

Black Box

A Porch Package Protection System



Department of Electrical Engineering and Computer Science
University of Central Florida
Dr. Lei Wei & Dr. Samuel Richie

Group 3

Nathan Chong	Nathanchong@knights.ucf.edu	EE
Adam Cuellar	Cuellar.adam@knights.ucf.edu	CpE
Jacky Li	Jackyli@knights.ucf.edu	CpE
Louis Rondino	Lrondino@knights.ucf.edu	EE

Table of Contents

1.0 - Executive Summary	1
2.0 - Project Description	2
2.1 - Project Motivation and Goals	2
2.2 - Objectives	3
2.3 - Requirement Specifications	3
2.3.1 - Gathering Requirements	3
2.3.2 - Hardware Requirements	5
2.3.3 - Software Requirements	6
2.3.4 - Cost Requirements	7
2.3.5 - House of Quality Analysis	9
2.3.6 - Entity Relationship Diagram.....	10
3.0 - Design Constraints and Standards.....	11
3.1 - Constraints Encountered.....	11
3.1.1 - Economic and Time constraints.....	11
3.1.2 - Environmental, Political, and Social	12
3.1.3 - Ethical, Safety, and Health.....	12
3.1.4 - Sustainability and Manufacturability.....	12
3.2 - Standards and Safety Concerns	13
3.2.1 - Soldering Standards	13
3.2.2 - Lead Solder Safety	15
3.2.3 - RoHS Compliance	17
3.2.4 - C/C++ Programming Standards.....	18
3.2.5 - USB Standards	20
3.2.6 - Barcode Standards	21
4.0 - Research Related to Project Definition.....	23
4.1 - Existing Similar Projects and Products.....	23
4.2 - Relevant Techniques	24
4.3 - Relevant Technology	25
4.3.1 - Serial Communication.....	25

4.3.2 - Wireless Communication	31
4.4.3 - Wireless Provisioning.....	38
5.0 - Project Hardware & Software Design Details	39
5.1 - Initial Design Architecture	40
5.1.1 - Microcontroller Considerations	40
5.1.2 - Wi-Fi Module Considerations.....	50
5.1.4 - Temperature Sensors	57
5.1.5 - Power Supply.....	60
5.1.6 - Battery	61
5.1.7 - Battery Measurement.....	65
5.1.8 - Barcode Scanner	65
5.2 - Software Design.....	73
5.2.1 - Application Software Design	74
5.2.2 - Embedded Software Design	78
5.3 - Parts Selection Summary.....	79
5.4 - Summary of Design.....	83
6.0 - Project Prototype Construction and Coding.....	84
6.1 - Overall Schematic	84
6.2 - Integrated Schematics	87
6.2.1 - Wi-Fi Module Connection.....	87
6.2.2 - ATmega2560 Programming via ICSP	88
6.2.3 - Temperature Sensor	89
6.2.4 - Power Supply.....	89
6.2.5 - Reset and Oscillator.....	91
6.2.6 - User and System Feedback.....	92
6.2.7 - Locking Actuation.....	93
6.3 - PCB Design	94
6.4 - PCB Vendor and Assembly.....	97
6.5 - Final Coding Plan.....	98
6.5.1 - Boot Loader	98
6.5.2 - The Boot Loading Process.....	99

7.0 - Project Prototype Testing Plan	100
7.1 - Hardware Testing.....	100
7.1.1 - Locking Subsystem.....	100
7.1.2 - Feedback Indicators Subsystem.....	103
7.1.3 - Wi-Fi Subsystem.....	103
7.2 - Software Testing	104
7.2.1 - Embedded Software Testing.....	106
7.2.2 - iOS Application Testing.....	112
8.0 - Administrative Content	113
8.1 - Milestone Discussion	113
8.2 - Project Management.....	117
8.3 - Budget and Finance Discussion.....	118
8.4 - Stretch Goals	122
9.0 - Project Summary and Conclusion	124
Appendix A Reference.....	A-1
Appendix B Permission To Reproduce	B-1

List of Tables

Table 1 - Design Constraints	4
Table 2 - Hardware Specifications.....	5
Table 3 - Hardware Enclosure Specifications	6
Table 4 - Software Specifications	6
Table 5 - Cost Specifications	7
Table 6 - Full Requirements Table	8
Table 7 - Technical Requirements.....	9
Table 8 - Soldering Temperature Chart [Vis19]	16
Table 9 - USB Advancement	20
Table 10 - Synchronous vs. Asynchronous Comparison	26
Table 11 - SPI & I2C.....	30
Table 12 - Wireless Communication Analysis.....	31
Table 13 - Band Frequency Summary.....	36
Table 14 - Microcontroller Consideration Table	40
Table 15 - CC3220x Series Basic Specifications.....	41
Table 16 - MSP430x Series Summary	44
Table 17 - MSP430 Specifications.....	45
Table 18 - ATmega2560 Specifications	46
Table 19 - Microprocessor Comparison Chart.....	49
Table 20 - Power Consumption Summary - ESP8266EX.....	52
Table 21 - Power Consumption Summary - CC3200MOD	54
Table 22 - Power Consumption Summary - WGM110.....	56
Table 23 - Final Wi-Fi Module Comparison	56
Table 24 - Temperature Sensor Considerations.....	57
Table 25 - STS3X-DIS Comparison.....	58
Table 26 - TMP451-Q1 Specifications.....	59
Table 27 - DHT11 Specifications.....	59
Table 28 - Final Temp. Sensor Comparison	60
Table 29 - Subsystem Specific Power Requirements.....	61
Table 30 - Barcode Scanner Module Specifications	68
Table 31 - Barcode Scanner Module Specifications	70
Table 32 - Barcode Scanner Module Specifications	71
Table 33 - Final Barcode Scanner Module Comparison	71
Table 34 - Fingerprint Scanner Comparison.....	72
Table 35 - Common Formats of Tracking Numbers [Trac]	79
Table 36 - Major Components List.....	80
Table 37 - Bill of Materials for Schematic	96
Table 38 - OSH Park Standard Services	97
Table 39 - Pin Summary	99

Table 40 - Voltage/Current Load Test Summary	101
Table 41 – Software Testing Time Line	105
Table 42 - Fingerprint Scanner Testing Results	111
Table 43 - Senior Design 1 Milestone Table.....	114
Table 44 - Senior Design 2 Milestone Table.....	115
Table 45 - Distribution of Labor	116
Table 46 - Black Box Budget	118
Table 47 – Budget Allocation.....	120

Table of Figures

Figure 1 - Entity Relationship Diagram	10
Figure 2 - Comfort Zone [THSM19]	13
Figure 3 - Stress Relief Examples	14
Figure 4 - USB Types	20
Figure 5 - Barcode Symbology	21
Figure 6 - UPC Example along with UPC- E	22
Figure 7 - USPS Representation of POSTNET Codes	22
Figure 8 - Amazon Locker Technology	23
Figure 9 - Land Port Technology Package Protection	24
Figure 10 - Typical UART interface	27
Figure 11 - Independent Slave Select [Ver19]	28
Figure 12 - Daisy-Chained Slave Select [EEHe]	29
Figure 13 - Master and Slave Scheme [EL19]	32
Figure 14 - OSI model [ASM]	34
Figure 15 - Frequency Bands [Stev]	35
Figure 16 - RFID Pictorial Representation [How19]	37
Figure 17- Texas Instrument CC3220 Development Board	42
Figure 18 - Functional Block Diagram [Mous]	43
Figure 19 - MSP430FR6989	44
Figure 20 - ATmega2560 Development Board	46
Figure 21 - ATmega1284 Development board	48
Figure 22 - ESP9266EX Wi-Fi Module	50
Figure 23 - CC3200MOD Wi-Fi Module	53
Figure 24 - Wizard Gecko WGM110 Wi-Fi Module	55
Figure 25 - Discharge Characteristics of Brand Alkaline AA Batteries [Powe]	63
Figure 26 - How a Barcode Scanner Works	66
Figure 27 - 1D (left) and 2D (right) Barcodes	67
Figure 28 - Waveshare Barcode Scanner Module	67
Figure 29 - QR Code Settings	68
Figure 30 - MCR12 CCD Barcode Scanner Module	69
Figure 31 - RB-Dfr-567 Barcode Scanner Module	70
Figure 32 - Software Logic Flowchart	73
Figure 33 - Collection of Major Components	80
Figure 34 - ATmega2560	82
Figure 35 - Outdoor Box for Black Box	83
Figure 36 - Black Box Schematic Design	86
Figure 37 - Wi-Fi Module Header	87
Figure 38 - Bootloader Schematic	88
Figure 39 - Temp Sensor	89

Figure 40 - Voltage Regulation	90
Figure 41 - VCC & Ground Schematic.....	90
Figure 42 - Reset and Oscillation	91
Figure 43 - User and System Feedback Schematics.....	92
Figure 44 - Locking Mechanism Actuation.....	93
Figure 45 - PCB Traces	94
Figure 46 - PCB Render	95
Figure 47 - Boot loading Visual.....	99
Figure 48 - Reverse Current Lock Step Response	102
Figure 49 - Lock Response to Microcontroller Impulse.....	103
Figure 50 - Tera Term Serial Port Setup	106
Figure 51 - Valid Barcode Testing	107
Figure 52 - COM5 Serial Port Response	107
Figure 53 - Locking/Unlocking via Software	108
Figure 54 - Local Server	109
Figure 55 - Wi-Fi Module Test Terminal	109
Figure 56 - Fingerprint Scanner Proof of Concept.....	110
Figure 57 - Gantt Chart [Gantt].....	121

1.0 - Executive Summary

About 26 million Americans have reported having their packages stolen” from their front door, or porch, according to a study reported by CBS News [Gib19]. Especially during the holidays, stealing packages from others has become common in today’s society.

According to security experts and postal officials, these are the methods recommended for package theft prevention:

- Have packages delivered to your workplace
- Have the package(s) delivered to your friends who will be present during that time
- Require a signature for packages you’re sending to yourself and family and friends
- Ask for the safest way from the receiver to have their packages delivered
- Choose alternative pickup and delivery options by visiting shippers’ websites such as UPS, U.S Postal Service, and FedEx.
- Home security equipment, such as surveillance cameras

There are many options that can be exploited; however, these require more effort from the person receiving the mail. It is more convenient if this person could have their package delivered safely without having to worry about it being stolen. Thus, the idea of the porch package protection system, uniquely named Black Box, was created to ensure the safety of deliverables.

It is crucial to keep a user’s package safe until the user can retrieve it. Only the mailman and the user can have access to opening it via barcode scanner or keypad entry. The user can access the box by using an app, the barcode, or a manual entry option will be provided such as a finger print option.

The Black Box offers a great alternative to having packages delivered safely due to its convenience to the user unlike the other methods described by security experts. In this day and age many of us do not shop in traditional brick and mortar stores, many of us will be online and ordering it from your online store. But as packages are being stolen every day. With the invention of this Parcel protection box to protect valuables to be stolen can impact the daily lives of many years to come, because there is no slow down anytime soon with online orders. Our team thinks this product we have will change the world one day and eventually discourage any potential Porch pirates

2.0 - Project Description

The Black Box can be the most significant enhancement to receiving personal mail in regard to security. Instead of baring the inconvenience of lack of security, individuals can now feel confident about receiving valuable mail. The description of this project is outlined below and includes:

- The motivations and goals of the Black Box.
- The objective of the Black Box's unique design.
- A comprehensive list of requirement specifications involved with the production of the Black Box.

2.1 - Project Motivation and Goals

Ever since the U.S. Postal system was established in 1775, mail deliveries continue to be made to this day. With advancements in technology, it is easier to purchase a package from online stores such as Amazon and have it delivered to the customer in a short period of time which is typically two to four days. However, there are occasions where he/she orders an item that is too big to fit into a typical mailbox or requires customer verification; thus, it is delivered to the customer's front door given that the mailman fulfills his/her duties. If any of the homeowners are not present to answer the mailman, consequently, the package is left at the front door. This is problematic since it leaves the item vulnerable to theft which causes issues for both the customer and the retailer.

Mail is unpredictable and can be delivered at any time. In some cases, mail can be delayed by unexpected environmental factors and it is unreasonable to attempt to make yourself available for such an erratic delivery time. One potential solution includes front door cameras; however, front door cameras and phone systems only provide information, not protection. If a theft occurs, even if it is caught on camera, there is little the police usually do because they are working on more important crimes.

To provide a solution to this, the Black Box project idea is formed to provide a countermeasure to theft. The Black Box is a porch package protection system that will allow postal carriers to place deliveries inside of it whenever the customer is not present. Once the package is placed, it will keep the package secured using magnetic locks to prevent itself from being opened. It will then notify the user that the package has been delivered and is safely secured waiting for it to be opened once the user is available.

The motive for this project is to demonstrate the knowledge and technical skills we've gained from attending the CpE/EE curriculum at the University of Central Florida. Additionally, our motivation arises from past experiences we've had from ordering an item online and having it delivered to us at inconvenient times. Leaving a package out in the open with the possibility of it being taken away is a

disadvantage to all customers and producers; thus, the idea of keeping that package secured is what inspired us to come up with this project idea.

The project design can be described as a three-dimensional metal container that is big enough to hold a few packages depending on the size of the package. A barcode scanner with numbers based on the package barcode will be used to unlock the protection system by the postal carrier. Mechanical locks will be used to keep the container locked and secured and are more ideal when compared to implementing an electromagnetic lock. Given that there may be heavy packages, the container should be able to carry a bulky weight. In addition to protection against theft, it should be able to withstand weather conditions such as thunderstorms, hurricanes, etc. The box will be tightly sealed, preventing any rain from entering it that could not only damage packages but the electronic components in it as well. Since the box will be placed in front of porches, it must be drilled into the ground securely to prevent the box itself from being stolen.

2.2 - Objectives

Our objective for this Senior design project is to create a parcel locker for residential use. Our motivation arose from watching endless porch pirates on the news stealing millions of dollars' worth of items each year. The typical victim of package theft loses about \$200 in value each time. In light of these statistics, our team believes it's our responsibility to change the world by eliminating the thefts. By knowing the average loss of per theft we can shoot for a goal of \$200 per locker to make it more attractive to buyers when brought to market. The Black Box will not only benefit the owner but also many merchants and postal services. Many merchants have to replace the item or provide a refund due to insurance and customer satisfaction. From the commercial standpoint we can say there's a lot more pros than cons, due to the fact that project managers from apartments can manage the delivery making the property more attractive to live in. In 2019, E-commerce is more popular than ever, people have started purchasing their products online more often due to the convenience.

2.3 - Requirement Specifications

Below are the requirements involved with the production of the Black Box. These requirements were discussed thoroughly and in great detail within the group to build the most efficient and secure Black Box possible.

2.3.1 - Gathering Requirements

In order to complete a project both efficiently and successfully, it is important to gather and establish the proper requirements. A requirement can determine the desired behavior of a product and focus on the needs of that product rather than the solution or implementation. Requirements define the expected behavior of a product without explicitly saying how that behavior is realized. A lack of established requirements is the major factor to a failing project. Without requirements, there is less user involvement, unrealistic expectations, lack of executive support,

changing specifications, lack of planning, etc. The main goal to establishing requirements is to enforce and establish the boundaries for a project. This allows for a greater understanding by the team and allows for quality progression throughout the project. Requirements should also be well documented; therefore, making referring back to the goal of a project a quick and easy task. Some requirements we've established for the Black Box include process constraints, design constraints, quality/nonfunctional requirements, functional requirements, and testing requirements. The following includes the technical requirements that we've determined are crucial in successfully completing this project.

Table 1 - Design Constraints

Classification	Description	Value/Units
D1	Device with Wireless Protocol - Unlock the box from a device using a wireless protocol.	
D2	Microcontroller - Used to implement locking/unlocking mechanism	
D3	Stored Energy - Box should operate continually for 1000 locking/unlocking cycles	Lock Cycles
D4	Cost - Affordable	Dollars
D5	Temperature Resistant - withstands extreme temperatures	Celsius
D6	Water Resistant - withstands light rain	Inches
D7	Electromagnet	Volts
D8	Wi-Fi module range	Feet
D9	Thermal sensor	Degrees
D10	Barcode Scanner	Values
D11	Durable package receptacle	Length x Width x Height
D12	Plugs into House power	
D13	Back-up battery when there is an outage	Runtime

Table 1 depicts the specifications necessary to implement the idea of the Black Box. We've chosen these features carefully to ensure the most efficient and useful product for protecting packages.

2.3.2 - Hardware Requirements

The requirements with hardware for this device are listed in Table 2 below. The given requirements are due to the necessary components for the proper implementation of the Black Box's features.

Table 2 - Hardware Specifications

Classification	Description
H1	The system will contain a box, a locking mechanism, RFID access, thermal sensors, a barcode scanner, and a camera
H2	The system will contain RFID capabilities to read input from the user
H3	The system will contain a barcode scanner for granting/denying entry
H4	The system shall be able to accept/deny access to unlock/lock the device respectively
H5	The power supply will be capable of supplying power to the PCB, sensors, and other accessories
H6	The power supply will be capable of supplying power to the PCB, sensors, and other accessories
H7	The camera shall take a photo every time the box is unlocked
H8	The camera shall take a photo every time the system is tampered with
H9	Temperature sensor is used to give live data to the user if the package is in an extreme hot or cold environment for temperature sensitive parcels
H10	Smooth integration with software
H11	The system will contain RFID capabilities to read input from the user
H13	The system will contain a barcode scanner for granting/denying entry

Table 3 - Hardware Enclosure Specifications

Classification	Description
H14	The enclosure will be slightly above the average package size sent by mainstream companies.
H15	The enclosure will be waterproof.
H16	The enclosure should have holes for securing to a single location.
H17	The enclosure will be made of a durable material.
H18	The enclosure will be easy to install and will not weigh more than 30 pounds.

Table 3 above specifies the ideal environment for the packages' enclosure. The information details how the Black Box will implement the safety of the package.

2.3.3 - Software Requirements

The specifications for software intended for the Black Block is detailed below in Table 4. These requirements will allow for easier interaction from the user's side as well as the proper integration of software and hardware.

Table 4 - Software Specifications

Classification	Description
S1	Smooth integration with hardware.
S2	Hashing/Unhashing sensitive information.
S3	Communicate with device to share information such as unlocking/locking timestamps, photos, pin, tracking number, etc.
S4	Communicate with device to set up device specifications
S5	Easy to use, intuitive, graphical user interface or GUI
S6	Smooth integration with hardware.

With this table we had to consider a smooth integration and testing with the hardware to resolve any of the failing tests we may induce in the future of building the software for this project. Hashing and un-hashing will require the current techniques that is used in today's security protocols.

2.3.4 - Cost Requirements

This section will describe the cost requirements for the design. Concepts of cost effectiveness and the quality of design come forth into the process of choosing proper components. Additionally, emergency back-up PCBs are a necessity because it allows the users to experiment on some while finalizing the design in another. It is important to keep the cost of making this product because if this product was to be sold to customers, it would be ideal if it can be sold at a competitive price.

Table 5 - Cost Specifications

Classification	Description
C1	The development cost of the project will cost no more than \$900
C2	Three PCB designs in the case of damaged/ineffective boards.
C3	All team members must commit to the purchase and use of cheap but effective parts.

Table 5 includes the requirements for each team member to take into consideration when researching or developing features for the Black Box. It is imperative to keep the production of the box within a reasonable cost.

Since this project is being designed by a group of four, it is only fair to divide the total cost evenly amongst the group. Each member has promised to commit to sharing the expenses.

Cost specifications are subject to change; however, will be adhered to as closely as possible. The development cost may exceed up to \$900 due to purchasing components in bundles. This will prevent future problems in developing the PCB design, as back-up components will always be available for use.

Cost will not be considered a limiting factor in production of the Black Box; however, the cost is heavily considered in the selection of parts; therefore, parts selected must be effective and cost no more than the average market value for similar technologies with functions applicable to our device. Table 6 below describes the functionality requirements of the Black Box that the team has agreed upon. This information can be used for reference when considering whether a part needs to be purchased or if a cheaper but equally efficient part can be bought.

Parts must also follow any restrictions provided to us by the ECE faculty at the University of Central Florida.

Table 6 - Full Requirements Table

Function Block	Description
Locking Mechanism/ Deadbolt	The locking mechanism will protect the owner's valuables from being stolen, the lock will be getting inputs from the DCU which authorize the locking/unlocking of the box.
Sensor unit / Environmental Temperature Sensor	Environmental Temperature Sensor is used for letting the DCU have a live data of the ambient temperature of the box.
Barcode Scanner	The Barcode scanner will scan a package barcode to send the decoded tracking number to the DCU for processing.
Device Control Unit	The Device Control Unit (DCU) will be the main operator of the product. The DCU will monitor any inputs from the Environmental Temperature Sensor, control the LED's, verify the required pin/barcode, and process/interpret data sent or received from the mobile application.
Keypad	Keypad will send a password combination in numbers to the DCU.
Indicators (L.E.D)	LED (Light emitting diode) will indicate the status of the black box. Green will indicate unlocked. Red will indicate locked.
Mobile App	Will control lock/unlock state and allow for setting up a passcode for keypad.
Wi-Fi Module	An addition to the DCU, sending/receiving information from/to the mobile application and the device.

Table 6 includes the overall scope of the Black Box's hardware components that the team is going to adhere to. As mentioned previously, this table will be used as a marker for final decisions in parts selection, hardware design, and software design. It is imperative that the team adhere to the guidelines above to ensure that all systems are integrated smoothly after individual development and the final iteration of the Black Box functions as initially intended unless any changes only improve the products function.

2.3.5 - House of Quality Analysis

The House of Quality diagram, represented in Table 7, depicts the correlation between engineering and marketing requirements for this project. The diagram design allows the team to determine the most efficient way to fulfill a certain requirement and to consider the tradeoffs in the project in attempt to create the best design possible. Taking the considerations of the design versed the requirements we have to make, we have to make a design that would fit into our design requirements. This house of quality provides us a consensus of the requirements the team believes we will need to be successful throughout the development of the Black Box.

Table 7 - Technical Requirements

	Speed +	Performance +	Ease of Use +	Energy Cost -	Sensor Accuracy	Wireless Range	Target Goal
Cost -	↑	↑	↑	↑	↑	Less than 40\$	≤ \$850
Size -	↓	↑	↑	↓	N/A	< 1 inch	< 23x11x13
Quality +	↑	↑	↑	↑	↑	↑	≥ 80%
Weight -	↓	↑	↓	↓	↑	+	≤150 lbs
Efficiency +	↑	↓	↓	↑	↑	More than 3 weeks	≥ 70%
Container Foot print	↑	↑	↑	↑	N/A	N/A	<32X22
Easiness to install	↑	↑	↑	N/A	N/A	N/A	<30 minutes

Key:

- ↑ = Positive Correlation ⤴ = Strong Positive Correlation
- ↓ = Negative Correlation ⤵ = Strong Negative Correlation
- + = Increase Requirement - = Decrease Requirement

When taking the house of quality into consideration of Table 7, we had to consider the design aspect of the project and also the requirements that would make this project considered a user-friendly device. We had to brainstorm and also optimize multiple designs to consider tradeoffs with doing one idea versus the other. With engineering as mentioned in many quotes “There is nothing that’s free” a perfect meaning for the house of quality, the quote means that if we take account and try to improve a section of anything, we do there will be costs on the opposite side.

2.3.6 - Entity Relationship Diagram

Below is an entity relationship diagram which describes the interactions between the user and the Black Box. It is important to define these interactions before the development of the Black Box to ensure that our design aligns with how we want the user to interact with our product.

