of a button from a mobile app. When your entryway lock lodging is mounted, you can press a nearby in our portable application and utilize the catch to open or close the lock.

4.9.2 Smock (Your Very Own Smart Lock)

Figure 31 Smock. Permission to reproduce requested.

Smock is another choice of a smart lock design we considered and researched for our locking mechanism. Similarly, this smart lock will be internet connected and will allow a user to both lock and unlock the box from a mobile device. Smock was conducted by using various programming and networking tools. The main aspects of Smock are the software, electronics, and hardware. Each part of these aspects plays a vital role for the overall construction of Smock. (Ranasoraus, 2015) Throughout our project writeups, we will be focusing mainly on the hardware aspects. Smock was constructed on a Raspberry Pi module. The Raspberry Pi module is a versatile microcomputer that is commonly used in the tech community. A Wi-Fi dongle was used to connect the Smock device to the internet through the Raspberry Pi module. Additionally, the Wi-Fi dongle enables the lock to gain access to a user's database and other online resources. A Stepper engine which will draw current from the PI's capacity and will have the option to move dependent on our Program setting singular pins on the board to High and Low. The components needed to construct this smart lock are shown in Table 22.

Table 22 Parts for SMOCK

Parts
(1) Raspberry Pi
(1) Stepper Motor
(1) L293D Driver
(1) GPIO Header
(1) Breadboard

4.9.3 Morning Industry RF-01SN Radio

Figure 32 Morning Industry RF-01SN. Permission to reproduce requested.

The Morning Industry lock is a remote-controlled door lock that is controlled via a mobile device or a controller. The Morning Industry Remote Control Deadbolt replaces existing deadbolt without additional drilling. (Innovative, 2018) Based on this lock's features, a user can remote control their lock to both lock and unlock the box for quick and secure entry to their packages. A user will be able to remote control their box's lock, up to 30 feet away. Opening your box from 30 feet allows for a quick and secure entryway to your delivered packages. To operate this automated lock, the lock uses four AA Alkaline batteries that are needed to power the lock. The lock's overall design makes it easy to install and fit within standard doors without any additional fittings. This progressive door handle perceives up to at least 15 remotes to concede access to various clients. The auto-lock highlight will secure your entryway consequently 10 seconds after each open section. Every unit accompanies two remote controls and two regular back-up keys for manual activity.

4.9.4 Lock-style Solenoid - 12VDC

Figure 33 Lock-Style Solenoid. Permission to reproduce requested.

A solenoid lock is an electromagnetic device that contains copper coils inside of its infrastructure. (takigen, 2018) Within these copper coils, there is an armature metal rod inside the middle. When voltage is applied to these coils, they become energized and pull the metal rod away from its original position. A solenoid lock is commonly used in the tech field as an electronic lock, which provides a bunch of versatility and usability. Based on the metal exterior of the lock, the solenoid is particularly reliable and robust. The lock-style solenoid has a slug with an inclined cut and a decent section to mount itself. It is essentially an electronic lock meant for an essential bureau or a way to secure someone's belongings.

Typically, the lock is dynamic, so any individual cannot open it once the solenoid slug stands out. It does not utilize any force in this state.

Table 23 Lock-style Solenoid Specification

Specifications	
Supply Voltage	12 VDC
Wire length	222.25mm / 8.75"
Activation Time	1-10 seconds
Weight	147.71g
Dimension	23.57mm / 0.92" x 67.47mm / 2.65" x 27.59mm / 1.08"
Max Current	650 mA

4.9.5 Locks Comparisons

When it came to doing our research for the locking mechanism, we were able to find four devices that would help reach our engineering needs or requirements. The three devices considered are the Smartphone-controlled Deadbolt Actuator, SMOCK, Morning Industry RF01 and Lock-style Solenoid. All the considered locks are electrically controlled and allow users to unlock their locks at safe distances. The Deadbolt Actuator contains a series of spur gears that are rotated by an electric motor. The last spur gear pushes a rack and pinion gear set that connects to the original actuator rod. Rack and pinion gearsets convert rotational motion to linear motion. A deadbolt actuator's motor spins the gears, which causes the clutch to swing out and lock the metal gear into large plastic gear. Through this movement of gears, the motor drives the door latch. Our choice for a locking mechanism came down to the Lock-style Solenoid. The Lock-style Solenoid is a pre-built device that has its infrastructure already established from its manufactures. A solenoid lock is commonly used in the tech field as an electronic lock, which provides a bunch of versatility and usability. A user will be able to remote control their box's lock, up to 10 feet away through the Lock-style Solenoid. In Table 24 we showcase a comparison between the different locking mechanisms considered.

Table 24 Locking Mechanism Comparisons

Specifications	SM-Controlled Deadbolt Actuator	Smock	Morning Industry	Lock-style Solenoid
Operating Voltage	5V	5 V	3.3 V	12VDC
Max Current	< 190mA	< 190 mA	<150 mA	650 mA
Locking Mechanism	Spur Gears	Spur Gears	Spur gears	Solenoid Copper Coil
Locking Range	30ft	28ft	30 ft	10 ft
Weight	2.1 lbs.	2.3 lbs.	1.3 lbs.	0.3256 lbs.
Remote Controlled	Yes	Yes	Yes	No
Price	$57.00	$65.00	$105.00	$14.95

4.10 Barcode Scanners

A barcode scanner will be used to scan the packages for a user. When a package has been scanned by the delivery driver, the box will open, which will allow the delivery driver to place the item inside. The overall goal for this design is to notify the user when there verified package had been scanned and delivered. Barcode scanners record and translate barcodes that have a striped image with alphanumeric digits underneath it. Scanners can read different types of barcodes that provide various types of properties and functionalities. The barcode scanner that we will be using, will send information to a database through a wireless connection. The digits that are below the striped patterns refer to a particular item. When a delivery driver scans the package, examining the numbers will pull up entry into a user's database. If the item or package is verified in a user's database, then the box will open for the driver to leave the package inside. There are various types of standardized tag scanners, a few scanners use lasers, while others use lights or cameras, to catch the standardized tag picture and transform it into an electronic code. In Figure 34, this image illustrates the background mechanism of how a barcode scanner works.

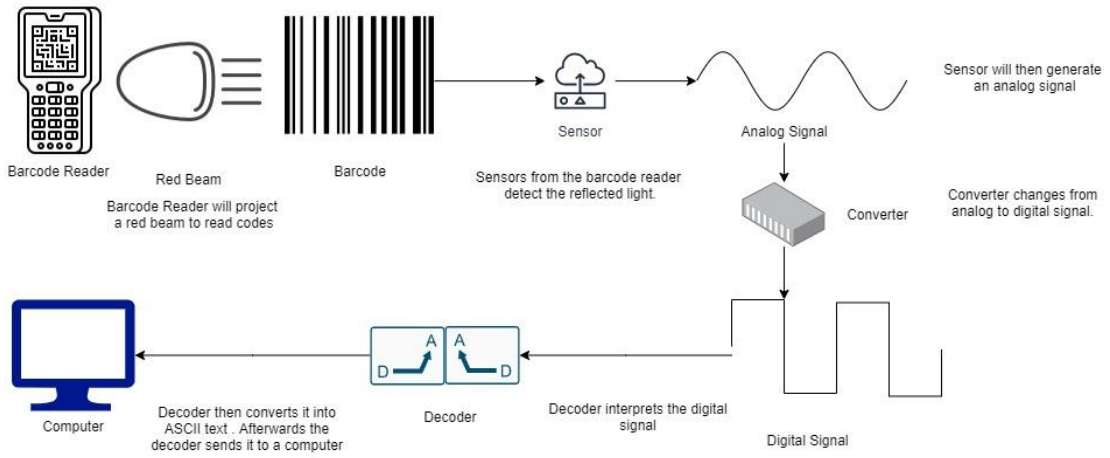


Figure 34 Mechanism of a Barcode Scanner

Barcode scanners are best used in businesses for looking up entries within a database. Additionally, a barcode is commonly used to track a user package or an auditing system. They can scan the alphanumeric digits and scan the barcode image from safe distances. Because of the way barcode scanners are structured, they have been built to last, which they can operate for many years. Barcode scanners commonly read two types of barcodes. These types of barcodes are 1D or 2D barcodes. (System_id, 2018) The 1D barcode is the most used image in many business and shipping companies. The images of a 1D barcode appear as a parallel line with some spaces. The spaces between a barcode are used to prevent the scanner from picking up incorrect details about a product. Likewise, a 2D barcode can store information both horizontally and vertically. The only primary difference a 2D barcode has over the 1D is that it requires a camera barcode scanner. In Figure 35, we showcase the two different types of barcodes side by side.

Figure 35 Left: 1D | Right: 2D

4.10.1 DYScan DE2120 – 1D/2D Barcode Scanner

Figure 36 DYScan. Permission to reproduce requested.

The DYScan Barcode Reader can read 20 different barcode symbology. (Sparkfun, 2017) Within these symbology, the reader can scan both 1D and 2D barcodes. The DYScan is equipped with a special camera to decipher barcode images using board image processing. (Sparkfun, 2017) Board image processing is a technique used to help a barcode reader identify and decode UPCs or QR codes. Two feature LEDs are found inside the structure of the DYScan. The first LED is used to illuminate the surroundings of the barcode reader. The second LED is used to project a red line that will scan different items from a distance. The red line seen from this barcode reader is also the same line we would see from laser-based scanners.

The DYScan module comes with a 12 position 0.5mm flat-flex cable. Since a 0.5 mm flat-flex cable is equipped for this module, it would be better to use a mating connector. Additionally, the DyScan is structured with two interfaces, which are TTL Serial and USB. The USB interface will operate by using a Full-Speed HID mode. The table below illustrates the DYScan module specifications.

Includes: DE2120 Barcode Scan Engine, 12POS 0.5mm Flat Flex Cable, Mounting Screws

Table 25 - Barcode Scanner Module Specifications

Specifications	
Supply Voltage	3.3 V
Interfaces	USB, UART
Scan Type	CMOS
Weight	< 4kg
Dimension	(L)16.0mm * (W)21.2mm *(H)12.0mm
Depth of field	25mm-400mm
CPU	32-bit
Max Current	190mA

4.10.2 Waveshare Barcode Scanner Module 1D/2D

Figure 37 Waveshare Barcode Scanner. Permission to reproduce requested.

The Waveshare Barcode module is a scanner that reads both 1D and 2D barcodes. (Waveshare, 2017) . The use of intelligent image recognition is a method used to decipher the barcodes on packages or items. The Waveshare module uses two types of communication interfaces. These interfaces are USB and UART, which connect directly to a device or computer. Since the Waveshare module has a small form factor, it can easily integrate into most devices. This module also includes an onboard light source that provides light for usage in dark lit areas. The table below illustrates the DYScan module specifications.

Includes: Barcode Scanner Module x1, USB type A plug to micro B plug cable x1, PH2.0 4Pin x1.

Table 26 Barcode Scanner Module Specifications

Specifications	
Supply Voltage	5 V
Interfaces	USB, UART
Scan Type	CMOS
Weight	< 4kg
Dimension	53.3mm x 21.4mm
Depth of field	25mm-400mm
CPU	32-bit
Max Current	135mA

4.10.3 MG65 1D 2D Code Scanner Barcode Reader

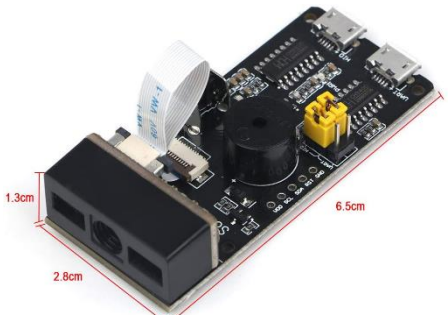


Figure 38 MG65 1D 2D Code Scanner. Permission to reproduce requested.

MG65 barcode reading module is a high-performance scanning engine that can read both 1D and 2D barcodes. (Banggood, 2016) The MG65 barcode module possesses a very high scan rate, making it easier for the reader to detect paper barcodes or barcodes on display. This barcode reader uses the same intelligent recognition software to discover or read items that have a barcode. Intelligent recognition software is built around advanced barcode decoding algorithms, that aids the reader in detecting valid and alphanumeric bar code symbols. Additionally, the MG65 module is structured in such a way that it uses low power consumption. The table below illustrates the DYScan module specifications.

Packages include 1 x QR Code Reader Module.

Table 27 Barcode Scanner Module Specifications

Specifications	
Supply Voltage	5 V
Interfaces	USB2.0, UART
Scan Type	CMOS
Weight	13g
Dimension	46.8 x 27.5mm
Depth of field	25 - 250mm
CPU	32-bit
Max Current	120mA

The MG65 contains a high-performance scanning engine that uses an intelligent image recognition algorithm. An intelligent image recognition algorithm aids the MG65 module in scanning barcode symbols easier. This barcode module supports both TTL232 and USB interfaces. The operating current for the MG65 is less than 150ma, which aids in power consumption. The structure of the module has an integrated design with a small size factor. MG 65 can read barcode types of UPC, EAN, ISSN, UCC Coupon, and many other symbology. The value of the barcode can is showcased on a text file, where we can compare the value to a user's database.

4.10.4 Barcode Scanner Comparisons.

For our project, we have considered the above three scanner modules. These scanner modules are the DYScan DE2120, Waveshare Barcode Scanner, and MG65 1D 2D. Based on the specs from each datasheet, we will compare each scanner to see which offers the best functionality. In Table 20, we showcase the comparison for the different barcode scanner modules.

Table 28 Barcode Scanner Comparisons

Specifications	DyScan	Waveshare	GM65
Supply Voltage	3.3V	5V	5V
Max Current	190mA	135ma	120 mA
Scan Type	CMOS	CMOS	CMOS
CPU	32 bits	32 bits	32 bits
Depth of Field	25mm-400mm	25mm-400mm	25 - 250mm
Weight	< 4kg	< 4kg	13g
Types of Barcodes	1D or 2D	1D or 2D	1D or 2D
Price	$29.95	$39.99	$29.99

We were able to successfully research different scanner modules for the functionalities needed for our project. When comparing the above three scanner modules, the DYScan or GM 65 barcode readers were among the best options. The DyScan module had a more considerable depth of field compared to the GM65. Alternatively, the GM 65 barcode module, has a faster image recognition response time than the DYScan. After analyzing the two different barcode scanners, we decided to use the DYScan module for its low power consumption and its wide depth of field. The DyScan barcode module has both USB and UART communication interfaces. These interfaces will aid us greatly when connecting all the components to our selected microcontroller. The main interface we will be using is the UART, which will be connected directly to a UART port on a microcontroller.

4.11 Fingerprint Sensor

A fingerprint sensor will act as one of secondary ways for the user to unlock the device if they do not have access to their smartphone. As a result, this is not intended to be a method of access for a delivery driver. Our main goal is to find a fingerprint sensor which is both secure and fast. The two products that we investigated were Adafruit 751 and Sparkfun SEN-1418.

4.11.1 Adafruit 751

The Adafruit 751 is an all-in-one fingerprint sensor which uses a high-powered DSP chip that does the image rendering, calculation, feature-finding and searching. The device has a power-on delay time of about 500-ms and it processes a fingerprint image in less than 1.0 seconds. The device communicates with a microcontroller through the UART protocol.

Each fingerprint file has a size of 256 or 512 bytes and the sensor can store up to 162 fingerprints within its own database. (ZhianTec, 2008)

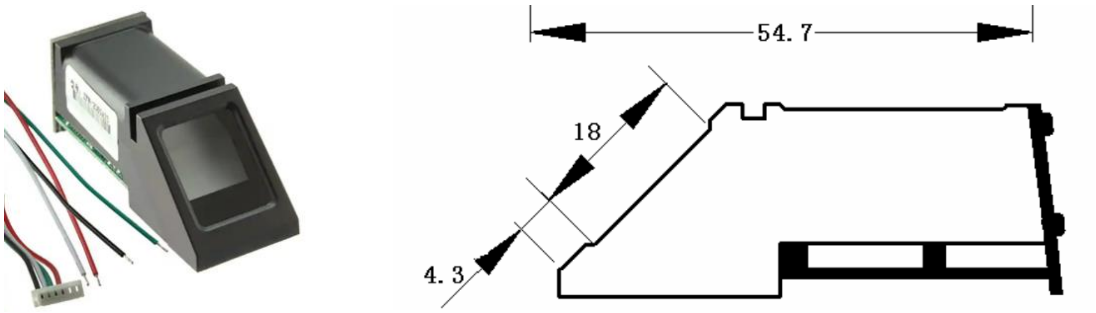


Figure 39 Adafruit 751 with housing dimensions in mm. Permission to reproduce requested.

The device will need to be powered with 5V and it has a max operating current of 120mA. Some of the important dimensions are given in the table below.

Part	Size
Fingerprint Window Area	14 x 18 mm
Fingerprint Window Housing	21 x 21 x 21 mm
Entire Device	56 x 20 x 21.5 mm
PCB	25 x 43 mm

Table 29 Adafruit 751 full dimensions

There are 5 safety ratings which allow the user to set a match percentage needed to open. To get an idea on how these security levels work, the device has a false acceptance rate of less than <0.001% on security level 3. The cost for the Adafruit 751 is $49.95. Putting everything together, the specifications of the Adafruit 751 are shown in the table below.

Specifications	
Supply Voltage	3.60-6V
Operating Current	100mA
Communication Interfaces	UART
False Acceptance Rate	<0.001%
Identification Time	<1 Second
Baud Rate	57600
Board Size	56 x 20 x 21.5mm
Manufacturer	Adafruit Industries LLC

Table 30 Adafruit 751 Specifications

4.11.2 Sparkfun SEN-14518

The Sparkfun SEN-14518 is a high-performance fingerprint module that can be woken up by a finger touching the metal frame of the sensor. The sensor includes a high performance, low power ARM Cortex M3 MCU.

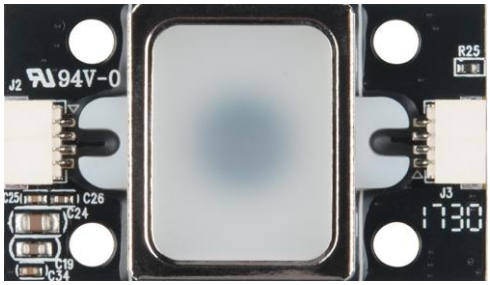


Figure 40 Front of Sparkfun SEN-14518. Permission to reproduce requested.

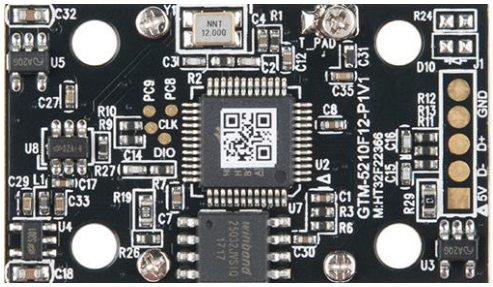


Figure 41 Back of Sparkfun SEN-14518. Permission to reproduce requested.

The SEN-14518 would communicate with our microcontroller through UART with a default BAUD rate of 9600bps. Each fingerprint file size is 512 bytes, and the sensor can store up to 200 fingerprints within its own database. (Sparkfun, 2020)

This sensor has a resolution of 450 dpi, and the images are taken at 258 x 202 pixels wide. The device will need to be powered using 3.3V-6V with an operating current of <130mA. Some of the important sizes are included in the table below.

Part	Size
Fingerprint Window Area	16.9 x 12.9 mm
Entire Device	36 x 21 x 4.38 mm

Table 31 Sparkfun SEN-14518 important dimensions

One of the benefits of this sensor is that it was designed to withstand rugged conditions. For this reason, the sensor works with dry, moist, or rough fingerprints. The surface is also anti-scratch. A second benefit is that the sensor can notify the microcontroller when a finger has been placed on the scanner. This gives us the option to put the device into a sleep until a finger is detected. When in a sleep state the device only draws 5uA of current.

The sensor is also highly accurate with <0.001% false acceptance rate. The scanner takes <1.5 seconds to process any fingerprint images. The speed for enrolling a new fingerprint is <3 seconds. Putting everything together, the specifications of the Sparkfun are shown in the table below.

Specifications	
Supply Voltage	3.3-6V
Operating Current	<130mA
Communication Interfaces	UART
False Acceptance Rate	<0.001%
Identification Time	<1.5 Second
Baud Rate	9600
Board Size	36 x 21 x 4.38 mm
Manufacturer	SparkFun

Table 32 Sparkfun SEN-14518 Specifications

4.11.3 Comparing Fingerprint Sensors

The table below show the comparison chart highlighting the main differences between each of the fingerprint scanners researched.

	Adafruit 751	Sparkfun SEN-14518
Baud Rate	57600	9600
File Size	512B	512B
Communication Protocol	UART	UART
Accuracy	99.999%	99.999%
Identification Time	<1 second	< 1.5 second
Operating Voltage	3.6-5V	3.3-6V
Operating Current	<120mA	<130mA
Cost	$49.95	$35.95
Size	56 x 20 x 21.5mm	36 x 21 x 4.38 mm
Manufacturer	Adafruit Industries LLC	SparkFun

Table 33 Comparing Fingerprint Specifications

The Sparkfun SEN was chosen over the Adafruit 751 for three primary reasons.

1. **Finger Sensing -** The device can wake itself up upon detecting a fingerprint on the metal of the sensor window. While the reader is in this low power state its current draw is <5uA.

2. **Price -** The Sparkfun SEN-14518 is 32.5% cheaper than the Adafruit 751 while also providing a larger feature set.

3. **Rugged Condition -** The Sparkfun is designed to read fingerprints even if they are wet. This is an important feature as the box will be located outdoors where there is a possibility of rain.

4.12 Cameras

A camera is a safety feature that would allow a user to receive a picture of the area around the S.M.A.C. The idea for the camera would be for a picture to be taken anytime the box is attempted to be opened. Our goal for this research is to find a balance between high resolution and low cost. The larger the pictures are, the more time and space required for transfers. The lower quality the picture is, the less useful they are to the user. In this section we will explore the Waveshare OV5640, The ArduCAM, and the OV7670.

4.12.1 Waveshare OV5640

Figure 42 Waveshare OV5640. Permission to reproduce requested.

The Waveshare OV5640 camera contains a video camera based on the OV5640 video sensor by OmniVision. The device takes a maximum of 5 Megapixel image (2592 x 1944). The main feature of this camera module will be an attached Fisheye Lens (STM32) which has a viewing angle of 170 degrees. The 170-degree viewing angle is suitable for security monitoring. The downside to this unit is that the board uses DVP for transferring the pixel data. The ATmega2560 does not have a native DVP interface, therefore a custom one would have to be implemented ourselves. I investigated this, and it is possible, but it would require the use of 12 digital input/output pins on the hardware side, and custom code on the software side.

The device requires a supply voltage of 3.3V and has an operating current of <140mA when operating with a 2592 x 1944 resolution. The device connects with the microcontroller through I2C. Putting everything together, the specifications of the Waveshare OV5640 are shown in the table below. (Waveshare, 2018)

Specifications	
Supply Voltage	3.3V
Operating Current	<140mA
Communication Interfaces	I2C, DVP
Max Resolution	2592x1944
Lens Size	¼"
Board Size	23.9 x 35.7 x 3 mm
Manufacturer	Waveshare
Output Format	• RAW RGB, • RGB565/555/444 CCIR656 • YUV422/420 • YCbCr422 compression

Table 34 Waveshare OV5640 Specifications

4.12.2 ArduCAM 2MP OV2640

Figure 43 ArduCAM. Permission to reproduce requested.

The ArduCAM is a 2MP general purpose SPI data camera which is compatible with Arduino. This device communicates with the microcontroller through both SPI and I2C. I2C for configuration settings, and SPI for data transfer. Since the ArduCAM communicates over SPI, we have the ability to connect multiple cameras to the same microcontroller. The device takes a maximum of 2 Megapixel image (1600 x 1200).

This camera uses an older image sensor (OV2640) as the previous camera researched (OV5640). The manufacturer of this chip is ArduCam. Some of the important features of this camera include single capture mode, multiple capture mode, JPEG compression, image sensor control, and low power mode. The ArduCAM has open source code libraries for the Arduino family which will make prototyping the camera seamless. Putting everything together, the specifications of the ArduCAM are shown in the table below. (ArduCAM, 2019)

Specifications	
Supply Voltage	5V
Operating Current	<140mA
Communication Interfaces	I2C
Max Resolution	1600 x 1200
Lens Size	¼"
Board Size	34 x 24 x 3 mm
Manufacturer	ArduCam
Output Format	• RAW • YUV • RGB • JPEG

Table 35 ArduCAM Specifications

4.12.3 OV7670 Camera Board

The final camera board researched is the OV7670 which is an image sensor designed by OmniVision. The purpose of this camera board is to be a simple camera which can take pictures and video from a development board for prototyping. This chip provides the full functionality of a single-chip VGA camera in a compact package. The size of the chip is 35.16 x 34.29 mm. The camera takes a supply voltage of 3.3V and has an operating current of <140mA.

As with the other cameras, the camera would communicate with our microcontroller over I2C. The lens size is 1/6” and its outputs 30fps with a resolution of 640 x 480. The camera works will in low light and has automatic image control functions such as exposure control, gain control, white balance, band filter, and black-level calibration. This camera would serve as a proof of concept and a more expensive camera can be substituted for the finished product. (Waveshare, 2016)

Figure 44 OV7670 Camera Board.
Permission to reproduce requested.