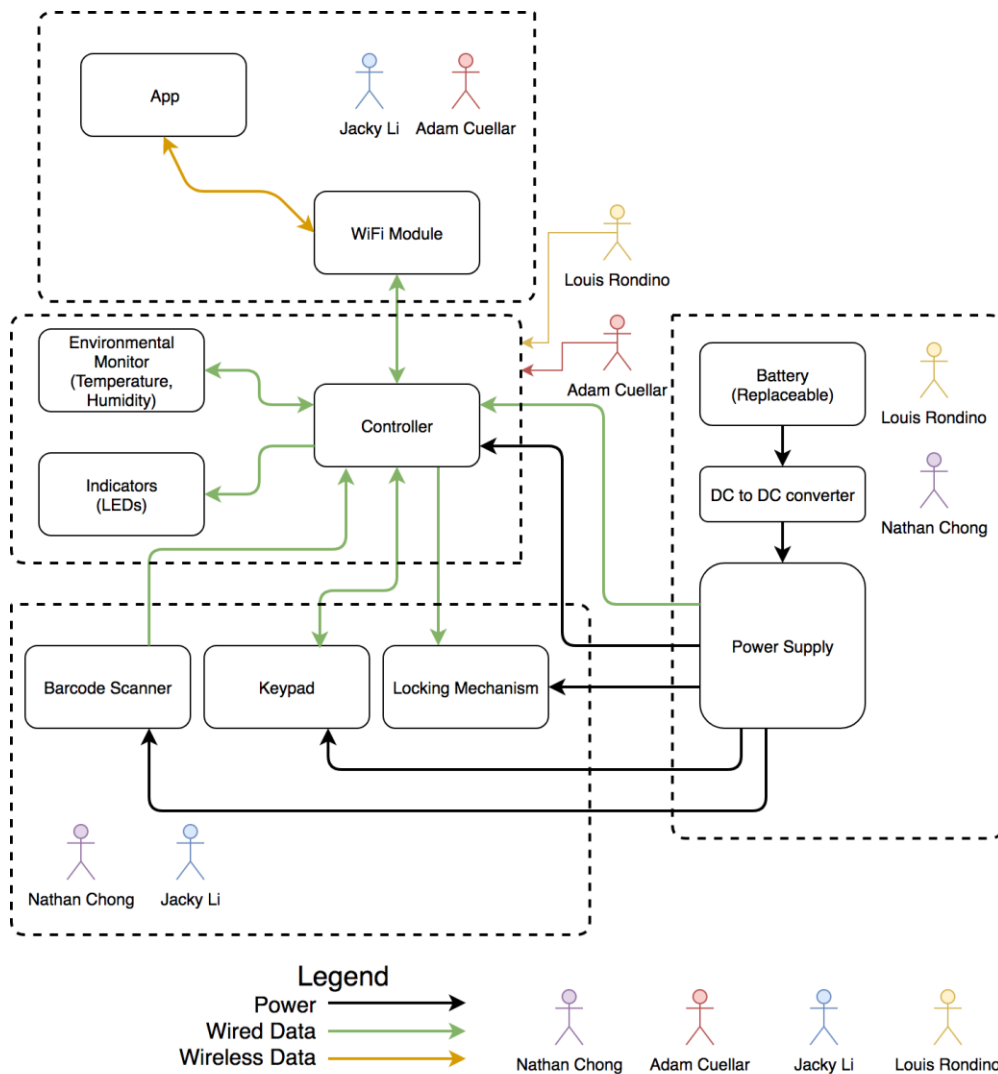


Figure 1 - Entity Relationship Diagram

The key modules of our system are represented in Figure 1 by each of the dotted lined boxes. Firstly, on the right, we have our power supply. We chose to utilize standard non-rechargeable batteries so that the user can instantly replace them without the need to plug in an immobile outdoor box. Our power supply will be able to report when the batteries are low, so the user can preemptively change them. Second, we have the physical access systems in the lower left box. This includes the lock and physical access systems such as a barcode scanner for the postman and keypad for any authorized user. Finally, in the upper left two boxes, we have the controller and virtual access methods. Our controller will be designed to connect with the user both physically and over a wireless network. Physically, the controller will be able to indicate whether the box is locked or unlocked via small LED light. Virtually, the controller will connect over Wi-Fi and be able to tell the user when a package is delivered and if their package is in any danger from environmental factors such as a high humidity or heat. All these systems combine to give the user an easy, intuitive, and secure way to receive packages.

It is important to note that this ERD is the initial design concept; thus, it is subject to change. Future changes will be made when new ideas occur and/or if a problem exists that would need an extra feature to implement to the Black Box.

3.0 - Design Constraints and Standards

For every engineering project there are design constraints. Design constraints are the conditions in which we need to satisfy to successfully complete our project. Some of these project level design constraints include:

- Economic
- Environmental
- Political
- Ethical
- Social
- Sustainability
- Manufacturability

Limitations also exist in the form of standards. Standards are the defined characteristics of a product, or project, with respect to dimensions, safety, performance, etc.

3.1 - Constraints Encountered

Manifesting the idea of the Black Box came along with taking into account the constraints we will encounter along the path to a fully developed product. Below are the anticipated constraints for the development of the Black Box.

3.1.1 - Economic and Time constraints

The economic constraints we've evaluated for the Black Box include the limit of the selection of parts which will be used in our design. Some of our ideas were not

considered due to the limited budget we had for the project. The Black Box budget is roughly \$800. The most advance, top of the line security sensors and components were not obtainable due to this constraint. In some cases, we had to find more budget friendly components for our project.

The time constraint we will be facing is the completion of the Black Box by the end of the summer semester. Research and testing of the design will be completed approximately by the end of April 2019. Schematics and visual description of any hardware will be completed by this time as well. In May 2019, the production and development of the Black Box's hardware will begin. The completed Black Box will be presented towards the latter half of the summer semester. This timeline allows for proper testing, implementation, and final changes. Time is a critical factor and in order to be successful we have to be on track as a team. We must adhere to our own schedule as well as meet the deadlines of the Senior Design 1 and Senior Design 2 classes.

3.1.2 - Environmental, Political, and Social

Social and political constraints do not apply to the production of the Black Box due to the nature of the product. Environmental constraints consist of the need for a reliable but environmentally friendly material to form the outside of the box's enclosure. This material will house the parcel package and the electronic components inside. The battery that is in the Black Box will be a rechargeable battery along with high capacity to sustain the longevity of the box and the amount of waste produced. After the battery ages it must properly be disposed and in order to reduce our physical waste, we've decided to implement the most efficient but environmentally friendly solution.

3.1.3 - Ethical, Safety, and Health

The safety of the Black Box must be considered due to the typical users age range. As the box will be available to the use of a typical residential family home, there will be ages ranging from children to elderly. The weight of the box's lid must be considered because many children and elderly users will not be able to lift a heavy lid on the box. The safety of a lighter lid will also reduce the amount of accidents of closing the lid on the user's hands. All the electronic components and electrical contact points will be covered and insulated to prevent potential electrocution. All electrical components will have to be properly grounded. The box should be of a lighter weight design along with being sturdy. Some users will want to move it in different places before installing the box in a permanent area; therefore, the weight of the box itself should be as light as possible.

3.1.4 - Sustainability and Manufacturability

The manufacturability constraints of this project will consist the restriction of a design that will be able to be manufactured in working condition. Manufacturing constraints will be the services that will be available in our area. The sustainability constraints are considered heavily due to the environment the Black Box will be

located in. Typically, the box will be located outside and exposed to many different weather environments over the years. Securing the user's packages is our top concern no matter how long the box is in use for. The manufacturing constraints along with the sustainability constraints caused some slight conflict in the overall design of the box. Considering the manufacturing constraints, we had to choose materials and equipment that we have access to; however, when taking in consideration sustainability many of these things had to be changed. It was imperative to find a compromise between the two constraints and the team has chosen what we believe will be the best path to the production of the Black Box.

3.2 - Standards and Safety Concerns

Various standards will be dealt with throughout the production of the Black Box. Many electrical and software standards are applicable to the production of the Black Box and therefore outlined below. Standards are rules and limits we can adhere to for acceptable benchmarks. The standards in this section will consist of laws and statues that are enforced by an administration, the government, or a adopted universal standard.

3.2.1 - Soldering Standards

The creation of the Black Box will involve soldering; therefore, it is imperative to understand the standards behind soldering to ensure the safety of our product as well as the safety of the electrical engineers on the team.

According to the National Aeronautics and Space Administration's document "Soldered Electrical Connections", there are several aspects to a proper soldering technique. Some of these include through hole technology, round lead termination, and other surface mount soldering techniques. Throughout the production of the Black Box there will be several surface mount components as well as small components such as resistors. In attempt to improve our knowledge and maintain safety we've used NASA's soldering standards as a guideline to soldering applications throughout our project.

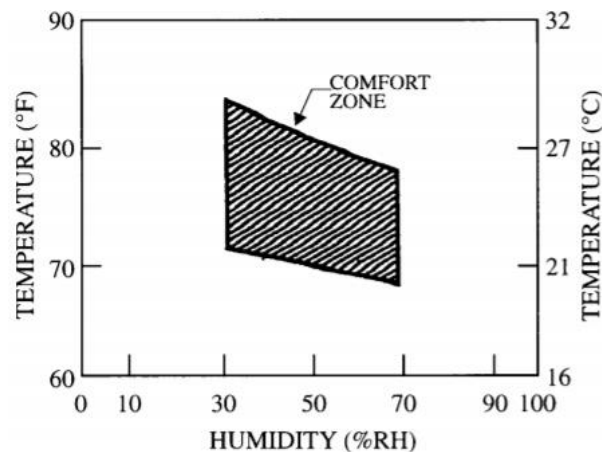


Figure 2 - Comfort Zone (Permission to reproduce submitted) [THSM19]

Soldering should be completed in a controlled environment. In order to limit the entry of contamination the soldering area should be controlled, and the temperature and humidity of the area be within the limits of what NASA defines as the comfort zone. In Figure 2, the comfort zone is defined as between approximately 70 to 85 degrees Fahrenheit with respect to approximately 30 to 70 percent humidity.

When mounting parts NASA, has detailed specifications on acceptance and rejection criteria. These criteria include: Stress Relief, Part Positioning, Visibility of Markings, and Glass Encased Parts. Stress Relief is the formed aspect of a conductor that minimizes stress between terminals due to its sufficient length. NASA believes all leads and conductors terminating in solder connections should provide freedom of movement of part leads/conductors between points of limitation. An example of this is provided in Figure 3 below.

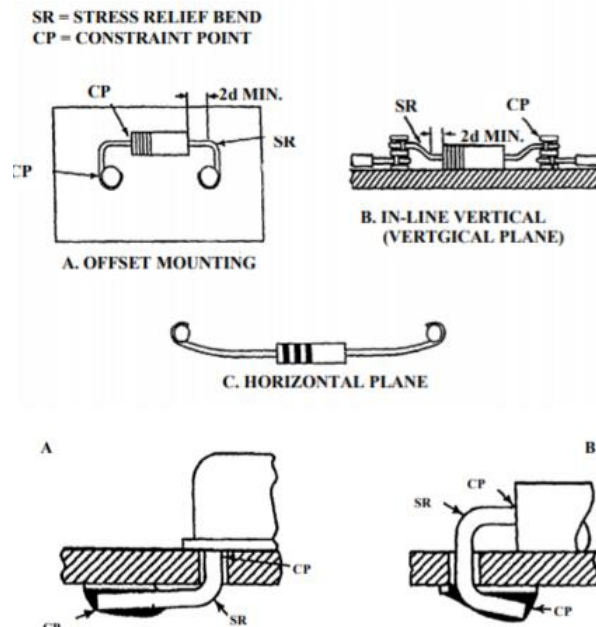


Figure 3 - Stress Relief Examples (Permission to reproduce submitted)

Part positioning is also an imperative aspect of parts mounting under NASA's standards. All parts will be positioned with respect to their engineering documentation and will be mounted so that terminations of other parts are not obscured. If parts must be mounted over one another and there are conductive materials, a suitable layer of insulation will be applied to ensure separation. All markings should remain visible as well. Parts mounted should be done in a manner in which value, part type, and other important markings are visible. If the marking will be covered regardless of position or orientation, the order of precedence below will take part for which markings will remain visible:

- Polarity

- Traceability code
- Piece part value and type

Glass encased parts will be covered with an approved material where damage from other sources is likely. These parts include items such as diodes, thermistors, or resistors.

3.2.2 - Lead Solder Safety

While welding is used in circumstances that require a strong load bearing joint, soldering is useful when a weak, malleable joint is needed. The intent of soldering is often for electrical contacts where the required connection will not be permanent and might be reversed later. Soldering facilitates the joining together of various electrical components such as transistors, MCU's, resistors, capacitors, etc. The various components are manufactured with metal legs or terminals which are used to connect the parts to the circuit. Soldering makes clean electrical connections between the components and the circuit board. The electrical components can be combined to create all sorts of useful devices, like calculators, gaming devices, watches, or computers. We will be using solder to connect components such as a microcontroller to its circuit board.

Through-Hole and Surface Mount soldering are the two types of soldering methods used to connect parts to boards. These processes have many major differences. Through-Hole soldering is a method where the lead is placed directly into drilled holes on an empty PCB [THSM19] The connections are known for being stronger and much more reliable. These connections are recommended for parts that might experience mechanical stress at some point. Despite their decrease in popularity during the rise of surface mount technology, they are acknowledged as being better for higher power components due to their greater surface area for conduction.

Surface Mount Technology was developed in the 1960s and since then has greatly increased in popularity. This method does not require holes to be drilled through the board and also allows components to be added on both sides of the PCB. Having double the surface area to work with has allowed for much more dense and compact boards to be designed. Due to the lack of drilling and time required, Surface Mount Technology has greatly reduced the cost of production for parts that require soldering. Unfortunately, surface mounts can be less reliable than the Through-Hole method when stress is applied to the mount. We will be using Surface Mount Technology because most modern microcontroller pin chips require a higher pin density for their added functionality. This technique will require us to receive assistance from professionals in the area.

The temperature that solder should be heated to is determined by the method of application being used, and the chemical composition of the solder. Table 8 below shows melting temperatures based on the solder type. Typical solder contains 60% tin and 40% lead, and has a melting point between 183 degrees Celsius and 188

degrees Celsius [Duck]. These temperatures are high and can easily harm operators if proper safety methods are not practiced.

Table 8 - Soldering Temperature Chart [Vis19]

Solder Type	Lead/Non-Lead	Temperature (C°)
63/37	Lead	183
60/40	Lead	183-188
50/50	Lead	183-212
45/55	Lead	183-224
40/60	Lead	183-234
96S	Lead	221
95A	Lead	236-243
Alloy No. 1	Lead	183-215
Alloy No. 2	Lead	183-190
HMP 5S	Lead	296-301
LMP 62S	Lead	179
TLS/5	Lead	296-301
TIN	Tin	232
99C	Non-lead	227
97C	Non-lead	230-250
SAC3	Non-lead	217-219
MC1	Non-lead	232
PJV-66	Non-lead	081266

Soldering can be very dangerous if proper safety measures are not observed. The University of Cambridge has an in-depth report about Soldering Safety and the risks associated with using soldering irons for electrical work [Univ]. They list many of basic recommendations, like never touching the elements of the iron, always keeping the cleaning sponge wet, and always turning off and unplugging the iron when finished. The greater dangers come from fumes and contact with the lead, since lead and rosin are used in the soldering process.

Lead is a primary component of most solder types, and also is known to bring rise to many serious chronic health issues. Since limited fumes are created in the soldering process, basic control methods should be enough to combat any fume issues. Good benches contain filter systems, which should be kept up to date and maintained regularly. These filters should vent to the outdoors. A major concern

with lead work is accidental ingestion from skin. To avoid this, the operator should wear gloves when handling solder and should always wash their hands with soap and water after they have completed the project.

Another major concern with soldering iron safety is maintaining electrical safety in the area. If there are any visible damages to the electrical cables of a soldering iron, it is advised that the operator does not use the iron. Also, all soldering irons should have electrical safety (PAT) testing every year. If not, there is no way to be certain that the iron is in its best condition. Keeping the soldering station free of electrical cables is another way to ensure that the heated tip will not cause fires within the area.

Finally, all solder waste must be collected in a lidded container. The container must be labeled with its contents, and no other materials should be placed in the container, to avoid fire hazards. Solder sponges and other used rags should be sealed into a waste bag and disposed of in a proper setting. For questions about the local toxic waste management methods, local Waste Management or Safety Office numbers can be found online. It is advised that anyone who solders more than once a week, especially if rosin is used in the solder methods, should contact local Safety Offices to ensure that all health and safety precautions are being taken.

3.2.3 - RoHS Compliance

RoHS (Restriction of Hazardous Substances) restricts the use of some particular materials when working with electrical components. It originated in the European Union and applies to products in the market after the date July 1st, 2006. The reasoning for the restrictions is that these materials are known to be dangerous to local environments and can cause major pollution in landfills. They can be harmful to wildlife in the area, and also to humans. Disposing of them properly can also be dangerous if workers are exposed to the materials without knowing to protect themselves with proper safety equipment.

The substances that are restricted by the RoHS include mercury (Hg), lead (Pb), polybrominated biphenyls (PBB), cadmium (Cd), and other phthalates. These regulations affect businesses that sell electronic products that contain the restricted materials. The restrictions also apply to those who use the materials in sub-assemblies of parts. Businesses that are not necessarily part of the EU must also comply if they sell components to EU businesses. If we choose to use lead solder for our circuit assembly, we may be subject to making sure we comply with RoHS standards.

The enforcement of RoHS laws is extremely important. Without constant enforcement, the laws will not be upheld, and harmful substances could begin to pollute the environments. Although it may be cheaper to purchase components from companies who manufacture parts that do not comply with RoHS laws, it is dangerous in the long run for operators to be handling these materials, and could

cause the company to have lawsuits later on. It is important that we practice these regulations within our own project, as a way to prepare us for the real world. Also if our product ever became bigger, having a track record of working with non-compliant materials could ruin a potential start-up company.

Many companies are upheld to strict standards by the RoHS. These standards require constant physical inspections, especially in the Netherlands and Belgium. Right now, there does not exist a test that provides absolute answers about the amount of heavy metals used in factories, but the International Electrotechnical Commission (IEC) is currently working on an international standard testing procedure that would test for the six different RoHS materials. Companies found guilty for use of these substances could face a monetary fine, which would depend on the location of the company. Another repercussion of not following the RoHS is having the company's' components imported back to where they were purchased. The company would be responsible for the transportation of the goods back to where they came from, and any costs that would be generated by this [RoHS]. Therefore, it is imperative we comply with RoHS standards.

3.2.4 - C/C++ Programming Standards

Using coding standards makes reading and maintaining code significantly easier. According to NASA [Shoa], the recommended style for writing C/C++ programs is defined as:

- Organized
- Easy to read
- Easy to understand
- Maintainable
- Efficient

Below are guidelines for organizing the content of any code involved with the Black Box. These standards can be used across multiple languages and, if necessary, will be.

Names

It is imperative to choose names that are meaningful. Meaningful variable or function names show clear relationships and reasoning within the code if fitted properly. General directions are outlined below.

- In situations where a programmer could use all upper-case abbreviation, either do so or instead use an initial upper-case letter followed by all lowercase letters.
- Avoid underscores or hyphens
- Consistency in naming style is paramount

Class names should contain a capitalized first letter of each word within the name. A GUI component class name will contain the suffix of the parent's name. Exception classes will contain the suffix 'Exception'. For class library names

namespaces should be used to avoid clashing and 'using' clauses should also be avoided. These clauses can be replaced with the scope operator '::' instead; however, in avoidance of clutter 'using' can be used. Function names should depict the action the function is performing. The name given to the function should be clear. Verbs should be written in mixed case starting with upper case. The name of the class should not be duplicated in the function name. A preferred method involves using the following prefixes to depict the action of a function.

- Is/Has/Can - used to as a question, usually returns a Boolean
- Get/Set - to get or set values
- Initialize - initializes an object
- Compute - computes

Arguments for these functions should follow the same guidelines for variable names. Using the same name for an argument as the class' type is allowed; however, can become overwhelming. Therefore, this method should be used scarcely.

For variables, it is preferred to declare only one variable per line of code. Variable names should begin with a lowercase letter. In the instance in which there is more than one word, the mixed case method should apply. If the name of the variable is not clear, then comments should be included to explain what the variable is for. Variable names should be declared at the level in which they are used. Therefore, if a variable is used throughout a procedure then it should be declared at the top of said procedure. If it is used only within a block, then it should be declared at the top of the block.

For pointers, the asterisk symbol should reside directly in front of the variable name rather than with its type. If a pointer is null, "NULL" should be used. Reference variables should include the ampersand in front of the variable name rather than with its type. Global variables should be used scarcely and only if absolutely necessary. A safer alternative are namespaces in C++. Constants should be all caps and underscores between words. Same applies to the #define statements; however, constants are preferred [iComp].

Braces should be used for all blocks and should appear on their own line. The opening brace should appear on the following line and lined up with the keyword. The closing brace should appear on the last line and lined up with the first brace.

Indentations should be implemented with 4 spaces. No tabs should be used throughout the code as tabs are defined differently for several editors. If several variable declarations are listed together, the variable names should be lined up. There should be one space after commas or semicolons. There should be one space around assignment operators. There should be one space between keywords and parentheses. There should be one space around conditional operators. There should be no spaces before parentheses following function

names. There should be no spaces between unary operators and their operands. There should be no spaces around the primary operators. Maximum characters per line should be 80 characters. Method/Function arguments may not fit all on one line. In this case, the first argument should be lined up in each line.

3.2.5 - USB Standards

With USB Standards This section will cover USB (Universal Serial Bus) this section is very important that was accounted for when starting this project. For USB 1.0 was first developed in 1996, the data rate transfer at a low speed moves at a rate of 1.5 Mbit/s and also at full speed it moves data 12 Mbit/s. When USB 2.0 was introduced it has given the tech word a very fast at the time 480 Mbit/s providing high speed for the user when transferring data. In 2011 we were introduced to USB 3.0 giving us a very fast speed for even today's time for moving data, USB 3.0 5 Gbit/s of fast data transfer. As time went on we had some revisions for USB 3.0, this gave us 3.1 with a data transfer speed of 10 Gbit/s, and lastly as of 2017 USB 3.2 giving us 20 Gbit/s.

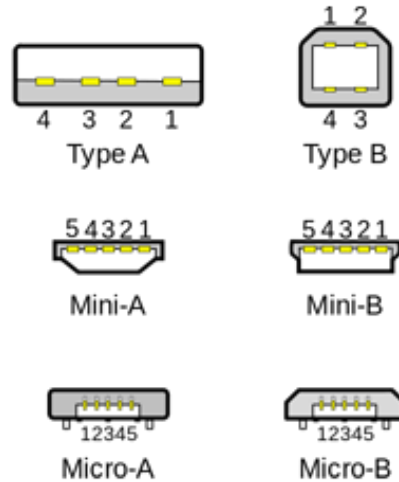


Figure 4 - USB Types (Permission to reproduce submitted)

Above is Figure 4 which shows the different type of USB's. It is essential to be able to identify which USB's are common if a USB connection will be implemented into the Black Box.

Table 9 - USB Advancement

Connectors	USB 1.0	USB 2.0	USB 3.0	USB 3.1	USB 3.2	USB 4.0
Year	1996	2001	2011	2014	2017	2019
Data rate	1.5 Mbit/s	480 Mbit/s	5 Gbit/s	10 Gbit/s	20 Gbit/s	40 Gbit/s

In Table 9, it shows the advancement of USB generations, helping us decide on USB speed for our senior design project.

3.2.6 - Barcode Standards

The following information depicts the standards that arise from including a Barcode scanner within the Black Box. Using bar codes is an extremely efficient tool for tracking packages and inventory in many organizations. However, in this day and age there are many different barcode types we have to consider. Below, in Figure 5, are the 3 different barcode symbology.









Symbol	Example	Character Set	Variable Length	Discrete/Continuous	Check Character	Application
Code 39 USS-39	 1 2 3 4 5 6	A	Variable	Discrete	Optional	In very wide use for many types of applications: Logmars, HIBCC
Code 128 USS-128	 1 2 3 4 5 6	Subset A,B,C	Variable (Even # of Subset C)	Discrete	Required	Widely used; excellent for many applications
UPC-A	 1 2 3 4 5 6 0 0 0 0 0 1	Numeric Only	12 Fixed (11 data + 1 check digit)	Continuous	Required	Retail product marketing in USA and Canada
UPC-E	 1 2 3 4 5 6 0 0	Numeric Only	7 Fixed (zero + 5 data + 1 check digit)	Continuous	Required	Retail product marketing in USA and Canada; compressed for products with limited label space
EAN-13	 1 2 3 4 5 6 0 0 0 0 0 5	Numeric Only	13 Fixed (12 data + 1 check digit)	Continuous	Required	Retail product marketing world-wide
EAN-8	 1 2 3 4 5 6 0 1	Numeric Only	8 Fixed	Continuous	Required	Retail product marketing in USA and Canada; compressed for products with limited label space
Interleaved 2 of 5	 0 1 2 3 4 5 6 7	Numeric Only	Variable (Even # of Digits)	Discrete	Optional	Very compact; encodes digits in pairs—total length must be even numbers of digits
ISBN	 9 7 8 1 2 3 4 5 6 7 8 9 7	Numeric Only	Variable (Even # of Digits)	Discrete	Optional	Very compact; encodes digits in pairs—total length must be even numbers of digits

Figure 5 - Barcode Symbology (Permission to reproduce submitted)

Numeric Barcodes

Numeric only barcodes are barcodes where only numbers are stored within the barcode. These are one dimensional barcode that encode specifically numbers. Numeric only barcodes represent data in a variation of width and also spacing to output a certain number. There are a couple different symbology's inside barcodes we researched before picking a bar code scanner for our project.

UPC Codes

UPC codes are the most commonly used symbology in the industry, as seen in Figure 6. Many retail stores use this symbology because it is a twelve-digit code that contains very basic information to look up a manufacturer's identification number for the product. More information can be researched if the user has the

service with UPC database, where it reveals the information code assignment organization. Some UPC Codes are basic and is called UPC-E where it only has a six-digit combination.

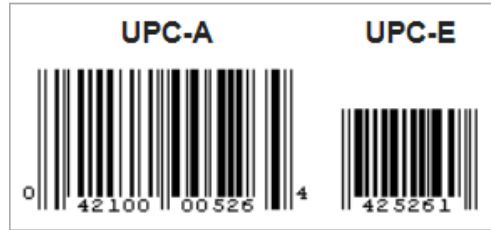


Figure 6 - UPC Example along with UPC-E (Permission to reproduce submitted)

POSTNET Codes

POSTNET codes is a Postal Numeric Encoding Technique, as seen in Figure 7. This symbology is used by the United States Postal Service to encode the variety of zip codes in the US and which allows for efficiency of directing the mail to correct geographical areas. Postal numeric coding technique symbology was developed to sort the mails into automation. POSTNET codes are not real barcodes as in the when barcodes are usually coded in variation of width and length to indicate each set of numbers. For POSTNET the numeric symbol uses five bars with two and three short and a four space between each character. The length of the bars, and the spaces between them is constant throughout the code. A single bar is used to start and stop the characters in the symbol to include a check character. POSTNET codes consist of lengths ranging from a small length of 32 bars all the way to 62 bars. The benefits to POSTNET is that its extremely easy to implement and print any code with any printer that is widely adopted. The limitations are most barcode scanners will not be abled to decode, due to the very limiting market use for POSTNET codes.

Digit	7 4 2 1 0	Barcode
0	1 1 0 0 0	
1	0 0 0 1 1	
2	0 0 1 0 1	
3	0 0 1 1 0	
4	0 1 0 0 1	
5	0 1 0 1 0	
6	0 1 1 0 0	
7	1 0 0 0 1	
8	1 0 0 1 0	
9	1 0 1 0 0	
Start/Stop	- - - - 1	

Figure 7 - USPS Representation of POSTNET Codes (Permission to reproduce submitted)

4.0 - Research Related to Project Definition

As the idea of the Black Box sprouted into a tangible product, the necessity of gathering information on similar idea's became crucial to developing a product that is unique and operable. Below are a few of the relevant technologies and techniques we encountered.

4.1 - Existing Similar Projects and Products

The lock box market is a wild west of innovation. There is really no product which has a large share or has cornered sales. Many companies are rapidly coming up with designs for these smart package protectors due to demand and any lack of major products already in the market space. Below are some popular existing designs that can currently be seen in use today.

Amazon Locker

The closest existing technology is the Amazon locker, shown in Figure 8; although it was difficult to find a product similar to the Black Box. The Amazon Locker, and other similar technologies, are usually located in apartment areas, share similar attributes, and are based on a commercial based business'; however, the Black box is intended for residential use. The Amazon Locker allows customers to select any locker at participating locations to retrieve their orders by entering a unique pickup code. The Amazon locker was invented to reduce the concerns for missing and stolen parcels in the mailing process.



Figure 8 - Amazon Locker Technology (Permission to reproduce submitted)

Land Port

Another existing technology is called the Land Port, shown in Figure 9, which is a metal box that is placed on the front porch, bolted down, and can store packages. There is a keypad on top of the lid that is used for inputting your own 4-digit code. Instructions can be given to the mailman that includes the code for the box to open.



Figure 9 - Land Port Technology Package Protection (Permission to reproduce submitted)

4.2 - Relevant Techniques

Many parcel lockers are unique in what they use, regarding smart electronic technology, ensuring the best and most secure experience. The main process of the lockers is broken down into three simple steps:

Step 1: Parcel Delivery

The mailman/courier can approach the parcel locker and enter their special employee passcode, which is to be provided by the owner of the parcel locker.

Step 2: Owner/Recipient is Notified

When package is delivered and is safely secured in the parcel locker, the owner will be notified via text, mobile application, or email. Due to the commercial aspect of these lockers, they are provided a unique code for every use; therefore, each package will generate a unique code to avoid theft.

Step 3: The Owner/Recipient Collects the Package

As the owner approaches the central hub of the lockers, he or she will enter their special passcode provided to them. Once the correct barcode or passcode is scanned a specific locker will be opened and the user will collect their package.

4.3 - Relevant Technology

Throughout the formation of the idea of the Black Box, our team has considered many technologies involving communication all through the product.

4.3.1 - Serial Communication

Embedded electronics use many protocols/techniques to interlink circuits which results in a collaborative system. For these circuits to communicate efficiently, they must share a common communication protocol. There are many communication protocols and they can be categorized into two groups: serial and parallel.

Parallel protocols transfer multiple bits at the same time. This is achieved through the requirement of parallel usage in having eight, sixteen, or more wires which act as buses of data. Data is transferred in a multitude of waves containing 1's and 0's. Serial protocols transfer data one bit at a time. It can do this by operating with one wire and usually never reaches more than four. Comparing the two interfaces, parallel is faster, straightforward, and easy to implement. However, by taking a look at a basic microcontroller such as Arduino Uno to a more complex MCU such as a Raspberry Pi, one can notice that the input and output lines are few. Therefore, it is more ideal to use serial communication even though it sacrifices some potential speed.

For the past couple of years, many serial protocols were created to satisfy the specific needs of embedded systems. Two of the most well-known serial protocols are the Universal Serial Bus, or USB, and the Ethernet cable. Other common ones are SPI, I²C and UART. These interfaces can be grouped into two categories: synchronous and asynchronous.

Synchronous vs. Asynchronous

Synchronous serial protocols will always pair data lines with a clock signal. This means that all devices on a serial bus will share a common clock. It is more straightforward and has a faster transfer rate than asynchronous with a downside of requiring an extra wire between devices communicating with each other. SPI and I²C fall under this category.

Asynchronous serial protocols have their data transferred without the support from an external clock source. This type of protocol is suited for minimizing the required wires and I/O pins with a downside of putting in extra effort towards transferring and receiving data reliably.

Since the asynchronous serial protocol does not use an external clock signal for data transfer, it must be accompanied with mechanisms that aid in transferring data with no errors. These mechanisms are:

- Parity Bits - low-level error checking, can be odd or even
- Data Bits - Bits that carry data; standard size is 8-bit byte; need to specify the endianness of the data - either most significant bit (MSB) or least-significant bit (LSB) is sent first.
- Baud Rate - how fast data is sent over a serial line
- Synchronization bits - Start bit and stop bits; details the beginning and end of a packet sent

It is imperative that both devices on a serial bus are configured with matching bits mentioned above. Not being careful with all the configurations will yield garbage values. In Table 10, a comparison is made for synchronous and asynchronous functions. This is made to view which type of transmission is ideal for the Black Box.

Table 10 - Synchronous vs. Asynchronous Comparison

Basis for Comparison	Synchronous Transmission	Asynchronous Transmission
Meaning	Sends data in the form of blocks or frames	Sends 1 byte or character at a time
Transmission Speed	Fast	Slow
Cost	Expensive	Economical
Time Interval	Constant	Random
Gap Between the Data	Absent	Present
Example	Chat Rooms, Video Conferencing, Telephonic Conversation, etc.	Letters, emails, forums, etc.

UART

UART stands for universal asynchronous receiver/transmitter and is a hardware device responsible for implementing serial communication. Many microcontrollers support and use UART as in intermediary between serial and parallel interfaces. They can also exist as an IC, just in case the given microcontroller does not have it integrated in it. One side of the UART is a bus that commonly has eight data lines and some control pins. On the other side, two serial wires exist which are called TX (transmitter) and RX (receiver). This can be seen in Figure 10. It is easy to tell that the serial interface looks easier to implement than the parallel interface; however, the parallel interface can take care of more data lines at the same time.

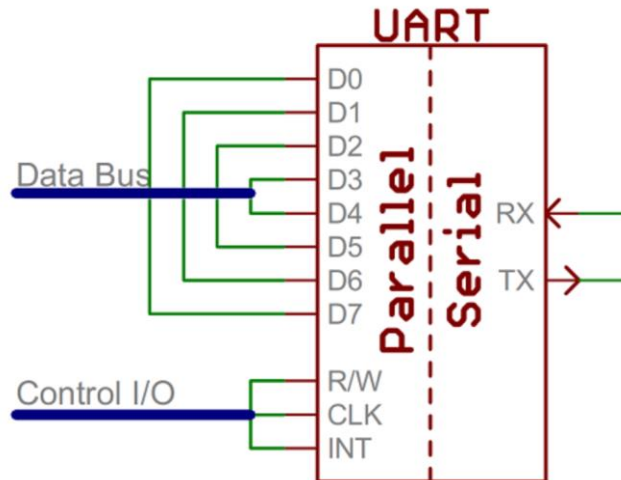


Figure 10 - Typical UART interface (Permission to reproduce submitted)

The transmitter side creates data packets (sync and parity bits) and sends them to the TX line with a timing that is based on the baud rate. On the receiver side, UART samples the RX line at a rate of the baud rate to pick out sync bits and expels the data.

SPI

Another serial communication interface is serial peripheral interface (SPI). It is known as an interface bus used to send data between peripherals such as sensors, shift registers, and microcontrollers.

SPI behaves slightly different when compared to the other forms of serial communication. It uses different lines for data and uses a clock that helps keep the transmitter and receiver in sync. A clock in this configuration can be described as an oscillating signal that directs the receiver when sampling bits on the data line. There are two options it can take: rising (low to high) or falling (high to low) edge of the clock signal. Whichever edge is used, the receiver will go to the data line and read the next bit. Since the clock is sent in conjuncture with the data, the speed of which data is sent and received is not important. There are four main wires that the buses in SPI use:

- Serial Clock - a clock signal that synchronizes generated by the master that synchronizes all SPI signals
- Slave Select - a signal used by the master to give choice to which slave is needed for communication
- Master In-Slave Out - abbreviated as MISO, it is a data signal with one direction only that comes out of the slave output pin to the master input pin
- Master Out-Slave In - abbreviated as MOSI, it is a data signal with one direction only that comes out from the master output pin to the slave input pin

The master in this case is the microcontroller and in SPI only one side generates the clock signal. When MOSI occurs, data is sent from the master to a slave on a data line that is called MOSI. If a slave needs to respond back, the master will generate clock cycles until it is appropriate for the slave to input data onto a third data line called MISO.

Slave select, or SS, is a data line that tells when a slave should be active and receive/transmit data. It is also used in cases where multiple slaves are present; thus, the master can freely select which slave it wants to talk to.

The slave select data line is normally active low, data line is held high, which disconnects slaves from the SPI bus. Before data is sent to a slave from the master, the data line is brought low, activating the slave. Once the slave is done being used, the line is held high again disconnecting the slave from the SPI bus. For multiple slaves, there are two ways of connecting them to an SPI bus:

- Independent Slave Select - where each slave has its own slave select data line connected to the master shown in Figure 11

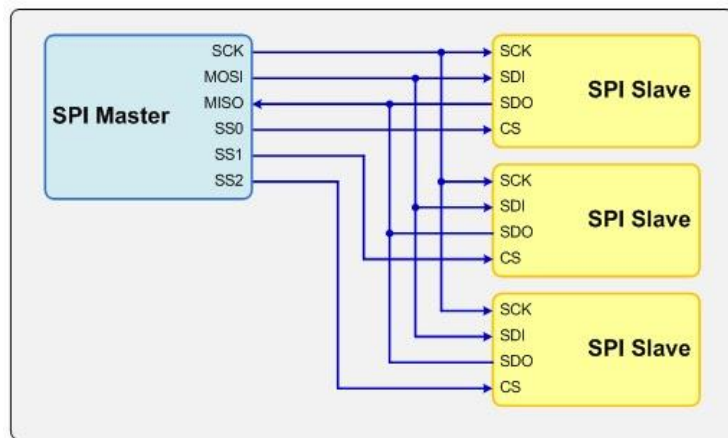


Figure 11 - Independent Slave Select (Permission to reproduce submitted) [Ver19]

From Figure 11, the SPI Master contains a slave select line for each slave. Each line must be set to low because doing the contrary will allow two or more slaves to be activated at the same time. This occurrence will result to garbled data. Having many slaves will require many slave select lines; thus, the user has to keep in mind the amount.

- Daisy-Chained Slave Select - where a single slave select line leads into each slave as shown in Figure 12.

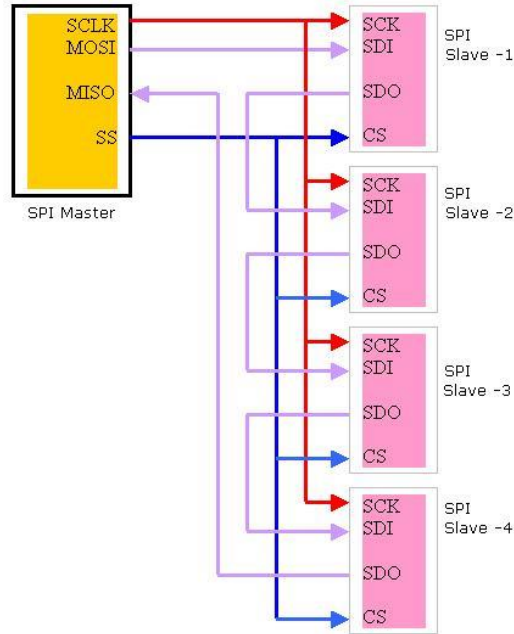


Figure 12 - Daisy-Chained Slave Select (Permission to reproduce submitted) [EEHe]

I²C

The Inter-integrated circuit protocol intends to allow multiple slaves to communicate with one or more master chips. The slaves and masters are digital integrated circuits also known as chips. I²C offers the best of both worlds when comes to comparing this protocol to SPI and UART. This is because I²C requires two wires (like asynchronous serial) and supports a master-slave configuration (like the configuration in SPI). The master-slave protocol is differed from the SPI in that I²C can support a multi-master system - there can be more than one master to communicate with all devices on the bus. The masters, however, cannot talk to each other the bus and cannot use the same bus lines at the same time.

I²C devices can communicate at high frequencies such as 400 kHz. It has multiple modes for data rates: 10 kilobits per second (slow mode), 100 kilobits per second (standard mode), 400 kilobits per second (fast mode), and 3.4 megabits per second (high speed mode). Hardware for implementing this communication protocol is much harder than SPI; however, it is easier to implement than asynchronous serial. For software purposes, it can be slightly difficult in coding it properly. Masters and sales may trade roles between transmissions. Four types of transmission can be characterized by their transmission behavior:

- Master transmission
- Slave transmission
- Master reception
- Slave reception

The buses on the inter-integrated circuit protocol uses 7-bit addresses and two signals in the two wires mentioned before:

- Serial Data (SDA) - this transmission line is a data signal that transmits bidirectionally between masters and slaves
- Serial Clock (SCL) - this transmission line is a clock signal that is generated by the master of the bus. Slaves can command this clock when the master is sending too much data. This is called clock stretching.

Serial data and serial clock lines use pull-up resistors because the I2C bus drivers can only pull the signal line low but not high again. These pull-up resistors usually have a resistance of 4700 ohms, but it can be adjusted down if it is necessary to do so. To transmit signals in I2C, messages are encoded in the signal. These messages are denoted as START, STOP, and ACK which means acknowledge. Once a master begins transmission, a start bit is sent along with the desired slave's 7-bit address. The last bit sent determines if data is read/written. Once the slave receives the message, an ACK bit is transmitted. The master will continue to operate in transmission or reception mode depending on the last bit encoded in the message. If a 0 is sent, the transmission is written. If a 1 is sent, the transmission is read.

Table 11 - SPI & I2C

Specifications	SPI	I2C
Number of Bus Lines	Four (SCLK, CS, MISO, MOSI)	Two (SDA, SCL)
Data Rate	Higher, about 10 MHz or more	Lower, about 100KHz or 400 KHz
Run Current	Lower, about 200 μ A at 4 Mbps, hence less power consumption	Higher, about 400 μ A at 400 Kbps, hence more power consumption
Preferable Application	Perform well in the single master and single slave configuration	Perform well in the multi master and multi slave configuration
Device addressing	Does not support	Support
Acknowledgement Mechanism (ACK)	Does not have the mechanism to support the receipt the data	It has mechanism to support receipt of data using ACK
Overhead in point to point connection	LESS	More
Application	Applications requiring continuous data steam transmission	Communication between device which not require continuous transmission, requiring occasional communication

Table 11 above addresses the SPI communications and I2C communication application. Depending on the type of communication that our MCU will support, this table is very important to view as reference.

4.3.2 - Wireless Communication

For Wireless communications we have gotten very far along the way from where we were 25 years ago. One of the biggest life changing type of communications are Wi-Fi, which was developed by Wi-Fi alliance in 1998, and Bluetooth which was introduced in 1994 by Ericsson. Table 12 below compares the current technologies for wireless communication.

Table 12 - Wireless Communication Analysis

	Wi-Fi	Bluetooth	RFID
Daily Use	Computer Internet connection & mobile connection	Small Media transfers, Audio streaming	Opening Doors, Materials Tracking, Electronic Tolls
Standard	IEEE 802.11n	IEEE 802.11.1	IEC 18000-2
Frequency Band	2.4 GHz, 5GHZ	2.5GHz	Big range from 120kHz - 10 GHZ
Wireless Mesh Network	30 Nodes	8	N/A
Range	30-50 Meters	2-10 Meters	1 Meter for Typical door RFID Readers
Type of Network	Point to Multipoint	Point to Multipoint	N/A
Battery Life	0.25 days to 4 days depending on battery	.5 days-7 days depending on battery	Years
Battery Consumption Level	High	Low	Very Low
Security	Max Security Wi-Fi Protected (WPA) TKIP encryption with 128 bit keys	Very Low 4 digit pin	Medium Hand held device that can read RFID may steal your Identifications

Why are we using a wireless connection in our project?

With wireless communications we can control our device over a wireless network which allows the device to be self-contained without the mess of wires and the convenience of opening the Black Box via Wi-Fi. The Black Box relies heavily on

wireless connection for the end user's sake. The box will transmit the sensor data to the mobile application showing the temperature and humidity, and also allow the user to open and lock the box.

Bluetooth

Bluetooth provides wireless communication across short distances by using ultra high frequency (UHF) radio waves in the 2.45 GHz frequency band with 79 unique channels. It was standardized as IEEE 802.15.1; however, the responsibility of maintaining this standard has recently been handed to the Bluetooth Special Interest Group (SIG). Their core responsibility is to assist the progress of creating low-powered and portable wireless personal area networks (PAN). This allows the communication between devices in an ad-hoc manner.

The difference between Bluetooth and Wi-Fi is that with Bluetooth, the operation distance range is much smaller and an access point is not centralized to provide communication to devices. Its focus is on the providing direct communication between devices by using a master-and-slave communication scheme. Common devices that use Bluetooth are smart phones, headphones, portable speakers, and many more.

The master and slave communication scheme can be described in the following Figure 13:

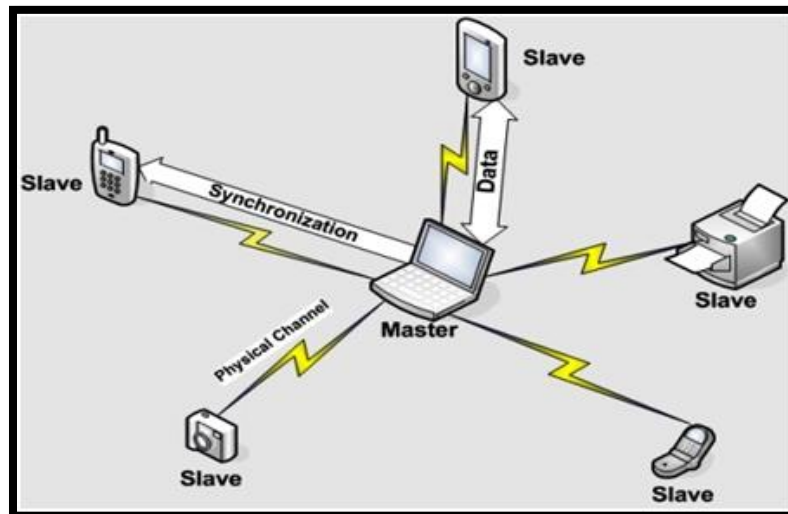


Figure 13 - Master and Slave Scheme (Permission to reproduce submitted) [EL19]

A master device is in control of the communications link and traffic between itself and the slave devices joined with it. It is also responsible for dictating when a slave device can transmit. A slave device is a device that gives a response to the master device by synchronizing either their transmitting timing or receiving timing to the master. When a group of Bluetooth devices are linked and sharing information with each other it is termed as a piconet. Thus, the master-and-slave scheme is a piconet. When two or more piconets link and share information, it is termed as a scatter net.

The amount of connections that Bluetooth can support between devices are up to eight. The master can communicate with the other slave devices; however, the slave devices cannot communicate with each other. The communications do not interfere with each other because each pair of devices uses one of the 79 unique channels. If devices want to communicate, they will randomly pick a similar channel. If there are devices that are already present in that channel, they will randomly switch to another channel. This technique is called spread-spectrum frequency hopping. To minimize the chances of interference, device pairs shift or hop about a thousand times per second.

When comparing Bluetooth versus Wi-Fi, it depends on what the user needs. For short distances and brief communication, Bluetooth is preferred since it is built for transmitting data between devices within a proximity. Additionally, it is relatively secure in which the user can restrict specific devices to trusted devices.

With that being said, however, Bluetooth would not be useful for the Black Box project due to one main reason. The fundamental flaw is that the mobile device will have to be within a close range to the Black Box to be notified about delivery details. This is counterproductive because the whole point of the Black Box is to safely secure a package when the user is not available. If the user must be close to the Black Box to be notified, then he/she can just receive the package from the deliveryman by themselves.

Wi-Fi

Wi-Fi is developed by Wi-Fi Alliance, a nonprofit organization that is responsible for the technology that allows devices to connect to a Wireless Local Area Network or WLAN. This communication protocol is known everywhere becoming a dominant communication standard present in many electronic devices. Due to its common existence, it is widely supported. Products made with Wi-Fi are based on the IEEE 802.11 standard. This standard is a series of protocols that describe the physical layer and media access control sublayer of a wireless network. It can be thought of as the backbone of the of the 802.11 protocol.

The physical layer (PHY) of the Open Systems Interconnection model (OSI) is the lowest level. This is where the hardware and direct manipulation of the bitstream takes place. The OSI model, shown in Figure 14, is a “conceptual model that characterizes and standardizes the communication functions of a telecommunication or computing system without regard to their underlying internal structure and technology” [ASM]. The physical layer provides many functions such as modulating transmission signals through the Physical Coding Sublayer (PCS). Another function is through the Physical Media Dependent sublayer (PMD). This allows a connection with the media access control (MAC) layer to the physical transmission medium (cable or antenna).

The MAC sublayer of the IEEE 802.11 is incorporated in the second lowest layer of the OSI model. This sublayer allows multiple devices to connect to a shared network by adding methods that address devices and access channels. Additionally, it contains full and half-duplex communications along with error checking. Because of the MAC sublayer, devices are able to connect to a network with a unique, assigned MAC address. This will allow the devices to accurately send and receive data packets. It is similar concept to that of homes with unique addresses which allow them to send and receive mail from other homes. Another important function that the MAC sublayer provides is the connection between the logical link control (LLC) sublayer to the physical sublayer. The LLC sublayer is a level that essentially allows high-level logical frameworks which includes functions such as the Internet Protocol.