Specifications	
Supply Voltage	3.3V
Operating Current	<140mA
Communication Interfaces	I2C, DVP
Max Resolution	640x480
Lens Size	1/6”
Board Size	35.16 x 34.29 mm
Manufacturer	Waveshare
Output Format	• Raw RGB • RGB (GRB4:2:2) • RGB565/555/444) • YUV(4:2:2) • YCbCr (4:2:2)

Table 36 OV7670 Specifications

4.12.4 Comparing Cameras

The table below show the comparison chart highlighting the main differences between each of the cameras researched.

	Waveshare OV5640	**ArduCAM**	**OV7670 Camera**
Communication Protocol	I2C	I2C	I2C
Max Resolution	640x480	1600 x 1200	2592x1944
Picture Format	RAW RGB, RGB, YUV, YCbCr	RAW, YUV, RGB, JPEG	RAW, RGB, YUV
Supply Voltage	3.3V	5V	3.3V
Size	23.9 x 35.7 x 3 mm	34 x 24 x 3 mm	35.16 x 34.29 mm
Manufacturer	Waveshare	ArduCam	Waveshare
Cost	$31.88	$39.99	$7.69

Table 37 Comparing Camera Specifications

All the camera researched use some type of video sensor developed by OmniVision, but the ArduCAM was chosen for a couple of reasons.

1. **Medium Resolution** – The 2MP (1600 x 1200) resolution is the right blend of quality and size. We cannot save an image that is too big, but on the other side, if the resolution is too small, the picture is useless.

2. **SPI** – The ArduCAM was specifically designed so that the DVP protocol between the video sensor and the cameras MCU is handled on the board. This allows us to use SPI to get the image files.

3. **Open Source Libraries** – There are many open source libraries for the OV2640 which will aid in the prototyping of our design.

4.13 Mobile Software Development

For our mobile Software Development section, we dive into the details and the technologies we will use to develop our mobile application to interact with our smart delivery system. In this section we detail a comparison of potential operating systems for our applications we could use and as well the potential options of programming languages we could use for our application. As well in this section we detail out of the types of databases we could potentially use for our application and compare the types for the databases we could use for our devolvement.

4.13.1 Mobile Application Operating System

For the mobile application we will be using android the mobile based operating system created by the open handset alliance in 2008. It Is based on the Linux kernel and many other open source software. Its main design was for touch screen mobile devices like many smart phones and tables available on the market today.

Some of the specifications and features we want for our mobile application include a smooth graphical user interface for the user to have a full array of tools for the delivery system, for security purposes we want to be able to store sensitive information such as their password, their address and all of their package number details in a safe manner through the standards of hashing for the security protocols, have the ability to communicate with the system and send information such as when to lock or unlock the device, the time stamps, the pins and even the tracking numbers. As well we want to just have an overall smooth integration with all the hardware of the delivery system.

When we began to take into consideration which operating system environment for our mobile application we weight out a multitude of factors to come to the clear and concise decision to use android. With android It is a complete open source project but as well safe and fast. The following benefits are some of the things we considered when using android as oppose to iOS operating system or even Microsoft operating system for mobile development.

- In 2016 alone there was around 90 billion android apps downloaded that year alone. This translate to massive pool of potential customers if your goal was to acquire a lot of sales.

- With android It comes with custom ROMS (Read-Only Memory). With this you can view, edit and recompile your code and adapt It to a multitude of android devices.
- A over whelming majority of android applications that exist on the marketplace are free meaning they are easily accessible to any socio-economic class you exist in.
- With the android development environment, we can use the java platform to create android apps that have the capabilities to port the application to different platforms.
- Android is highly flexible, since it originated as an open source project in 2008 It has immense amount of functions and feature that users' supreme customizations features for the user.
- When android released android studio, it increased the scalability and as well the flexibility of the operating system.
- With android It Is easy and fast to learn because It has a wide range easy and available resources that make it efficient to develop the android application.

4.13.2 Database

When choosing our database, we had to decide whether we wanted to a MySQL database or a NoSQL database. When choosing between MySQL databases and NoSQL database we had to weigh out the key differences on both and decide which would be the most applicable to our project.

4.13.2.1 Database Type Comparisons

Some of the key differences between that wo we considered MySQL databases they are relational databases meaning the data base would store and allows for access to data points that are related to each other with tabular design. The NoSQL database is non-relational by design and It instead has a document-based design to It. The MySQL database has been a well establish a database for a long time that has been implemented across the IT industry as the NoSQL are slowly gaining momentum and gaining popularity over time. With the MySQL database we have an abundance of tutorials and resources available to us because of how long it has been around but even NoSQL data bases there is larger and larger community that is growing with the ever-expanding tutorials and resources for users.

In terms of scalability with the databases, MySQL databases can't as easily scale because of its constraints with the schema, whereas NoSQL database can scale up to size very easily because of the nature of the schema. The schema for a NoSQL database is dynamic. As well in terms of how flexible the databases are, MySQL databases are not that flexible. When you want to insert a new column or field It will affect the existing design of it. For NoSQL databases when we want to insert new columns or fields, we can do that without affecting existing design set up. With performance NoSQL database is superior to MySQL database. With NoSQL databases it can process and handle large amounts of data in an efficient manner as a MySQL data base can suffer a lot from huge amounts of data. For all these reasons for it being a NoSQL database, to its performance, to how easily a NoSQL database can scale. We decided to go with the firebase data base.

4.13.2.2 Firebase

With further research we stumbled upon firebase, A NoSQL data base that can store and sync data into a cloud database. The data is synced from the clients in real time. Doing this allows for you to access the client's data even when the app goes offline. The data is stored as a JSON and is synchronized in real time with all its clients. One of its advantages is it is cross platform meaning all your clients share the same database meaning apps with iOS, android or JavaScript can all access the same database. How firebase essentially works is the data exists locally and as while offline but when real time events happen it continues to fire, which will give the user a real time responsive environment. The data base has flexible and has an expression-based rules language that is named Firebase Realtime Database Security Rules. The Firebase Realtime Database Security Rule defines exactly how the data will be structured and when you will be able to read from or write to the database When it is created the developer will be able to the permissions to access the data and how they can access it.

4.13.2.3 Firebase Key Capabilities

Real Time

When using firebase, we avoid the completely a typical paradigm of the HTTP request protocols we are so used to. With Fire it Is a real time database that implements data synchronization. Whenever there is a change to the data any device that is connected receives and retroactively updates the data within seconds. With firebase it gives the developer an ease of experience because now the developer does not have to think about the structure of the program or networking code.

Offline

When using firebase apps, they remain responsive even when your device is offline. Fire base can do this since the SDK persist the data from your device to the disk. Once the device goes from offline to online, the device then gets updated with the data that was changed while the device was offline, it is synchronizing the device with the current state of the server.

Accessible from Client devices

With firebase the data can be accessed directly from a web browser or a mobile device with it you do not need an application server. With the database we have built in data and security validations which are made available with the Firebase Realtime Database Security Rules. The rules are all based on expressions that are ran when the data is written to or read in from.

Scale across multiple databases

When using firebase, you can use the app’s data and scale it up by splitting your app’s data amongst a multitude of instances the database within in the same firebase project. As with the firebase authentications system we can streamline authentication and authenticate all

the users amongst every database instances. With firebase we can as well control the access to the data in each one of the databases instances with the real time database rules.

4.13.2.4 Interesting Firebase Bar scanner API

In firebase they have created a whole array of valuable API that are highly useful within its environment but with the creation of their machine learning kit's API they have one specifically used for bar code scanning. The API can read in the encoded data from most basic standards of barcodes. When the barcodes get scanned it does it on the mobile device and doesn't not need a required network connection to do so.

Some of the specifications and key capabilities of the API are listed below as so:

Reads most standard formats

- Linear formats: Codabar, Code 39, Code 93, Code 128, EAN-8, EAN-13, ITF, UPC-A, UPC-E
- 2D formats: Aztec, Data Matrix, PDF417, QR Code

Automatic format detection

When the barcode is scanned the scanner supports all barcode formats at once. With the API you do not have to specify which format the scanner is looking for; it will identify it for you. The option to specify which format you want is available to you when you do this it will provide you with better performance because you're not searching for all the formats there.

Extracts structured data

The structured data that is stored when you are using one of the available two-dimensional formats automatically get parsed. Some of the supported information include URL's, contact information, email addresses, phone numbers, WIFI connection information, SMS message prompts etc.

Works with any orientation

Regardless whether it is upside down, right side up or sideways the barcodes will be recognized and scanned. As well the scanning of the barcode happens all on the device and no on a network connection.

Example of the API inputs and outputs:

Input:

Figure 45 Example of the Input of the Barcode Scanner API

Output:

Corners:	**(87, 87), (612, 87) (612, 612) (87, 612)**
Raw Value:	WIFI: S: SB1Guest; P:12345;T:WEP;;
WiFi information	SSID: SB1Guest
Password	12345
Type	WEP

Figure 46 Example of the output of the Barcode Scanner API

4.13.3 Software Language Option 1

When deciding which programming language, we wanted to use we had a couple options available one of them being java one of the most fundamental languages known. Java is a general-purpose programming language that is objected oriented meaning it is class based. Java was designed to have very little implementation dependency as possible. It was created to allow developers to write code once and virtually run it anywhere. When a java program is compiled It is turned into bytecode so that it can run under a java virtual machine regardless of which computer architecture.

When it comes to mobile development java ranks as one of the most known language for android developers. It is used by developers for a multitude of reasons. Some being is how secure java the programming language can be. Since java is compiled as byte code and ran on a java virtual machine it has no security threats since it gets executed in the java virtual machine environment. Another reason java is used amongst android developers is has

object-oriented paradigms. Since java is a well-developed established language has it has an abundance of frameworks and classes that exist outside its core library. The external classes and features are used for things like networking, I/O operations etc. Java and android have an open source environment to it meaning you have a wider access to tools that you can potentially use.

4.13.4 Software Language Option 2

Another option we weighed out was the programming language Swift. Swift is general purpose, compiled, multi-paradigm language. Swift was created by apple for the iOS operating system their iPadOS, macOS, watchOS. Swift was created to work alongside with apple's cocoa and the extensive base or objective c code that is were already in apple products. Swift is the predominant language in the iOS software world, so it highly sought after because of how highly applicable it is to all devices that use the iOS operating system.

With swift there is a whole array of advantages that exist between many languages here is a list:

Table 38 Table of Advantages with the Swift language.

- **Swift out of most languages available have always held a competitive advantage that being it has always had better run time performance than its competitors.**
- **Swift was made with a type checking system that defends against typical memory hazards that a lot of languages don't.**
- **Swift does not have much boiler plate code like java, does to start any sort of project within a java environment you will need an immense amount of import statements just to even get your project up and running. As in swift is exceptionally quicker at doing it.**
- **Generally speaking, it is a very readable language compare to others in the community.**
- **This is an overall benefit that most languages enjoy is that it has a strong open source community. When you have a strong open source community you generally come to better results because the community is bouncing ideas off each other all the time and the community as a whole learns overall how to develop.**

4.13.4 Software Choice:

After doing all the proper research on the differences between the main languages and the different operating systems we could use we came to conclude to use the android. We took into considerations our most important features we would want in our software environment and chose it predicate on those needs. For our group all our local machines use the windows operating system which android studio supports. If we were to choose to develop our application in swift and use the iOS operating system we would have to set up the whole environment in a virtual machine that had to iOS operating system making the task less accessible for no good reason.

When it came to a specific language to pick with android, our main choice was between kotlin and java. With Kotlin it first appeared in 2011 so it's a relatively new language we didn't want to steer away from languages no one else new so we framed away from kotlin simply from the fact that most the group is comfortable with java because of how much we have used it in our classes. As well we value how much resources are available to us with java in terms if using android with it so that as well was a major weight to our decision. For these simple reasons we decide to use the programming language java with the android operating system for our mobile application.

4.13.5 Sending and Accepting Data from MCU

Overview of how it works:

With the ESP 8266 the Wi-Fi module we chose we would connect it to our microcontroller that we would use to give it access to a Wi-Fi module. The board would connect to our home Wi-Fi network and then begin to run a program that would repeatedly call on home to REST API of firebase that would use the paradigms of HTTP request's. Then there is a response in the form of a JSON object that will then inform the board what its current state it is right now. Every single time we will activate the Wi-Fi module it will be able to now read and write to the JSON object that is stored in the firebase database updating it each change brought to the database

Reading and Writing data from the device:

With fire base it has API's for iOS, JavaScript and Android apps. With firebase it can be accessed through a series a relatively easy REST API's from any devices you desire that can do a HTTP request to firebase server. To read from a firebase database you would need to do a simple HTTP get request on the URL name of your database so you can access it. Luckily when reading in some of the JSON objects they can be deeply nested with data in them, but Arduino has made a library to parse the JSON for us instead. When reading in the data and we decide to want to use different database keys for different devices it will work a little bit different with firebase. When doing so we have each device read in data from the database by a key mapped to a unique identifier. How most identify their unique identifier is using the device mac address of the Wi-Fi radio located on the board since it unique to the device. It a 6-byte ID that is associated to the board. We can write to the firebase database by having a reference to the entry within the database that you want to modify then use the set method on that reference you're pointing to.

5. Hardware Design

This section will cover how the projects components come together. It will focus on physical hardware design and implementation. As we have already researched the main components S.M.A.C. will use now we will draw up a PCB and schematics to show how they will interact together to achieve their intended functions. This section will explore all the possible conflicts and options for a successful pairing and use of the components. The design methodology and topology are vital for a complete and robust design experience.

5.1 Housing Design

In this section we will delve deeper into the design of the box and how its going to be constructed. As previously mentioned, we have chosen wood as our main material for the box. This is because of its malleability, great aesthetic possibilities, and resistance to the elements. The dimensions of the box as determined by our research are going to be 28 inches in length by 18 inches in width by 15 inches in height. This should be more than enough to fit a large variety of the most common boxes in the market. The box will compromise of three critical parts which are the components compartment, the sensor locations, and the lid mechanism. For practicality and aesthetics purposes we need to try and have all the components as close together as possible to run wires through the least places possible.

The first part we are covering is the false bottom which will house most of our equipment and wiring for the project. This section which will be made at the base of the box and will house our PCB, AC to DC transformer, and any excess wiring needed for our box to operate. This compartment will be about 1.5 inches in height so that our AC to DC transformer could fit comfortably. We decided to go with a false bottom instead of a boxed section or ward inside the box as it will disrupt the space where packages will be placed. The packages could also hit on this enclosure and maybe damage it which is why we thought this would be the most secure and safe environment for the most critical components of S.M.A.C. On the corners of this false bottom there will be a small hole to be able to run cables to the components that will be in the outside of the box.

The components that will be exposed in the box will be placed in the top right corner of the box in the front panel. This section will include the fingerprint scanner, barcode scanner, camera module, and the latch sensor. I wanted to keep all components close to each other to have the least amount of wires going to multiple places of the box. The fingerprint sensor, barcode scanner, and camera module will all be arranged in a row starting at the top right corner of the box's front panel. Holes of the appropriate size will be made for each component to pear thru the box so it can interact with the outside world. The latch sensor will be attached to the front most part of the right panel instead of the front to reserve the front panel space for critical components that will be interacting with the outside world.

We now move to the last critical part of the box's design, the lid. The lid will open upwards and will be held by hinges that are attached to the inside of the back panel of the box. The lid will have two components attached to it. These components include the UV light and the solenoid lock. The wiring for this part will come from the false bottom just like it did

for the front panel components through a small hole made at one of the corners of the false bottom. The wiring will ride through the back side of the box up to the lid through were the hingers are located to create the least stress possible on the cables. Towards the area closest to the hinges the UV light will be located. This area was chosen as it will not cause the bulb and its housing to be hit by incoming packages into the box maximizing its life span. The solenoid lock will be centered and position toward the front of the box where it will act just like a door latch mechanism. A small depression on the wood will be made to the box on the front panel to allow the lock to close and secure the box properly. This will all nicely fit into our design and will also allow for the box to serve as a bench and be inconspicuous to potential thieves. The top side of the lid can be equipped with a cushion for extra comfort and appearance.

5.2 UV Light

The cold cathode UVC light we have chosen operates with a standard 120V AC directly from a wall outlet. However, due to the nature of our project it will have a 5V relay to control it directly from the ATMega2560. This relay module will act as a smart switch to control when it is on. More granular modifications of its operation, like time and modes, will be done from the ATMega2560 code. So, in retrospective this component will have two different power supplies and voltages coming into it. One to power the light itself and another to power the relay which will interact with our ATMega2560. In this section we will talk about both connections and essentially how it will be designed to work with the project. The pins out for this section will be the following:

- 120V AC
- 5V DC
- Signal
- GND 120V AC
- GND 5V DC

5.2.1 5V Relay

To understand how the UV light will function we need to understand what essentially a 5V relay is. A 5V relay is an electric switch that uses low level triggered controlled signals to turn on or off the relay. Triggering the relay with a HIGH signal closes the switch turning the device on or when a LOW signal is given it opens the switch and turns off the device. This signal will be given by one of the digital pins on the ATMega2560 which we would configure as an output pin and just send either a high or low signal through it.

On the right side of figure 40 we can see this is where the relay is powered and controlled. Relay requires a 5V DC source and digital signal pin from the ATMega2560. On the left side of figure 47 we can see this is were the power to be controlled is fed to the relay. NC

Figure 47 5V Relay

stands for normally closed and NO stands for normally open. Normally closed acts as a closed circuit when the signal is LOW and as an open circuit when signal switch is HIGH. Normally open acts as an open circuit when the circuit signal is LOW and as a closed circuit when the signal is activated with a HIGH output. For this application we will be using the normally open and C pin out as it is safe practice to normally use the NO pin as this configuration in the event the relay loses power it will automatically open the circuit and turn off the device and prevent any further malfunction or unwanted operation.

5.2.2 Wiring and PCB Schematic

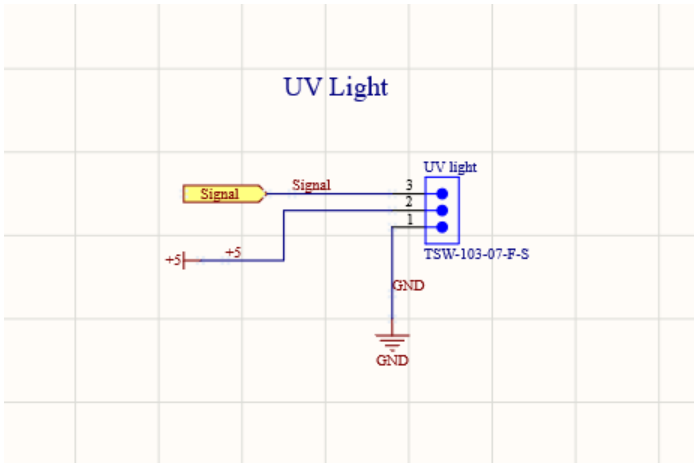


Figure 48 UV light schematic

5.3 Power System

The Power system is going to be a little complex due to the multiple different voltages the components need. We are working with four different voltage needs for the components. These voltages are 120V AC for the UV light, 12V DC for the solenoid lock, 5V DC for 3 components, and 3.3V for 2 components. This imposes a challenge. The first one and most important one will more than likely being able to supply enough current so that all the devices can power at their theoretical maximum current draw and leave room for current spikes. We will be using power directly from a wall outlet meaning we will be drawing 120V AC directly from the wall. From here we connect the in parallel the UV light circuit that feeds the 5V relay and a AC to DC 12V transformer capable of outputting at least 12V DC that we can use to power our solenoid lock. This 12V DC line can then be used with Buck converters to give us a 5V DC line and a 3.3V DC line. To recap we will be using the following power sources:

- 120V AC
- AC to DC 12V transformer
- 5V buck converter
- 3.3V voltage regulator

5.3.1 AC to DC transformer

To be able to feed our devices we must first transform the raw 120V AC voltage from the wall outlet to something usable which starts at 12V DC output. The reason we choose the highest voltage in our list of voltages is because of how much easier and efficient it is to convert to lower voltages rather than stepping up to higher ones. An AC to DC transformer works by converting Alternating Current which oscillates into a Direct Current which is steady. This is achieved by first using a transformer to lower the AC voltage and then using a full wave rectifier to achieve the current conversion to DC.

5.3.2 Buck Converter

A buck converter is chosen to convert our remaining voltages as it is the most efficient way to convert our DC to DC voltages without significant current loss. A voltage divider does not bode well with varying currents and since the components will all be varying currents depending on use it does not work in our scenario. A linear regulator works better in this scenario as it can take varying currents however it dissipates a lot of current as heat making it as inefficient as a voltage divider. A buck converter on the other hand only draws enough charge to maintain the required output voltage and current needed making it extremely efficient.

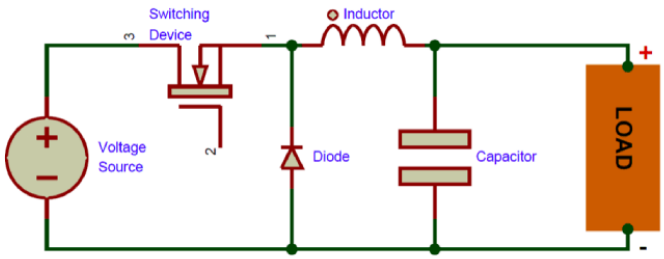


Figure 49 Buck Converter

In figure 49 we can see how a buck converter works by using a switch that charges up the capacitor and inductor for a short period of time and the charging current cause the inductor to create an opposing voltage as it is charging thus creating a net voltage drop across the load.

5.3.3 Wiring and PCB Schematic

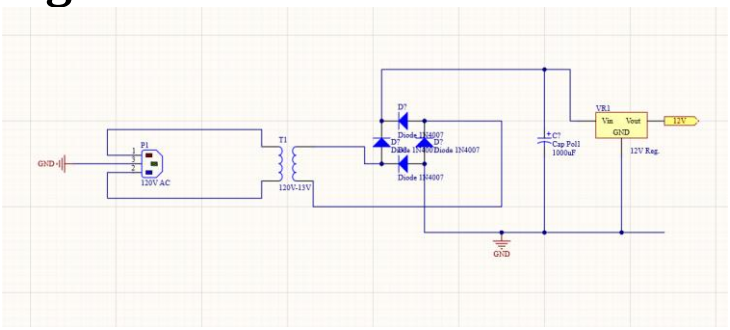


Figure 50 AC to DC converter

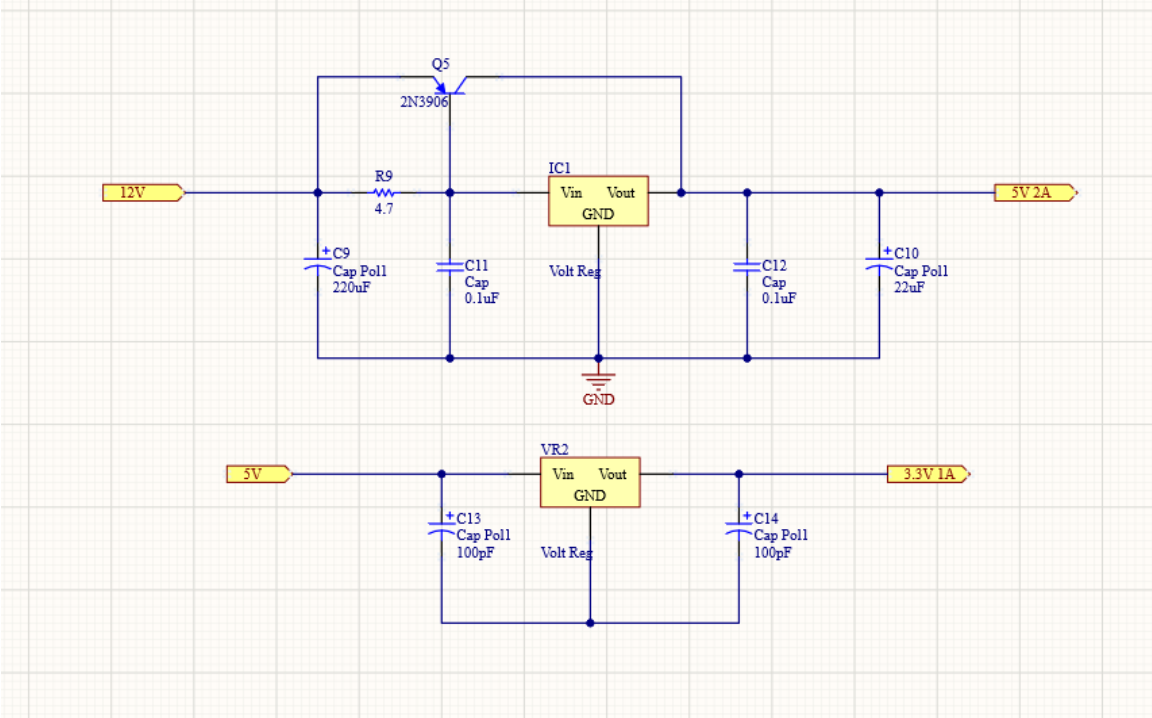


Figure 51 DC to DC converters

5.4 Wireless Communication

The ESP32 will be responsible for S.M.A.C.'s communication to the outside world. This is no small task for such a small chip. One could say this will be the most important part of the project's functionality. The ESP32 will have to run in tandem with the ATMega2560. They will communicate using UART. The only issue present is that the ATMega2560 and the ESP32 do not operate at the same voltage. The ATMega2560 operates at a 5V logic level and the ESP32 I/O pins operate at a 3.3V logic level. This will require the use of a logic level converter to be able to safely communicate between both boards. The ESP32 board will be responsible to connecting to the internet and in turn our mobile application. This connection will require a two-way street for the ATMega2560 and the ESP32 to communicate. Each will have to send commands to each other which is why we require UART. Using UART we can send serial commands over the TX and RX pins to be able to send data back and forth. This board can be powered with either 3.3V or 5V. This will allow us to run the ATMega2560 and the ESP32 on the same voltage rail. The pin outs that will be using for this will be the following:

- TX
- RX
- GND
- 3.3V DC or 5V DC

5.4.1 Logic Converter

The logic converter will be crucial in this setup. Even though the ESP32 can be powered with 5V that does not indicate that the logic level it operates at is the same. An internal component on the ESP32 converts that 5V input into usable 3.3V power for the chip to use and in turn the I/O pins to output. To solve this miscommunication issue we will use something that is called a logic converter which will convert the outgoing 5V data logic

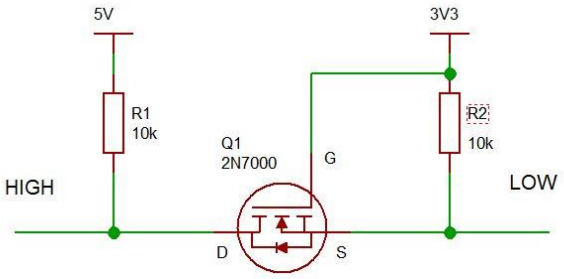


Figure 52 MOSFET Logic Level Conversion

into 3.3V data logic that the ESP32 can read and also turn the 3.3V data logic that the ESP32 pushes out and amplify it to 5V for the ATMega2560 to be able to read. There are two ways we can achieve this conversion. The first one is using a simple voltage divider using resistors and drop the higher voltage down to 3.3V. This method is only effective when converting the 5V line down to a lower voltage but not vice versa which we also need for the ESP32 to communicate with the ATMega2560. This is where we use a MOSFET and a couple of pull up resistors. This method uses a MOSFET and relies on the parasitic diode between Source and Drain to allow full bidirectional communication with low and high side voltages. In 51 we can appreciate how the system works. how the system works.