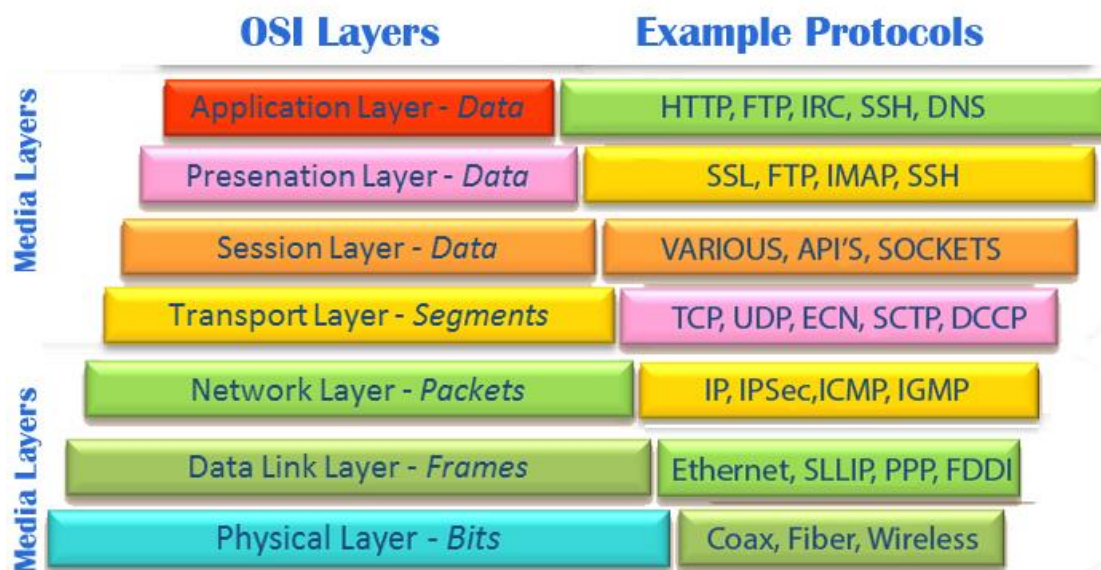


Figure 14 - OSI model (Permission to reproduce submitted) [ASM]

After receiving many revisions to wireless communications over specific frequency bands, the IEEE 802.11 protocol allows wireless communication over the industrial, scientific, and medical (ISM) frequency bands, which are immune to FCC licensing. There are two main frequency bands that devices mainly operate on: 2.4 GHz and 5 GHz. These bands are distinguished as follows and is shown in Figure 15:

- 2.4 GHz band
 - Allows “for only three non-overlapping 20 MHz channels” [Stev]
 - Provides coverage at a longer range
 - Transmits data at lower speeds
 - Devices are more prone to interference from other devices sharing this frequency such as garage door openers and microwaves
 - Has 11 channels for devices to use
- 5 GHz band

- Provides less coverage since higher frequencies cannot penetrate solid object such as floors and walls
- Transmits data at faster speeds
- Devices are less prone to interference since many devices do not use this band
- Has 23 channels for devices to use

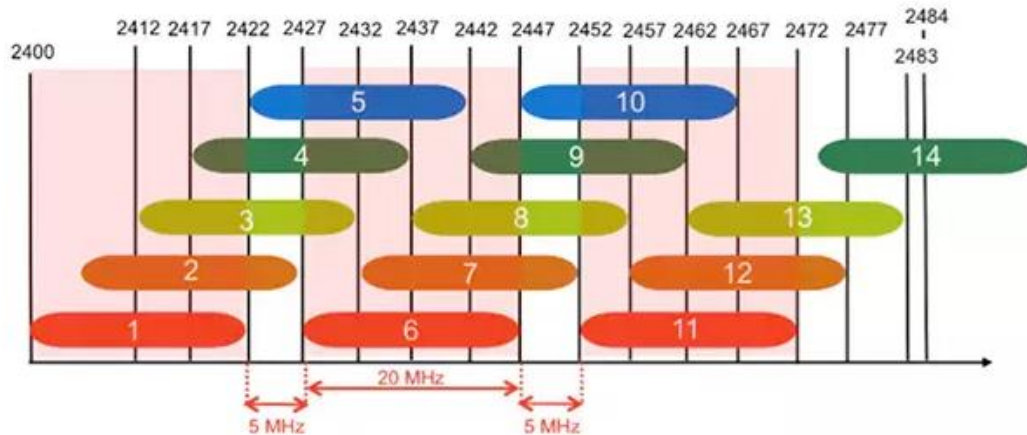


Figure 15 - Frequency Bands (Permission to reproduce submitted) [Stev]

To connect over these bands, wireless routers are devices compliant to 802.11 standard which grants wireless access points (WAP) for other devices. When devices are connected through a network, it is commonly known as a node. They connect to the routers by using transceivers known as wireless network interface controllers which are responsible for sending and receiving data through and from a network. This process can be done in two methods, infrastructure or ad-hoc.

Infrastructure mode offers a scaled, centralized security management and a higher reach than ad-hoc. However, the disadvantage of using this mode is the additional cost to purchasing AP hardware. Ad-hoc networks use peer-to-peer to connect devices; thus, wireless access points or routers are not needed for two or more devices to reach out to each other. Typically, infrastructure is used for a more permanent implementation of a network such as homes, schools, and businesses. It is not recommended for these designations to use ad-hoc due to the peer-to-peer method it uses. For applications of an ad-hoc network, it is seen used by devices that need to share files that are too far from a network to make it work.

For the Black Box design, Wi-Fi is a necessary communication protocol. It would be advantageous for the user to know when their package is delivered or when the Black Box has been tampered with. To do this, a mobile application will be created which will serve as a tool for the user to be notified. The Black Box must be able to communicate and connect to a household router, providing the user information regarding package delivery, tampering, etc. Additionally, this communication is

preferred more than the other types of communication due to how easy it is for the user to receive updates on their package and/or the Black Box.

Downsides for Wi-Fi communication are security vulnerability and network traffic. Perpetrators can easily gain access to a network and take advantage of the devices connected to it. To combat this, networks with a strong password must be made to prevent any harm that can be done on the Black Box. Network traffic can pose a problem since there are many Wi-Fi enabled devices such as phones, gaming consoles, smart TVs, and more. With many devices connected to the same router, network performance can easily be swayed. Download speeds will be slower with an increased number of devices simultaneously using the Wi-Fi network. This can delay the notification time of when the Black Box receives a package or when it is being tampered with. To avoid this delay, the Black Box can connect to the 5 GHz band, since there is generally less traffic and the speed is faster. However, the router will have to be placed strategically closer to the Black Box because the higher frequencies cannot travel as far compared to the general use of the 2.4 GHz band.

Table 13 - Band Frequency Summary

Band	2.4 GHz	5 GHz
Spectrum Allocation	ISM (83 MHz)	ISM + UNII (725 MHz)
Channels	3 Non-Overlapping	23 Non-overlapping
IEE 802.11 Standard	b, g, n	a, n, ac
Range	Longer	Shorter
Data Rate	Lower	Higher
Interference	Higher	Lower

Table 13 above shows the difference in Frequency band, providing us the range and, also the data rate each band is capable of transferring at.

RFID

Radio Frequency Identification (RFID) is a system using radio frequency to identify devices for tracking purposes. This type of technology is in the group known as Automatic Identification and Data Capture (AIDC). Methods used in this group automatically analyzes objects, gathers data about them, and enters the data into a computer system. In the case for RFID, it contains information on an integrated circuit which can be read without physical contact.

This is made possible by the transceiver and an antenna within the RFID system. The transceiver emits a radio frequency signal through its own antenna which then travels to the integrated circuit via an attached antenna (also known as the RFID tag). By modulating the RF signal, the information can be transferred to the interrogator. This technology has become common throughout all industries including, retail, pharma, logistics, and many more. Within those areas, RFID can perform tasks such as personnel tracking, access control, toll gate systems, etc.

The Black Box can use this technology as a gateway to grant/deny access to RFID systems. In this case, it will need an RFID reader with an antenna and an RFID tag/transponder that will be integrated into a credit-card sized card. The reader contains a radio frequency module, an antenna coil that will generate high frequencies inducing electromagnetic fields, and a control unit. The tag/transponder will also consist of an antenna coil which will detect the reader's induced electromagnetic field from its' antenna to power the transponder microchip. A voltage is generated from the reader's coil which is how the microchip in the tag will be powered. Figure 16 below demonstrates this occurrence.

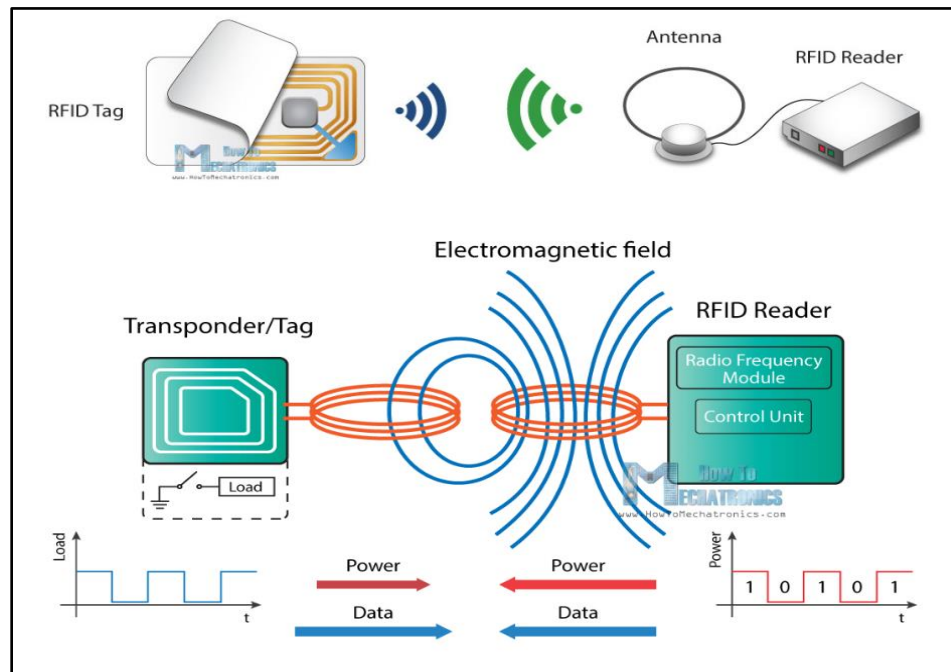


Figure 16 - RFID Pictorial Representation (Permission to reproduce submitted) [How19]

There are two ways data can be transferred once the tag and reader reach a close proximity. The tag can obtain the transmitted message from the reader through a technique called inductive coupling. This technique involves switching on and off the load in the tag's antenna which affects the reader's antenna altering the power consumption. A voltage is induced due to the reader's coil and by changing the power consumption, a voltage drop can be measured. The voltage drop will be read as ones and zeros; thus, data is transferred from the transponder to the reader. Inductive coupling is a near field effect; the tag and reader must be about 1-15 centimeters away from each other. Additionally, this technique operates at low RFID frequencies typically below 13.56 MHz.

The other method is capacitive coupling which uses electric currents instead of electromagnetic fields. Capacitance is present between the tag and the reader through which a signal can be transmitted. Alternating current generated by the reader is detected and rectified within the transponder and powers the devices

within the tag. To return data, the load is modulated to successfully reach the RFID reader. The range of using this technique is less than of the inductive coupling. Both the reader and tag must be within 1 centimeter of each other. The project design will use either technique, based on testing results and compatibility.

4.4.3 - Wireless Provisioning

Since this device is a headless unit it will have limited physical user input, it is then necessary to use a different method for a user to easily configure wireless credentials when they first set up their device. Often times this is referred to as wireless provisioning and can be done using many methods.

Keypad

A simple to create, but very difficult for the user, provisioning method would be one that utilizes the keypad planned to be included in our design. Since a traditional keypad includes groups of three letters in addition to numbers, theoretically, instructions could be included to describe the user how to enter a complex string for the wireless SSID and password, similar to texting on keyboard-less cell phones. Though this would be time consuming, the setup process is only required once or whenever the device is required to change networks. An advantage of using the keypad would be that the user may manually tell the device to go into provisioning mode when they know they need to reset the wireless credentials. As opposed to a program on the device guessing if it needs to go into provisioning mode when it cannot find the wireless network.

Access Point

A widespread method for provisioning Wi-Fi devices is to act as a traditional Wi-Fi access point. In this method, when in provisioning mode, the device broadcasts a Wi-Fi network alongside the user's wireless internet connection. This separate Wi-Fi network would have an SSID which uniquely identifies it to the user. Using any Wi-Fi enabled device, the user would then:

1. Connect to the device like they would a normal access point
2. Visit a simple webpage to enter the credentials of the Wi-Fi they want the device to connect to
3. Submit the credentials to the device allowing it to go out of provisioning mode and connect to the specified network
4. Reconnect to their existing wireless network
5. Verify their device is present on the existing wireless network

This method could be considered difficult for the user since they are required to disconnect from their Wi-Fi and reconnect after visiting a webpage. However, this method is often implemented in apps where the application handles all transfers between different Wi-Fi networks. Despite this application-based method, the user is still temporarily disconnected from their original network which provides an internet connection.

Provisioning with Bluetooth

Another common method for provisioning is adding a second wireless communication method which does not require prior configuration such as Bluetooth. Bluetooth removes the need for the user to disconnect from their current network or use an app. Though an app-based solution is still possible with Bluetooth. However, on smaller single MCU systems, like our design, there is only enough computing power to manage one network interface at a time. As a result, the Bluetooth connection would have to be terminated to attempt a connection to the Wi-Fi network. If the provided network credentials were incorrect, the user would not know since Bluetooth could not provide a confirmation without reconnection. Ignoring some of these technical difficulties, Bluetooth is often chosen because it is the easiest method for the user.

Texas Instruments SmartConfig

An alternative to these traditional methods, Texas Instruments, TI, offers a simple prebuilt application that utilizes access point provisioning to easily and securely configure one of their chips. Alongside example code useful to a developer, this system employs their own custom mobile app or self-hosted web interface that is available on any wireless device. For security, they employ 128-bit encryption and automatically delete any previous Wi-Fi profiles to ensure the proper network is connected to every time. After the initial connection to the TI device, SmartConfig will automatically use current Wi-Fi settings for the TI Wi-Fi profile and after confirming connection, redirect the user over their original network to the TI device on board application. If employed in a final product, SmartConfig is a well-polished application for both a developer or general user.

5.0 - Project Hardware & Software Design Details

In this section, the design details will be discussed in terms of the hardware used and how the system will be programmed to configure the hardware. The subsections that will be discussed are the microcontrollers considerations, Wi-Fi Modules, temperature sensors, power supply, batteries, and barcode scanner modules. One of the important features for many electronic devices is the ability to save battery life while still operating as intended to. The initial method of powering the Black Box is by using a series of batteries. All of hardware must be able to have low-power functionalities to extend battery life as much as possible. Since this project uses batteries, the power system design should not be too complicated and will use DC-DC converters to connect power to the PCB. The PCB will have to be designed intricately, however, since it will house all the hardware. The microcontroller section considerations section will provide details about potential microcontrollers that fit ideally to the Black Box design. It will need to have many pins that serve as serial communications for modules to be added. Additionally, the different variety in serial communications such as an MCU having multiple UART, I2C, and SPI support is necessary because it is too early to decide which sensors, with a given serial communication support, is going to be chosen.

5.1 - Initial Design Architecture

One of the most difficult things about designing electrical schematics is the sheer amount of parts there are to choose from. Many different semiconductor or parts companies provide various integrated circuits and other components that are mostly identical. Therefore, it is necessary that a good electrical or computer engineer know how to sift through all these components in order to find the right one for the task.

5.1.1 - Microcontroller Considerations

The microcontroller is the main component that allows for integration between modules. The Microcontroller will be used for the monitorization of the sensors that will be linked to a microcontroller intended for the purpose of sending messages to the mobile device and also back to the database to store its status and temperature readings.

When selecting a microcontroller it is important to select one with the most user friendly chip and also one that is not too small in memory size or clock speed, which may limit some of the sensors read speeds and also input output processing speeds due to the lack of memory and speed for processing some of the task we give it . Some tasks for the microcontroller include connecting and getting data from the Wi-Fi module allowing for a connection to our Mobile application. The microcontroller will also connect to the deadbolt giving the signal to open or close after the verification of the passcode.

Table 14 - Microcontroller Consideration Table

	Description
1.0	Power Consumption for efficiency
1.1	Easy to learn and support for their product
1.2	Input and output ports
1.3	Cost
1.4	Technical Specifications

Table 14 above describes the information we looked for when considering the proper microcontroller for the Black Box. The table shows what will rank as in the most import from ranging 1.0 to 1.4 with 1.0 being the least important all the way to 1.4 being the most important. The top 2 of the most important aspect is the cost

and also the technical specifications of the microprocessor we picked, we wanted low cost along with the highest performance.

CC3220x Series

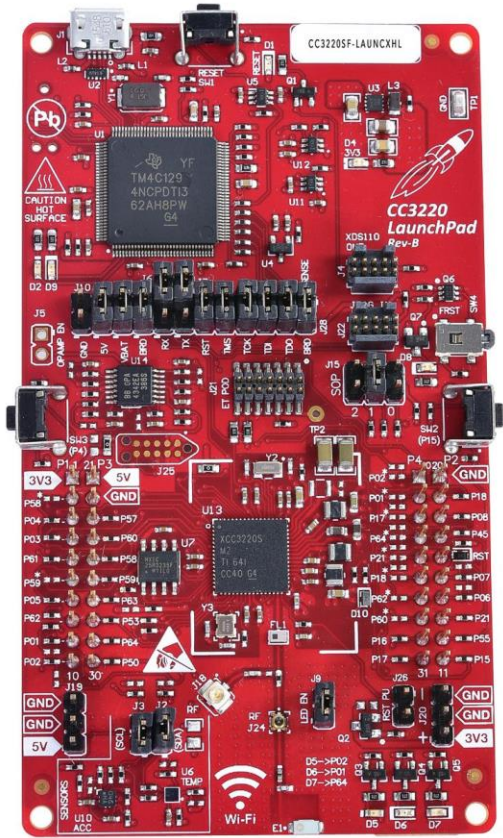
These are the microcontrollers considered for this project. The basic benchmarks for considering these MCUs all follow the need for great clock speed, RAM and program memory size, serial communication flexibility, and low-power modes.

As shown in Table 15, the CC3220x series has a dual-core architecture, highly-integrated Wi-Fi Network Processor and a User-Dedicated Wi-Fi Network Processor. It offers a variety of IoT security features including enhanced IoT networking security, asymmetric keys and unique device identity, software IP protection and secure storage. For battery purposes, the MCU series grants advanced low-power modes and a built-in power management subsystem which is essential for our project. Furthermore, the CC3220x series is part of the SimpleLink MCU platform, which is a common, user-friendly, development environment that supports Wi-Fi, Bluetooth, and host MCUs. These microcontrollers serve as a great application to video surveillance, video doorbells, low-power camera, and building security systems including smart locks, as shown in Figure 17.

Table 15 - CC3220x Series Basic Specifications

Specification	Value	Interest Summary
Main Clock Speed	80 MHz	Great Speed
Data RAM Size	256 KB	Good Memory Size
Program Memory Size	1 MB	Might need large memory for complex code
Communication Peripherals	SD, SPI, I2C, UART	Provides multi-purposed communications
Clock Source	40.0 MHz crystal with internal oscillator 32.768 kHz crystal External RTC	Allows sharper controls for clock speed matching
Other Technical Specs	Built-in Wi-Fi Processor Built-in DC/DC converters Advanced Low Power Modes Built-in Camera Sensor Crypto Engine	A built-in Wi-Fi capability is ideal

Based on the datasheet, the advanced low power modes allow the CC3220x to draw in only 1 microamps to 4.5 microamps in shutdown and hibernate modes, respectively. These modes can serve a purpose for when the Black Box is waiting for a package delivery. It can exit the low power mode once it is being interacted with. A noticeable consequence to this is the difficult implementation of this feature within the software. It must be able to access and exit modes properly in order for the clock sources to remain stable and not cause unwanted behavior.



Additionally, the built-in power management subsystem includes integrated DC/DC converters which supports a wide range of supply voltage. This is ideal since the Black Box will be powered by regular AA batteries. It also provides ease for the hardware designer since voltage can be regulated without the need of an additional circuit to compromise that function.

Similar to most TI microcontroller products, this device makes great use of pin multiplexing. This allows the CC3220x to accommodate the large number of peripheral functions in its' small frame. A combination of register control and hardware configuration controls pin multiplexing; thus, the software designers are responsible for the correct pin configuration. In total, there are 64 pins that we can use for our liking.

Figure 17- Texas Instrument CC3220 Development Board (Permission to reproduce submitted)

the C3220SF. In that order, the level of complexity and additional features increases making the C3220SF the most powerful of the three.

The CC3220x series comes in three variants: the CC3220R, CC3220S, and

- CC3220R - Grants 256 kilobytes of RAM, contains internet of things networking security, and has device identity/keys
- CC3220S - Uses the features of CC3220R and builds upon the MCU level by including a file user IP (MCU image) encryption, file system encryption, and secure boot and debug security.
- CC3220SF - Builds on the CC3220S by integrating a user-dedicated one Megabyte of executable flash

Based on these descriptions, the CC3220SF has been considered, granted that it contains every function that the others have with an additional memory coming from the executable flash. The extra memory may be needed due to how complex the code may be when considering all the functions that the Black Box must be able to do. Figure 18, below summarizes all the basic functions of the CC3220SF MCU.

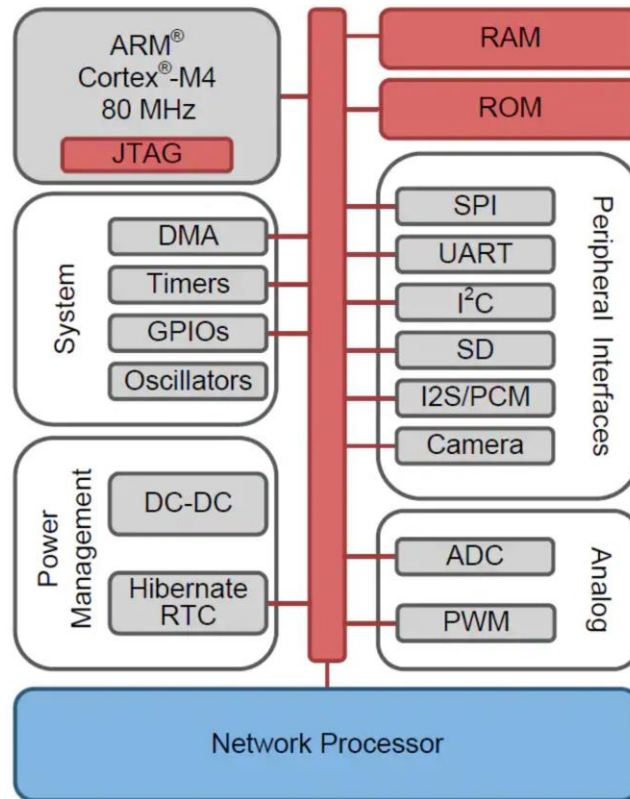


Figure 18 - Functional Block Diagram (Permission to reproduce submitted) [Mous]

MSP430x Series

The MSP430 series are known for their general-purpose usage, low-cost, and low power modes. Their forte excels in sensing and measurement applications such as heat and gas flow measurement, capacitive touch buttons, and more. Additionally, this series has many MCUs to choose from, giving many options to satisfy a specific need. Since the MSP430x series focuses on sensing and measuring, this line of MCUs can be considered for the thermal sensors needed for this project.

Table 16 below summarizes the basic specifications of the MCU line. The value line and general purpose of the MCU will include a 12bit digital to analog converter. Also the MSP 430 series has features built in for support with capacitive touch sensing microprocessors. This is an ultrasonic sensing analog front end with a less than 3uA wake up touch power. The current draw during idle mode can be less than 1uA, and most of this MSP430 family CPU is 25MHz, it can be throttled

for lower power consumption. There is six low -power modes to conserve the amount of power draw.

Table 16 - MSP430x Series Summary

Specification	Value	Interest Summary
Main Clock Speed	16 MHz	Decent Speed
Data RAM Size	0.5 to 256 KB	Good Memory Size
Program Memory Size	0.5 KB to 128 KB	Small room for code implementation
Communication Peripherals	SPI, I2C, UART	Provides multi-purposed communications
Clock Source	High Frequency Crystal 32.768 kHz crystal Low power low frequency internal clock (VLO)	Provides options for clock speed matching
Other Technical Specs	Integrated LCD driver Advanced Low Power Modes IR Modulation ADC channels 16 to 100 pins	Many hardware to choose from providing flexibility to fulfill a given need

Since there are many MCUs to choose from in this line, a few microcontrollers are considered.



MSP430FR6989

The MSP430FR6989, shown in Figure 19, is a microcontroller that is part of the FRAM line with essential points on low power modes and quick response times. It contains a non-volatile memory of 128 KB integrated within the chip. Key features that are included are five 16-bit timer modules with up to seven capture and compare registers, three-channel internal direct memory access, 32-bit hardware multiplier, extended scan interface (ESI) for background, water, heat, and gas volume measurement, and many serial communications. The serial communications it supports are UART with automatic baud-rate detection, SPI, and I²C with multiple-slave addressing. Many low power modes exist in this MCU and are optimized, giving the user many options to choose a specific mode for a certain task. In this case, a low power mode can be chosen such that the thermal sensor will only be on when a package is inside of the enclosure. If the timer runs on ACLK and a flag is raised (indicating that the

Figure 19 - MSP430FR6989 sensor is on), low power mode 3 will be needed to draw the least amount of power

since it shuts down all the available clocks except the ACLK. During active mode, the MCU draws in about 100 microamps per MHz at standby mode, the average currents it draws is about 0.4 microamps while in shutdown mode, the average current draws about 0.02 microamps. The main clock speed on its highest mode is 16MHz, it is very efficient MCU for its size. The programming memory size provided too little space for multiple sensors and devices given it only contains a 128kb programming memory size. With many low power modes to choose from, the programmer would have to choose and appropriately use one of them to satisfy the Black Box's need.

Table 17 - MSP430 Specifications

Specification	Value	Interest Summary
Main Clock Speed	16 MHz	Decent Speed
Data RAM Size	2 KB	Good Memory Size
Program Memory Size	128 KB	Small room for code implementation
Communication Peripherals	SPI, I2C, UART	Provides multi-purposed communications
Clock Source	High Frequency Crystal 32.768 kHz crystal	Provides two options for clock speed matching
Other Technical Specs	Extended Scan Interface for heat measurement Integrated LCD driver with Contrast Control Many low power modes	Additional hardware provides flexibility and utility

Table 17 above shows the specifications of the MSP430 that we've taken into consideration for the Black Box. The MSP430FR6989 is an MCU that was used in previous coursework taken by all of us; thus, familiarity with this product is a bonus. We are all experienced with the various timer module modes such as continuous and up mode as well as configuring interrupts. These concepts are important for setting up future installments to the Black Box.

ATMega2560

The ATMega2560 can be viewed as a multipurpose chip for our project. It offers more memory size than the MSP430FR6989 and the communication protocol it supports is SPI. The main difference between the two microcontrollers is the available libraries for reference materials and the community support that the ATMega2560 has versus the community for the MSP430. Many people use the ATMega2560 device in many ways, which gives it a boost in being user-friendly for execution. Table 18 summarizes the technical details of this MCU.

Table 18 - ATmega2560 Specifications

Specification	Value	Interest Summary
Main Clock Speed	16 MHz	Decent Speed
Data RAM Size	8 KB	Good Memory Size
Program Memory Size	256 KB	Small room for code implementation
Communication Peripherals	SPI	Only has one communication protocol; not much flexibility
Clock Source	8.0 MHz RC oscillator 128kHz RC oscillator Various clock speeds	Provides many options for clock speed matching
Other Technical Specs	Many sleep modes supported Two low power modes Supports PWM JTAG	Additional hardware provides flexibility and utility

Table 18 describes the data overview for the ATmega2560. The main clock speed is relatively fast with 16 MHz processing power. This will allow us to have a quick and swiftly computing power when the various peripherals that will all be in using during the operation of the box. The data ram size can be better however with the operations we are going to implement, and by seeing the performance data this is a very perfect ram size for our project. I would not be very great if more sensors and components are added such as a camera that will require a lot more ram storage. The programming memory size is very decent for the application we will do for our project, we have an overall idea that it should take no more than 10,000 lines for all the peripherals we have for our project. What is best about this ATmega2560 is that there is many power modes for this microprocessor, there is a standby mode where it only draws very low amount of power to reduce the battery size we will need for the longevity of the device, and also the user friendliness of the user not having to replace batteries frequently.



Figure 20 - ATmega2560 Development Board
(Permission to reproduce submitted)

In Figure 20 the ATmega2560 Development Board It contains many sleep modes which reduce power consumption by shutting down unused modules in the

microcontroller. Having multiple sleep modes provides flexibility in choosing which modules should turn off/on which greatly impacts power consumption. Going further in to saving battery power, the ATmega2560 has two ultra-low power modes:

- Active Mode: 1.8V: 500 microamps @ 1MHz
- Power-down Mode: 0.1 microamps @ 1.8 V

Some downsides to using this MCU is the limited communication protocols it has. Only being limited to SPI provides little flexibility and will have to explore more options if other forms of serial communication is required. Additionally, the voltage range is smaller than the other two mentioned. It has a voltage range of 4.5 Volts to 5.5 Volts at 16 MHz

ATmega1284

The ATmega1284 is a very low power 8-bit microcontroller based on the AVR RISC architecture. This microprocessor makes a good contender to our project due to its low power draw and its 128 Kilo Byte flash memory. There are 2 serial USART which enables you to use either UART or USART compatibility. There are only 3 SPI interface and one I2C along with 32 programmable Input and Output lines. The programming instructions are stored into non-volatile flash memory, although the Mcu are all 8 bit each instruction takes 1 or 2 16bit words.

The ATmega1284, with consideration, there is a problem with the number of SPI connections on this microprocessor for us consider another Atmel product. The ATmega1284 is JTAG compatible it is also IEEE 1149 compliant interface. The peripheral features contain two eight-bit timers and counter with separate pre-scaler and compare mode. Also, it contains two sixteen timer and counters with separate pre-scaler, compare mode, and capture modes. There is a very useful two programmable serial USART communication pins for most sensor devices and 2 is more than enough for the project objective.

The special features also include a power on reset and programmable brown out detection.

The ATmega1284 also include many modes to conserve power, it has built in six sleeping moves, idle, ADC Noise Reduction, Power save, Power down, Standby, and lastly Extended Standby. This feature provides us to maximize our battery life on our Black box device. There is 32 Programming Input and output lines, 40 pin PDIP, 44 Lead TQFP and 44 Pad VQFN/QFN. By reducing the amount of power draw the ATmega1284 also provide us a feature that has speed grades along with an average power needed, from 0 to 4MHz the power can run between 1.8v to 5.5v, from 0 to 10 MHz the power can run between 2.7v to 5.5v , and lastly for the final speed grade which is the fastest speed at 0 - 20 MHz the power can run between 4.5 to 5.5v.

This chip is very low cost which will give us the utilization to spend more money in many other places such as sensors and the overall appearance of our Smart lock box. This microprocessor is also the its most used board with the Arduino microcontroller brand boards. The key features are that the it is a flash memory along with a limited amount of program memory size very comparable to the MSP430FR6989. For the comparison with the MSP430FR6989 this chip shows more potential with its size and also forum support if we do have problems on implementing this with our design in the future of this project.

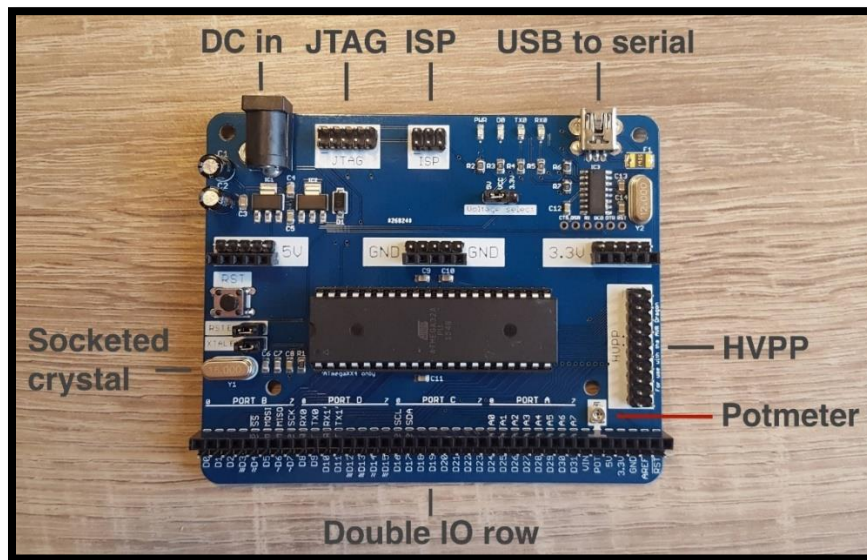


Figure 21 - ATmega1284 Development board (Permission to reproduce submitted)

Figure 21 shows the standard developers board for the ATmega1284. It is a very common board popularized by the Arduino brand. There is also a 5.0volt dc to dc voltage regulator for many sensor component applications. Also, the board comes with a 3.3-volt dc to dc regulator on the opposite side to reduce the heat dissipated from the dc to dc conversion. There is a connected HVPP which is a high voltage protocol for any ATmega microprocessor part. The HVPP as shown has a support for 19 signals along with the consideration of power and ground. The timing of the power pin matters and the software controlled through SVCC. As shown the board also has JTAG support which allows us to use and test access of printed circuit boards using the boundary scan, this can help programming the microprocessor and it is a way to communicate with each other in the outside world. In order to run any boundary scans it is necessary to have some implementation with JTAG enable devices on the connected board. It shows the Socketed crystal shared in most Atmel microprocessor supported boards from Arduino, along with a USB to serial communication for computer connectivity.

Table 19 - Microprocessor Comparison Chart

	ATMega 1284	ATMega 2560	MSP430FR6989	CC3220SF
Program Memory Type	Flash	Flash	Flash	Flash
Program Memory Size	128 KB	256 KB	128 KB	1 MB
CPU Speed (MIPS/DMIPS)	20	16	16	80
SRAM Bytes	16,384 KB	8192 KB	2 KB	256 KB
Data EEPROM/HEF	4096	4096	N/A	N/A
Digital Communication Peripherals	2-UART 3-SPI 1-I2C	4-UART 5-SPI 1-I2C	2-UART 4-SPI 2-I2C	2-UART 4-SPI 2-I2C
Capture/Compare/PWM Peripherals	1 Input capture, 1CCP, 6PWM	4 Input Capture, 4CCP, 16PWM	7 input capture, 4CCP, 16PWM	10 input capture, 4CCP, 16PWM
Timers	2X8 Bit or 2X16 Bit	2X8 Bit or 4X16 Bit	5 X 16bit timers	5 X 16bit timers
Number of Comparators	1	1	16	16
Temperature Range (C)	-40-85	-40-85	-40-85	-40-85
Operating Voltage Range (V)	1.8 - 5.5	1.8 - 5.5	1.8 - 3.6	2.1 – 3.6
Pin Count	44	100	83	40 pin

Table 19 shows the comparison chart between the differences of the microprocessors we researched. This chart is very helpful for us to choose the microprocessor we will use, this is attached to show the reason why we chose the **ATMega2560**.

The **ATMega2560** is a perfect fit for our project because it is 256kb program memory size, more than enough for what we need. Also, it has 4 UART communication pin out. It gives us more components to use, when most of our

sensors and parts may use UART. Along with this the ATmega2560 is also the one with the most amount of SPI communication pin out.

5.1.2 - Wi-Fi Module Considerations

A Wi-Fi module is a SoC (system on chip) devices that allows control and management functions on traditional serial devices and MCUs that lack Wi-Fi capabilities. Looking at it from a hardware point of view, data is transmitted and received via radio frequency using technologies such as MAC, microcontrollers, RF, and baseband. Viewing it from a software perspective, a Wi-Fi module implements the 802.11 standard protocols such as transmitting/receiving packets, gaining access/allowing access to networks, etc.

Wi-Fi capabilities are a necessity for the Black Box Design as it will allow the project device to connect to a Wi-Fi network and provide useful information to the user about package delivery details. The Wi-Fi modules considered in this section will be analyzed with the following benchmarks:

- Serial Connection Peripherals
- Low-Power Modes for optimized battery life
- Prices and delivery options
- User-friendliness

ESP8266EX

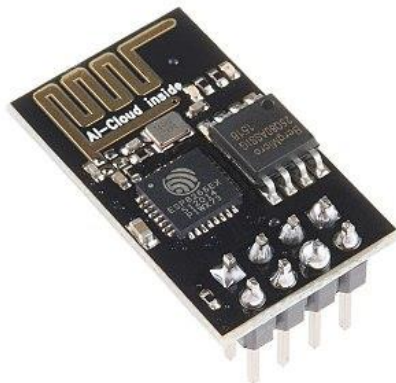


Figure 22 - ESP9266EX Wi-Fi Module (Permission to reproduce submitted)

This Wi-Fi module, as shown in Figure 22, is a popular module that is suitable for our project for microcontrollers that do not have a Wi-Fi module embedded into it. Due to its popularity from both professionals and hobbyists, a ton of online support is existing for troubleshooting making it more user-friendly. Additionally, the device comes with a Software Development kit which provides sample codes to many applications.

It can be operated as a standalone application or can serve as a slave for a host MCU. Since this Wi-Fi module is chosen for the purpose of integrating it with an MCU, the ESP8266EX can be applied to it by using SPI or UART interfaces.

ESP8266EX 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 ESP8266EX has a Wi-Fi support infrastructure that can be configured in three modes:

- Station Mode
- Soft Access Point Mode
- Station Mode + Soft Access Point mode

In station mode, the ESP module gets connected to a Wi-Fi network established by an access point. Station mode grants many features that manage Wi-Fi connectivity. For a case when connection is lost, the module will automatically re-establish a connection used by the last access point when it is available. This is done by saving the previous authorization used from the last used access point to its flash memory.

For coding purposes, a begin function is used switching the module to station mode. This function needs access to common restrictions such as the service set identifier (SSID) and the password to connect to the specific access point.

In soft access point mode, the module provides access to Wi-Fi to other devices. These devices that connect to the module can be described as stations, similarly, to when the ESP module is set to station mode. In general, modules that have an access point functionality can connect stations further by using a wired network. However, for the ESP8266EX Wi-Fi module, it does not have to be interfaced with a wired network. This is fundamentally what soft access point (soft-AP) is. The number of stations that can connect to the soft-AP is a maximum of five.

For coding purposes, only one parameter is needed for the soft-AP function to work. In this case, the SSID is needed while other parameters such as password, channel, and hidden are optional.

Station Mode and Soft Access Point Mode is a Wi-Fi supported infrastructure that combines the functions of the two previously mentioned modes. Essentially, the ESP8266EX can connect to a Wi-Fi network and provide network access to other devices simultaneously.

The clock can run at a maximum speed of 160 MHz using a Tensilica L106 32-bit RISC processor. This type of processor helps achieve lower power consumptions while still having a relatively high clock speed.

For memory cases, the ESP8266EX holds a RAM size of about 50 KB. Unfortunately, there are no accounts of programmable ROM in this chip; thus, the

user program must be stored in an external flash using SPI. In this case, the SPI flash can support up to 16 MB.

The data sheet exhibits all the pins that this Wi-Fi module has. It contains 17 GPIO pins which can be selected for numerous functions with proper programming of the appropriate registers. ESP8266EX has the three common serial communication peripherals:

- SPI - Pins 18-23 (Slave/Master SPI), Pins 9-10, 12-13 (Slave SPI)
- UARTs - UART0 (used for communication) Pins 12-13, 25-26
- UART1 (used for printing log) Pins 14 & 23
- I2C - Pins 9 & 14

Power management options are available for this Wi-Fi module. Its' low-power architecture engages in four low power modes:

- Active mode - ESP8266EX is fully operating
- Modern-sleep mode - CPU is operating; Wi-Fi and radio are disabled
- Light-sleep mode - CPU and all peripherals are paused; interrupts such as MAC, host, RTC timer, or external wake-up events will cause the chip to awaken
- Deep-sleep mode - Only the RTC is operational; everything else is powered off

Table 20 - Power Consumption Summary - ESP8266EX

Power Mode	Current Drawn
Active Mode	56-170 mA
Modern-sleep	15 mA
Light-sleep	0.9 mA
Deep-sleep	20 μ A

As shown above in Table 20, with the many options to choose from in power management, this Wi-Fi module provides flexibility in reducing power to fully optimize battery life.

Overall, the ESP8266EX contributes to the MCU greatly by granting it excellent Wi-Fi capabilities with many options to choose for connection. The extra modes in saving power is a huge bonus too. With a small price averaging about \$2.50 a chip, this module is worth considering.

CC3200MOD



Figure 23 - CC3200MOD Wi-Fi Module

The CC3200MOD, displayed in Figure 23 above, is a powerful Wi-Fi module created by Texas Instruments. It accommodates an ARM Cortex-M4 MCU giving the user the ability to create and develop their own application.

Similar to the ESP8266EX, its subsystem contains an 802.11b/g/n radio, baseband, and MAC technologies along with a fast-crypto engine that secures internet connections. Additionally, it can be configured to station, access point, and Wi-Fi Direct mode, which is unique to this chip when compared to the ESP8266EX.

Wi-Fi Direct certified devices enable Wi-Fi devices to directly connect to each other without joining a home, office, or hotspot network. For application examples, mobile phones, PCs, and gaming devices can connect to each other directly to transfer any content or similar applications.

The data sheet displays all the available pins for either a specific or general use. Not only there are many pins offered in this chip, the CC3200MOD takes advantage of pin multiplexing to consist of more peripheral functions. To distinguish a peripheral configuration, a combination of hardware configuration and register control is necessary.

Pin multiplexing in this chip can be demonstrated by how the user chooses which serial communication he/she is needing. Pins GPIO10 and GPIO11 have an I2C and UART function; thus, it is up to the user to choose which serial communication to use by choosing the correct register control for the corresponding peripheral interface.

CC3200MOD has various low-power modes for the user to choose from. Below is a list of modes and its' description that this module can operate in and Table 21:

- Active Mode - The Wi-Fi module is executing its' programmed code at 80-MHz state rate
- Sleep Mode - The Wi-Fi module's clocks are gated off while the entire state of the device is maintained. This mode can be instantly woken up by

interrupts. These interrupts can be either the internal fast timer or from GPIO line activity.

- Low-Power Deep Sleep - Specific register configurations are retained; however, state information is lost. This can be interrupted by external peripherals or the timer. Only certain parts of memory can be retained.
- Hibernate Mode - All digital logic is power-gated. Real-time clock is running, and the module can be woken up by interrupt events described in the other low power modes.

Table 21 - Power Consumption Summary - CC3200MOD

Power Mode	Current Drawn
Active Mode	15.3 - 278 mA
Sleep Mode	12.2 - 275 mA
Low-Power Sleep Mode	0.875 - 272 mA
Hibernate Mode	7 μ A

In terms of user friendliness, the datasheet provides a recommendation for pin multiplexing configurations for a desired application. For example, if the user were to use the CC3200MOD Wi-Fi module for designing a home security high-end toy, the datasheet contains a table detailing the specific pin locations and numbers, and specific serial communications needed. Additionally, since this is a TI made product, there are resources online that the user can ask from experts.

Overall, the CC3200MOD Wi-Fi module is a great chip for granting an MCU, Wi-Fi capabilities. It does its' job and more, making it versatile for the user to experiment around with. Because of the extra features and the fact that it is design by Texas Instruments, the chip is quite expensive but powerful. One drawback from having those extra features/peripherals is that it is not needed for this project. For example, the Wi-Fi direct mode feature may be nice to have; however, the Black Box simply needs to connect to Wi-Fi. Furthermore, having those extra peripherals might be a reason to why the power consumption is higher than most Wi-Fi modules, making it less battery-efficient.

Wizard Gecko WGM110

Figure 24 below shows the WGM110 chip. The Wizard Gecko is similar to our other options; however, more details are given below.

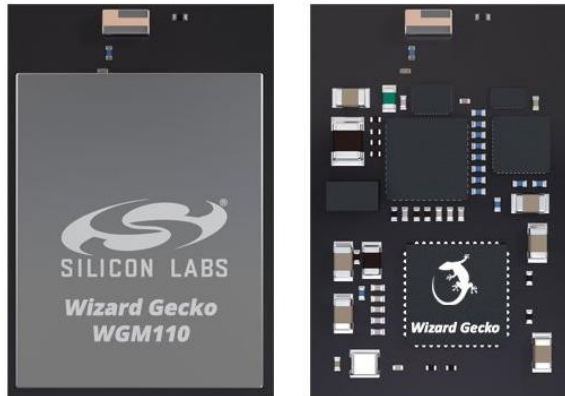


Figure 24 - Wizard Gecko WGM110 Wi-Fi Module (Permission to reproduce submitted)

The Wizard Gecko WGM110 Wi-Module is a chip created by Silicon Labs that excels in radio frequency performance, low-power consumption, and user-friendly application development. Standard Wi-Fi elements include the 802.11b/g/n radio, Wi-Fi and IP stacks, HTTP server, and multiple protocols such as TCP and UDP. Additionally, it has different variants that allow for external Wi-Fi protocols. These variants are:

- WGM110A (chip antenna)
- WGM110E (u.FL connector)

The range for Wi-Fi connection capabilities extends to 450 m. Additionally, this module can be configured to be a Wi-Fi client or be used as an access point. These configurations are a similar concept to the station mode and soft access mode mentioned in the ESP8266EX section.

It can be its own MCU or can be configured to be controlled by a host MCU. Since this module is going to be an addition to the MCU of our choice, WGM110 must be configured to a mode called Network Co-Processor mode. This allows the TCP/IP protocols to be left taken care of by the module while the host MCU can focus on processing only the user's application tasks. To connect this module to the MCU, it can be connected by using UART, SPI, or USB through the proper pins. Based on the datasheet, this chip takes advantage of using pin multiplexing as well. The peripherals can be configured into many locations for the user to decide, providing flexibility.

For low-power options, WGM110 comes with three options to choose from:

- Powered off state - CPU can either be executing the program, remain idle, or in the deepest powered down state
- Sleep unassociated state - CPU and Wi-Fi chip are in sleep state
- Associated idle state - 100 ms beacon period

Table 22 - Power Consumption Summary - WGM110

Power Mode	Current Drawn
Continuous Transmit	81 - 261 mA
Powered Off	23 μ A - 169 mA
Sleep, unassociated	160 μ A
Associated, idle	1.2 - 2.5 mA

Final Wi-Fi Module Analysis

The three Wi-Fi modules that we felt are worth considering are the ESP8266EX, CC3200MOD, and the WGM110 modules. Each of them has the basic requirements needed for the Black Box; however, one of them stands above all. In Table 23, comparisons of each Wi-Fi module are summarized.

Table 23 - Final Wi-Fi Module Comparison

	ESP8266EX	CC3200MOD	WGM110
Serial Interface(s)	SPI, UART, I2C, I2S	UART, SPI, I2C	I2C, UART, SPI, USB
Baseband Support / Frequency Support	802.11 b/g/n 2.4 GHz	802.11 b/g/n 2.4 GHz	802.11 b/g/n 2.4 GHz
Max Data Rate	54 Mbps	54 Mbps	72.2 Mbps
Voltage Range	2.5 to 3.6 V	2.3 to 3.6 V	2.7 to 4.8 V
Current Drawn Range (From lowest power mode to active mode)	20 μ A - 170 mA	7 μ A - 278 mA	23 μ A - 261 mA
Price (Digi-Key)	\$1.85	\$15.50	\$16.86

ESP8266EX is the Wi-Fi module our group decided to use for this project. There are more reasons to choose this module than the others. One noticeable reason is the price difference. The ESP8266EX can be ordered in multiple bundles while being cost effective. If the other Wi-Fi modules were to be ordered, it would become very expensive. Using the same money to buy one WGM110 Wi-Fi module is equal to about 10 ESP8266EX.

This brings forth the next reason in that the modules share similar characteristics in terms of their serial interfaces, baseband support, and operating frequency. Why spend more money on either the CC3200MOD or WGM110 when the ESP8266EX can basically operate in the same manner? There are differences in design; however, these differences do not outweigh the amount of money it would go in to purchasing them.

Other valid reasons for using the ESP8266EX are lower current drawn from usage which makes it slightly better in efficiency for battery life and even though the other

Wi-Fi modules have technical support, the ESP8266EX is more popular than them. This means that more demonstrations and online support are available through forums/videos. Additionally, Arduino IDE contains libraries for this module.

5.1.4 - Temperature Sensors

A temperature sensor is a device used for measuring temperature or heat through the use of electrical signals. These sensors play an important role in many applications and they need to excel in responsiveness and accuracy when it comes to heat detection for optimized quality control. For that reason, temperature detection can be seen as a preventative reliability, allowing the user to know what temperature he/she is operating in. The Black Box will need a temperature sensor to constantly measure the heat inside of it. This is to ensure that special packages such as medicine will be protected from damage as the temperature sensor will notify the user that it may be too hot for the package to be kept inside it.

If we use the operating ambient temperature of an iPhone as a standard for electronic devices, then we would have to keep in mind of that temperature range which is 32 to 95 degrees Fahrenheit [Ste19]. There are other types of electronics; however, this range can be generally used to provide proper feedback. For medicines, the temperature ranges will vary depending on the type of medicine. High temperatures will change drug uniformity and drug release in ointments, lose does uniformity in oral suspensions such as Pepto-Bismol and change dissolution time in antibiotics. Cold temperatures can prevent activations of the active ingredient in solutions.

Obviously, the Black Box is not made for the intention of storing products but for protecting them until the user gets it. However, it would be convenient to know what temperature the package is sitting in if the user is away and busy (such as working). This is to keep the user aware of the potential damage that can happen to their package if it is exposed to the extreme ends of the temperature range for prolonged periods of time. For consideration purposes, Table 24 describes key characteristics we are looking for in a temperature sensor:

Table 24 - Temperature Sensor Considerations

	Description
1.0	Temperature Range Measurement
1.1	Accuracy
1.2	Power Consumption for efficiency
1.3	Cost

STS3x-DIS

The STS3x Series is a board mount temperature sensor designed by SENSIRON. Its functionality includes enhanced signal processing, two distinguished and selectable I2C addresses, and a communication speed of 1 MHz max. The series comes with **three different sensors** offering options for the user to opt in. For consideration purposes, if a temperature sensor is chosen from this series, only **one** sensor will be used. The following table describes key characteristics we are looking for in a temperature sensor:

Table 25 - STS3X-DIS Comparison

	STS30-DIS	STS31-DIS	STS35-DIS
Accuracy Tolerance	±0.2	±0.2	±0.1
Temperature Measurement	0 - 65 °C	0 - 90 °C	20 - 60 °C
Voltage Range	2.15 to 5.5 V	2.15 to 5.5 V	2.15 to 5.5 V
Current Drawn (when sensor is measuring)	600 - 1500 microamps	600 - 1500 microamps	600 - 1500 microamps
Price	\$1.93	\$2.22	\$3.07

Table 25 above compares the three sensors that are in the STS3x Series lineup. Each sensor has different ranges for temperature measurement and the range for the STS30-DIS makes more sense for the Black Box to have. Zero to 65 degrees Celsius corresponds to 32 to 149 degrees Fahrenheit. The average weather temperature nationwide ranges from a low 26.6 degrees Fahrenheit to a high of 70.7 degrees Fahrenheit. The STS31-DIS has a higher temperature range than STS30; however, the extra range is not needed because weather temperatures cannot reach or approach 90 degrees Celsius (194 degrees Fahrenheit).