5.4.2 Wiring and PCB Schematic

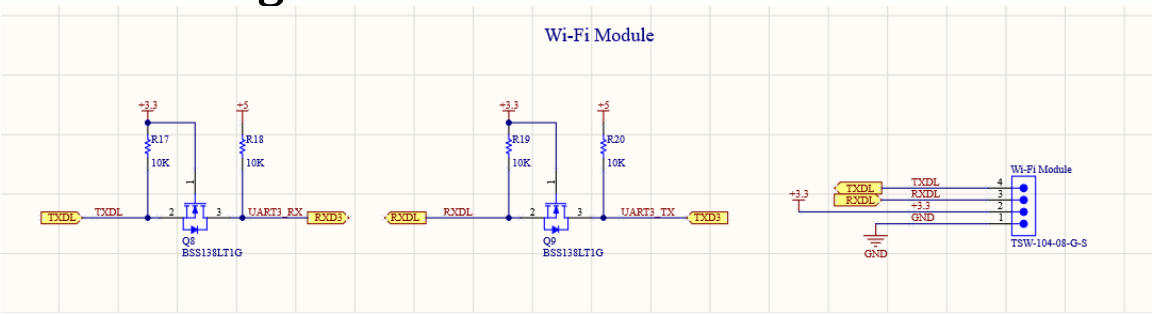


Figure 53 Wi-Fi Module Schematic

5.5 NFC

With our NFC it will be able to detect the smart card IS01443A that will be used as an authentication method. The only person who will have the smart cards are the mailman who are set to do the delivery. When the intended mail man who is supposed to deliver the

package swipes his card on the RFID module then the smart box will open allowing them to properly store the package in the box.

Two for power

- VCC
- GND

Supported communication

- SPI
- High- Speed UART
- II2, TX, RX

To test our hardware, we use the default communication mode for the module which is the HSU mode. We will use a set of males to male jumper wires to solder in the male header pins from the bottom of the breakout board.

We will connect the wires like so:

PN32 NFC RFID MODULE		**ARDUINO**
VCC	→	**5V**
GND	→	**GND**
TXD	→	**PJ1**
RXD	→	**PJO**

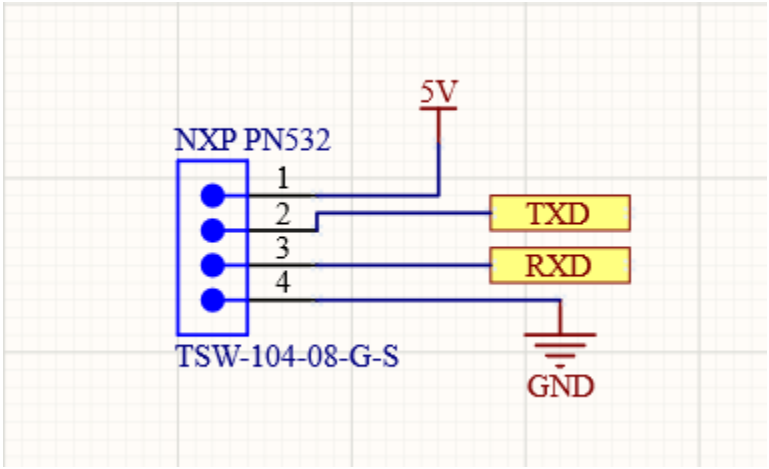


Figure 54 NXP PN532 Module Schematic.

5.6 Open / Close Sensor

For the open and closed sensor, we will use it to determine whether our system door is either in an open state or in a closed state. With the reed switch we will need a magnetic field to interface with the reed switch for the switch within the component. When we move our magnet, the magnetic field will open up the switch and when we move our magnet near the reed switch it closes the switch reed. That is basics of how the hardware for the reed switch works.

Two for power

- No external power is needed
- GND (ground)

Two for communication

- RX
- TX

The connections for our open/close sensor are simple. We will need to connect a pin from the reed switch to the one of the Arduino grounds. Then we have to connect the other reed switch pin to the one of the Arduino input pins to make a complete circuit.

- When the magnet is in a near the Reed Switch, the value will of the Arduino's input pin will be LOW (LOW is just a Library defined token that represents the integer value 0).
- When the magnet is away from the Reed Switch, the value will of the Arduino's input pin will be HIGH (HIGH is just a Library defined token that represents the integer value 1).

To check the state of the door, we simply need to check what state of our Arduino input pin is in.

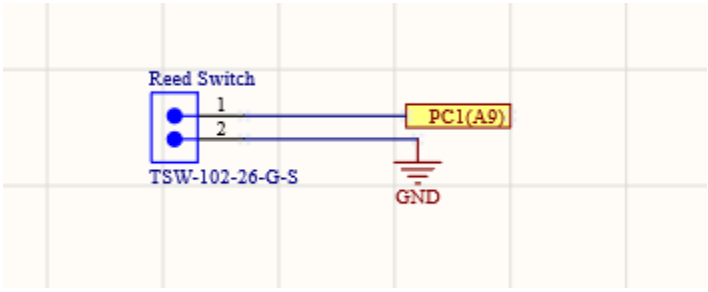


Figure 55 Schematic of the reed switch

5.7 Electronic Lock

The electronic lock chosen will work on a force high or low voltage to initiate the locking system. A voltage of 12V should be applied to the solenoid lock to adequately charge the metal solenoid coils. Moreover, the lock will require a 650mA current over its unit to trigger the lock. The utilization of a 12 VCC flexibly voltage and GND wires will be incorporated into the usefulness of the locking framework. The solenoid lock likewise can open from a separation of more than 30 feet. This opening element will likewise be thought of, on the off chance, that we grandstand the separating highlight for our lock. This solenoid has a strong and robust structure which aids in keeping a user's package secure when locked.

Two for power

- VCC
- GND

Using our mobile application, a signal relays to a Wi-Fi receiver which will communicate with our Arduino microcontroller. The microcontroller will then control a selective voltage of 12V which would cause the door to unlock.

5.7.1 Pin Description

A voltage of 12VDC needs to be supplied to the solenoid to move the embedded metal slug to operate this electronic lock. This slug manages when a large amount of current is sent through the unit to charge the coils. Once loaded, the metal slug shifts downward to an open state. The lock solenoid will be connected via two pins—one that controls the flow of the 12VDC and another that goes to ground. In figure 56 we showcase the pin selection for our electronic lock.

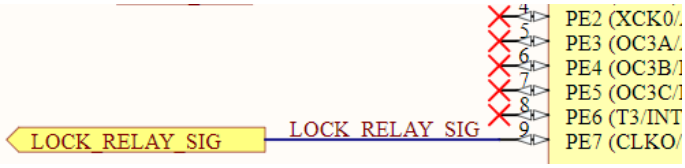


Figure 56 Lock Relay Selection

The locking mechanism will provide feedback to the ATmega2560 to indicate the status of the lock. Once this status has been received, the MCU will relay this message to the ESP8266 Wi-Fi module, which would send a response to the mobile application. This response will entail information about the different time period when the box has been locked or open.

5.7.2 Electronic Lock (5V Relay)

A relay switch and a diode will be expected to wire it to our ATmega2560 microcontroller to drive our electric solenoid. A sensibly decent force flexibly drives a solenoid, as a great deal of current will hurry into the solenoid to energize the electro-magnet coils which would reach current values of over 500mA. Since the lock is an inductive solenoid requiring 12V, a 5V relay switch will be used to activate the locking mechanism. The figure below illustrates the locking mechanism of a solenoid lock.

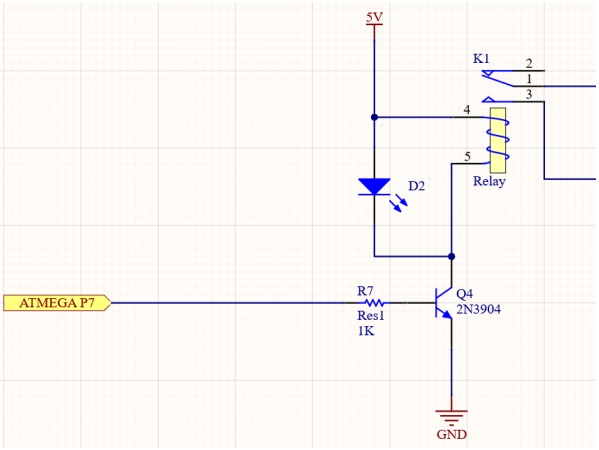


Figure 57 Electronic Lock Relay

5.7.3 Wiring and PCB Schematic

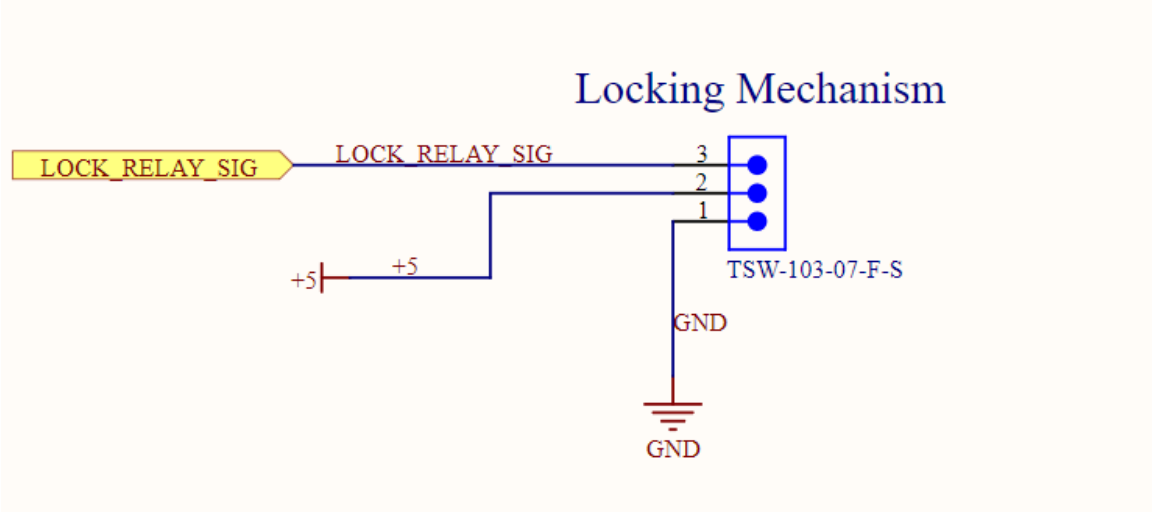


Figure 58 Electronic Lock (5V Relay) Header

5.8 Barcode Scanner

The DYScan Barcode Scanner module can scan both 1D and 2D alphanumeric digits that appear on packages. We will be able to locate and match different package information against our database through image recognition. A supply voltage of 3.3V will be needed to function the barcode module. Additionally, the DYScan module has specific communication interfaces that we need to consider. Embedded within this module's infrastructure are the interfaces of UART and USB. Together as a group, we concluded that we would be using the UART connection to combine this module with our selected microcontroller.

Two for power

- VCC
- GND

Two for communication

- RX
- TX

The design of the DYScan module, contains a high brightness and dark environment setting which are used to scan packages during low lighting. Because of its mini size, this barcode module is widely used in wireless machines and self-service terminals. PIN 9 will have the VCC function attached to it. Whereas PIN 5 is used for the GND functionality. Lastly, this barcode module uses a CMOS scan type mechanism to read different barcode digits.

5.8.1 Pin Description

The DyScan Barcode scanner comes with a built-in breakout board that makes it easy for us to connect and use all of the capabilities of the scanner module. Located on the breakout board, there is a LED and buzzer which are connected to the module by various drive circuits. Additionally, this breakout board contains a pushdown tactile switch which provides a trigger pin on the board. To incorporate this DyScan module to our MCU, we

will leverage the pin header for direct access to the TTL serial pins, power pins, and trigger input. In Figure 59 we illustrate the different pins we will be using for our barcode scanner.

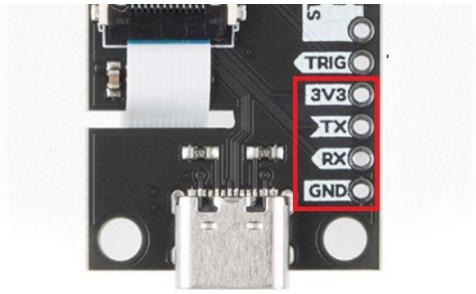


Figure 59 DyScan Barcode Pin Layout

5.8.2 Level Shifting (Barcode Scanner)

Similarly, to our fingerprint sensor, a level moving circuit will be expected to downscale the voltage for our standardized identification scanner. The ATmega2560 is developed to yield just 5V on its RX and TX pins. While the standardized identification scanner can just work at a voltage of 3.3 V. To amend this voltage issue, we will step down the voltage between the RX pin and TX pin to the ATmega2560. You can adequately set your high and low voltages and venture down between them safely on a comparable channel. We will be going into great detail about our level shifting circuit in section 4.9.1.2.

5.8.3 Barcode Scanner Hardware Design

The DYScan modules come with their PCB module built around the design of the barcode scanner. The barcode module contains an image recognition software that will be operated by applying a voltage of 3.3 V to the unit. The structure of the barcode module will be positioned on top of the container where an individual will be able to access it freely. Due to its small size, the barcode module makes it easy for us to fit the module into a relatively comfortable position. The barcode scanner's location plays a vital role when an individual is trying to get a package scanned. A long wire will be wired along the box's inner insides to connect power to the barcode module. The PINs nine and five are taken into consideration when we are powering the unit. Scanner tag filtering gives clients an advantageous option to physically looking for item data or entering information. Examining standardized tags can assist clients with straightforward recognizable proof and substance errands, for example, looking into an item to buy, getting to occasion subtleties, or finishing a structure. In figures 60A and 60B below, we illustrate the design dimensions of the DYScan module, we will be implementing.

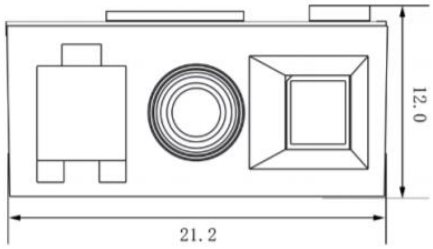


Figure 60A DYScan Front View

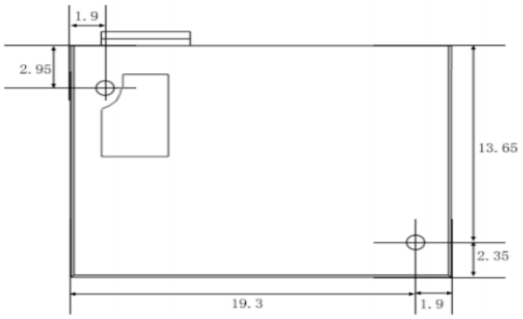


Figure 60B DYScan Top View

The PCB board that comes with the barcode scanner makes it easier to explore the vast capabilities of the DE2120. With the help of a PCB, there will be no need to deal with finicky flat flex cables to connect components. Onboard the barcode scanner, there are two interfaces built with its design. The USB interface can be easily seen since it is exposed primarily at the end of the unit. We will be disregarding the use of the USB interface and will switch our focus to the UART communication interface for our design. The Arduino Mega board contains four UART ports, which aids us significantly to connect the DyScan to the microcontroller. On the PCB board module, there is a buzzer and LED which are implemented using appropriate drive circuits and a pushdown switch. To connect our DYScan to our other devices, we can leverage the 5-pin header for direct access to the UART pins. This barcode module has a fantastic feature where it can be configured to use either serial interfaces or scan barcodes located in a data sheet. The DE2120 has the one-of-a-kind capacity to list as every one of the three conventions including a CDC sequential driver so the gadget shows up as a standard COM port.

4.8.4 Wiring and PCB Schematic

A plan obstacle is that the unique barcode scanner can't be legitimately associated with our principle PCB. Therefore, to rectify this connection issue, a four-pin header will be used to connect the barcode module to our ATmega2560. The first 4 x 1 header design used for UART is shown in the image below.

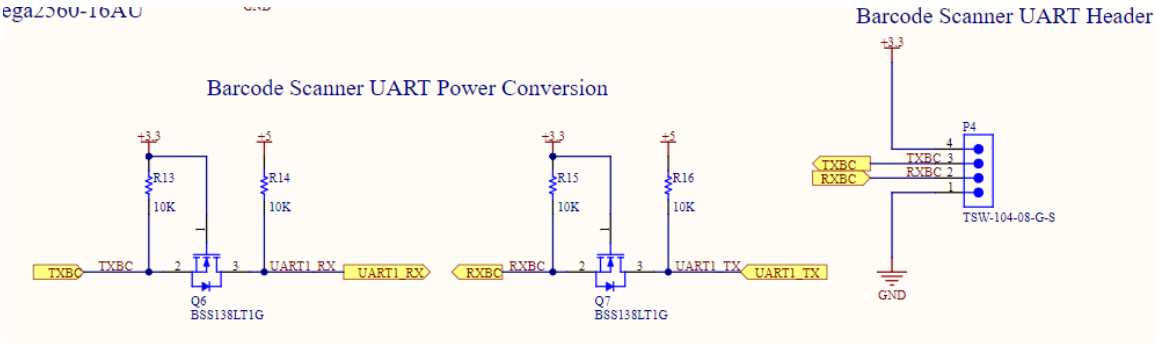


Figure 61 Barcode Scanner Header

The header pins for the barcode scanner are showcased below:

Pin	Label	Type	Description
1	VCC	POWER	3.3V Power Supply
2	TX1	Output	Transmission Line
3	RX1	Input	Reception Line
4	GND	Ground	Power Ground

Figure 62 Barcode Scanner UART header pin descriptions

5.9 Fingerprint Sensor

In this section we will take a deep dive into the hardware connections for the Sparkfun fingerprint sensor. The sensor comes as a pre-built PCB which supplies 8 pins total.

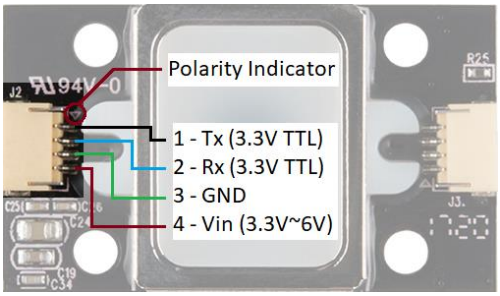


Figure 63 UART wire connections. Permission to reproduce requested.

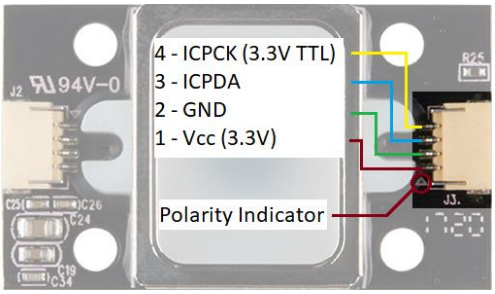


Figure 64 Finger sense connections. Permission to reproduce requested.

The first four as seen in the image on the left are for UART communication. The second four as seen in the image to the right are for finger sensing.

5.9.1 Wiring and PCB Schematic

A design hurdle is that the fingerprint scanner cannot be directly connected to our project's main PCB. The reason is that the scanner will need to be placed in a custom location on the box. For this reason, we will integrate two Molex 4 x 1 headers within our PCB design so that we may make all connections from the scanner and the ATmega2560.

5.9.1.1 UART Header

The first 4 x 1 header design used for UART is shown in the image below.

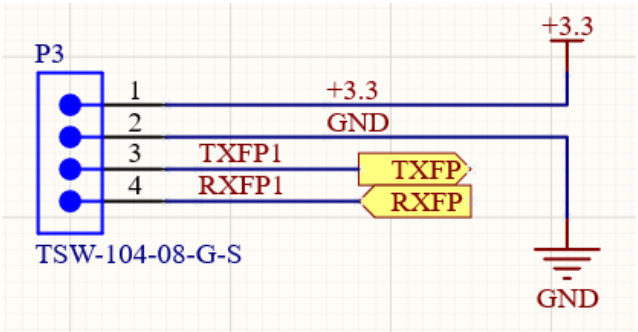


Figure 65 Fingerprint Sensor Header

The headers pins are described in the table below.

Pin	Label	Type	Description
1	VCC	POWER	3.3V Power Supply
2	GND	Ground	Power Ground
3	TX	Output	Transmission Line
4	RX	Input	Reception Line

Table 39 Fingerprint UART header pin descriptions

The 3.3V VCC supply voltage and GND wires will be supplied through the power system. The RX pin of the ATmega2560 will connect to the TX pin of the fingerprint sensor. Similarly, the TX pin of the ATmega2560 will connect to the RX pin of the fingerprint sensor.

5.9.1.2 Level Shifting

The ATmega2560 outputs 5V on its RX and TX pins, but the Sparkfun fingerprint sensors UART pins require a maximum voltage of 3.3V. Therefore, the voltage will need to be stepped down between the RX pin of the ATmega2560 and the TX pin of the fingerprint scanner. A similar process will need to be done for the TX pin of the ATmega2560 and the RX pin of the fingerprint scanner. This can be accomplished using a bi-directional logic level converter as seen in Figure 66.

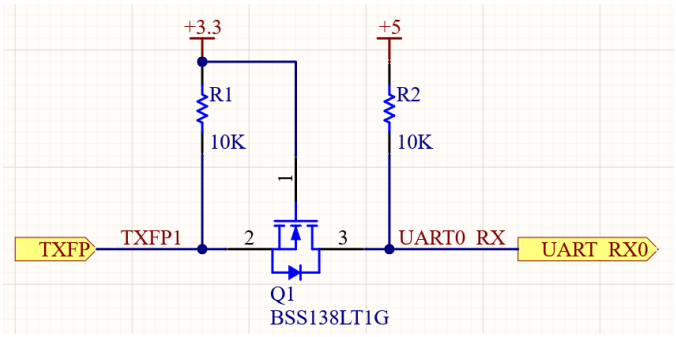


Figure 66 Bi-Directional 3.3V to 5V Converter

A converter can be made with an N-Channel MOSFET with the lower voltage attached to the gate, and the higher voltage attached to the drain, and a diode. Here, when TXFP is low, the 3.3V source turns on the MOSFET creating a short from TXFP to RX0, thus pulling RX0 low as well. Once TXFP is high, the gate-source voltage (Vgs) will be less than what is needed to turn on the MOSFET, and the MOSFET will turn off, causing it to be an open and RX0 is pulled up to 5V.

5.9.1.3 Finger Sensing

One of the benefits of the Sparkfun fingerprint scanner is that it can sense if a finger is placed on the optical sensing area using the four wires seen in Figure 67. When a finger is placed on the sensor, the ICPCK pin will output 3.3V. This value will be read through one of ATmega2560's general input pins. The full state logic table for finger sensing is shown in the table below.