In terms of accuracy tolerance, the STS35-DIS is more refined than the others; however, the other sensors are still adequate. Only being ±0.2 degrees off does not make a huge difference as whole numbers for temperature are more important for our project.

With all comparisons in mind, the **STS30-DIS** fits better as a temperature sensor for the Black Box. Its' range is within the range we want to consistently measure, adequately accurate, it draws small current, and it is the lowest price of the three.

Keep in mind, that this temperature sensor on its own, will be compared to other potential temperature sensors offered from alternative companies.

TMP451-Q1

This temperature sensor is a highly accurate, low powered device designed by Texas Instruments. Its' temperature accuracy is $\pm 1^{\circ}\text{C}$ when placed under typical operating range. The interface it uses is I2C since it uses SDA and SCL lines for the SMBus communication protocol.

Operating voltages range from 1.7 V to 3.6 V and the operating temperature range ranges from -40°C to 125°C . For power modes, it consists of one low mode called shutdown mode that draws 3 microamps. While it is operating, the current drawn is 27 microamps.

Table 26 - TMP451-Q1 Specifications

TMP451-Q1	
Accuracy Tolerance	± 0.1
Temperature Measurement	$-40 - 125^{\circ}\text{C}$
Voltage Range	1.7 V to 3.6 V
Current Drawn (when sensor is measuring)	600 - 1500 microamps
Price	\$1.93

DHT11

The DHT11 is a humidity and temperature sensor typically seen in projects that involve an Arduino. Thus, this temperature sensor would be the easiest to interface with, if ATmega MCUs are used because of the libraries integrated within the Arduino IDE. In Table 27, the technical details are summarized.

Table 27 - DHT11 Specifications

DHT11	
Accuracy Tolerance	$\pm 2^{\circ}\text{C}$
Temperature Measurement / Humidity Measurement	$0-50^{\circ}\text{C} / 20-90\%$ RH
Voltage Range	3 V to 5.5 V
Current Drawn	0.2mA - 1mA
Price	\$2.00

With decent specs, user-friendliness, and a decent price, the DHT11 is a temperature sensor worth considering.

Final Temperature Sensor Analysis

The three temperature sensors that caught our attention are the STS30-DIS, TMP451-Q1, and the DHT11. With the specs given, Table 28 below compares all the specs and highlights which temperature sensor will be selected for the Black Box. With the following consideration with voltage compatibility for our board and the ease of implementing it with our design, the current draw was considered for the most Minimized current draw for efficiency for our battery and the user friendliness of our product.

Table 28 - Final Temp. Sensor Comparison

	STS30-DIS	TMP451-Q1	DHT11
Accuracy Tolerance	± 0.2 °C	± 0.1 °C	± 2.0 °C
Temperature Measurement	0 - 65 °C	-40 - 125 °C	0 - 50 °C
Voltage Range	2.15 to 5.5 V	1.7 to 3.6 V	3 to 5.5 V
Current Drawn (when sensor is measuring)	600 - 1500 microamps	3.0 - 27.0 microamps	0.2 mA - 1 mA
Price (Digi-Key)	\$1.93	\$2.48	\$2.00

Out of all the considerations, the **STS30-DIS** is chosen because of many reasons. First, it contains an operating voltage range that is far larger than the other two temperature sensors. This is important because it provides flexibility for the user to choose a voltage within the range to make the sensor operate.

Additionally, it is the cheapest of the three temperature sensors making it more cost effective. Other reasons include the great accuracy it has when compared to the DHT11 and the valid temperature range it measures when compared to the TMP451-Q1. The temperatures recorded by the STS30-DIS, are temperatures that give enough information that the surrounding area is either too hot/cold for a package to be in. A temperature range of -40 to 125 °C is too wide of a range making the sensor seem more auxiliary if anything. For these reasons, the STS30-DIS is the recommended temperature sensor for measuring temperature inside the Black Box.

5.1.5 - Power Supply

Picking a good battery or power system stems from knowing component requirements. Table 29 presents a simple summary of the ideal voltage and minimum available current for each device. Though lower or higher voltages and currents can be used by the device, these are the manufacturer recommended values. As such, they are good estimates for designing the power supply circuits and picking out batteries.

Table 29 - Subsystem Specific Power Requirements

Component	Voltage	Max Current
MCU	5V	200mA
Lock	12V	2A
Barcode Scanner	5V	135mA
Keypad	5V (Logical 1)	Passive
Indicators	2.2V	16mA
Environmental Sensor	5V	1.5mA
Door Switch	5V (Logical 1)	Passive

5.1.6 - Battery

Below are the types of batteries we've considered for the Black Box.

Rechargeable

Batteries which are able to be recharged can be advantageous as they hold a lower cost of ownership. Most people can agree that finding or purchasing standard replaceable batteries is frustrating. However, rechargeable batteries come with a "down time" when they do need to be recharged and also add a limited overall time to the lifespan of a product as they are usually not able to be replaced when worn out.

Lead-Acid

Most popular in high capacity solutions to this day, Lead-Acid batteries are usually seen in cars or computer battery backup equipment, as we "still have no cost-effective alternatives" [Batt]. Unique to our use case, lead-acid batteries are large and heavy. Since our product is meant to sit in one place and also prevent theft, a lead acid battery could be a good way to add weight to the product. However, the power requirements of our project are magnitudes smaller than what a typically Lead-Acid battery can supply making the large capacity unnecessary [Batt]. Therefore, we will look to other alternatives for batteries and power storage.

Nickel-Cadmium

Most popular for high performance applications, Nickel-Cadmium batteries are usually seen in devices which are regularly fully discharged such as AA or AAA rechargeable batteries. Also seen in other applications like airlines for their low temperature performance or devices which need to have fast charge times, Nickel-Cadmium batteries have been popular for good reason. However, with all this performance comes a penalty. Nickel-Cadmium batteries are plagued by a

“memory effect” [Batt2] which can severely cut the capacity of the battery if not well maintained. This “memory effect” can be prevented by often fully discharging the battery but this maintenance is not ideal for most end users. Since our design would be left to run until the batteries need recharging or replacing, Nickel-Cadmium batteries could be a good option for our smart lock box. Another disadvantage though, is that the typical cell voltage of these batteries is 1.2 volts, lower than that of the average cell [Batt2]. Therefore, we would need to add more cells for a higher nominal voltage which become an unnecessary added capacity.

Lithium-Ion

Most popular for Internet-of-Things type devices, Lithium-Ion batteries are commonly seen everywhere. From laptop and phone batteries to power tool batteries, Lithium-Ion has quickly become an industry standard. As a need for so many devices, many form factors have been created for this technology of battery varying from different sized “flat packs” to the industry standard 18650 sized cell. In comparison to Nickel-Cadmium batteries, Lithium-Ion are essentially an improved version.

Not only do they perform very similarly to Nickel-Cadmium in load characteristics and discharge rates, but they have “twice the average energy density” [Batt3]. In addition to these advantages, the single cell of a typical Lithium-Ion battery is 3.6 volts or three times that of its Nickel-Cadmium counterpart. A result of all these advantages, however, is that Lithium-Ion batteries are almost 1.5 times the cost of Nickel-Cadmium. All this performance packed with energy density also creates an unsafe operating range. For Lithium-Ion batteries, it is then required that extra circuitry be implemented to regulate their output voltage and limit current [Batt3]. Popular for their low user maintenance and good performance, it is easy to see why Lithium-Ion batteries the choice for many applications are.

Replaceable

Batteries which are meant to be replaced add a level of cost to the ownership of any device. Continuing to operate requires batteries be purchased occasionally by the user. However, this is not a big issue if the batteries only need to be replaced occasionally in the given lifetime of the device. The real advantage of replaceable batteries comes with the allowance of minimum downtime when charge is low. Instead of having to wait for a recharge, the user can instantly give the device new life. In addition to a completely new charge, fully replacing the battery extends the overall lifetime of the device since it is not subject to battery wear.

AA

AA batteries can come with many different internal chemistries. But for the sense of this discussion being aimed at rechargeable versus non-rechargeable, we will be focusing on the popular single use store bought alkaline batteries. The common discharge characteristics for a Duracell branded battery and Radio Shack branded battery are shown in Figure 25 [Powe] However, non-rechargeable lithium AA batteries have become popular too due to their expanded lifetime. Though slightly

more expensive, lithium-based non-rechargeable AA batteries have twice the current capability as their alkaline counterpart. If chosen, lithium-based batteries would be included for the user and recommended for purchase when time to change them comes. However, because this choice for purchase is up to the consumer, we will have to make sure alkaline batteries are still usable with our circuitry.

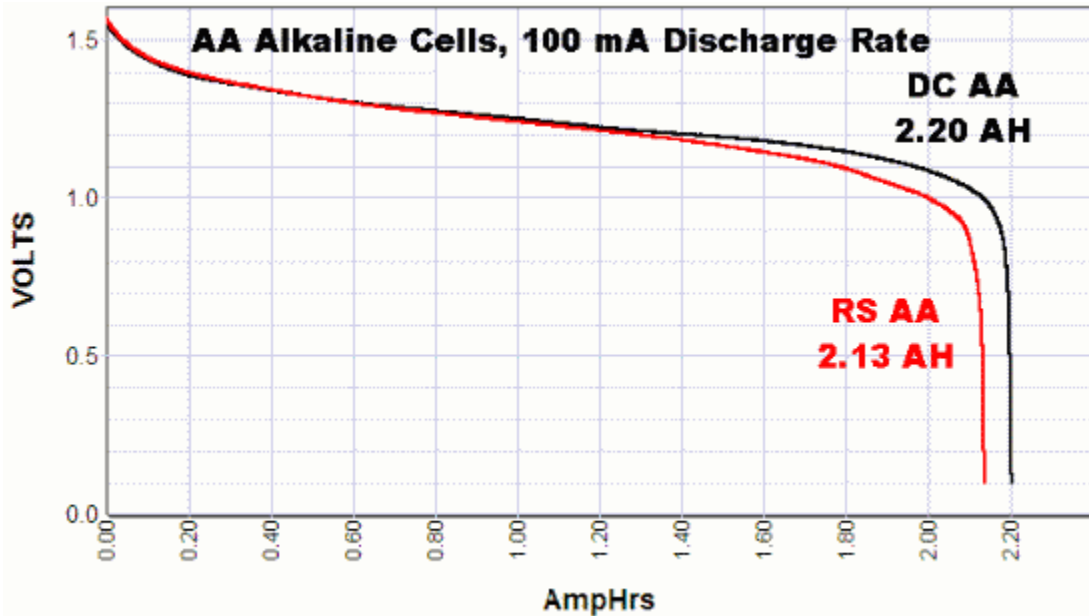


Figure 25 - Discharge Characteristics of Brand Alkaline AA Batteries (Permission to reproduce submitted) [Powe]

An ethical dilemma arises when considering disposable batteries. Disposal of single-use AA batteries has an equal amount of options as rechargeable batteries do. However, single-use alkaline batteries are not advised to be recycled, but rather just thrown in the regular trash to save time. This is due to an act passed in 1996 that made mercury use in alkaline batteries obsolete. It is now completely safe to dispose of the batteries in landfills. With the rise of environmentally friendly movements, a consumer may be off put by a device which uses single use non-recyclable goods.

AAA

The brother of the AA battery, AAA batteries come with the same characteristics in a smaller size. Still able to pack a nominal 1.5V, their smaller size is an indication of their smaller capacity. The usage of the two types of batteries varies, but most of the time AA batteries are used in devices that require more power. This can easily be seen by the consumer in the types of household items that require AA versus AAA batteries. Things like calculators or clocks don't use high power, and therefore usually use AAA batteries. Other things like electric toothbrushes require AA batteries, since the continuous turning of the motors against the resistance of teeth drains power much faster.

Since we have the interior space for AA or even bigger batteries, we will not consider AAA unless other concerns arise. Especially when comparing their availability not only in the store but around the house for most people. More and more devices are opting to fit AA batteries even if AAA would be more ideal. It is not wrong to think that the AAA form factor of battery could be obsoleted in years soon to come, especially in favor of rechargeable batteries for small devices like television remotes or calculators.

9V

When higher voltages than the standard 1.5V or 4.5V from other cells is required, 9V batteries become a good choice. Though common in stores, these batteries tend to be pricier compared to their AA or AAA counterparts making them less ideal for consumers. When total voltages greater than 12V are needed from standard store-bought batteries. Two 9 volts could be a better choice than a large series of 1.5V AA batteries. If our 12V locking mechanism required a regulated voltage, then we would need a supply voltage greater than 14V, and 9V batteries could be a good way to achieve this. However, their larger voltage means they have a smaller current capacity. Different brands and chemistries can have between 300mAh to 500mAh, sufficient for powering a microcontroller but not for actuation a lock which may require up to 2A of continuous power for a half a second. Due to this current limit, 9V batteries will no longer be considered as an option to power our device.

Choice (Rechargeable vs Replaceable)

Choosing a good battery system starts with knowing the requirements. For the most time, our circuit will draw a low amount of power. This means we need to look for a battery which can last for a long time with a constant small current draw. Though this is the case for a majority of the device's lifetime, in certain circumstances like the lock being opened or closed, the circuit may demand up to two or three amps from the batteries. This means we need a battery which can last a long time under small load but still be able to occasionally deliver a higher load on demand.

Of the battery technologies researched, replaceable is the obvious solution given our device is made to be permanently fixed in one place. Therefore, bringing it somewhere for charging is not ideal and bringing the charger to it is inconvenient. Furthermore, the downtime of the smart box must be as small as possible given that a package could be delivered at any time and the box needs to be able to protect this package all the time.

AA then follows as a good choice for battery format. The size is widely available in a number of chemical variants and easily understandable by the user. This format also optionally allows the owner to use rechargeable batteries that they can easily swap and recharge at their leisure. Since we require 12 volts for our locking mechanism, at least 8 AA batteries at a nominal voltage of 1.5 volts will be required. The voltage point where a battery is considered dead varies from

technology to technology, though for alkaline AA batteries, around 1.2 or 1.3 volts is considered fully discharged. For 1.2 volts, 8 AA batteries still add up to 9.6 volts. This is greater than the brown out voltage of 6V for the lock and the minimum input voltage of 7V for a standard LM7805 voltage regulator making it a good choice for powering our device.

5.1.7 - Battery Measurement

Since our device relies on constant operation and battery power, it is necessary to alert the user when the batteries are running low, so they can change them in a timely manner. To facilitate this, we will use one of the microcontrollers analogs to digital converter pins. The ATmega2560 features 16 different analogs to digital converter inputs for a resolution of 10 bits through successive approximation. The resulting 10-bit number is a value between 0000000000 and 1111111110 representing its difference between 0 volts, ground, for the lowest digital value or the reference voltage for the highest 10-bit value. The ATmega2560 provides internal reference voltages of 1.1V, 2.56V, or the externally applied VCC voltage. Since the lowest voltage of our battery will be much greater than this, we can scale down the input voltage with a voltage divider.

Doing so will allow us to be able to use the internal reference voltages versus the external reference, making it unnecessary to provide a more accurate external reference than VCC. The exact accuracy is trivial, however, as we are only attempting to tell when the battery is starting to become low which already has values that can vary wildly from battery to battery. This addition, though small, will make a large impact on the usability of our device.

5.1.8 - Barcode Scanner

A Barcode Scanner is a stationary input device that scans and reads a barcode containing important information. This technology is available and used all the time: making a purchase in a retail store, managing inventory, validating movie admission tickets, etc. Since it is used in a variety of industry fields, barcode scanners are made with diverse capabilities to fit the criteria. These include but not limited to:

- Pen barcode reader
- Laser scanner
- Camera-based readers
- Charged coupled device (CCD) readers

A typical scanner consists of three main parts, the sensor, the illumination system, and the decoder. It scans the black and white elements of a barcode by highlighting the code with a red light. The intensity of the light reflected from the light source is detected by the sensor and generates an analog signal that is sent to the decoder. It does this by converting optical impulses into electrical energy. Next, the decoder deciphers the analog signal, certifies the barcode by checking each digit, and translates it into text. Typically, the text is converted into ASCII Text which is

formatted and sent to a computer for interpretation. Figure 26 below depicts the inner workings of a barcode scanner.

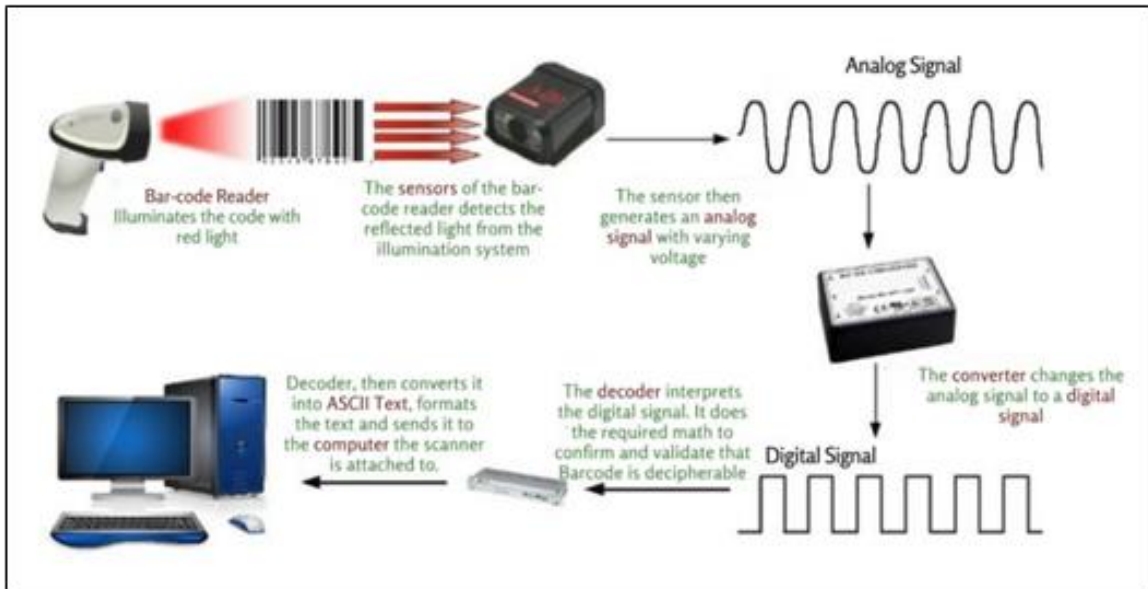


Figure 26 - How a Barcode Scanner Works (Permission to reproduce submitted)

For consideration purposes, each type of barcode scanner will be compared to how user-friendly they are, the costs between them, and accuracy of data scanned. It is important to minimize costs and maximize efficiency and this can be done by choosing a scanner that does enough of what it needs to do. Pen barcode readers look like a small wand stick that consists of an LED light and photodiode in its top. The user will pass the tip over a barcode in which the LED light will brighten. The photodiode will detect the amount of reflection of light based on the width and color of each bar. It is the cheapest of the four considered; however, it does pose problems in accuracy. Due to the shakiness of human use, the pen may scan errors in the data scanned.

Laser scanners work similarly to the aforementioned scanner. The difference is that a laser beam is shot at a mirror inside of it. The laser is able to sweep across the barcode in a straight line due to the movement of the mirror. The reflected light is measured similar to the pen barcode reader. Compared to the pen, it is more accurate and user friendly with a slightly higher price. Camera-based readers function like a video camera when compared to the others. It uses a video camera to take a picture of the barcode and is compatible with one-dimensional and two-dimensional barcodes. Additionally, camera-based readers can capture the barcode in any way the user places it in front of it.



Figure 27 - 1D (left) and 2D (right) Barcodes (Permission to reproduce submitted)

The downside to using this type of scanner is the price. Camera-based readers are more expensive when compared to the other scanners. Charged-coupled device (CCD) readers are known as LED scanners because they use hundreds of tiny LED lights arranged in a row like pattern. These lights are shot directly to the barcode. Instead of the sensors measuring the reflection of the light onto the black and white bars, the sensors in a CCD reader measure voltage of the ambient light directly from each light. Sensors use this voltage measurement to construct a digital snapshot of the barcodes. These scanners are highly accurate, but very expensive. After evaluating each type of potential barcode scanners, the laser scanner seems more reasonable to use out of the four. Many packages do not use 2D barcodes; thus, scanners with them are unnecessary. Additionally, the laser scanner provides good enough accuracy, user-friendly, and is offered in decent prices. However, many products offered in the market use a combination of the technologies mentioned. For example, laser scanners are not known for scanning 2D barcodes; however, there exists laser scanners that have this functionality. Either way, if a barcode scanner exists with the functionalities needed for the Black Box and is at a competitive price, that scanner is ideal.

Waveshare Barcode Scanner Module



Figure 28 - Waveshare Barcode Scanner Module (Permission to reproduce submitted)

The Waveshare Barcode Scanner Module, shown in Figure 28 is capable of decoding one-dimensional and two-dimensional barcodes on paper or screen with great accuracy. It is versatile in the sense that it can be plugged through its onboard USB and UART interface. Additionally, due to its small dimension of

53.3mm × 21.4 mm, the device can be easily integrated into types of devices. Table 30 describes the overall features of the Waveshare Barcode Scanner Module. A white light is used with a light intensity of 250 lux and can scan a typical serial barcode such as a Code 39 at a maximum distance of 25.0 centimeters. In general, the minimum distance required for scanning is about 6 centimeters.

Table 30 - Barcode Scanner Module Specifications

Specifications	
Operating Voltage	5 Volts
Operating Current	135mA
Standby Current	58mA
Sleep Current	2mA
Operating Temperature	0°C~60°C
Operating Humidity	5%~95%(Non-condensing)
Interfaces	UART, USB
Scan Angle	Tilt 360°, Skew ±65°, Pitch ±60°
Field of View (FOV)	28° (Horizontal), 21.5° (Vertical)
Dimension	53.3 mm × 21.4 mm

For setting up the barcode scanner module, the datasheet provides an easy method to choosing the user's desired configuration. To set the configuration, the user has to scan a QR code that represents that configuration. For example, to choose which type of serial communication to choose (in this case, it is either UART or USB), the user would have to scan either of the codes showing in Figure 29 below:



UART Output



USB PC Keyboard

Figure 29 - QR Code Settings (Permission to reproduce submitted)

By scanning these modes, it makes configuring the barcode scanner user friendly. Since an MCU is being used for our project, the UART serial communication is necessary. The default parameters of the interface are a baud rate of 9600 bps, data bit: 8, and a stop bit: 1. Overall, this barcode scanner provides great versatility and works well with the MCU chosen. There are multiple features to choose from such as single scanning time, lighting options, warning tones, and barcode ID configurations.

MCR12 CCD Scanner

The MCR12 CCD Scanner, as shown in Figure 30, can decode any one-dimensional barcode that uses a tiny camera to take 100 photos per second. This method is highly accurate compared to the laser scanning mirror assembly. Additionally, it is less likely to get damaged or misaligned.



Figure 30 - MCR12 CCD Barcode Scanner Module (Permission to reproduce submitted)

To scan a barcode, the user must place the barcode serial number at around 4 inches or more. Additionally, the barcode scanner comes with many customizable features that the user can choose from. There are three scanning modes: manual, continuous, and trigger delay.

The manual mode is default where the user sets a trigger pull to access the decoding session. In continuous mode, the red light will always be on, reading any barcodes that come within proximity. Trigger delay mode allows the red light to be on and scan for 1-9.9 seconds, depending on the trigger timeout settings that can be accessed through scanning barcodes in the user manual.

Unfortunately, the drawbacks to using this barcode scanner module are the lack of low power modes and that its limited to USB as serial communication. The PCB design will have to incorporate a USB support if this module is used. In Table 31, the technical details are summarized.

Table 31 - Barcode Scanner Module Specifications

Specifications	
Operating Voltage	5 Volts
Operating Current	80mA
Standby Current	None
Sleep Current	None
Operating Temperature	0°C~50°C
Operating Humidity	20%~95%(Non-condensing)
Interfaces	USB
Dimension	44 mm x 30 mm x 19.2

RB-Dfr-567 barcode scanner

In Figure 31, the RB-Dfr-567 barcode scanner module is displayed and is designed to scan one-dimensional barcodes that can be from either an LCD screen or paper. It scans about 100 times per second at a distance of 500 mm. The scanner is a CCD device and is ideal for third party and custom plug-ins.