Touch State	ICPCK Pin Status
Finger Initially Touching the Frame	LOW →HIGH
No Finger Touching	LOW → LOW
Finger Touching the Frame	HIGH → HIGH
Removing a Finger from the Frame	HIGH → LOW

Table 40 Finger sense logic table

5.9.1.4 Finger Sensing Header

The second 4 x 1 header design, used for finger sensing, is shown in the image below.

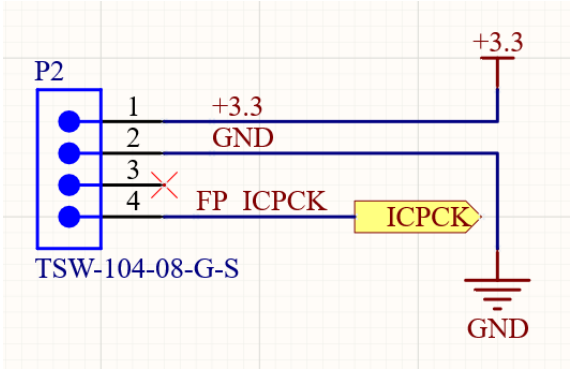


Figure 67 Fingerprint Finger Detection Header

The headers pins are described in the table below.

Pin	Label	Type	Description
1	VCC	POWER	3.3V Power Supply
2	GND	Ground	Power Ground
3	x	x	x
4	FP_ICPCK	Output	Hi / Lo output

Table 41 Fingerprint sensing header pin descriptions

The 3.3V VCC supply voltage and GND wires will be supplied through the power system. The ICPCK pin of the fingerprint sensor will connect to a general-purpose input / output pin of the ATmega2560. The ICPDA pins is used for updating firmware and is not necessary to connect.

5.10 Camera

The ArduCAM camera comes as a pre-built PCB. The camera cannot be directly connected to the main PCB as it will need to be placed in a custom location on the box. Instead we will have a header on our PCB which connects to the camera's pins.

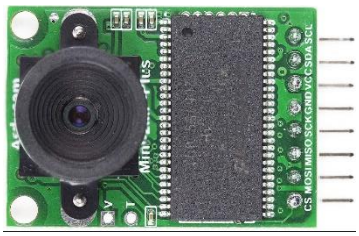


Figure 68 ArduCAM Front. Permission to reproduce requested.

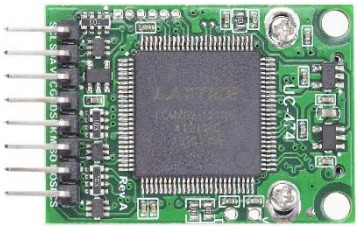


Figure 69 ArduCAM Back. Permission to reproduce requested.

5.10.1 Pin Descriptions

The ArduCAM provides 8 pins, all of which have an important role in the function of the camera. The pin descriptions are given in the table below.

Pin	Label	Type	Description
1	CS	Input	Slave select
2	MOSI	Input	Data output from master
3	MISO	Output	Data output from slave
4	SCK	Input	Serial Clock from master
5	GND	Ground	Power Ground
6	VCC	POWER	5V Power Supply
7	SDA	Bi-directional	Serial Interface Data I/O
8	SCL	Input	Serial Interface Clock

Table 42 ArduCAM pin descriptions

5.10.1 Wiring and PCB Schematic

The camera cannot be directly connected to our projects main PCB as it will need to be placed in a custom location on the box. For this reason, we will integrate one Molex 8 x 1 header within our PCB design so that we may wire the camera to the ATmega2560.

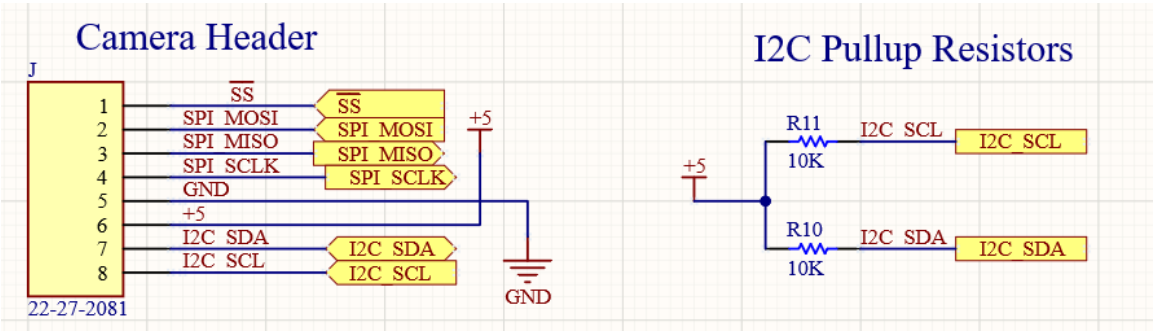


Figure 70 Camera header PCB design

As seen in the schematic above the camera communicates with the MCU through both SPI and I2C. I2C is used to configure the camera, and SPI is used to transfer the image data. The 5V power supply and GND wires will be supplied through the power system. However, since the data pin connections have more depth to them, I have described them in the table below.

Table 43 Camera Pin Connections

ATmega2560	Pin	Direction	Pin	Camera
SS	19	→	1	CS
MISO	22	←	2	MOSI
MOSI	21	→	3	MISO
SCK	20	→	4	SCK
Power System		→	5	GND
Power System		→	6	VCC
SDA	44	←→	7	SDA
SCL	43	→	8	SCL

5.12 External Clock Source

The clock for the ATmega2560 is sourced from either the External Clock, Crystal Oscillator, Low-frequency Crystal Oscillator, or a Calibrated RC Oscillator.

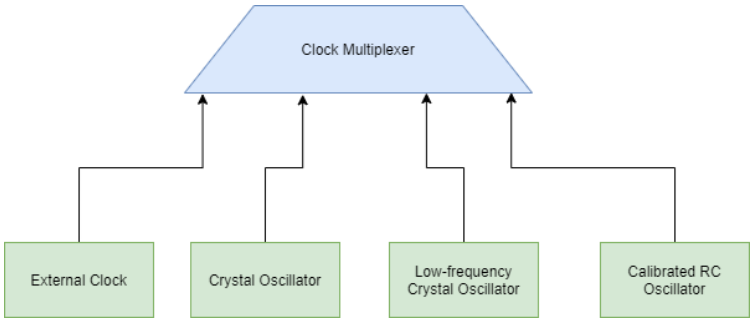


Figure 71 ATmega2560 Clock Sources

The default clock source is an internal RC oscillator at 8.0MHz. This input is then divided by 8 resulting in a system clock of 1.0MHz. The clock can be divided by 1, 2, 4, 8, 16, 32, 64, 128, or 256 depending on the clock prescaler select. A more accurate oscillator may be installed externally through XTAL1 and XTAL2 pins.

Table 44 XTAL Descriptions

Pin	Description
XTAL1	Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.
XTAL2	Output from the inverting Oscillator amplifier.

The frequency of the oscillator can be between 0.4-16 MHz. We implemented a 16 MHz Full Swing Crystal Oscillator into our design. The recommended range for capacitors C1 and C2 is 12-22pF.The schematic for the clock is given in the figure below.

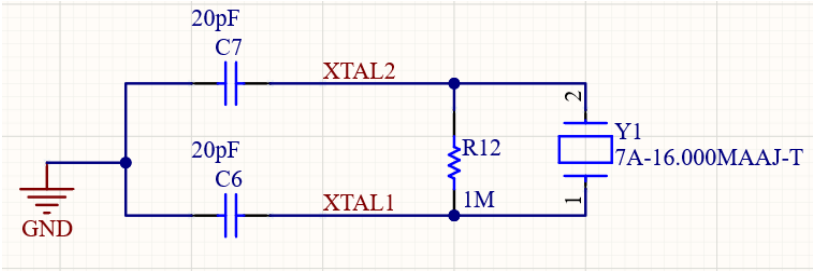


Figure 72 ATmega2560 external clock schematic

5.13 Reset Button

A reset may be applied to the microcontroller through the reset input pin. A low level on this pin for longer than the minimum pulse length (2.5µs) will generate a reset, even if the clock is not running. Some important reset parameters are included in the table below.

Table 45 REST Button Specifications

Parameter	Min	Type	Max
RESET Pin Threshold Voltage	0.2V		0.9V
Minimum pulse width on RESET Pin			2.5µs

For our design, we attached a button for resetting the microcontroller as seen in figure 73. Included in this circuit design is an RC filter network designed to debounce the button.

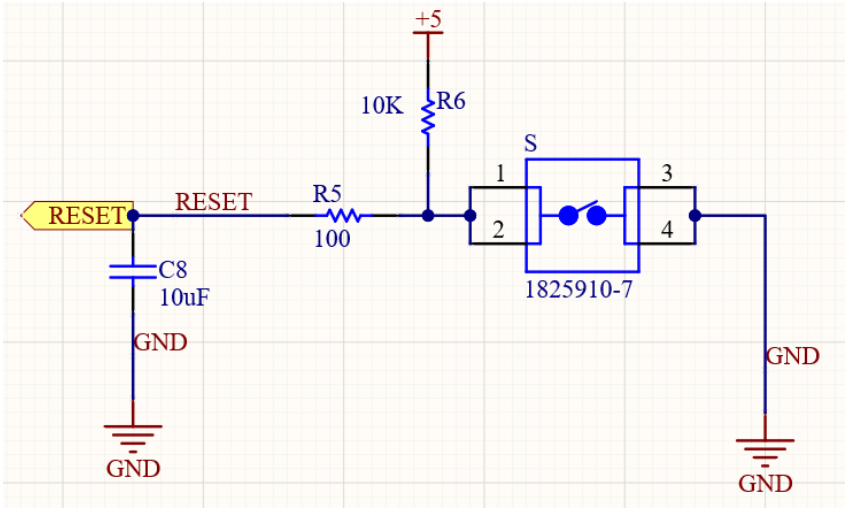


Figure 73 Reset button schematic with debouncing

When the button is not pressed, the reset line is pulled up to 5V. At the same time, the capacitor is also charged up to 5V. When the button is first pressed, the capacitor will still have enough charge to prevent bounces. Then, after the capacitor has discharged, we will get a single HIGH to LOW signal resetting the MCU. The same idea applies in reverse as the capacitor cannot charge up to 5V instantaneously, it will filter out the false positives occurred from bouncing.

5.14 Overall Schematic

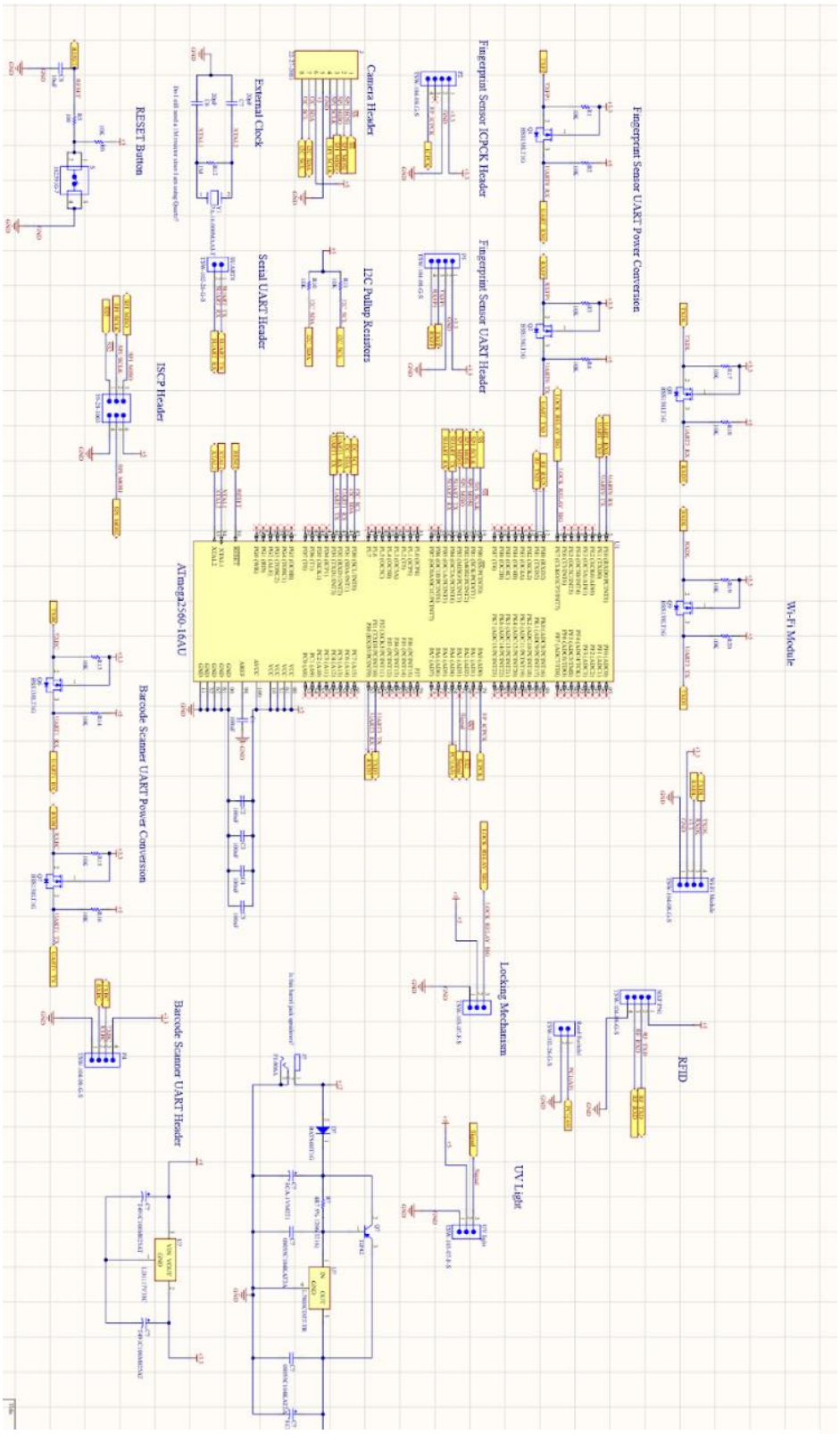


Figure 74 Overall PCB Schematic

5.15 PCB Board Design

Printed circuit boards (PCBs) are the main building blocks of most present-day electronic gadgets. (Circuits, 2019) PCB's have standardized and electrical qualities that make them ideal for electrical purposes. Most PCB's manufactured in the world are inflexible. Generally, 90% of the PCBs produced today are unbending sheets. Some PCBs are adaptable, permitting the circuits to be twisted and collapsed into shape. PCBs utilize flexible circuits to endure a considerable number of flex cycles, with no break in the circuits. These flexible PCBs generally involve 10% of the market. Ordinary PCB's can be as straightforward as a solitary layer of hardware or can go to fifty sheets or more. They comprise of electrical segments and connectors connected utilizing conductive circuits, generally copper, to defeat electrical signals and force inside and between gadgets. Contrasted with conventional wired circuits, PCBs offer various favorable circumstances. Their little and lightweight structure is proper for numerous advanced devices, while their unwavering quality and simplicity of support suit them for combination in complex frameworks. Moreover, their ease of creation makes them an exceptionally savvy alternative.

All PCBs are specifically structured in such a way for their application. (Circuits, 2019) Regardless of whether basic single-layered unbending sheets, PCB's are structured utilizing extraordinary programming called CAD for PC helped plan. The fashioner uses this product to put all the circuits and association focuses, called vias, all through the board. The product knows how every one of the segments needs to cooperate, and necessity also, for example, how they should be bound to the PCB. When the creator is done, the product trades two necessary parts, with which we will manufacture their sheets. The first is called Gerber documents, which are electronic work of art records that show each circuit in the PCB, precisely on every layer of the board. The Gerber documents will likewise contain drill records, giving us precisely where to bore the gaps to make all the employing associations we talked about before. They will also include patch veil and classification documents, which are discussed later, just as a record tells us precisely the best way to remove the edge of their board. The essential materials utilized in the assembling of PCBs are fiberglass or plastic substrates, copper, patch veil, and terminology ink.

5.15.1 Custom Printed Circuit Boards

All PCB creators, whether if it's fixed, adaptable, or unbending flex, all use Gerber documents to convey to PCB makers precisely how they need their sheets assembled. They incorporate one other thing that is basic for the PCB fabricators. This item is called a PCB creation print. The creation prints cautiously subtleties all the prerequisites of the sheets, that isn't in the Gerber records. The creation print will detail what materials we are to utilize to build our board. Additionally, the creation prints reveals what size bored gaps we might want for any exceptional assembling directions or particulars we have to meet for our design. The details may include different data like shading patch veil or classification we might need for our design's overall structure.

With these two segments, we can fabricate a custom board that meets the client's prerequisites precisely. As PCBs are exceptionally adaptable, they can be structured and produced to different adaptabilities, sizes, and setups to fit practically any application. A PCB will help us to connect different components to a mainboard that already contains all the gaps holes and wiring embedded with the board. The overall structure of the PCB aids us when we are connecting our compatible components. There are three types of PCBs. These types include rigid, adaptable, and rigid flex.

5.15.2 Types of Printed Circuit Boards

1. Rigid PCBs

Rigid PCBs are developed of a rigid fiberglass substrate, making them functional and cheap, yet unyielding. (Circuits, 2019) They are simpler and more affordable to produce than their increasingly adaptable partners. However, they are significantly less flexible and difficult to fit into abnormal geometries or little zones.

2. Adaptable PCBs

Adaptable PCBs are used to highlight moderately significant bowing and collapsing capacities to fit into kept and strangely formed spaces. (Circuits, 2019) This quality makes them exceptionally flexible and ready to be utilized to bundle littler electronic gadgets. Moreover, as they are exceptionally versatile, the item doesn't need to be worked to fit around the PCB's limitations. Contrasted with unbending PCBs, they can offer more exceptional protection from heat.

3. Rigid-Flex PCBs

Rigid-flex PCBs have a mixture of both the characteristics of the unbending and adaptable PCBs. (Circuits, 2019) As opposed to the following two sorts of circuit sheets, these PCBs contain the aggregate of the electronic interconnectivity secured inside the board, diminishing the board's weight and all-around size. They are an excellent choice when ultra-light packaging is a critical need. Also, they are continuously durable and trustworthy while holding different quality and flexibility.

5.15.3 Composition

The essential materials utilized in the assembling of PCBs are fiberglass or plastic substrates, copper, patch veil, and terminology ink.

1. Fiberglass and Plastic Substrates

The base material, or substrate, is by and large fiberglass. Genuinely, the most generally perceived designator for this fiberglass is "FR4". This stable place gives the PCB its rigidity and thickness. (Circuits, 2019) There are, moreover, versatile PCBs dependent on versatile high-temperature plastic. Ordinary plastic substrates for versatile circuits join polyimide, liquid valuable stone polymer, polyester, and polyethylene naphtholate. The inspiration driving the substrate is to give a non-conductive base, at which point the conductive circuits can be assembled and shielded from one another.

2. Copper

Copper has a high electrical conductivity. Due to Coppers conductivity, it is the most used driving material for equipment in PCBs. (Circuits, 2019) The spreads portrayed over; all go with thin sheets of copper foil overlaid to one of the different sides of the plastic. The fabricator by then uses the Gerber records gave by the originator, to picture and to etch the circuits to meet the customer's necessities. The depth and number of layers are dependent upon the application for which the PCB will be used. Multilayered PCBs are worked by turning layers of copper equipment and securing materials to complete the PCB.

3. Solder mask

Solder mask is a liquid, by and large, an epoxy material that is applied onto the outer layers of firm PCBs. (Circuits, 2019) It is in like manner conventionally used on the resolute territories of stiff flex PCB's. Solder mask is mainly expected to shield the copper circuits on outer layers from oxidation from the earth. Solder masks propose a way to control and hold the weld movement when the PCB parts are gathered. Without a solder mask, the liquid tie could stream out onto the outside of the PCB, interfacing two coterminous circuits and short out the board.

4. Nomenclature

Once the solder mask layers are done, perceiving information, marks, and occasionally, scanner labels are engraved onto the solder mask. These engravings are called characterization, and they will similarly be described by records that were consolidated with the other Gerber layers. (Circuits, 2019) They are engraved onto the fixed spread to help ensure the exact social affair of the PCB.

5.15.4 Why Are Printed Circuit Boards Used?

Printed Circuit Board's little and lightweight configuration is reasonable to be utilized in numerous advanced gadgets, while their dependability and simple support suit them for reconciliation in complex frameworks. Their full determination of PCB plans can be used to serve a decent scope of businesses around the world, including military, clinical, aviation, PC, broadcast communications, and instrumentation. Below we offer a far-reaching diagram of PC circuit sheets to deliver relevant foundation data for what they do.

Clinical

Clinical equipment has chiefly benefitted from the introduction of PCBs. (Circuits, 2019) The contraptions in PCs, MRI machines, imaging structures, and radiation gear all despite everything advances in development from the electronic limit in PCB's. The slimmer and more diminutive size of versatile and unyielding flex PCBs thinks about creating continuously limited and lightweight clinical devices, such as convenient speakers, pacemakers, implantable gadgets, and little cameras for inconsequential prominent techniques. Unyielding flex PCBs are ideal for reducing the size of refined clinical contraptions, as they clear out the need for the flex connections and connectors that consume significant room in other diverse systems.

Aviation

Unbending, adaptable, and inflexible flex PCBs are generally utilized inside the aeronautic trade for instrument boards, dashboards, flight controls, flight the executives, and security frameworks. (Circuits, 2019) The developing number of advances in aviation innovation has expanded the necessity for littler, progressively complex PCBs to be utilized in airplanes, satellites, drones, and other aviation hardware. Flexible and inflexible flex circuits offer extreme solidness and mission survivability on account of the end of connectors. This makes them appropriate to be utilized in high-vibration applications, while their little and lightweight configuration lessens the gear weight and, subsequently, fuel utilization prerequisites. For applications where constancy is vital, they work an exceptionally dependable arrangement.

Military

In the military area, PCB's are utilized in hardware now and again presented to overwhelming effect, stun and vibration applications, similar to military vehicles, ruggedized PCs, current weapons, and gadgets frameworks. (Circuits, 2019) As military innovation advances to fulfill changing client requests, higher gear coordinates progressed automated innovation, requiring both the electrical and mechanical presentation that is natural in flex and unbending flex bundling. These sorts of electronic bundling can withstand a great many pounds of g-power without disappointment.

5.15.5 Computer Aided Design (CAD)

To design these printed circuit boards, we will have to create a virtual schematic from a computer-aided design software (CAD). The computer-aided design utilizes a whole model in a nonexistent space, permitting you to envision properties like stature, width, separation, material, or shading before the model is utilized for a specific application. A CAD helps designers in implementing layout from circuits. Electrical engineers are the most common individuals who use the features of a PCB designer software. A CAD allows engineers to leverage the software to create designs from a vast electrical library of components. PCB design is frequently utilized by building firms and groups that engage with item improvement. For a product to be categorized as a PCB design software, it needs to modify PCB layouts and circuitry. (G2, 2015) The list below includes four computer-aided design software we will consider for our project design.

Altium Designer

Altium Designer is the most commonly used PCB designer software out of the four listed in this section. (G2, 2015) This PCB design tool creates a strong interaction between the PCB design process that will establish a seamless connection to every aspect of our design. Within Altium, you can construct schematics, PCBs, layouts, documentations, and simulations. Using these features, we will be able to create top tier, design models. The following schematic and configuration reuse exist in one durable, simple to-explore UI that guarantees straightforwardness while you work. Plan unbending flex in full 3D and affirm that the 3D segment, case get together, and PCB clearances meet every mechanical prerequisite.

Autodesk EAGLE

Eagle is an electronic design automation (EDA) software that permits creators to make printed circuit sheets (PCB) consistently by utilizing schematic charts, segment situation, PCB steering, and great library content. Eagle contains a schematic director highlight that can be utilized for organizing circuit diagrams. Schematics are taken care of in reports with. SCH extension, whereby parts are portrayed in contraption libraries with. (G2, 2015) LBR enlargement. Components can be put on various sheets and related together through ports. The PCB group article supervisor stores board records with the development. BRD. It allows back-remark to the schematic and auto-coordinating to follow subjects to the affiliations described in the schematic. It has a friendly user interface and substantial schematic diagram work. The product is not free, but its expense is acceptable. It has a vast community base that always gives back positive remarks.

KiCad EDA

KiCad is an open-source electronic design automation software that aids in handling schematic capture, PCB layouts, and Gerber file outputs. KiCad operates on most operating systems, such as Windows, Linux, and macOS, licensed under GNU GPL v3. (G2, 2015) Tools exist within Kicad that work on an electrical list of packages to create bills of components, artworks, 3D models of circuits, and Gerber file outputs. The KiCad software divides up into five main sections:

- kicad - The Project Manager.
- eeschema - The schematic capture editor.
- cvpcb - The footprint selector for components
- pcbnew - The PCB layout program.
- gerbview - The Gerber Viewer.

Contrasted with its rivals, KiCad has similar highlights and capabilities. KiCad unravels all phases with the same interface; Schematic Capture, Gerber age/perception, PCB format, and library altering. There are in like manner gadgets to help with acquiring parts from other EDA applications, for instance, EAGLE. There are many electrical components available within its infrastructure that make KiCad easier to manipulate and use. Record positions are acceptable content and all-around archived, which is useful for CVS or Subversion and create robotized part age content.