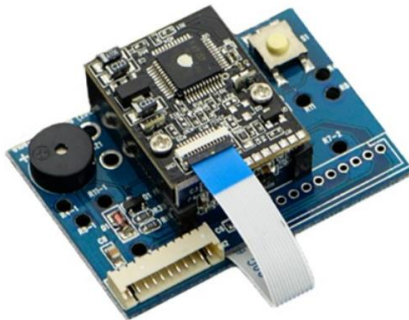


Figure 31 - RB-Dfr-567 Barcode Scanner Module (Permission to reproduce submitted)

Similar to the modules previously mentioned, this module has many modes in which the user can scan to fit the proper situation. These include the various reading modes, baud rate parameters, scan speed, etc. In Table 32, the technical details are summarized:

Table 32 - Barcode Scanner Module Specifications

Specifications	
Operating Voltage	5 Volts
Operating Current	150mA
Standby Current	50mA
Sleep Current	None
Operating Temperature	0°C~50°C
Operating Humidity	20%~95%(Non-condensing)
Interfaces	USB (RS232)
Dimension	20 mm x 25 mm x 11.5 mm

Overall, the barcode scanner is decent at it allows itself to enter standby mode after not being used for a period of time, which saves power. However, it is only limited to the USB type RS232 as serial communication which the MCU will have to incorporate.

Final Barcode Scanner Module Analysis

The three barcode scanner modules that we considered are the Waveshare Barcode Scanner, MCR12 CCD Scanner, and the RB-Dfr-567. Given the specs from the datasheets, Table 33 compares the barcode scanner modules.

Table 33 - Final Barcode Scanner Module Comparison

	Waveshare Barcode Scanner	MCR12 CCD Scanner	RB-Dfr-567
Serial Interface(s)	UART, USB	USB	USB
Types of barcodes it can decode	Most 1D codes Some 2D codes		
Operating Voltage	5 V	5 V	5 V
Current Drawn (from lowest power mode to operating mode)	2 mA - 135 mA	80 mA	50 mA -150 mA
Price	\$50.00	\$69.95	\$49.95

Generally, it was difficult to find other options for modules that contain barcode scanning functionality. Additionally, it more difficult finding a barcode scanner that has other methods of using serial communication. Most of these modules use USB or a variant of USB (such as USB type RS232) for communicating with other computers/microcontrollers. Therefore, the Waveshare barcode scanner is considered amongst the scanners found. It has a unique function of using UART to communicate while the other modules are limited to USB which is a drawback for designing the PCB. If any of the other modules are used, a USB serial port will

be needed for the MCU and the particular barcode scanner module to communicate. Furthermore, the Waveshare Barcode Scanner has more low power options that ultimately draws the least amount of power. This makes it more battery efficient when compared to the rest.

Fingerprint Scanner

The fingerprint scanner will be used for the user to store their unique thumbprint. These thumbprints will be stored in order for our MCU to verify that the user is attempting to access and unlock the Black Box. If the thumbprints match the previously stored ones, then the MCU will send a signal to unlock the Black Box. The fingerprint scanner is an additional feature our group thought of to give more access options to the user only while also providing keen security by preventing others from accessing it.

When choosing a fingerprint scanner for this project we had to find a device that would be compatible with our controller and one that would scan with accuracy along fitting within our budget. We went with the Flash Tree due to the number of features it has along with a decent resolution for only \$18.99.

Table 34 - Fingerprint Scanner Comparison

Finger Print Scanner	EM 406	R303	GT-511C1R	Flash Tree
Type	Optical	Capacitance	Optical	Capacitance
Manufacturer	HF Security	Grow	-	Flash Tree
Voltage	5.0V	5.0V	5.0V	5.0V
Resolution	509	508	450	324
Cost	\$42.99	\$38.74	\$32.50	\$18.99
Finger Print Storage	1000	1000	20	168
Current	<100mA	<55mA	<100mA	<95mA

Table 34 describes the various features that are important for making a decision on which finger print scanner to use. One notable factor that is displayed in the table is the price of Flash Tree scanner we chose. It is definitely the most cost effective design out of the others. Additionally, we did not need a fingerprint scanner with deluxe features which is why this fingerprint scanner is inexpensive.

5.2 - Software Design

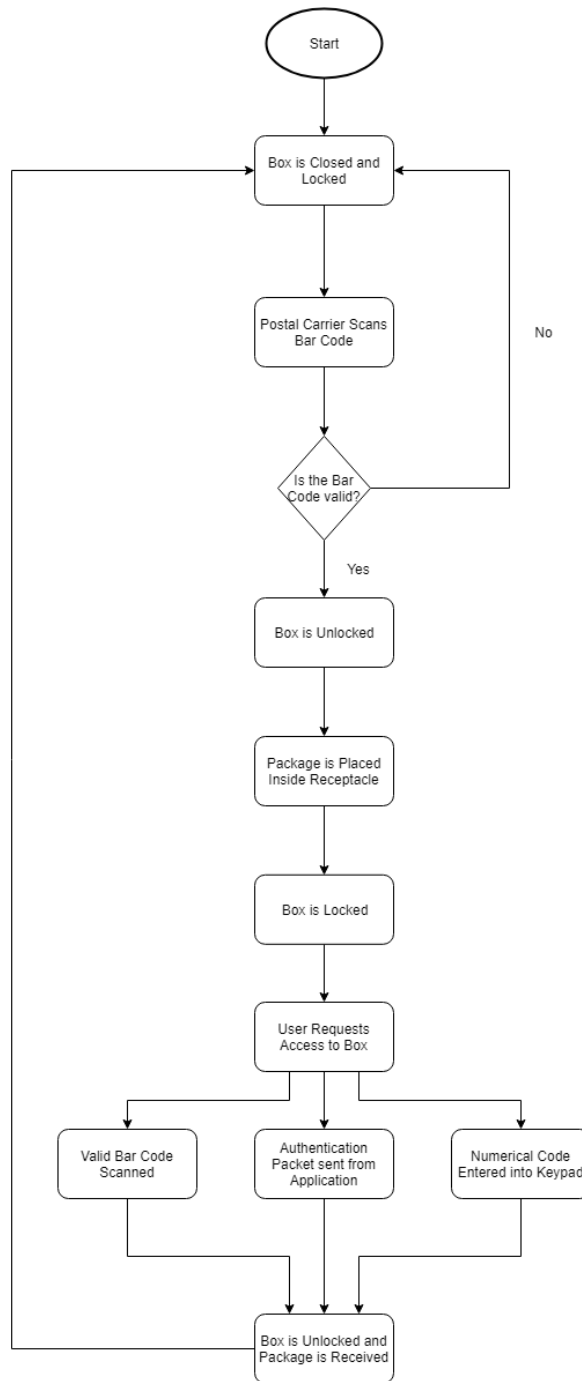


Figure 32 - Software Logic Flowchart

Figure 32 displays a broad-spectrum of how our project is intended for use in regard to software. To explain how the “Black Box” will work, it initially starts in the closed and locked stage. When the package delivery arrives, the postal carrier scans the barcode to verify that the requested package is delivered then places it

inside if it is unlocked. The box proceeds to keep the item locked and informs the user that the package is received. The user has two ways of unlocking it, via barcode or a numerical code entered into the keypad.

5.2.1 - Application Software Design

Below describes the functionality and design behind the iOS application software. The application will work in conjunction with the Black Box and its components to inform the user of any changes.

Functionality

As depicted in Figure 32 above, the main functionality of the application involves sending an authentication packet to the Black Box from the users' smart phone. This packet will include the keypad pin to signify the box to allow entry. A returning information packet will include details such as logged entries and photos taken at the scene of the Black Box. In order to accomplish this, there must be a web interface that can hold this data and keep that data secure. Below are the features necessary to properly implement the application.

Web Stack

In order to create an efficient application to communicate with the Black Box a web stack will be needed. A web stack is a collection of software necessary for development on the web. A web stack usually contains each of the following: an operating system, a programming language, a database software, and a web server. These stacks are necessary for implementing websites and web applications. Each piece of the stack is critical for efficiency. The operating system is the center of each part of the stack. The web server aids in providing information to the user and the databases help store large amounts of data needed for the website or application.

The programming language acts as a script interpreter and helps with the user interface of the application. The assemblage of these items, including proper hardware, helps in the creation and usefulness of a website. The browser is where it all starts; however, the web stack is what handles everything behind the scenes. A request is sent from the client's side to the web server where it is determined what type of request was sent. If the request was for a file it is sent to the application server and interpreted and sent back accordingly. If the request needs to access data than it is pulled, edited, deleted, or saved from the database and sent back to the client.

Depending on the project, different variations of web stack configurations are available to fill a project's needs. There are many variations of these web stacks; however, the three most popular include LAMP, WISA, and MEAN. When deciding to choose from one of these stacks you should consider the following: scalability, the development team's strength, what platform is available, and what database is necessary. Each one of these stacks has benefits as well as their own drawbacks when considering these topics.

LAMP is an acronym for the web stack that is comprised of Linux, Apache, MySQL, and PHP. LAMP is a free and open-source web development stack. This stack was developed in 1998 by a German scientist Michael Kunze. Linux is the operating system of the stack, Apache is the server, MySQL manages the database, and PHP is the programming language. LAMP is most well known for being open-source, its support by online communities, and the customization it provides. Using LAMP provides for more scalable websites or applications when the demands placed on it require it to shrink or grow. Another benefit to using this stack is that the platform independence of the stack also allows for the code to work on a range of systems.

Due to its large online community, using the LAMP stack allows for fast development with the available open supply of information. The high security of the stack attracts developers as well. Consistent updates, a secure foundation, etc. provide a safety net for developers. A large drawback to the stack is the fact that it is bound to a Linux operating system and does not allow for the use of many useful Windows programs. Another drawback is having to know multiple languages to get the server running. Having to switch between PHP and JavaScript/HTML can be a challenge.

An alternative to the LAMP stack is the MEAN stack. Often times the LAMP and MEAN stack are comparable due to both being open-source. The MEAN stack includes MongoDB for handling of the database, it is known to be great for its scalability in both storage and performance. E for Express.js is formed in the back end of the stack, it is designed to be very efficient to handle processes without cluttering the application. A for AngularJS is used for the front-end frame work of web development, this help developers build a user face side of the application and it pairs nicely with the other Mean Stacks, and N for Node.js the back bone of the stack it is built to work on top of Node.js, It can be scaled very large due to the cloud application framework. The pros for Mean stack are starting to get popular due to the scalability of the stack, also it is very well known for its flexibility in the language and for any web development for the cloud this is one of the best stacks to learn and know how it works.

The cons will be that the Mean stack package is still fairly new, so the support is considered limited compared to LAMP stack. Therefore, it maybe takes a technical person more time to learn the Mean stack compared to a Lamp, due to the helpful forums and years of information with Lamp. Express for the web application framework that runs on Node.js, AngularJS for the web application framework that runs in browser JavaScript, and NodeJS as a server environment. This stack differs from the others due to its lack of dependency on an operating system. It uses Node.js to provide the functionality of the web server. The advantages to this stack are the increased flexibility of its database for smaller applications and the fact that JavaScript is used at all levels of the stack. MEAN can run on any OS as well; however, the drawback to this stack is that it has much less support compared

to the LAMP stack and the learning curve is quite steep to new programmers. Choosing MEAN over a LAMP stack would provide a development team with benefits including: enhance speed for the retrieval of data, flexibility in deployment, and the use of one language, JavaScript, from front to end. JavaScript can also be seen as a disadvantage as well. The use of JavaScript can be disabled by users rendering the website or application developed with the MEAN stack completely useless.

WISA is an acronym that stands for Windows, IIS, SQL, and ASP.net. Windows is the operating system, Internet Information Services (IIS) is the web server, the Microsoft SQL server handles the databases, and ASP.net is the web application framework. The benefits of WISA is that it is more targeted toward enterprises or large corporations. ASP.net is also very useful for web developers as it enables code hiding, allows for increased performance, and has many features for the developer to expand upon. However, the largest drawback of the stack is the lack of portability. This means that the server is bound to a Windows operating system. Each stack has its pros and cons, LAMP is great for security and widespread support, WISA is great for enterprise projects, and MEAN is fast, easy to scale, and isn't bound to a specific operating system. For our project, the LAMP stack will be used. Due to its high security and widespread support we believe this stack will be most beneficial for the Black Box.

Security

Implementation of secure software for the Black Box is imperative to its design. Secure code is written to prevent users, and malicious third parties, from gaining access to software or information they should not normally have access to. In this instance, data such as tracking numbers, keypad pin, timestamps, and pictures will be kept secure and only accessible by the user.

The sensitive data tied to the Black Box will all be saved on the server; therefore, malicious attacks such as SQL injections can be detrimental to the user's information. A SQL injection is when a malicious user intentionally crafts an input that is known to be used in an SQL query. To prevent SQL injections the software will implement sanitizing data which will cause any freeform user inputted data to be interpreted as plain text.

The configuration of the SQL database will also be performed very carefully. Many database management engines contain an admin account which, if not properly configured at set up, can become a major liability and lead to disastrous data leakages or corruption. In order to prevent this, our database system will be secured with access to editing data from only one account with protected credentials.

Language

The application will be developed using Swift in the XCode IDE for iOS applications. Swift was created by Apple in 2014 and therefore is backed by one

of the most influential technology companies. Swift has become the dominant language for development for iOS. Swift is also open source, safe, and fast. In consideration of the development of this application included the language of Objective-C; however, the following benefits of Swift and disadvantages of Objective-C made the choice for the language of our application much easier.

- The performance of Swift is similar to that of C++. C++ is considered the fastest in algorithmic calculations.
- Swift is faster than Objective-C due to Objective-C's dependency on and the limitations of the C language.
- Swift was designed with safety in mind. Swift excludes and avoids mistakes with features such as generics, optional, and type interference for better stability.
- Swift is easier to read. The colon (;) is optional in Swift.
- Swift has less code for similar actions compared to Objective-C.
- Swift integrates with memory management. Using the Automatic reference counting the management of the memory of digital objects is easily handled.
- Swift is open source and more portable than Objective-C.
- Swift is an ongoing focus within Apple. The programming language is constantly evolving.

Sending/Receiving Information

In order to communicate with the Black Box from any location, our application must connect to the same server as the MCU of the Black Box. Therefore, a server that is only accessible through our application, and the hardware itself, will be beneficial for the construction of the Black Box. Communication will be done through HTTP GET and POST requests. HTTP, or Hypertext transfer protocol, is designed for communication between clients and servers. The MCU of the Black Box as well as the iOS application will act as the clients that are speaking to our hosted server. The GET method is used to request data from the server and is one of the most common HTTP methods along with the POST request. 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. GET requests can be cached and remain the browser history; therefore, their use should not involve sensitive data. POST requests, however, are never cached and have no restrictions on data length.

For the user to communicate with the Black Box, the application will send a request to the server for information, or to update information, and the server will transmit that request to the MCU, the MCU will then transmit the result of that request back to the server which is then forwarded to the user. For the Black Box to communicate with the user, the MCU 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 MCU speaking with the server to forward messages

to the user is to inform the user of the conditions of the box; therefore, the MCU doesn't necessarily need a response back. However, if during development the necessity of a response back to the MCU is necessary it will be implemented.

The Server

The server is where the LAMP stack will be utilized. As previously stated, the server itself will only be accessible to the user through the mobile application as well as from the MCU. The intentions of this server are to store and update information. It will also send information to and from the MCU or iOS application. Therefore, there will be no accessible website with a user interface for anyone to interact with outside of the mobile application.

5.2.2 - Embedded Software Design

The software on the CC3220 MCU will integrate the other hardware components with the MCU as well as allow for communication between the MCU and the server. One of the greatest benefits of using a Texas Instruments microcontroller is the well documented information provided to the developer. Using TI's software development kit for the CC3220 has provided great insight on the potential routes for implementation of the above-mentioned features of the Black Box.

For communication to and from the server, the microcontroller will be programmed to parse and generate JavaScript Object Notation data. JavaScript Object Notation, or JSON, is language dependent and therefore easy for machines to understand. JSON data is usually represented in a collection of name and value pairs. This can be realized as a struct, dictionary, associative array, etc. in various languages. However, in C the most common realization is the struct data type. The code will then send or receive this JSON data through HTTP requests.

In order to successfully connect to the server, the MCU must be provisioned to a Wi-Fi connection. This provisioning will be handled throughout the code in several different ways. For testing, the provisioning can be done over a serial connection using UART. For the final product, the provisioning will be done over a Bluetooth connection. Therefore, the MCU will integrate a Bluetooth module and parse the data sent from to provision the board to a Wi-Fi connection. A limit on provisioning will be set due to the limitations of the Simple Link Wi-Fi module.

The reconfiguration of the Simple Link Wi-Fi is configured at runtime and should only be performed when necessary due to the process involving flash writes. This can impact system lifetime, or flash write endurance, and power consumption. Another important aspect of the software on the microcontroller is the ability to distinguish the validity of a tracking number scanned by the barcode scanner. Tracking numbers have several configurations, usually dependent on the postal service providing the shipping, and can be easily be detected by the barcode and transmitted to the microcontroller where the validity of the number will be tested. The different shipping carriers we wish to implement for is mainly with USPS, FedEx and also UPS. These types of carriers all have different styles of barcode

for each area code they ship to and also with USPS. For example, USPS uses all numeric values for their barcode. FedEx and UPS include both alphabetical and numerical values.

What the software will consider as a valid barcode is determined by our research of the most popular postal services and their formatting for tracking numbers is depicted in Table 35. Along with integration of the barcode scanner with the microcontroller, the software will decode the information provided and accept or reject the tracking number that was scanned. If the number is accepted, the software will then send a signal to unlock the locking mechanism. If the number is not accepted, the software will provide information to the owner of the box and reject the deliverer.

Table 35 - Common Formats of Tracking Numbers [Trac]

Company	Description
FedEx	Most common tracking number format is 12 digits or 15 digits. Less common is 20 or 22 digits.
UPS	Most common tracking number format is 18 alphanumeric characters. Usually starting with 1Z. Although, some other less common formats exist.
USPS	Most common tracking number format is 20 digits or 13 alphanumeric characters. Usually the tracking number starts with two alphabetic characters, followed by 9 digits, and ending with "US". Some other less common formats exist.

Given the information in Table 35, the microcontroller will be programmed to match the scanned barcode with regular expressions representing the provided popular tracking number formats. If it is accepted, it will then indicate which the company the tracking number identifies with and notify the user of the incoming package with the details.

5.3 - Parts Selection Summary

When considering the hardware needed, we must consider our design requirements and constraints. As the internet has a vast majority of hardware components available, we had to make a pros and cons list to make the difficult decision. The hardware requirements had to be centered around giving the customer security and the practicality for the average user to know how to use it without reading the manual. A motion sensor for the camera, deadbolt, and a siren are critical components we have considered in our design.

Our smart box also has to include a component that would be able to communicate information to and from the device, this will give the end user a status on the box and implemented with the use of a Wi-Fi component. The power for the smart box will be powered by a battery with a huge capacity in order to sustain the longevity of the box and also give the end user the mobility to move the box to the desired

location without compromising its function. Below in Figure 33 are the major components we have decided to pursue to develop of the Black Box. After careful considerations and comparisons, the locking mechanism, barcode scanner, fingerprint scanner, wireless module, environmental sensor, and microcontroller will be the final components used in our design. As we are still in the prototyping stage, the basic components such as the MOSFET, resistors, and capacitor type components are still subject to change if we run into complications.

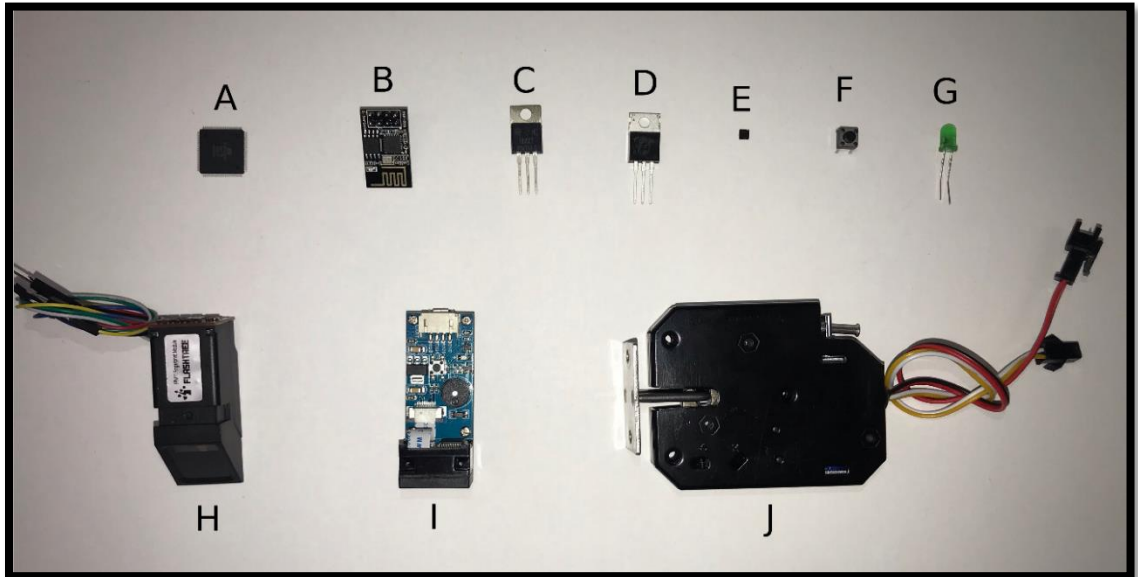


Figure 33 - Collection of Major Components

Figure 33 shows the major components that was ordered for our senior design project. The photo contains most of the components needed to get started with our reference design and for testing with our development board. To get started we had to use a developer’s board to do our testing for capabilities. We first ordered the barcode sensor and lock to test its functionality and also the physical size to see if we can prototype a design to mount on our Suncoast 22-gallon resin deck box.

Table 36 - Major Components List

Reference	Part Name
A	ATMega2560 Microcontroller
B	ESP8266EX Wi-Fi MCU Module
C	LM7805CT 5V Voltage Regulator
D	IRF630B Power MOSFET
E	STS30-DIS Temperature/Humidity Sensor
F	Push Button
G	User Feedback LED
H	Fingerprint Scanner
I	UART Barcode Scanner
J	12V Solenoid Locking Mechanism

The microcontroller, shown as part A, is the main component that will interface with the other modules which will go into our system. The main job for the microcontroller is to connect and get data from the Wi-Fi module, part B, and connect to a server that transmits data to and from the mobile application. The microcontroller will also connect to the deadbolt, part J, giving the signal to open and close after data verification.

Part B - the ESP8266EX is chosen over all the other Wi-Fi modules because of the synergy it has with the ATmega2560. Both components use the Arduino IDE which makes it easier to code. Additionally, there are tons of examples and resources online to help us fully unlock the potential of our Wi-Fi connection. The single board package provides us compatible wireless antennas and pin connections that will allow for easy interfacing to our mainboard. Finally, the ESP8266EX is offered at a competitive price compared to the others.

Part C – The LM7805CT 5V Voltage Regulator is used to step down the 12 V batteries that will be used to power the hardware. The MCU uses 5 volts; thus, this voltage regulator is a must-have.

Part D – The IRF630B Power MOSFET is used because it has a gate to source voltage of 4 volts which provides a good enough threshold for determining a read of high or low.

Part E – The STS30-DIS is the temperature sensor chosen to interface with the ATmega2560 using I2C serial communication. It provides decent ranges in temperature measurement when compared to the other sensors and it is the most cost effective.

Part F – The push button will be used for enabling the barcode scanner (Part I) to turn on.

Part G – The green light emitting diode is used for notifying the user/deliveryman that the lock has been unlocked.

Part H – The finger print scanner will enable the user to record his/her thumbprint to record. The fingerprints will be registered by this module to give future access to the user(s). It can record multiple thumbprints that can be assigned to different identification numbers if the user wants to allow others to use their fingerprint to open the Black Box.

Part I – The Wave share barcode scanner is chosen to scan the ordered package's barcode serial number. It will interface with the ATmega2560 by using UART serial communication. This barcode scanner is chosen out of all of the other scanners because of its ability to communicate through UART.

Part J - The Atoplee inductive solenoid lock is chosen due to its simplicity in design and the overall price. The lock contains red and black wires which signify the power line connections while the yellow and white lines are for the limit switch. The limit switch lines are important to the MCU as it tells the MCU when the locking mechanism is opened or closed. It is normally closed (locked) which means that the switch is on; thus, a signal is transmitted to the MCU. When it is opened, the switch is off, and the transmission signal is no longer being carried to the MCU. This way, if there is an issue with locking, for example if the mail carrier does not fully close the box, the user can be alerted that the lock is not set.