SolidWorks PCB

SolidWorks PCB is powered by the Altium system, which provides the best PCB design technology with an integrated mechanical solution. (G2, 2015) Through the use of SolidWorks PCB, we are able to create models or designs of Printed Circuit Boards productively. SolidWorks system enables users to quickly create models that collaborate between the electrical and 3D mechanical design teams. Additionally, it aids designers to provide a bright bit of leeway where ECAD cooperation is essential for achievement of electronic item plan. Version control is another feature embedded within the SolidWorks software that manages and compares all history and changes conducted on the project design. The new SolidWorks PCB is powered by the Altium design tool. Through the usability of Altium, we can have power and speed with the 3D expertise of SolidWorks.

The SolidWorks PCB software can oversee changes to sheets structures, electrical segments, mounting gaps, and patterns for wiring positions. The open blocks' plans are made of copper, which makes it simpler to direct propelled examination of warm or vibration reenactments. The UI for this PCB planner is structured in a manner to expand profitability and convenience. (G2, 2015) A client will have the option to appreciate a cutting-edge interface, that will help explore the various alternatives SolidWorks gives. Join and manage your schematic and PCB part libraries together in the alone zone. Reuse portions that you know and trust from past productive endeavors. Open neighborhood SOLIDWORKS archives in SOLIDWORKS PCB, and open PCB records in SOLIDWORKS. Imported mechanical models are precise and hold your exceptional arrangement information. Lastly, SolidWorks PCB features a streamlined editing environment that offers all the of necessary tools to bring your design to life. Within this program, you can easily reuse design elements from its vast electrical library manager. Once completed with your design, you can keep it all in an organized hierarchical multi-sheet design.

6. Microcontroller and Sensor Software Design

The goal of researching the microcontroller software design is to understand the flow of information. This is an opportunity to research the tools we will be working with.

6.1 Microcontroller Software Design

As we will be using an ATmega2560, we will investigate the development environments available for it. The ATmega2560 uses the Arduino Software (IDE) as suggested by Arduino. It is a compact software which allows users to write code and upload it to the device through USB. To assist in the development process, Arduino includes many libraries which make coding more straightforward. Some notable libraries are listed below.

- **SPI** – Communication with devices using the SPI bus
- **Wi-Fi** – Connection to the internet using Arduino Wi-Fi
- **Wire** – I2C reading and writing over multiple devices / sensors
- **Serial** – Connection with devices through UART

These libraries are installed directly into the Arduino Software IDE where they will be available to use. One of the other benefits is that the serial data being sent from the Arduino board is done directly within the IDE. This is a huge help when dealing with the UART protocol.

6.1.1 The Loop

Once initialized, the microcontroller sits in a loop where it lives. The loop continuously checks to see if any peripherals have contacted the MCU. This is the most straight forward approach to checking for input, but it is unreliable and uses the most energy.

6.1.2 Interrupts

Interrupts are an advanced method of handling processes. They benefit in two ways:

1. **Precedence** - When an important signal is detected, the processor immediately handles the task, then resumes the original task. This strategy allows precedence to be given to specific peripherals.
2. **Power Saving** - The MCU may be set to be in a sleep state until one of the sensors triggers an interrupt, provoking it to awake and handle a given task, then returning to sleep.

The interrupt execution response for all interrupts is five clock cycles minimum. We will be using interrupts for every component in which we can. Some important interrupt sources are given below.

- SPI Serial Transfer Complete
- Timerx Compare
- External Interrupt Request
- UARTx RX Complete
- UARTx TX Complete
- 2-wire Serial Interface

6.1.3 Sleep Modes

To take advantage of the interrupts power saving benefits, we will want to put the microcontroller into a sleep mode when there is no urgent task to be completed. Some of the modes we will be interested in are:

Table 46 Sleep Modes Specifications

Available Modes	Active Clocks	Active Oscillators	Wake-Up Sources
Idle	IO, ADC, ASY	Main Clock, Timer Osc	Pin Change, Address Match, Timer2, SPM, ADC, WDT, I/O
Power-down			Pin Change, TWI, WDT
Power-save	ASY	Timer Osc	Pin Change, TWI, Timer2, WDT
Standby		Main Clock, Timer Osc	Pin Change, TWI, WDT

6.2 Sensor Software Design

In the following sections we will discuss the software design for all sensors and peripherals in in the project. These primarily include UV light, RFID, REED, Solenoid Lock, Barcode scanner, fingerprint sensor, and camera. The main topic of discussion is the software decision making in the form of a flowchart.

6.2.1 UV Light

The UV light will mostly be controlled from the app that accompanies S.M.A.C. most of its features and timed schedules will be controlled from there. The UV light's on and off feature will be controlled from a relay module which will be controlled from the ATMega. The ESP32 will be the medium from which the app and the ATMega will communicate. A HIGH and LOW signal will be sent to the relay which would either turn on or off the UV

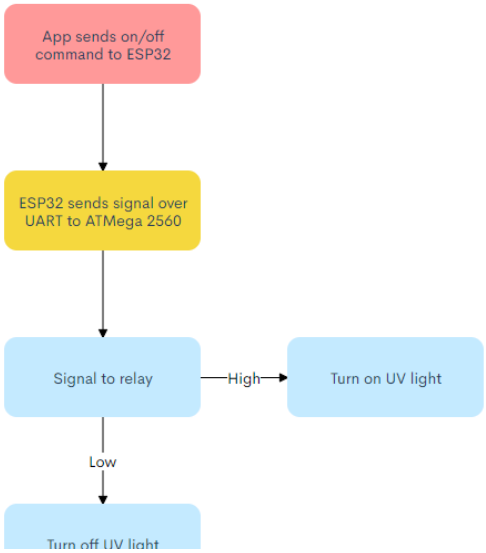


Figure 75 UV light flowchart

light. This will allow for complete control of the light from the app which will allow to create schedules and timed disinfecting hours.

6.2.2 NFC/RFID

For our NFC/RFID sensor we want to have the delivery man to have an ease of access to the smart box. Whenever they swipe their card the box will automatically open for them so they can place their package in the box. Below we will lay out the logic to how the software for opening the box will be for the person who will be delivering.

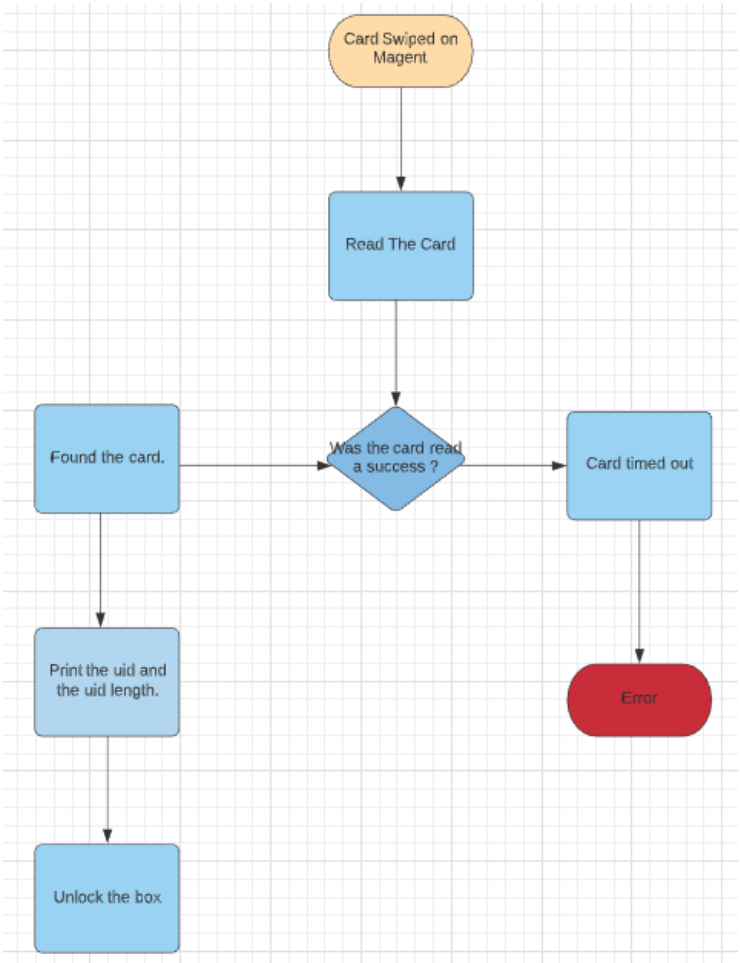


Figure 76 NFC/RFID Software Flowchart

6.2.3 Open/Close Sensor

For our open and closed sensor, we show the logic for the reed switch we will be using to detect whether our door to the smart box is open or not. For our design whenever the door is open, we want for the sensor to detect when its open and send back a response to our Arduino telling us that the door is now open. We will do the same reasoning and logic for when the door is closed. We want for our reed switch to send back a response in real time

whenever our door is closed to our Arduino. With the Arduino that we will be using it has a library that they have created to specifically work with reed switches to have an ease of use. With the given library we are able to assign what state the door is in and as well the transitions from when to is going from closed to open or vice versa. Whenever the state is LOW the door is closed. When the state is HIGH, the door is open.

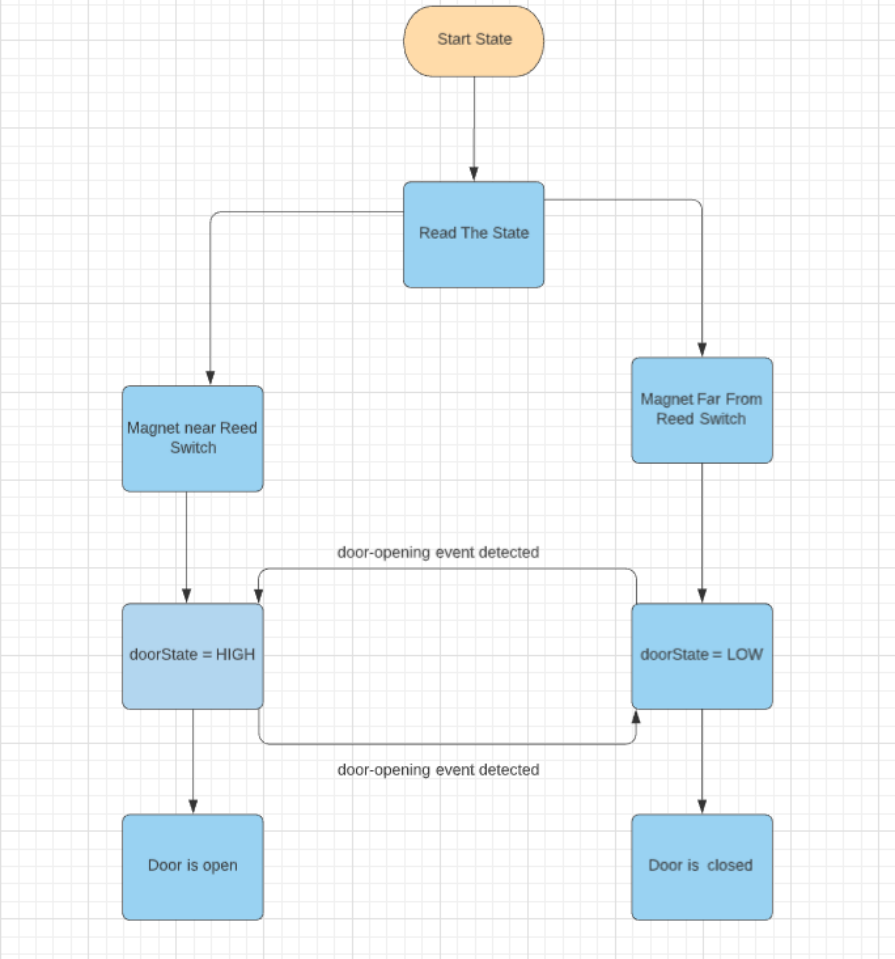


Figure 77 Open/Close Sensor Software Flowchart

6.2.4 Lock

Our electric lock design uses an electromagnet coil to convert electrical energy to mechanical energy. To make our container's door hold, we will be creating a custom door latch for the solenoid lock. The electronic solenoid lock has already been prebuilt by the manufactures. This prebuilt design will help us easily position and connect all the necessary parts from the lock to the MCU. To operate the solenoid lock, a voltage of 12V needs to be applied across the unit. This voltage will aid in powering the coils and will start the

functionality of the electric lock. We will currently incorporate a transistor with the lock to act as a limit switch that would activate the lock. When a high voltage signal of 12V is sent to the lock, the lock will unlock the box. Likewise, if a voltage low is posted to the solenoid lock, the box will lock and secure a user's package's contents. The lock will be positioned in front of the container. This position will make it easier for individuals to know the status of the lock. Once the lock has been activated, a sound from the coil will be heard to indicate that the lock is successfully moving its gears and the metal slug. The barcode scanner chosen will be used to safely lock or unlock a user's package. Based on the design of the locking mechanism, the lock will bring security and comfort to a houseowner. Additionally, the lock will host many capabilities towards the owner of the box. A portion of these accessible capacities will empower us to:

1. Lock and unlock a user's package.
2. Control the lock via a mobile app.
3. Lock the box after inactive for 30 seconds.
4. Monitor when the lock has been unlocked or opened.

The software process for locking our container is given in the flowchart below.

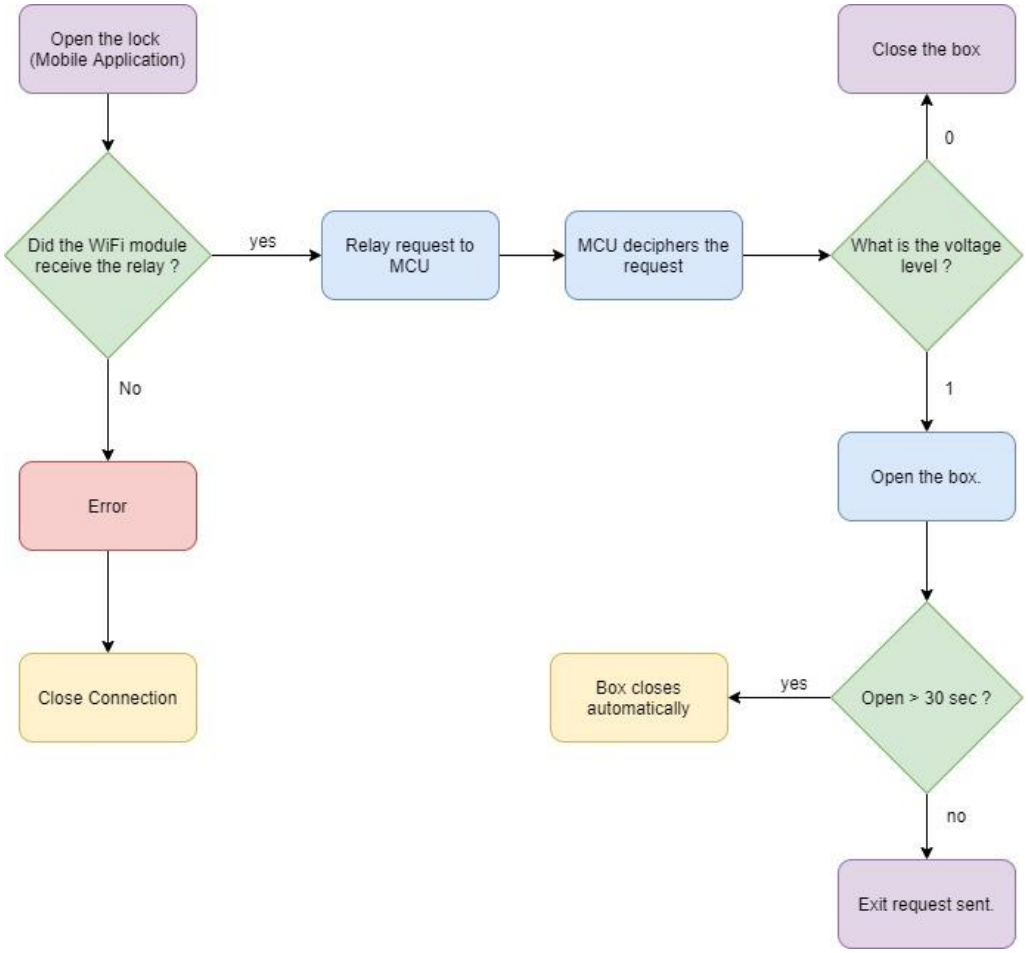


Figure 78 General power lock software logic

6.2.5 Barcode Scanner

The barcode scanner chosen has an opensource library for Arduino based stages. This library permits a client to make a working standardized identification scanner that will yield the alphanumeric digits from a bundle. A portion of these accessible capacities will empower us to:

1. Scan a package with a barcode.
2. Relay the barcode to the MCU.
3. Broadcast a red beam to read the alphanumeric digit.

The software process of scanning a verified package's barcode is showcased below.

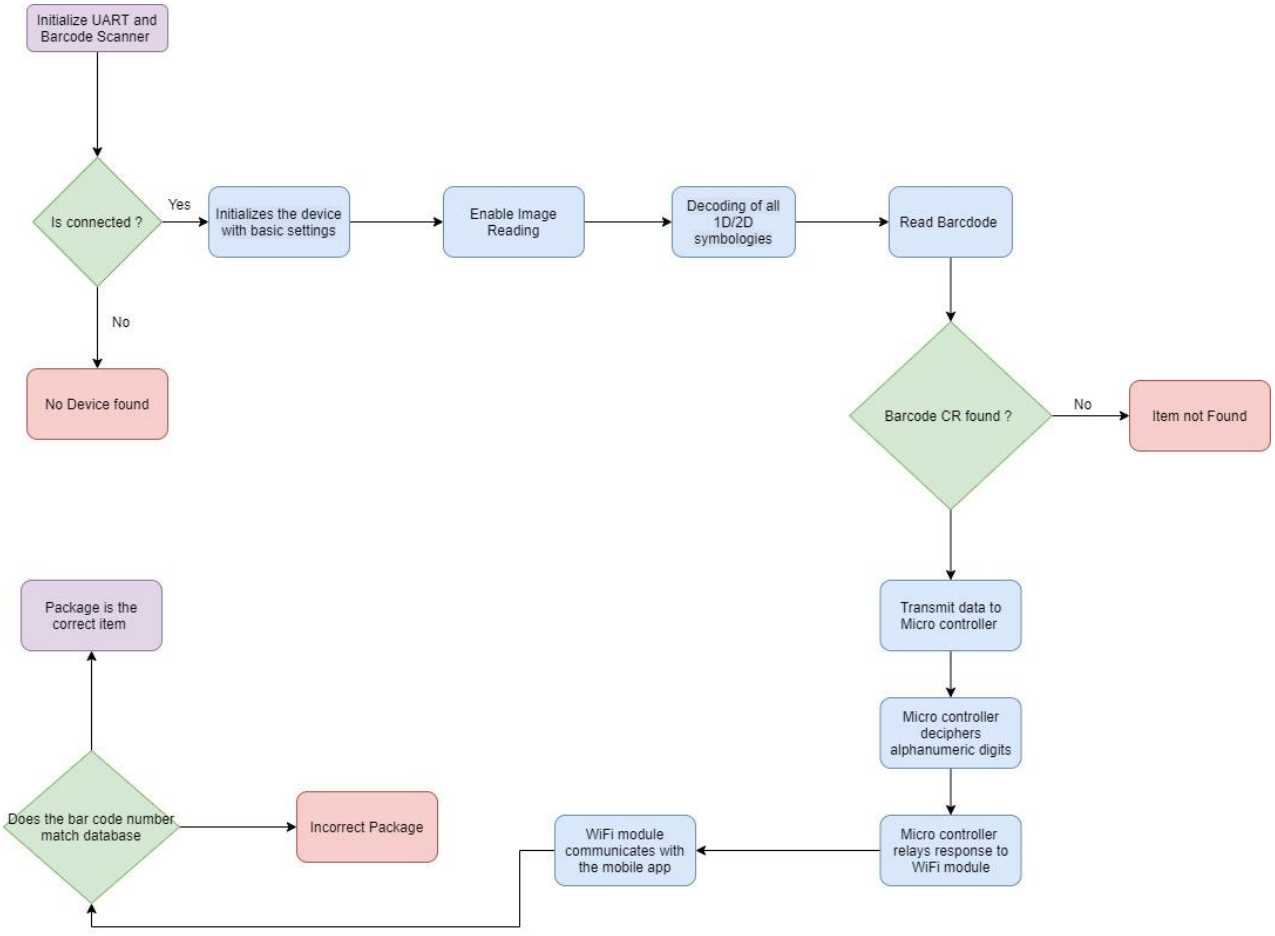


Figure 79 Barcode Scanner Software Flowchart

6.2.6 Fingerprint Sensor

The fingerprint scanner has its own database which we may interact with using the opensource library provided for Arduino. These interactions include but are not limited to:

- **Enroll** – Add a new fingerprint to the database
- **Delete** – Remove a known fingerprint from the database
- **Read Fingerprint** – Scan and check to see if the fingerprint is valid.

The software process for scanning a known fingerprint is given in the flowchart below.

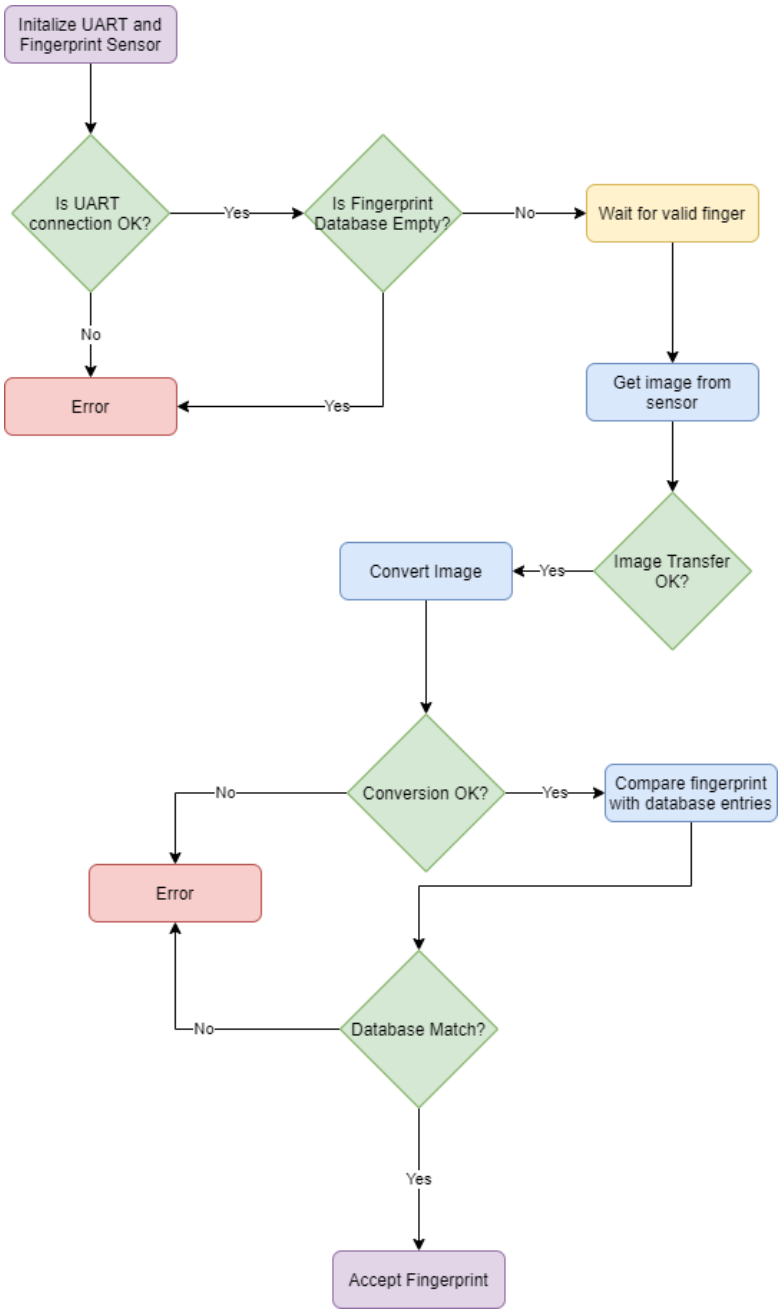


Figure 80 General fingerprint scan software logic

At every step of the way, the responses from the sensor are more verbose what is shown in the flowchart.

Connection

- Found fingerprint
- Did not find fingerprint

Fingerprint Database

- Contains Fingerprints
- Is Empty

Database

- Found a print match
- Communication error
- Did not find a match
- Unknown error

Get Image from Sensor

- Image taken
- No finger detected
- Communication error
- Imaging error
- Unknown error

Conversion

- Image converted
- Image too messy
- Communication error
- Could not find fingerprint features
- Unknown error

Although there is a database built in, some administrative functions will need to be performed from the mobile application. The app should be able to communicate with the MCU to perform the following actions:

- Delete known fingerprints
- Enroll new fingerprints

6.2.7 Camera

The general camera logic is shown below. We want the camera to take a picture every time a barcode is scanned, no matter if it allows access or not. The camera selected has an opensource library for Arduino based platforms. This library allows the user to create an ArduCAM object which may be interacted with. Some of them available functions provided allow us to:

1. Set the sensor to output type
2. Set the picture resolution
3. Capture and buffer the image

The software process for taking a picture is given in the flowchart (Figure 81) below.

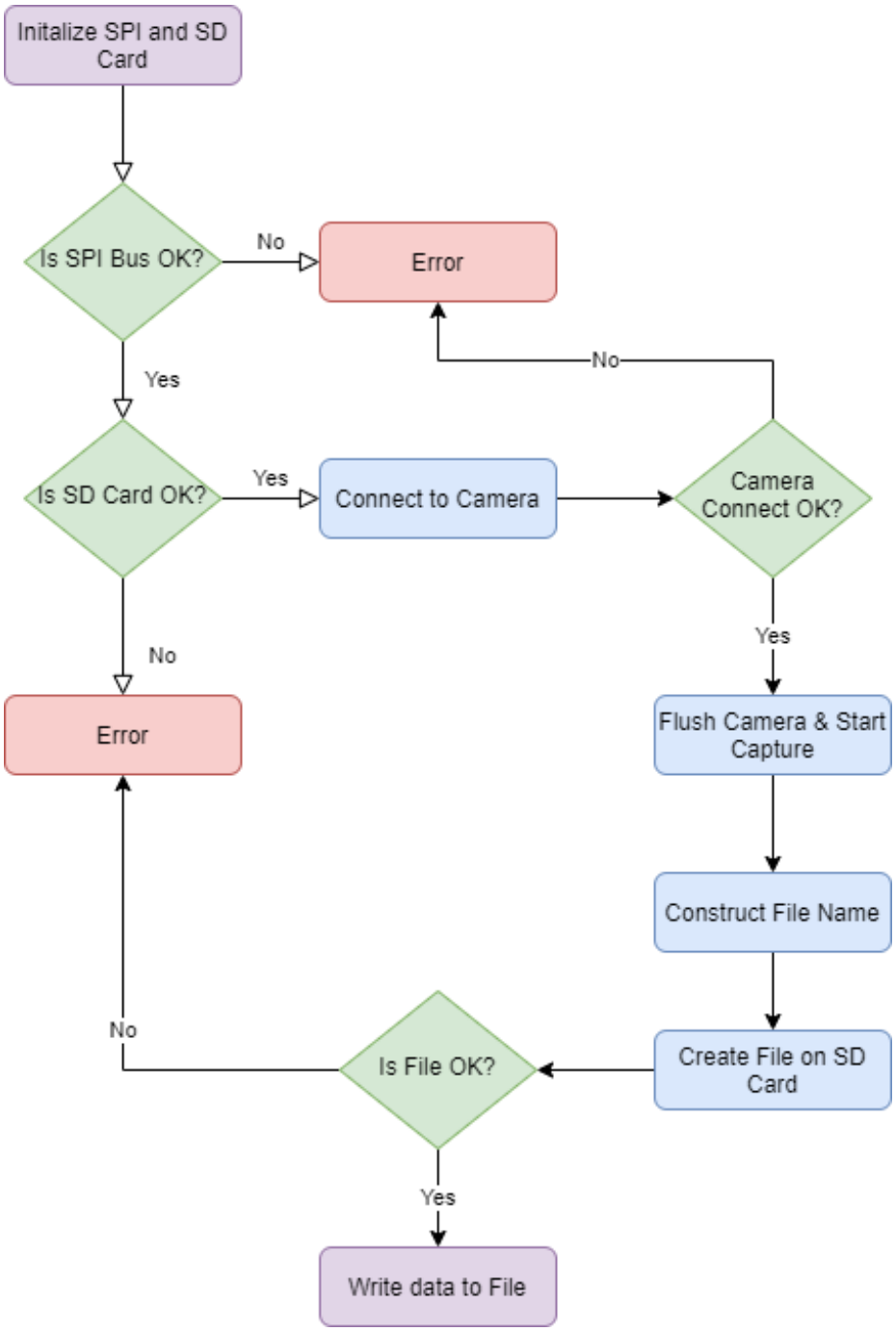


Figure 81 General camera capture software logic

6.3 Wireless Communication Software Design

This section will cover how the ESP32 will communicate with the outside world. Which is a crucial part for the extended features the box will contain from the accompanying app. JSON will be used to parse most of the data collected from the sensors attached to the ATMega.

6.3.1 JSON For Transmission

JSON is crucial for transmitting data between our server and the ATMega. JSON allows for a text-based representation of structured data that is based on key-value pairs and ordered lists. JSON is commonly, but not exclusively, used to exchange information between web clients and web servers. The popularity of JSON has also resulted in native JSON support by many databases. JSON is a generic data format with a minimal number of value types: strings, numbers, Booleans, lists, objects, and null. Although the notation is a subset of JavaScript, these types are represented in all common programming languages, making JSON a good candidate to transmit data across language gaps. JSON data is stored in files that end with the .json extension. In keeping with JSON's human-readable ethos, these are simply plain text files and can be easily opened and examined. just about every language you can name can read and process plain text files, and they are easy to send over the Internet.

6.3.2 JSON Read and Write

Communication will be done through HTTP GET and POST requests. HTTP, or Hypertext transfer protocol, is designed for communication between clients and servers. The GET method is used to request data from the server and in turn read the JSON file. This is one of the most common HTTP methods along with the POST request which is used to write information into in this case a JSON file. The POST request is used to send data to a server to create or update resources. For GET requests, the URL contains the information being requested whereas POST requests have the data stored within the body of the HTTP request.

6.3.3 ESP32 Server

For the user to communicate, the application will send a request to the ESP32 which will act as a server for information, or to update information, and the server will transmit that request to the ESP32, the ESP32 will then transmit the result of that request back to the server which is then forwarded to the user. For ESP32 to communicate with the user, the ESP32 will send a request to the server for information, or to update information, and the server will transmit that request to application, the application will then transmit the response of that request back to the server. The intention of the ESP32 speaking with the server to forward messages to the user is to inform the user of the conditions of the box.

7. Mobile Application Software Design

For this section of our paper we will dive into the actual design, format, layout of our application. We want for our app to have a great client-side experience for the user to feel secure with their package whenever it gets delivered, it should have ease of access with all the tools that will be available to them. As well it should have great performance for whenever the user uses the application for any action or events that happen.

7.1 Database Design

For our data base design, I think what would best illustrate how we potentially want our application to work would be with an ERD (Entity Relationship Diagram). With an ERD we can property display all the proper information we will need from our users but as well how all the event will happen. With the ERD we can represent each state of the program and what information each state has on it and what will be sent to which part of our application.

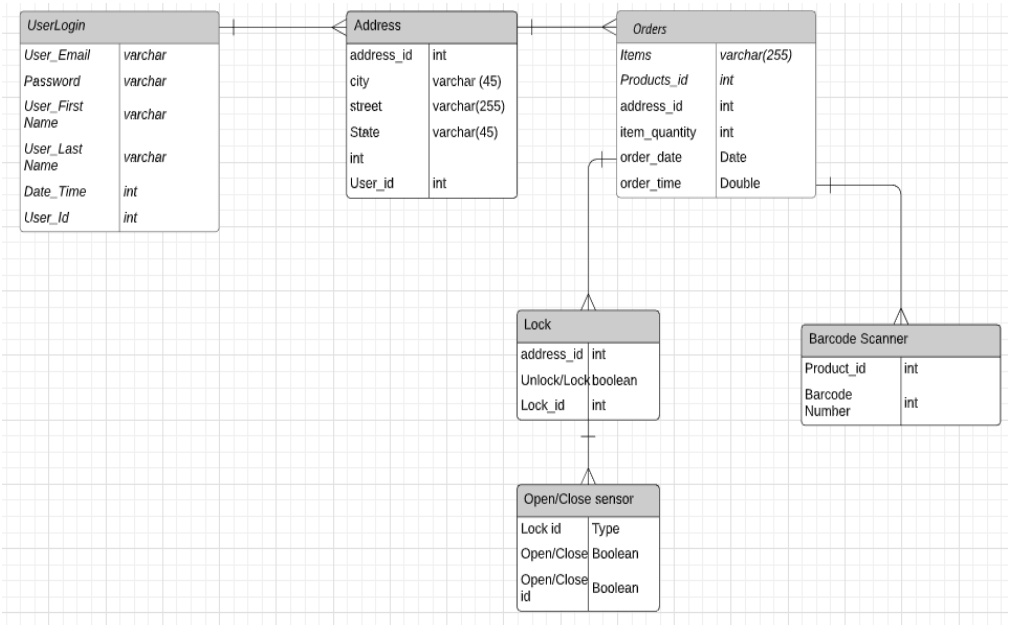


Figure 82 Our Database design and layout

7.2 Mobile Application Design

In this section we will explain in detail how we will design and implement our mobile application for the smart box. For the design we want our app to have ease of use but yet have great performance for the app. With it we want user the be able to register an account with the application, be able to securely login and use their password, be to unlock their smart box from their mobile application, view their pictures, see the tracking number, have account settings and see their delivery history. Down below we will get into the exact design of the app and the logic behind it that will map out how the app works.

7.2.1 Login / Register

For our login we want the user to be able to write in their username then their password. When the user will type in their user and password, we will check whether the user and password exist in the database then let them securely login, if they were to end up not having a valid user and password it should throw an exception error at the user giving them another chance to login. If the user does not have account, we will have a register button right down below that will allow the user to create an account for them. With the creation of an account we will ask the user for their first name, last name, email, and an address, we will then ask them to create a user and password for the account.

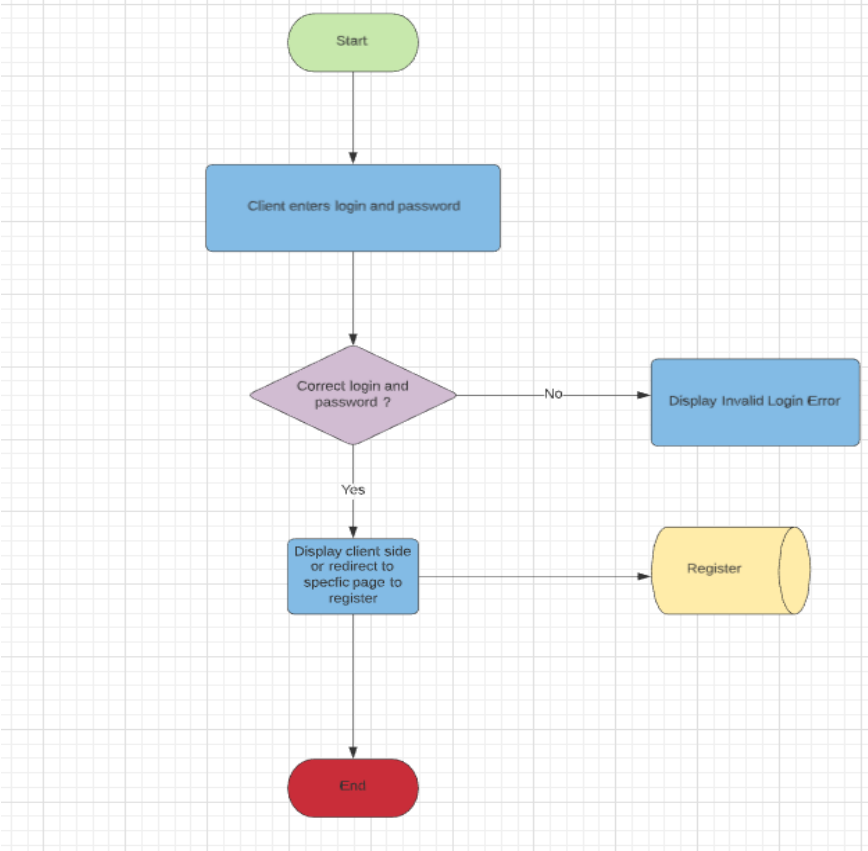


Figure 83 Login/Register Software Logic Flowchart

7.2.2 Unlock Box

For the user to unlock the box we wanted it to be extremely simple. For a user to unlock the box all what they will need to do is press the unlock button and they will unlock the box. As well the user should be able to do the exact opposite. Close the box then lock it with a lock button on the application.

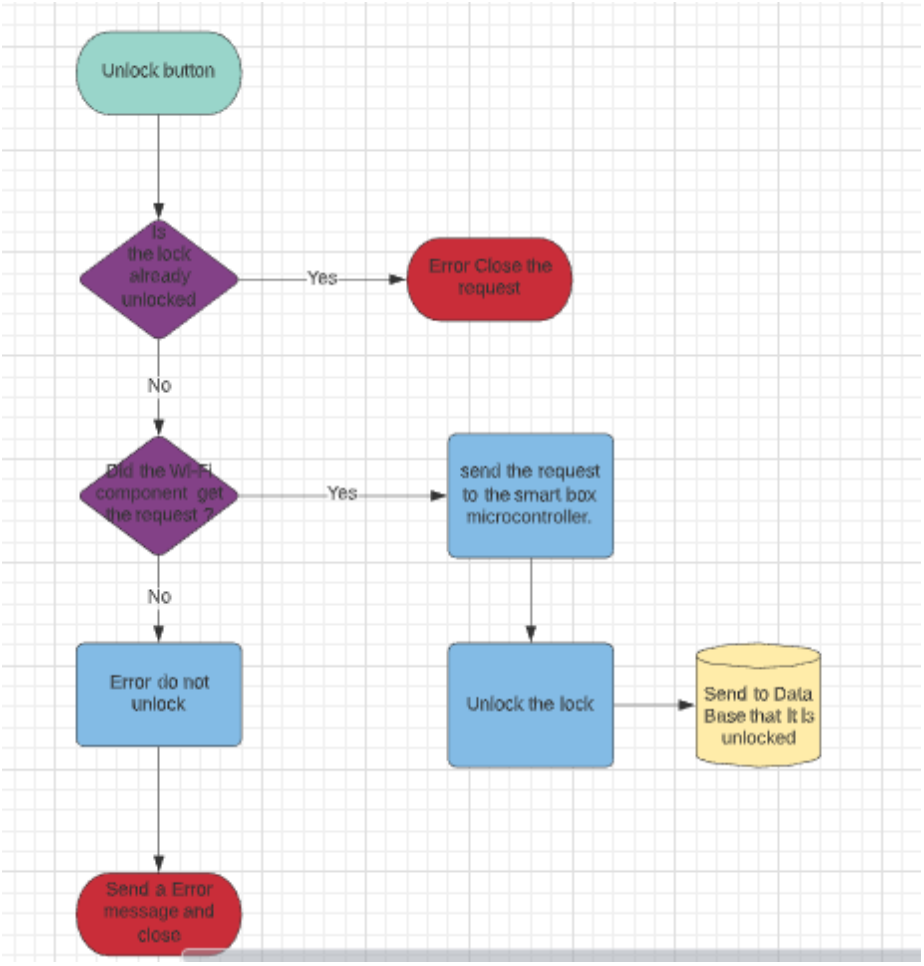


Figure 84 Unlock Button Software Logic Flowchart

7.2.3 View Pictures

Another feature we want to do is when the box unlocks or when an item is scanned , the unlock or the scanner will trigger our camera to take a picture from where we mounted the camera and send that image to the database and have the app pull from the database and alert the user that the box has been unlocked and send them a picture of the person who has unlocked the box. We want to do this because we think it a good protocol for security measures so the user feels safe whenever their box is opened and in the event that someone unlocks the box who is not supposed to, we now have saved in the database a picture of who opened the box.

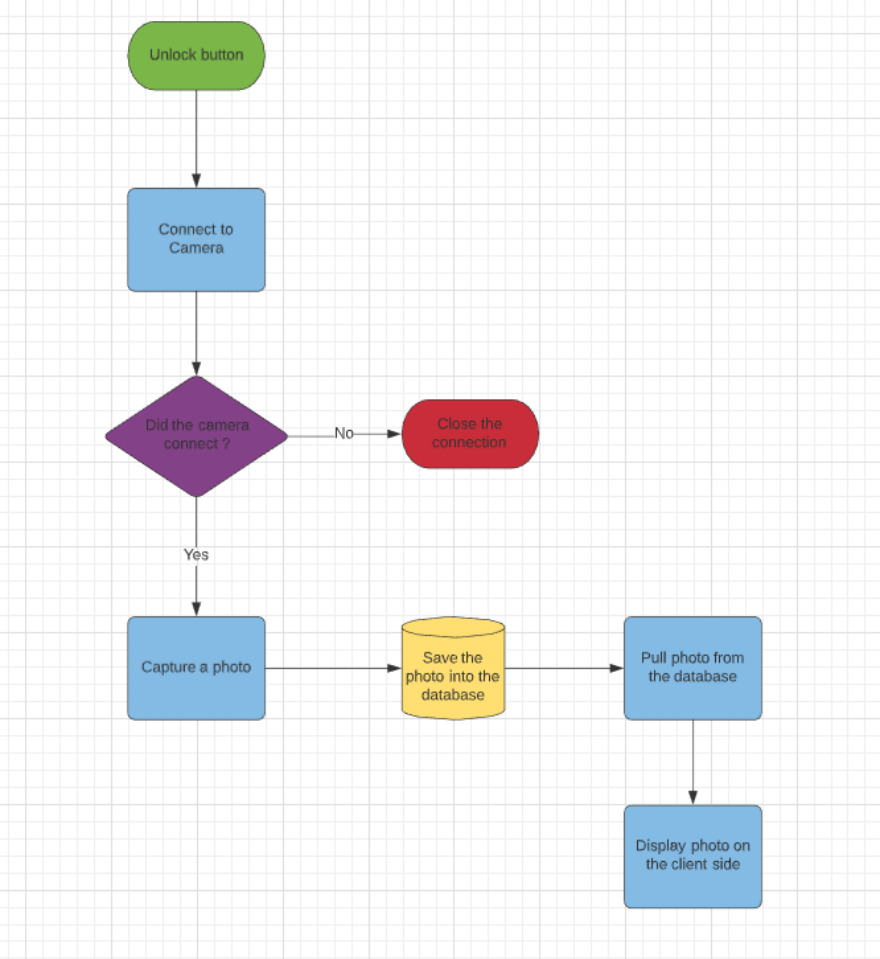


Figure 85 View Pictures Software Logic Flowchart

7.2.4 Enter Tracking Number

For this section of the paper we will detail how we will display the tracking number to the user but as well save the tracking number into our database for other uses with in our application. For this section, the main method we will be using to input our track number will be the

7.2.5 History

For this section we want the user to have a saved history section for their account so they can see exactly what has been ordered and delivered within in the past and all the details associated within the order. When a user clicks on the history button, we will send them to a page where all the orders they have ordered have been delivered. We will provide them a list of all the orders and display as well as the date it was delivered, the time it was delivered, and the tracking number associated with that package.

8. Testing

In this section, we will discuss the testing of all the peripherals used to create S.M.A.C. Each peripheral will be individually tested in both its wiring and its software.

8.1 Wi-Fi Testing

The main purpose of the Wi-Fi module testing will be its ability to transmit data between our ATMega and ESP32 using UART. The second most important part we are testing is for the device to successfully pair to the internet by connecting to a host. Once connected the last part we are testing is the ability to use GET and POST commands and successfully read and update an http server.

UART connectivity

The testing ideology for this part will be to send a string from the ESP32 to the ATMega and see the printed statement on the ATMega console. This step will be repeated in the other way. The ATMega will send a string to the ESP32 and the ESP32 will print it in its console. A successful test will signal the UART connection working successfully and should be able to accomplish all the features needed for the project.

Successful internet connection

This testing porting is the simplest. The fundamental portion of this section is to connect to the internet. A connection to an available 2.4GHz will be made. Once connected to the internet a PING will be made to a website and print the ping time if a successful ping was made it will print in the ESP32 console. A successful connection will signify that the Wi-Fi module will be able to connect to our server and transmit data between the app and our devices.

GET and POST commands

The last portion of the testing will be to test the GET and POST commands. We will start by using a POST command to write into an API a JSON string. Once successful we will have string that outputs into the console the successful POST command. The next step is a simple GET command. From the POST command we used we are going to try and print the same information that was POST into the console from the API we used. This will allow us to confirm the POST command and the GET command are working properly.

Successfully completely all these tests will allow us to build upon our main features for this project. This will help us begin to shape our API for our app and the logistical connections between all of our devices.

8.2 UV Light Relay Testing

The UV light will serve to disinfect packages for safe handling of the user. The downside of this primitive mechanism is that alone it has no capabilities of being controlled by our ATMega which takes huge hinderance to the safe use and effectiveness of this product. As result of this dilemma we have added a relay module to be able to control the flow of electricity towards the light and thus turning it on and off essentially acting as a switch. In the testing for this component we would connect the relay to a digital pin and set it to output mode. A HIGH signal will activate the relay thus let current through and a LOW signal will turn of the relay in turn turning off the light. To test this a multimeter was set to record the voltage passing through the relay. A program was created to switch the relay from HIGH to LOW signals every 10 seconds and voltage was tested across the relay. A simple 5V feed was given to the relay to simulate voltage.

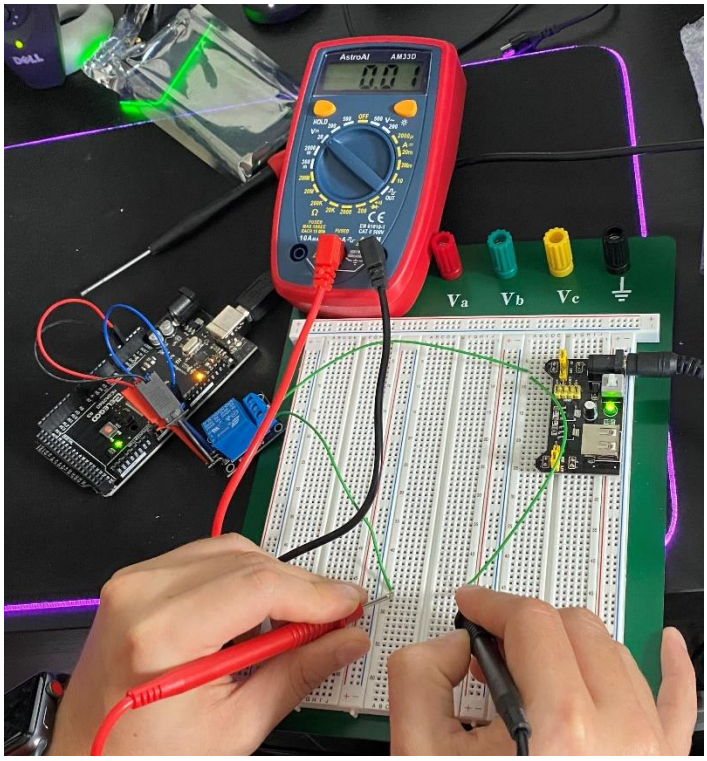


Figure 86 Relay testing off

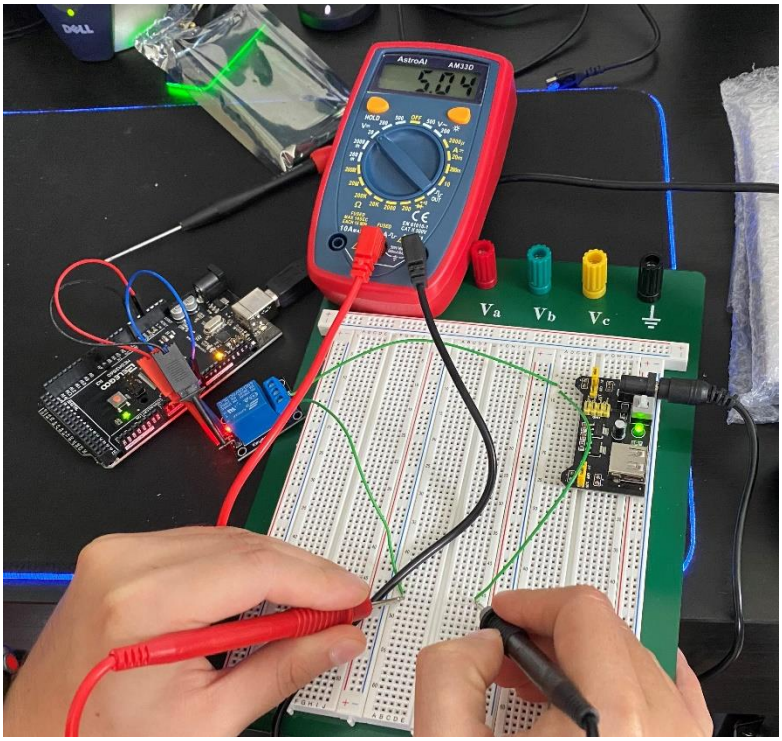


Figure 87 Relay testing on

8.3 Power Component Testing

The main points we must stress in the power testing is that all voltages are being converted efficiently. We have three different voltages that are needed for the project to work correctly. A 12V source that will feed the solenoid lock. A 5V source that is going to power 3 components. Lastly, a 3.3V source is needed for power an additional 3 components which will make up all of S.M.A.C.'s components. The following will be what will be tested to make sure the voltage conversions and connections are acting as intended:

- Correct voltage from buck converter, voltage regulator, and transformer
- Can support current needed under load
- Correct heatsinks are provided for voltage regulators

The most critical part of this testing is to make sure that we are getting the correct and accurate voltage without a lot of variation so that the components are not damaged. Most of the components do not have a protective measure when being fed raw voltage into the boards. A wrong input could possibly damage the components or prevent them from working properly.

Another important point that will be tested is the current necessities needed for all the components to work. We have 7 different components with different current requirements that need power. If the circuit is overloaded, it can cause the components to shut off or malfunction which is why current will be an important metric to keep a close eye on. With the components that are being used I have calculated that at least 2A are needed to power all components with room for current spikes.

To finish off the testing portion of power system is to make sure that the heatsinks being used for each of the voltage converters are enough to cover the power dissipated. Without the correct heatsinks the voltage regulators can melt or even catch fire if under enough stress.

8.4 Barcode Scanner Testing

The Barcode Scanner will be used to unlock our smart box when a user's unique package is scanned. Within the structure of the scanner module, a red light is flashed when the scanner is triggered. This will then extract the alphanumeric digits, where we compare the barcode values to our user's database. The use of serial communication will be used to transmit and receive data from our scanner module to a PC or a database. In Figure 88 below we showcase the testing of our DyScan Barcode Scanner.

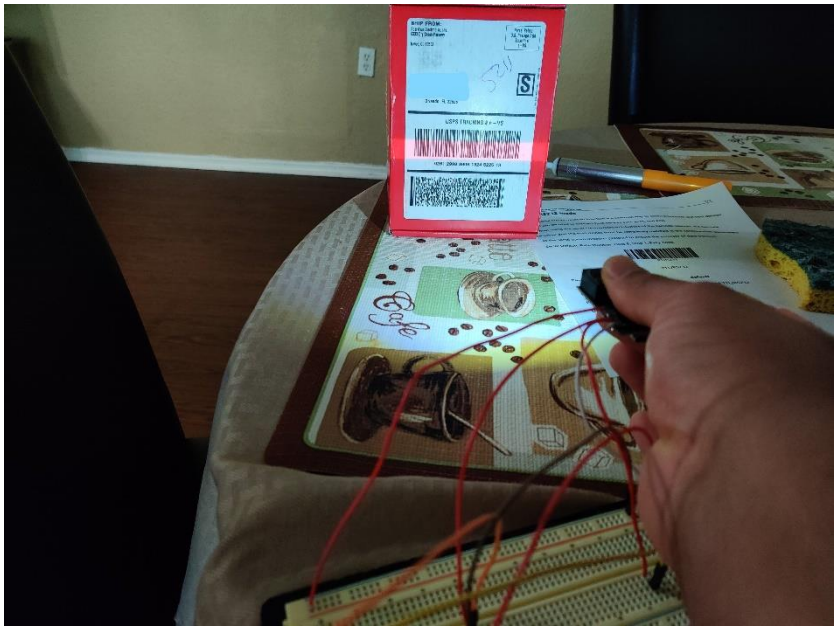


Figure 88 Barcode Scanner Triggered

Before any scanning can be performed, the barcode scanner needs to be set up to use serial TTL communications. As stated from before, these serial communication pins are the main forms of communications for both the fingerprint scanner and barcode scanner. This will place the module in the right mode to get and transmit sequential data. The image below illustrates the set-up process for initializing the barcode scanner into its TTL mode.

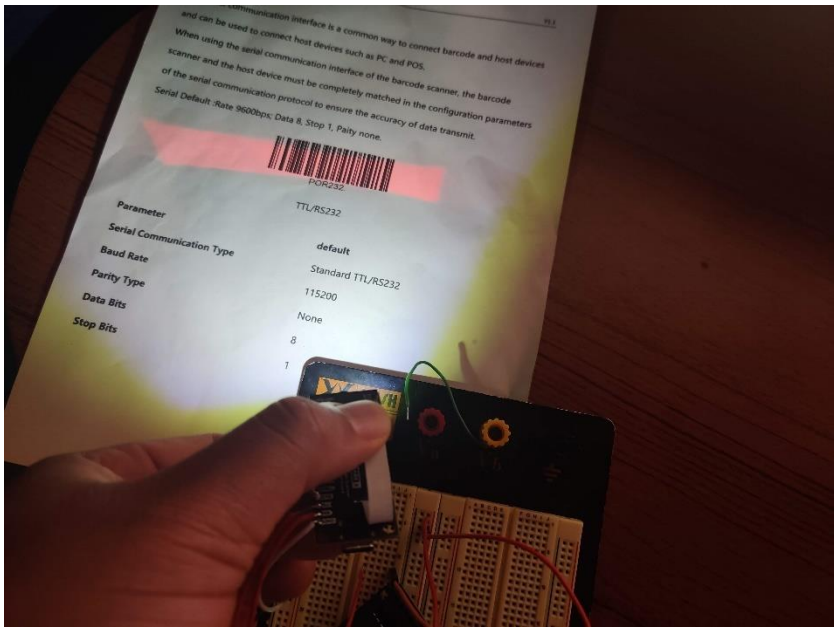


Figure 89 Barcode Scanner TTL Mode Setting

Through our testing, we were able to successfully obtain the barcode numbers on a package. In Figure 90, we showcase our results from scanning an item with a barcode.

Figure 90 Extracted Barcode Numbers

8.5 Lock Testing

The lock for our device will be one of the main components that offers security towards a user's package. Precise testing will be conducted to meet the satisfactory needs we created from our initial idea development stage. The smart lock implemented will only open or close, based on several different scenarios. The lock's primary unlocking feature encompasses the use of a mobile application to relay a message to our microcontroller to unlock the box. Within the testing stages, we will be testing the following operations.

- Unlock/ Lock via a power source
- The box secures itself when the lid is closed.
- The lock firmly latches unto the custom fasteners created.

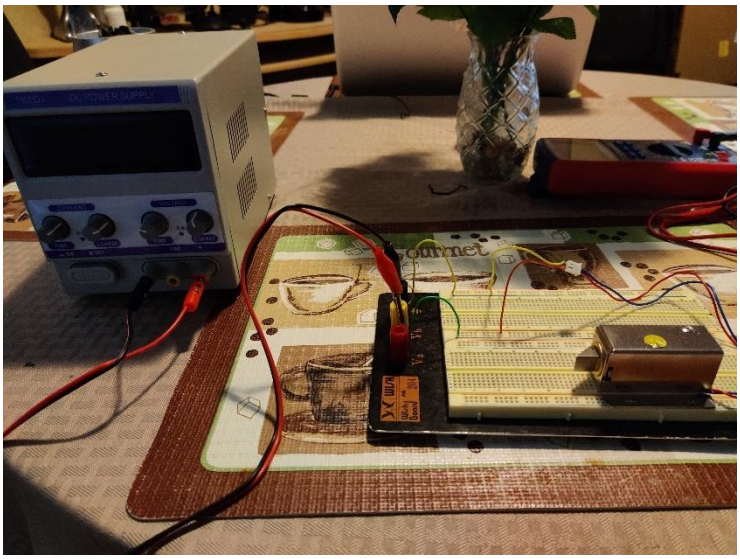


Figure 91 Barcode Scanner in Lock Position (0V)

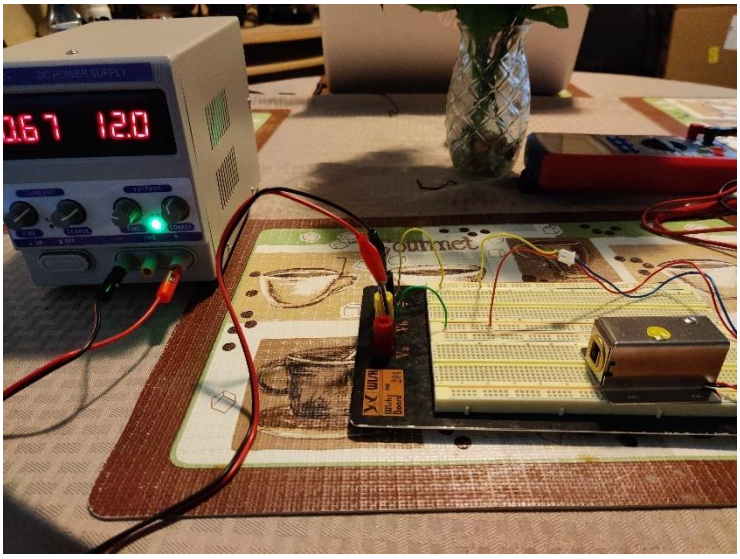


Figure 92 Barcode Scanner in Unlock Position (12V)

Based on the figures illustrated above, a 12V power source is needed to operate the solenoid lock. Additionally, a large amount of current is needed to charge the metal coils inside of the lock design. The amount of current required to charge these coils range from between 500ma to 650ma. Once the lock is charged, the solenoid lock converts the electrical energy into mechanical energy to shift the metal slug inside of the lock. This movement of the metal slug will cause our box to be in either a lock or unlock state.

8.6 Fingerprint Testing

The fingerprint sensor uses UART to communicate with the microcontroller. Tested first was the UART transmission signals leaving the ATmega2560. This was done using the logic analyzer included in the Digilent Discovery 2. The string “Hello, world.” was sent through TX1 using hardware serial. The transmission was read correctly as each character was interpreted properly by the logic analyzer.

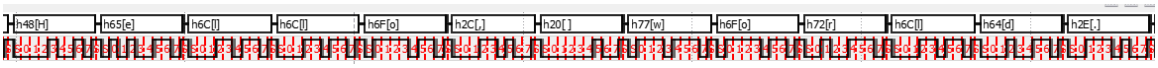


Figure 93 UART Logic Communication Verification

The next item to test is that the TX line outputs 5V. The measured output is given below.

DC	4.85V
True RMS	4.85

Table 47 Measured TX Output

This is a problem as the RX pin on the fingerprint sensor requires at least 2.0V, but less than 3.3V to be recognized as high voltage.

Min Low (VIL)	0.8 V
Min High (VIH)	2.0 V
Max Voltage	3.3V

Table 48 Fingerprint UART Voltage Requirements

A prototype solution is to create a simple voltage divider with a 10k and 20k ohm resistor. The resistor values were measured and are given in the table below.

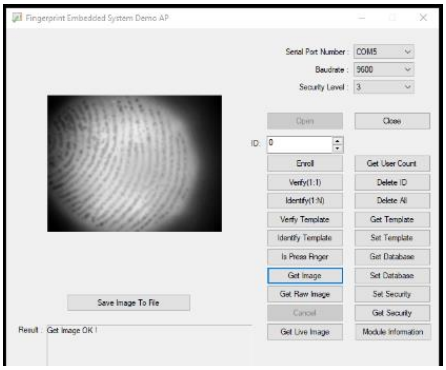
	Theoretical	**Actual**
R1	10k	9.75k
R2	20k	19.3k

Table 49 Voltage divider resistor values

The new measured voltage from the Arduino’s TX pin is 3.223V.

DC	3.223V
True RMS	3.223V

Table 50 Measured TX Output with Voltage Divider



The same logic analyzer test was performed again, and “Hello, world.” was read correctly. The remaining hardware connections were made, and the device powered on. Now, the features of the sensor were tested. First, a connection to the sensor was made using the open source test software provided by Sparkfun. (Sparkfun, 2019) As seen in the figure to the left, the connection was successful, and a picture of a fingerprint was downloaded from the sensor.

Figure 94 Picture of fingerprint on demo software. Permission to reproduce requested.

The fingerprint shown above was enrolled into the database without issue. This process involved placing a finger on and off the sensor three times.

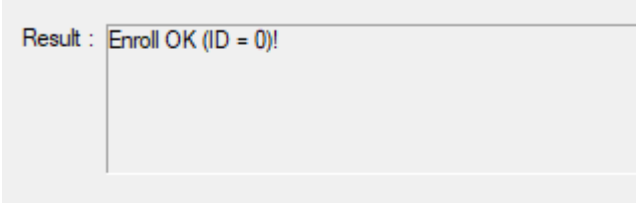


Figure 95 ID Enrolled OK.

Then, the fingerprint was verified against the database entries. It was a match, and the device took 422ms to process the image.

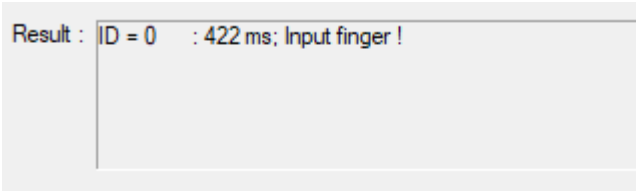


Figure 96 ID Identified within Database.

A fingerprint which was not in the database was tested against the database entries, and it was NG (no good).

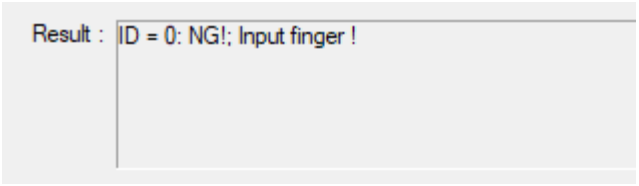


Figure 97 Finger not in database rejected.

8.7 Camera Testing

The cameras capture functions were tested using the open source library and the ArduCAM windows host program (Figure 98). (ArduCAM, 2019) The program allows us to interface with the camera through a GUI. The ArduCAM Mini 2MP Plus test code was used to connect to the camera, set configurations, then capture images. It displays if the SPI bus connections are good as seen in the image below.

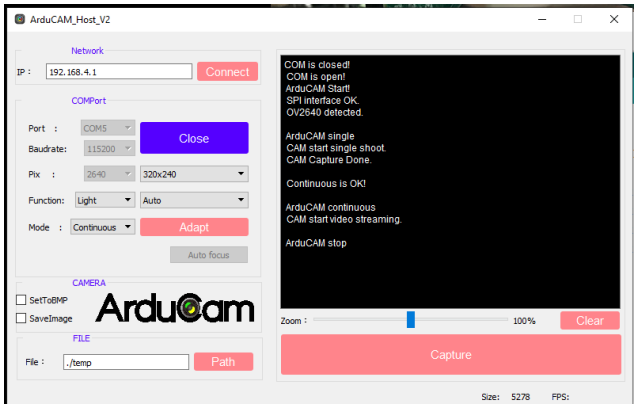


Figure 98 ArduCAM Windows Host Program. Permission to reproduce requested.

The images were sent to the MCU through SPI, then to the computer through serial. The wiring and test picture are shown in the images below.

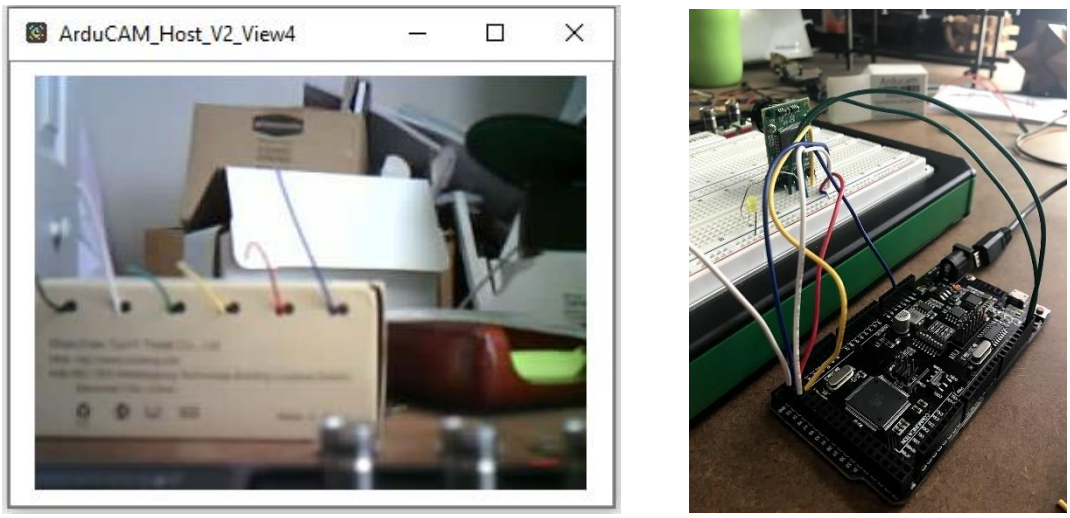


Figure 99 ArduCAM test picture and hardware wiring

The highest quality image (1600x1200) was able to be retrieved from the device with minimal delay. However, video streaming with this resolution had a large delay which would not work in a production environment.

8.8 RFID Testing

For the testing section for our RFID module we will test whether we can properly use the module. With the test we will need to connect our module to our Arduino and use code within the Arduino IDE to test functionality of our component. Within our test we are mainly looking for whether our module will detect the smart cards.

Our first image is of our Arduino connected with the RFID module. This was just a test to see if the component were to work correctly with the component.

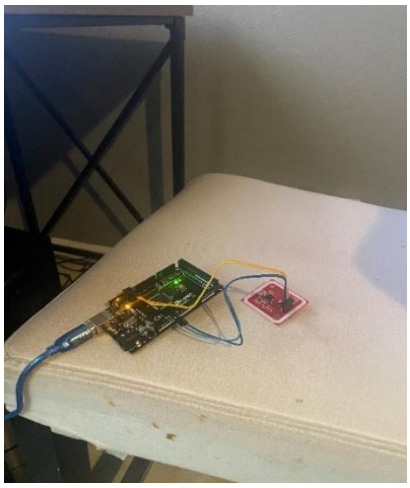


Figure 100 Arduino connected with the RFID module.

The second demonstration we have is when we hover our smart card over the module. The first state we have in our serial monitor below is when our smart card is not near the module, then our second state is when our smart card is hovering over the RFID module.

```
Hello Maker!
Found chip PN532
Firmware ver. 1.6
Waiting for an ISO14443A Card ...
Timed out! Waiting for a card...
Found A Card!
UID Length: 4 bytes
UID Value:  0x29 0x21 0xB0 0x23
Found A Card!
UID Length: 4 bytes
UID Value:  0x29 0x21 0xB0 0x23
Timed out! Waiting for a card...
```

Figure 101 Serial monitor of our RFID module working with our Smart card.

8.9 Reed Sensor Testing

With the reed sensor we tested whether it would detect whether the door was opened or closed by varying the distance between the magnets. To do so we connected the reed switch with the Arduino with the proper connections then input the code into the Arduino IDE to compile.

In the first picture it demonstrated what the reed switch looks like on the Arduino whenever it is in a closed state meaning there is only one magnet present.

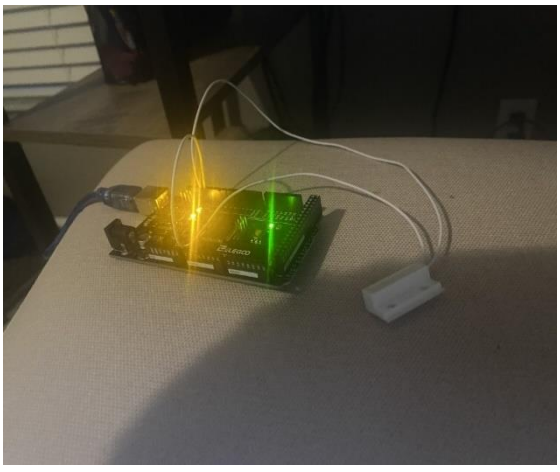


Figure 102 The Arduino with the reed switch in an open state.

For the second photo we demonstrate what the Serial monitor is printing. In our photo you can see the reed switch is within an open state. Meaning the door is open.

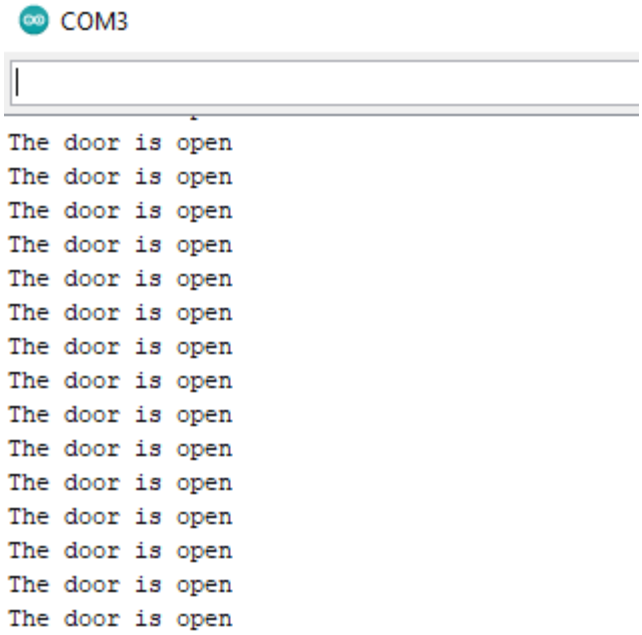


Figure 103 The serial port monitor showing the reed switch is in an open state.

For the third demonstration we tested when the reed switch magnets become connected meaning the door is closed.

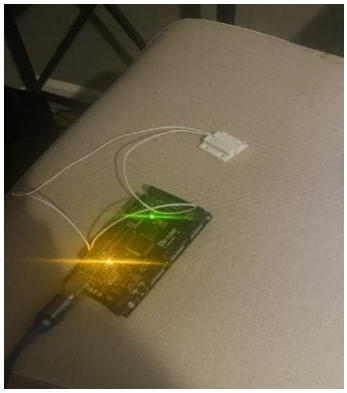


Figure 104 The Arduino with the reed switch in a closed state.

For the fourth photo we demonstrate what the Serial monitor is printing. In our photo you can see the reed switch is within an open state but then make the transition into the closed state the moment we connect the magnets together. Once this is done, we detect that the door is closed.

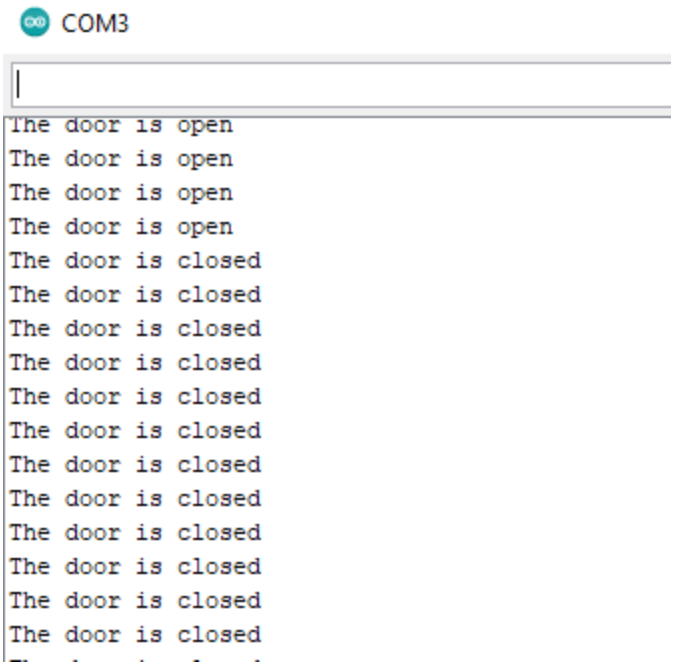


Figure 105 The serial port monitor showing the reed switch is in a closed state.

8.10 Mobile Software Testing

For our mobile testing section, we just wanted to demonstrate a simple app that is connected to the firebase database. Within it we create an android studio project to create the app. Within the app we just have a simple UI to display to show that android studio is up and running. We just created it to print out Hello World.

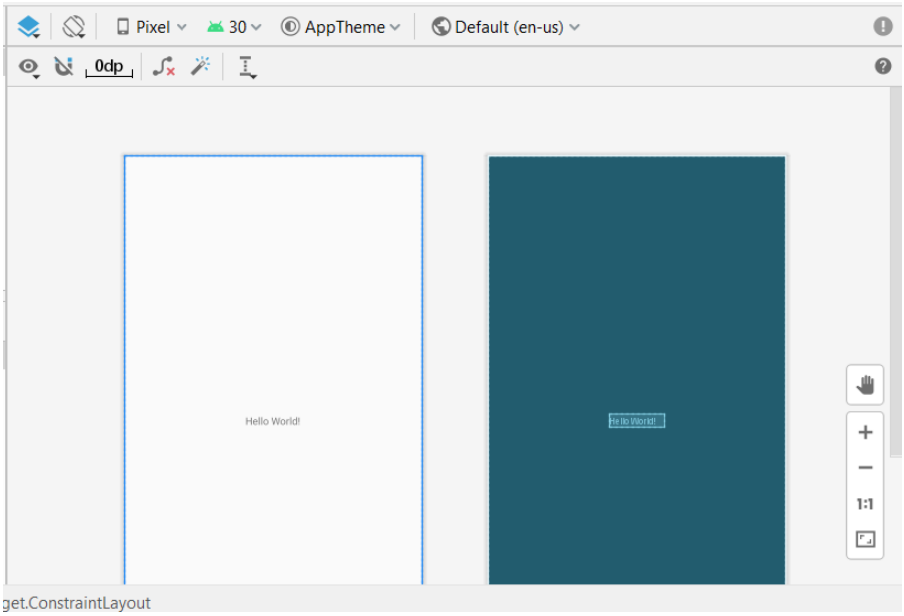


Figure 106 The UI display of the simple Android we created for testing

Another area of the software we wanted to test was an emulator. Since I will be doing most of the mobile application development, I will need android emulator since I'm primarily an apple user. Below is the emulator in action.

Figure 107 The android studio emulator of an android phone.

The last part of our testing was to a create a simple testing app and connect it to our firebase database, within our code we had it connect to the database and initiate the app. Below is the initiation of the database with our test app.

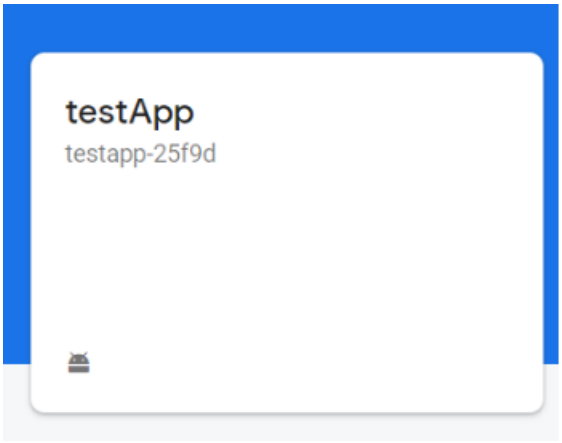


Figure 108 Example showing the database connected and was created.

9. Administrative Content

In this section, we will discuss our timeline plan as well as our finances. Our timeline was managed using a Gantt chart, and our budget was created as we sourced and purchased the prototype sensors / peripherals for our project.

9.1 Gantt Chart

The overall schedule for Senior Design I is shown below in figure 109. The reason we used a Gantt chart is so that the progress for the project was monitored throughout the semester. At any point in time we could take a quick glance and see what each group member was currently working on, and how far they have gotten. Our goal throughout the project was to write 0.68 pages per person daily before the 60-page draft, then 0.83 pages per person daily until the due date.

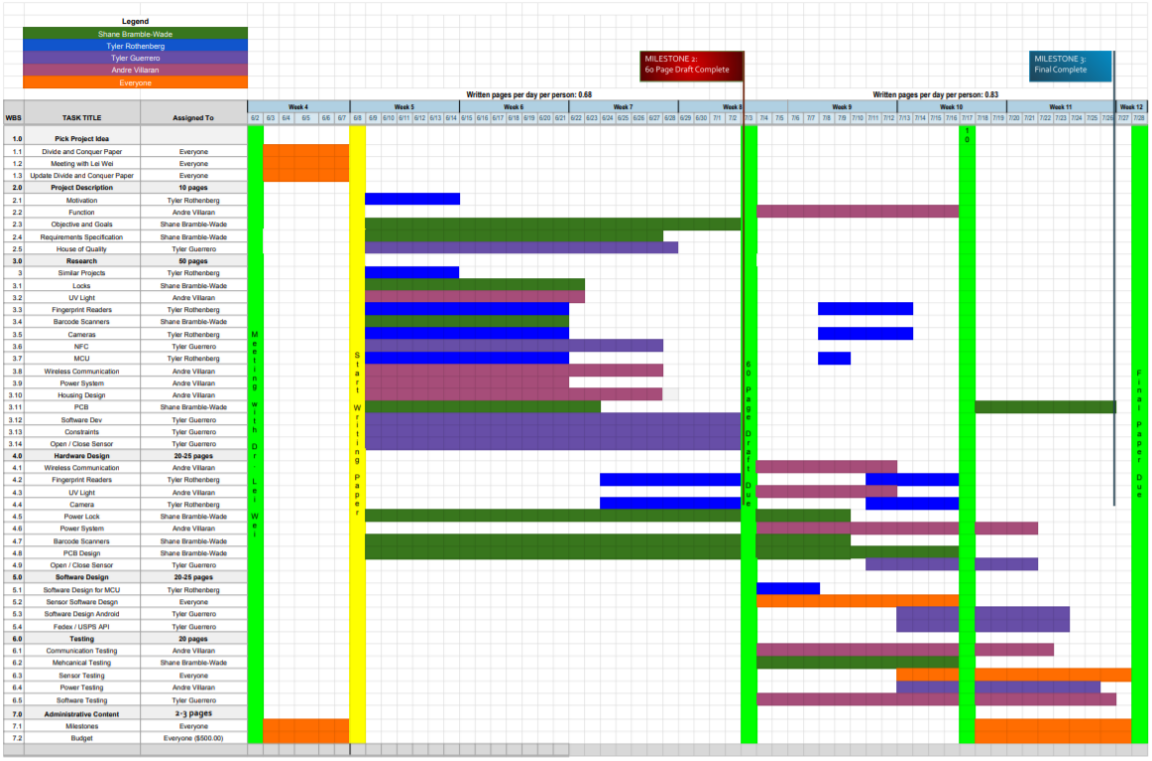


Figure 109 Senior Design I Gantt Chart

9.2 Budget

The project will be financed by the group members equally. The tables below show the actual retail costs for the parts we purchased. As a team, we have tried to split up the costs as evenly as possible. The project, in total, cost us as a group $1,505.46.

Tyler Rothenberg Expenditure

Item	Amount	Cost
ESP32	1	$10.99
Arduino Mega	1	$13.31
Waveshare Camera	1	$31.05
Sparkfun Fingerprint Sensor	1	$43.04
ArduCAM	1	$33.04
Quiic Cable Breadboard Jumper	2	$3.00
Sparkfun Logic Level Converter	1	$2.95
Breadboard Jumper Cables	2	$11.58
Solder Practice Kit	1	$9.99
Flux	1	$8.99
PCB Board Tester	1	$30.29
PCB Components	1	$9.16
SMD Practice Kit	1	$6.98
Solder Wick	1	$6.88
Bluetooth Serial	1	$7.39
USB to TTL Adapter	1	$11.99
ESP32 CAM	1	$10.99
Laser Infrared Thermometer	1	$23.11
Lowes Supplies	1	$11.39
Walmart Supplies	1	$7.62
	TOTAL:	**$293.74**

Shane Expenditure

Item	Amount	Cost
Sparkfun Barcode Scanner V1	1	$29.95
Sparkfun Barcode Scanner Module	3	$134.85
Lock-style Solenoid - 12VDC	1	$14.95
ELEGOO MEGA 2560 R3	1	$15.99
Mega +WiFi R3 Module ATmega2560	1	$15.74
Youngneer 5v Relay Board Relay Module 1	1	$11.99
PCB Manufacturing	1	$77.07
BNTECHGO 22 Gauge PVC 1007 Solid Electric Wire	1	$12.98
NTE Electronics SW02-10 No-Clean Solder Wick	1	$6.88
PCB Board Kit | Jumper Wires | Solder | Flux	1	$49.87
Digi-Key Mounting Components	1	$26.74
PCB Mounting Components	1	$101.00

0-2A 0-15V DC Power Supply	1	$31.99
Carpet Flooring	1	$12.59
5.1 Zener Diode (Through hole)	1	$2.25
RES SMD 4.7 OHM 1% 1W 0805	1	$6.50
Neiko 01902 Adjustable Helping Hand With Magnifying Glass	1	$8.15
PCB Power System Design	1	40.51
	TOTAL:	**$600.00**

Tyler G Expenditure

Item	Amount	Cost
ELEGOO MEGA 2560 R3	1	$15.99
REED Switch	2	$14.58
RFID Module	2	$19.38
Heatsink	1	$9.80
TIP42	1	$6.99
PCB Printing	2	$185.96
PCB Components	1	$108.56
TIP42	1	$4.99
ESP32	1	$10.29
Random Components for Andre	1	$22.00
	TOTAL:	**$398.54**

Andre Expenditure

Item	Amount	Cost
ESP32	1	$10.29
3.3V Voltage regulator	3	$5.99
TIP42 PNP power transistor	4	$3.01
DC Power Jack 2.1mm	5	$5.44
Tolako 5V relay module	1	$5.50
9V 1A power supply	1	$6.79
Breadboard power supply	5	$7.49
ATMega2560	1	$15.99
Digital Multimeter	1	$10.30
Logic Level Converter	10	$7.49
Home Depot	1	$94.58
Fans	1	$16.54
UVC Ozone lamp	1	$23.77
	TOTAL:	**$213.18**

10.0 References

03, G. (2019, August). *A Porch Package Protection System*. Retrieved from BLACK BOX: http://www.eecs.ucf.edu/seniordesign/sp2019su2019/g03/index.html

08, G. (2019, May). *Secure Outside Package Box*. Retrieved from SOPBOX: http://www.eecs.ucf.edu/seniordesign/fa2018sp2019/g08/

ArduCAM. (2019, March 23). *Quick Start Guide*. Retrieved from ArduCAM: https://www.arducam.com/docs/spi-cameras-for-arduino/software/quick-start-guide/

Arduino. (2010, October 20). *ATmega640/1280/1281/2560/2561 Datasheet*. Retrieved from Microchip: http://ww1.microchip.com/downloads/en/DeviceDoc/ATmega640-1280-1281-2560-2561-Datasheet-DS40002211A.pdf

Arduino. (2012, January 13). *Arduniogetstarted*. Retrieved from Arduniogetstarted: https://arduinogetstarted.com/tutorials/arduino-door-sensor

Banggood. (2016). *GM65 1D 2D Code Scanner BarCode Reader QR Code Reader Module*. Retrieved from Banggood: https://www.banggood.com/GM65-1D-2D-Code-Scanner-BarCode-Reader-QR-Code-Reader-Module-p-1440532.html?cur_warehouse=CN

Circuits, P. (2019). *What is a Printed Circuit Board (PCB)?* Retrieved from Printed Circuits: https://www.printedcircuits.com/what-is-a-pcb/

Cricuit Basics. (2016, January). *TURN ANY APPLIANCE INTO A SMART DEVICE WITH AN ARDUINO CONTROLLED POWER OUTLET*. Retrieved from Cricuit basics: https://www.circuitbasics.com/build-an-arduino-controlled-power-outlet/

Espressif Systems. (2020, January). *ESP32 Datasheet.* Retrieved from Expressif: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf

FedEx. (2020). Retrieved from FedEx: www.FedEx.com

Freeman, J. (2019, October 25). *What is JSON? A better format for data exchange*. Retrieved from Info World: https://www.infoworld.com/article/3222851/what-is-json-a-better-format-for-data-exchange.html

G2. (2015). *Best PCB Design Software*. Retrieved from g2: https://www.g2.com/categories/pcb-design

Hall-Effect Sensor - US1881 (Latching). (2015, May 14). Retrieved from Spark Fun start something new: https://www.sparkfun.com/products/9312

Health, N. I. (2020, March 24). *Study suggests new coronavirus may remain on surfaces for days*. Retrieved from National Institutes of Health: https://www.nih.gov/news-events/nih-research-matters/study-suggests-new-coronavirus-may-remain-surfaces-days

Innovative. (2018). *Morning RF Series Remote Control Deadbolt*. Retrieved from Innovative Home Systems: https://innovativehomesys.com/products/morning-rf-series-remote-control-deadbolt-w-o-insteon-interface

Kokkula, M. (n.d.). *How the 5v relay works*. Retrieved from IOT – BUILDING RELIABILITY: https://mounishkokkula.wordpress.com/how-the-5v-relay-works/comment-page-1/?unapproved=483&moderation-hash=3ba73668bdd2c8faf177344956372f6f#comment-483

Lofton, J. (2019, November 22). *Holiday 'porch pirates' on the rise, how to prevent package theft*. Retrieved from Michigan Live: https://www.mlive.com/news/2019/11/holiday-porch-pirates-on-the-rise-how-to-prevent-package-theft.html

Ranasoraus. (2015). *Smock (Your Very Own Smart Lock)*. Retrieved from instructables circuits: https://www.instructables.com/id/Smart-Lock-Microsoft-Azure/

Research, C. (2019, November). *2019 Package Theft Statistics Report*. Retrieved from C+R Research: https://www.crresearch.com/blog/2019-package-theft-statistics-report#

Schoeffler, M. (2018, January 5). *Arduino-Tutorial: How to use the RDM630/RDM6300 RFID reader*. Retrieved from https://www.mschoeffler.de/2018/01/05/arduino-tutorial-how-to-use-the-rdm630-rdm6300-rfid-reader/

Schoolov, K. (2020, January 11). *With package theft at an all-time high, Amazon and others are fighting back*. Retrieved from CNBC: https://www.cnbc.com/2020/01/10/package-theft-how-amazon-google-others-are-fighting-porch-pirates.html

Security. (2019). *Nearly 4 in 10 have packages stolen*. Retrieved from security.org: https://www.security.org/resources/stolen-packages-survey/

Shack, H. (2017, 6 23). *Smartphone Connected Home Door Lock*. Retrieved from Hackster: https://www.hackster.io/hackershack/smartphone-connected-home-door-lock-69944f

Sparkfun. (2017). *2D Barcode Scanner Module - DE2120*. Retrieved from sparkfun: https://www.sparkfun.com/products/16410

Sparkfun. (2019, Febuary 28). *Fingerprint Scanner TTL*. Retrieved from Github: https://github.com/sparkfun/Fingerprint_Scanner-TTL

Sparkfun. (2020). *Fingerprint Scanner - TTL (GT-521F32)*. Retrieved from Sparkfun Start Something: https://www.sparkfun.com/products/14518

Stone, W. (2020, July 22). *Scientists Want to Know More About Using UV Light to Fight COVID-19 Spread*. Retrieved from Kaiser Health News: https://khn.org/news/scientists-want-to-know-more-about-using-uv-light-to-fight-covid-19-spread/

System_id. (2018). *What is a Barcode Scaner and How Does it Work?* Retrieved from System_id: http://www.systemid.com/learn/barcode-scanners-and-how-they-work

T.K, .. H. (2015, August 27). *PN532 NFC RFID Module — A Quick Introduction*. Retrieved from ElectroSchematics Logo: https://www.electroschematics.com/nfc-rfid-module-pn532/

takigen. (2018). *SOLENOID LOCK*. Retrieved from TAKIGEN: https://www.takigen.com/products/list/14020#:~:text=SOLENOID%20LOCK-,SOLENOID%20LOCK,be%20used%20selectively%20for%20situations.&text=The%20power%2Don%20locking%20type,the%20solenoid%20is%20powered%20on.

Tech, C. . (2016, March). *uCella: The Smartest Package & Delivery Mailbox*. Retrieved from Indiegogo: https://www.indiegogo.com/projects/ucella-the-smartest-package-delivery-mailbox#/

Texas Instruments. (2014, June). *Mixed-Signal Microcontrollers datasheet*. Retrieved from Texas Instruments: https://www.ti.com/lit/ds/symlink/msp430fr6989.pdf?ts=1595601840501&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FMSP430FR6989

Texas Instruments. (2015, February). *Texas instruments corporation.* Retrieved from https://www.ti.com/lit/ds/symlink/cc3200.pdf?ts=1595896958667&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FCC3200

United Parcel Service. (2020). Retrieved from UPS: UPS.com

United States Postal Service. (2020). Retrieved from USPS: USPS.com

Waveshare. (2016, November 3). *OV7670 Camera Board*. Retrieved from Waveshare Wiki: https://www.waveshare.com/wiki/OV7670_Camera_Board

Waveshare. (2017). *Waveshare- share awesome hardware*. Retrieved from WAVESHARE: https://www.waveshare.com/barcode-scanner-module.htm

Waveshare. (2018, October 10). *OV5640 Camera Board* . Retrieved from Waveshare Wiki: https://www.waveshare.com/wiki/OV5640_Camera_Board_(C)

ZhianTec. (2008, September). *ZFM user manual*. Retrieved from Adafruit: https://cdn-shop.adafruit.com/datasheets/ZFM+user+manualV15.pdf

11.0 Permission Request Emails

Tyler Rothenberg
Fri 7/3/2020 12:45 PM
To: Nathanchong@knight.ucf.edu; Adam Cuellar; Jacky Li; Louis Rondino

Hey there former Group 3,

I hope this email finds you well. My name is Tyler Rothenberg I am a current senior CpE student at UCF. I was emailing to ask one of you for permission to use the following pictures of the Black Box for the research section of my senior design paper?

3.0.2 Black Box

Figure 2 Black Box housing

Respectfully,
Tyler Rothenberg

Tyler Rothenberg
Fri 7/24/2020 12:56 PM
To: ClientSuccess@microchip.com

Hello Microchip!

I hope this email finds you well. My name is Tyler Rothenberg and I am a senior computer engineering student at the University of Central Florida. I am emailing you to ask for permission to use a picture of the ATmega2560 chip you manufacture in my senior design paper.

Figure 10 ATmega2560

Respectfully,
Tyler Rothenberg

Reply | Forward

Tyler Rothenberg
Fri 7/3/2020 12:50 PM
To: hello@comingsoon-tech.com

Hello ComingSoon-Tech,

I hope this email finds you well. My name is Tyler Rothenberg and I am a senior computer engineering student at the University of Central Florida. I am emailing you to ask for permission to use pictures of uCella in my senior design paper.

Figure 3 uCella housing design

Respectfully,
Tyler Rothenberg

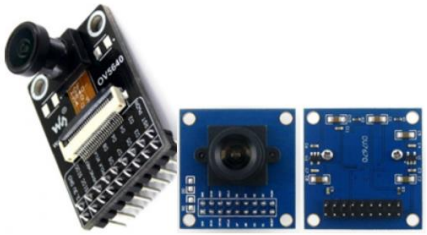
Reply | Forward

TR

Tyler Rothenberg
Fri 7/24/2020 12:59 PM
To: service@waveshare.com

Hello Waveshare!

I hope this email finds you well. My name is Tyler Rothenberg and I am a senior computer engineering student at the University of Central Florida. I am emailing you to ask for permission to use the pictures of the OV5640 and the OV7670 boards as seen below for my senior design research paper.



Respectfully,
Tyler Rothenberg

Reply | Forward

Permission Request - GM 65 Barcode Scanner

SB

Shane Bramble-Wade
Fri 24/07/2020 16:33
To: cservice@banggood.com

PERMISSION REQUEST LETTER

[07/24/2020]

Dear Banggood,

Gooday Sir/Madam, how are you doing? I am a senior student at the University of Central Florida who is currently conducting my Senior Design project. I am writing to request permission to use the below picture for the purpose of our Senior Design project. I will provide a complete citation and display your acknowledgement of the permission to use such material.

This request is for a royalty free, irrevocable, worldwide permission for usage of the material as stated above. I would also very much like an electronic version of this material.

Figure 1 GM65 Barcode Scanner

If you require more information regarding this request, feel free to call at (289)-236- 6914 or email at sbramwade@msn.com .

Regards,
Shane Bramble-Wade

Permission Request - Lock Solenoid

SB Shane Bramble-Wade
Fri 24/07/2020 16:58
To: support@adafruit.com

PERMISSION REQUEST LETTER

[07/24/2020]

Dear Adafruit,

Gooday Sir/Madam, how are you doing? I am a senior student at the University of Central Florida who is currently conducting my Senior Design project. I am writing to request permission to use the below picture for the purpose of our Senior Design project. I will provide a complete citation and display your acknowledgement of the permission to use such material.

This request is for a royalty free, irrevocable, worldwide permission for usage of the material as stated above. I would also very much like an electronic version of this material.

Figure 1 Lock-style Solenoid - 12VDC

If you require more information regarding this request, feel free to call at (289)-236- 6914 or email at sbramwade@msn.com .

Reply all | Delete Junk Block ···

Permission Letter - DE 2120 Barcode Scanner

SB Shane Bramble-Wade
Fri 24/07/2020 16:44
To: support@sparkfun.com

PERMISSION REQUEST LETTER

[07/24/2020]

Dear Sparkfun,

Gooday Sir/Madam, how are you doing? I am a senior student at the University of Central Florida who is currently conducting my Senior Design project. I am writing to request permission to use the below picture for the purpose of our Senior Design project. I will provide a complete citation and display your acknowledgement of the permission to use such material.

This request is for a royalty free, irrevocable, worldwide permission for usage of the material as stated above. I would also very much like an electronic version of this material.

Figure 1 DyScan DE2120 Barcode Scanner

If you require more information regarding this request, feel free to call at (289)-236- 6914 or email at sbramwade@msn.com .

Regards

Permission Request - SMOCK

SB Shane Bramble-Wade
Sat 25/07/2020 08:00
To: support@instructables.com

PERMISSION REQUEST LETTER

[07/24/2020]

Dear Ranasoraus,

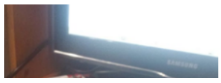
Gooday Sir/Madam, how are you doing? I am a senior student at the University of Central Florida who is currently conducting my Senior Design project. I am writing to request permission to use the below picture for the purpose of our Senior Design project. I will provide a complete citation and display your acknowledgement of the permission to use such material.

This request is for a royalty free, irrevocable, worldwide permission for usage of the material as stated above. I would also very much like an electronic version of this material.
Source Link: https://www.instructables.com/id/Smart-Lock-Microsoft-Azure/

Smock (Your Very Own Smart Lock) : 3 Steps - Instructables

Smock (Your Very Own Smart Lock) : How Cool would it be To have a lock that could be controlled anywhere around the world? For example you were away on a business trip and you had to let someone into your house, and NOW you will be able to do just that with a tap of your phone ^.^....

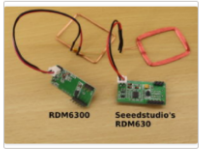
www.instructables.com



To M mail@mschoeffler.de Bcc

Cc

Use of Picture



Hello Micheal,
My name is Tyler Guerrero, I am current Computer Engineering Student here at the University of Central Florida, For my senior design paper we are required to ask permission for any picture we use that is not ours is it okay if I use the RDM630/RDM6300 RFID reader picture ?

To S sales@digikey.com Bcc

Cc

Use of picture

Hello Digikey,
My name is Tyler Guerrero, I am current Computer Engineering Student here at the University of Central Florida, For my senior design paper we are required to ask permission for any picture we use that is not ours is it okay if I use the Reed Switch picture ?

To S support@sparkfun.com. Bcc

Cc

Use of Picture

Hello sparkfun,

My name is Tyler Guerrero, I am current Computer Engineering Student here at the University of Central Florida, For my senior design paper we are required to ask permission for any picture we use that is not ours is it okay if I use the Hall - effect- sensor picture ?

M mail@mschoeffler.de
Mon 7/27/2020 7:13 AM
To: Tyler Guerrero

Hi Tyler,

Of course, you can use it for your paper.

Best wishes for your work!

Michael

...

Reply | Forward

S

noreply@salesforce.com on behalf of
Support <support@sparkfun.com>
Mon 7/27/2020 2:29 PM
To: Tyler Guerrero

Hello!

Thank you for reaching out to SparkFun! As an open source company we allow usage of anything we post online under the creative commons statute and only request that you provide us with the credit for the original work.

Please let us know if there is anything else that we can help you with.

Have a great day!
Aubrey
Customer Service Representative
SparkFun Electronics
www.sparkfun.com <http://www.sparkfun.com>

Mail - Andre Villaran - Outlook - Google Chrome

outlook.office.com/mail/deeplink?version=2020072004.04&popoutv2=1

Reply all | Delete | Junk | Block | ...

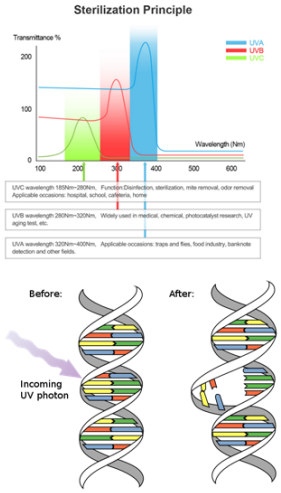
Permission to use image

AV

Andre Villaran
Mon 7/27/2020 8:19 PM
To: info@wikimedia.org

To whom it may concern,

I am requesting permision to use some of the pictures in your website for my senior design report that i am working for in the university of central florida. as part of the requirements of the project we have to ask permision to use any pictures that we attach to the report. Attached are the images that i intend on using on the report:



Regards,
Andre V.

pvillaran97
a few seconds ago

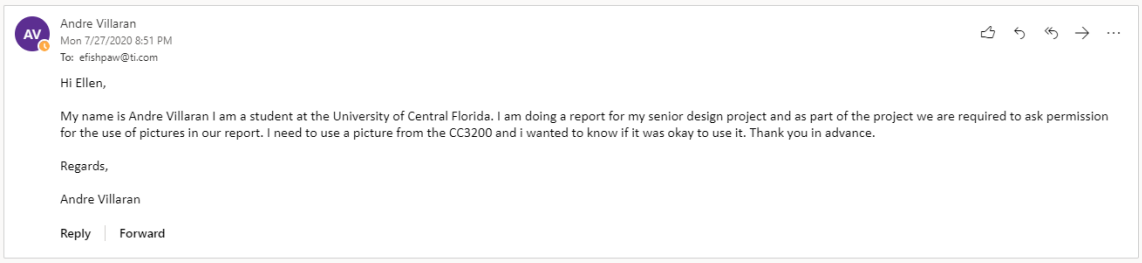
Hi Jefferson,

I wanted to ask permission to use some of the pictures here for my senior project. A part of the requirements is that i must ask permission for every picture i use in the project and some of the pictures here are perfect for it. Thank you in advance.

Regards,

Andre Villaran

Thank • Reply



Andre Villaran
Mon 7/27/2020 8:51 PM
To: efishpaw@ti.com

Hi Ellen,

My name is Andre Villaran I am a student at the University of Central Florida. I am doing a report for my senior design project and as part of the project we are required to ask permission for the use of pictures in our report. I need to use a picture from the CC3200 and i wanted to know if it was okay to use it. Thank you in advance.

Regards,

Andre Villaran

Reply | Forward

Andre Villaran
JULY 28, 2020 AT 1:01 AM

Your comment is awaiting moderation. This is a preview, your comment will be visible after it has been approved.

To whom it may concern,
I wanted to ask permision to use one of the pictures in this article for my senior design project. One of the requierements of this project is to ask permission for the use of any pictures in our report. Is it okay if I include one of the pictures here for my project?

Regards,
Andre Villaran

★ Like