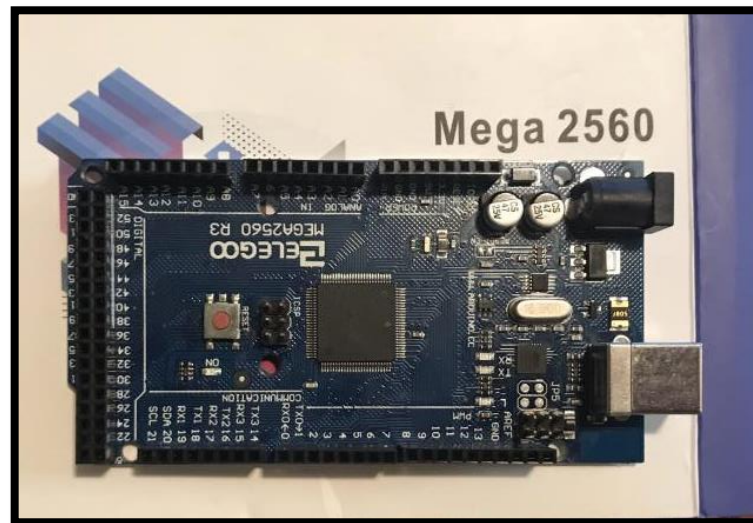


Figure 34 - ATmega2560

Labeled as Part A, the ATmega2560 is chosen out of all the MCUs because of how cost effective and how easy it is to fabricate into a PCB. This MCU only requires 2 layers for designing the PCB while some of the other MCUs such as the CC3220SF may require up to 4 layers to fabricate. The more layers needed to fabricate into a PCB, the more money it costs and more issues we may encounter. Additionally, the ATmega2560 is efficient for the features we want the Black Box to have. Shown above in Figure 34 is the Elegoo Mega2560, a development board based on the ATmega2560. We purchased this board as it will allow us to rapidly prototype our different designs without needing to print a PCB for just the chip. Verifying the design of some of our subsystems using this development board allowed us to be confident in our microcontroller chip selection.

Sun Cast Resin Deck Box

This Resin Deck box was the perfect fit for our project, we were looking for something that was decorative and functional to complete our project. This was also an outdoor rated deck box that will store our parcel for years to come. The Sun Cast Resin box is a 22-gallon deck box that is perfect for storing outdoor accessories or the vastly average sized box.

The box features a weather resistant resin construction that would allow very easy to little maintenance among with its very elegant high-end design comforting any homes front door piece. The Deck box was very simple to install, we had to consider this because for our user we had to make sure it took little to no tools to assemble this box. The Box came in 6 pieces that will snap into its grooves by simply following the instructions.



Figure 35 - Outdoor Box for Black Box

Figure 35 shows the purchase of our outdoor box for our project. This is a very durable outdoor storage box to withstand the outside elements.

5.4 - Summary of Design

In summary the Black Box's design is very simple and will be easily managed by the user. Below is the summary of both hardware and software design plans for the Black Box.

Hardware

The Black Box will be having an enclosure that is approximately slightly larger than the size of the standard package. This enclosure will encase the electrical components as well as have room for the inserting and removing of a package. The lid of the box will contain a keypad for the user to interact with and on the front of the box there will be a barcode scanner for the delivery service to interact with. The electrical components will consist of a microcontroller, specifically the ATmega2560, as well as a temperature/humidity sensor, a mechanic lock, etc.

Software

The Black Box will have software programmed within the MCU, on an iOS application, as well as on a web server. The iOS app will be able to communicate to the box as well as receive information from the box. The MCU will be able to communicate to the iOS app as well as receive information from the iOS app. The ability of this transfer of knowledge will be mediated by a webserver to ensure that the user can access the Black Box from anywhere with an internet connection. The software on the MCU will also allow for the integration of hardware components. This will allow for the transfer of knowledge between component and microcontroller to monitor the status of the Black Box.

6.0 - Project Prototype Construction and Coding

A successful project is defined by its integration between hardware and software. Both are equally important systems for any device, but in today's fast paced electrical world, one cannot be successful without the other. Good software is made great by the hardware it is run on and vice versa. Therefore, it is necessary to prototype the project in such a way that software and hardware are optimized together.

6.1 - Overall Schematic

Figure 36 below is the schematic that the team has produced for the development of the Black Box. Each component has been chosen and integrated to its relative specifications. The schematic shows how every component will interact with the ATmega2560 MCU. This schematic is intended to be our final design for the electronic component aspect of the Black Box; however, the design is subject to change under any circumstance in which the Black Box can be improved. In order to bring the overall schematic to physical form, we have to outsource and custom print a circuit board within our specifications and connected correctly with our various sensors and peripherals.

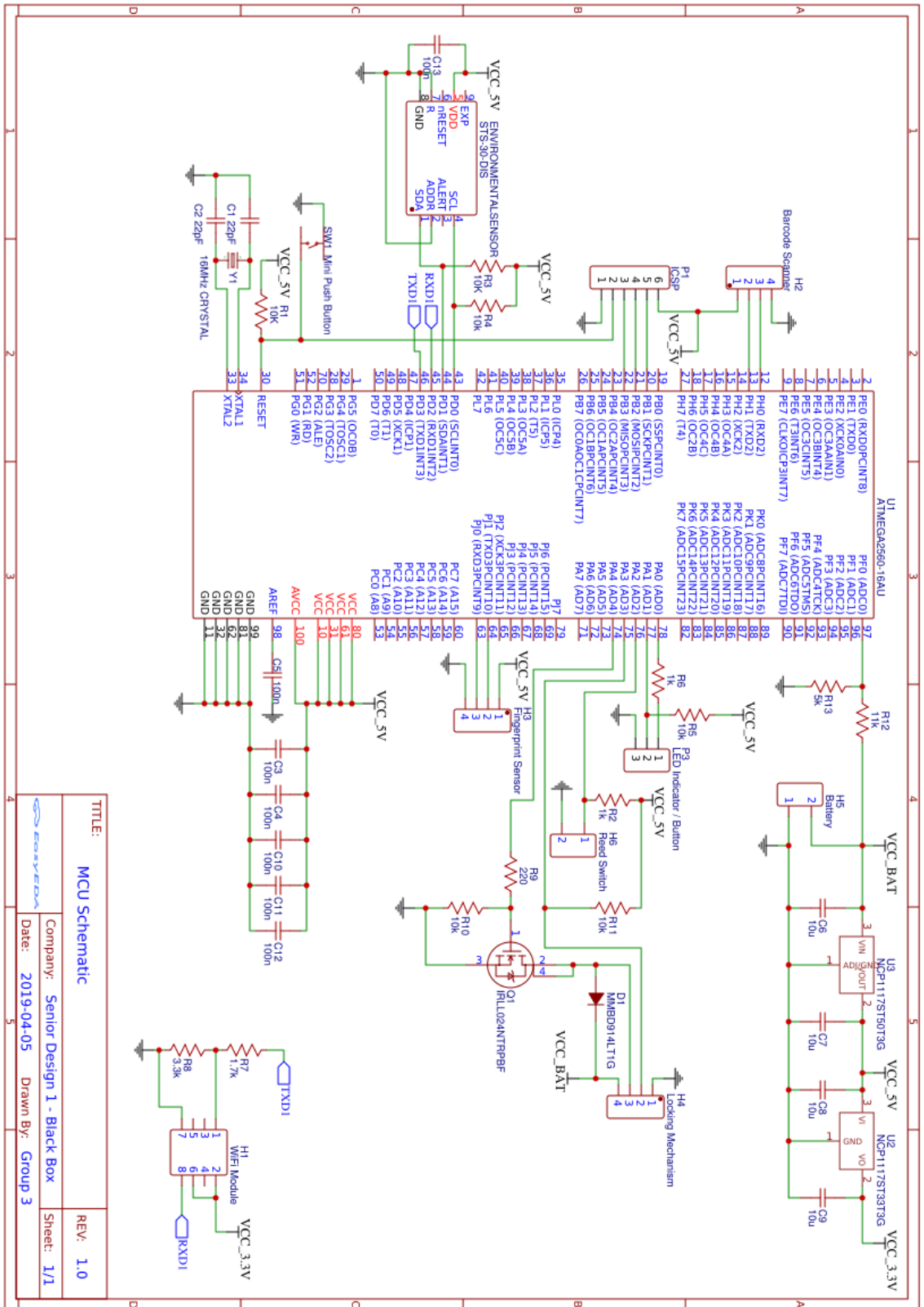
For each and every component that requires a decoupling capacitor we placed it close to the corresponding components as possible, when doing this we had to see the common placement and usage of this certain component before proceeding to the next. On the board we do have two linear regulators, which will step down the voltage for our application. The lock we have operates at a constant 12 volts. Our electrical engineer team Nathan and Louis has decided to operate the lock at 12 volts and also the board. When connected to the board a linear regulator will step down the voltage from a constant 12 volts to 5 volts. These components were thought out very well, we have the linear voltage regulator power supply placed far from other components due to the heat it will generate for long periods of time. A heat sink was considered for the linear voltage regulator, however looking at the datasheet and operation temperature. We have noticed

and tested that a heat sinks may be an over kill for this component. For the headers and terminal connectors, we have chosen the most standard and industry grade quality connectors. We wanted a secure connection because we considered that the Black box will be moved occasionally. These connections will be screwed tighten with the terminals we picked out. Also, we will have considered to Silk screen the PCB for ease of populating the board and also giving it a very standard PCB for potential improvements of the board.

It was also important to separate the temperature sensor from other heat producing components on the board. To accomplish this, we put the sensor in the bottom right of the board. Unfortunately, this also puts it closer to the Wi-Fi module, which since it transmits and receives lots of wireless data will generate a lot of heat. This could possibly be avoided by creating a mechanical and physical barrier of some sort that can redirect the heat from the Wi-Fi module and MCU away from the temperature sensor. However, since it is not necessary that we measure exact values for temperature and humidity, the heat effect might not matter that much in our design. We can also study the pattern of heat the Wi-Fi module and MCU create and account for their error in the code.

One difficulty of designing the PCB was that the schematic was created first. When designing the schematic, we did not necessarily have the PCB layout or routing in mind. Because of this, any of our components were connected to the pins of the MCU in ways that were convenient and looked nice in the schematic designer. The result of this, is that a lot of the pins we used ended up on opposite sides of the board from where those microcontroller pins were. Though a two-layer board mitigates this issue, it may still be best practice to use pins that are closer on the MCU to the physical location of where the components are. Since we aren't using every pin, moving which input and output pins we use can help us abide by the design recommendation of placing filter capacitors on VCC and GND of the MCU as close to the MCU as we possibly can. Overall, since this is our first time designing a PCB, we did the best we could and there is certainly a second revision to be done in Senior Design 2.

Another consideration for the PCB design is the mechanical considerations when affixing integrated circuits and chips to the board. Many components have extra-large back tabs that are not only used to dissipate heat but also strongly and securely fix the chip to the board in case of shock or vibrations. It is important to decide whether or not to use these back tabs, as they are usually always electrically connected. In the case of the MOSFET, the back tab is tied to the drain pin. And in the case of the low drop out voltage regulators, the back tab is tied to the voltage output. It is also important to determine whether these extra electrical connections are useful or not. Should they be tied to the net or left isolated and simply as mechanical connections? Depending of the application, such as how much power is required or the reduction of magnetic field noise, either scenario could be considered. When studying and designing our board, these were situations we encountered and had to account for.



TITLE:	MCU Schematic	REV:	1.0
Company:	Senior Design 1 - Black Box	Sheet:	1/1
Date:	2019-04-05	Drawn By:	Group 3

Figure 36 - Black Box Schematic Design

6.2 - Integrated Schematics

This section identifies each component of our schematic, our goal is to have our microcontroller interface with our temperature sensor and other peripherals to obtain the goals of our project. Our PCB has to include a voltage regulator for the Temperature sensor operating at 3.3v, and also have a 2 by 4 header for the WI-FI module for a direct and easy connection to the board. Communication with the multiple peripherals will consist of I2C, SPI, and UART, we plan to operate the board mainly on 5v for the ATmega2560 however, the stretch goal is to have 3 types of voltages which is 5.0v, 12.0v, and 3.3v.

6.2.1 - Wi-Fi Module Connection

Figure 37 below shows how the connection for our wireless module is configured. We used a standard size 2.54 female 2 by 4 header opposite of our male Wi-Fi module header, this way the ESP8266 can be plug and play when the PCB is printed. The idea of this is if the WI-FI module have a possibly of being faulty we can easily change the device out by just pulling it off and plugging back a new WI-FI module back to the 2x4 header. Additionally, having the antennas for our Wi-Fi on a completely separate board will help us avoid unwanted signal loss or noise due to the inductive nature of our locking mechanism and MOSFET.

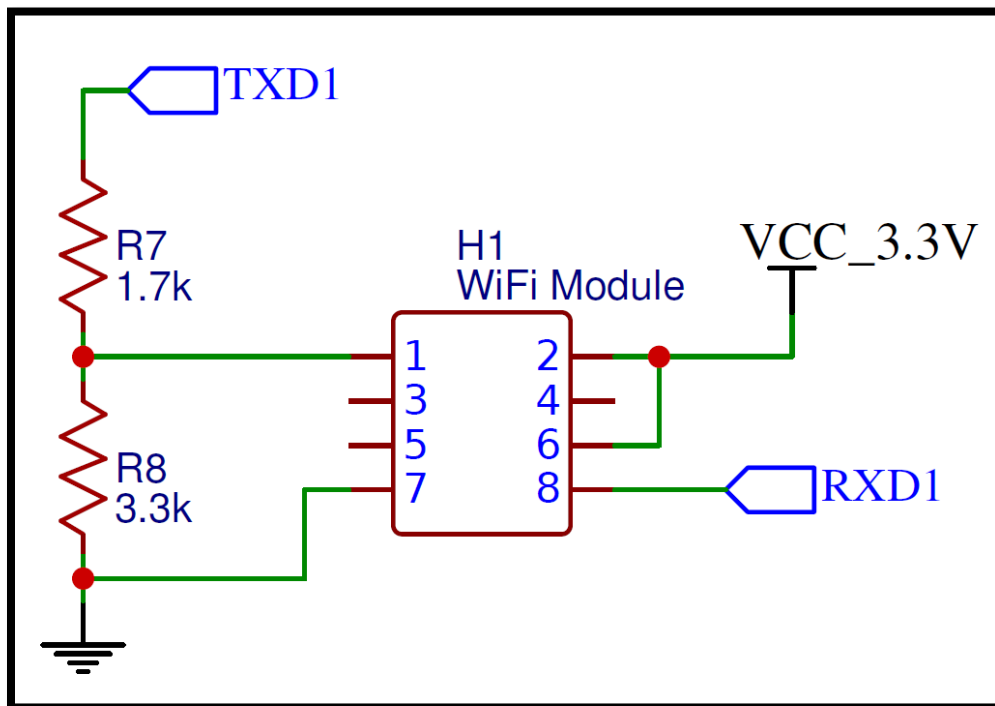


Figure 37 - Wi-Fi Module Header

Pin 1 on the header will be connected to the TXD1 on the ATmega2560 through a voltage divider to step down the 5V signal to a 3.3V signal, this line will act as a receiving data line. Pin 2 on the header is connected to our voltage regulator stepping our VCC voltage from 5V down to a constant 3.3V providing the ESP8266

optimum power as specified in the datasheet. Pins 3, 4, and 5 can be left floating because for Pins 3 and 5 it is connected to a GPIO input connection which we will not need for this application for our project. Pin 4 is connected on the ESP8266 as a reset for the chip, we do not need it as we are loading the code to the ATmega2560 and can disconnect the ESP8266 to reprogram. Pin 6 on the header is pulled high to enable communication for the ESP8266.

6.2.2 - ATmega2560 Programming via ICSP

Figure 38 shows the plugin of pins 20, 21, and 22 on the ATmega2560 in order to boot load the microprocessor. We also implemented a standard size 2.54 2 by 3 header to easy access and clean design after when we complete the boot loading process, we can disconnect the wires. This will allow us to easily and efficiently program or reprogram our board after it is completed.

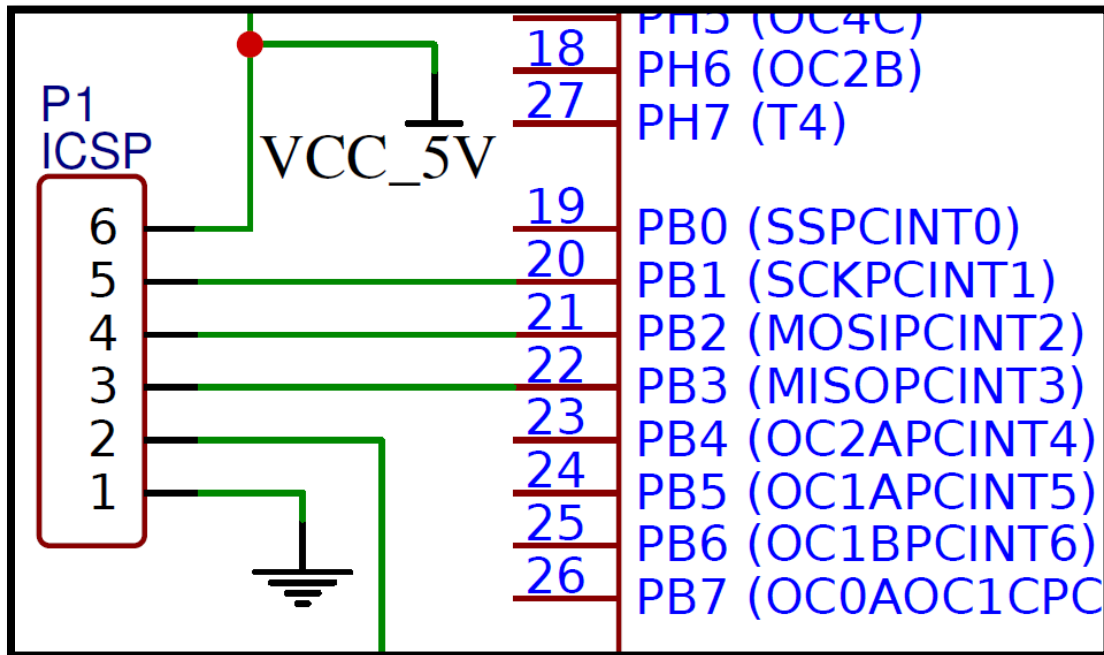


Figure 38 - Bootloader Schematic

On the header Pin 3 is connected to PB3 Pin 22 which is the MISO SPI connection, this will be the master line for sending out the data for the Arduino Uno. Pin 6 on the header will require the 5V connection for the ATmega2560, during this process we have to make sure the power is not connected. If connected this can fry the ATmega2560 MCU. Pin 5 on the header is connected to pin 20 which is the SCK pin, this will generate a clock pulse to synchronize the data transmission generated by the master pin. Pin 4 on the header will connect to pin 21 which is the MOSI pin, this pin is the master line for sending the data from out Arduino Uno to our ATmega2560. Pin 1 on the header is to ground. Finally, Pin 2 is connected to the RESET pin which is pulled up by a connection to VCC_5V and can be pulled down either by the ICSP programmer or push button on the board.

6.2.3 - Temperature Sensor

For the temperature and humidity sensor we chose to use the STS-30-DIS shown configured in Figure 39.

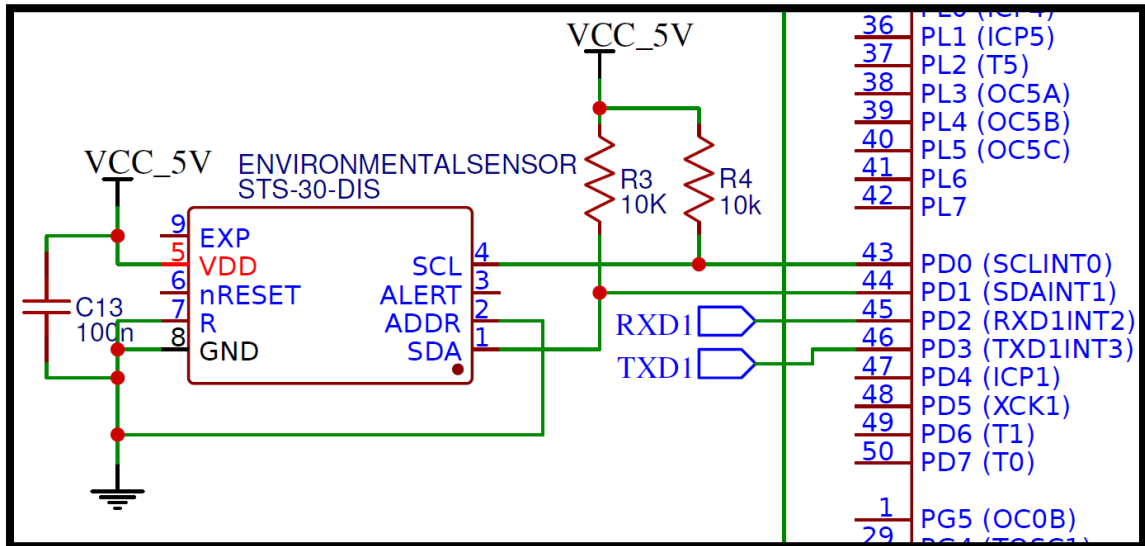


Figure 39 - Temp Sensor

Pin 1 is connected along with a 10k pull up resistor for the SDA pin which is the serial data pin where it will send and receive data. Pin 2 is connected to ADDR which can specify a different address depending on whether you pull it high or low, we arbitrarily pulled it low for the address 0x4A. Pins 3 and 6 can be left untouched since they are used if the STS-30-DIS is preprogrammed for an alert threshold. Pin 4 is connected to SCL along with a 10k pull up resistor as required on the data sheet provided. Pin 8 is dedicated to ground. Pin 7 is R which is has no function but needs to be connected to our VSS or GND. Pin 5 is connected to a constant 5V source which is our VCC pin. Lastly EXP pin is consider the die pad so this pin is also connected to the ground or floating. It doesn't matter but it is recommended that it be soldered to the pad for mechanical reasons. C13 is required to reduce noise going into the chip power, which could adversely affect the readings we receive.

6.2.4 - Power Supply

Since we are using batteries in our project, which can vary wildly in voltage, a series of voltage regulators are required to provide clean power to our components. For our main components which use a logic level of 5V, a 5V VCC power is required. Though our Wi-Fi chipset requires 3.3V power since it communicates using that lower level of logic, we need an additional voltage regulator. We could have used a switching regulator straight from our battery power, but it seems to be more efficient to use a low drop out voltage regulator in series with our 5V voltage regulator. Shown below in Figure 40 is our schematic for converting 6V-20V battery power into logic level power we can use in our system.

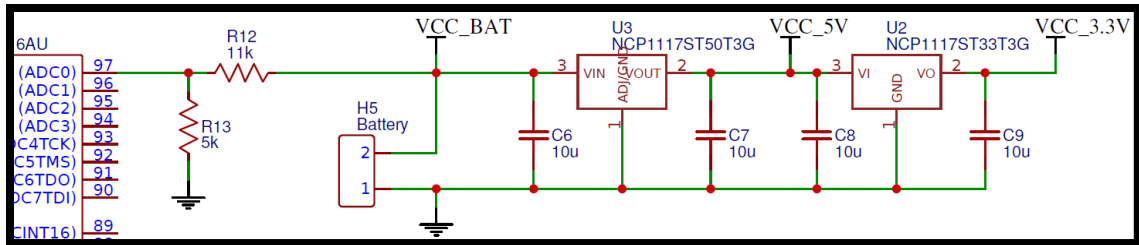


Figure 40 - Voltage Regulation

We picked two low drop out voltage regulators offered by ON Semiconductors, specifically the NCP1117ST50T3G for up to 20V to 5V conversion and the NCP1117ST33TG to convert 5V to 3.3V. These two voltage regulators have a dropout voltage of approximately 1V. This was required for the 3.3V regulator, as its input will be 5V, however, we have a much higher voltage for our battery before the 5V regulator. We still chose a low drop out voltage regulator in order to account for voltage drop when the locking mechanism is actuated. If the voltage drops below the dropout voltage of a normal regulator, it could brown out the rest of our circuit causing unwanted operation.

With a low drop out voltage regulator, our batteries can run much longer and at a much lower voltage while allowing the rest of the circuit to operate normally. Pin 1 for both regulators is connected to ground and if connected through a voltage divider, can adjust the output voltage level. Pin 3 on both regulators are the voltage input and Pin 2 is the voltage output. The tab can be connected to ground or left floating, but it should be connected to the board for mechanical reasons and to help dissipate heat. Two 10µF capacitors are connected on either end to help filter out noise. In addition to the 100nF capacitors located at each VCC and GND pair input, these should be sufficient to minimize any unwanted cross talk or radio frequency noise, especially from our Wi-Fi module.

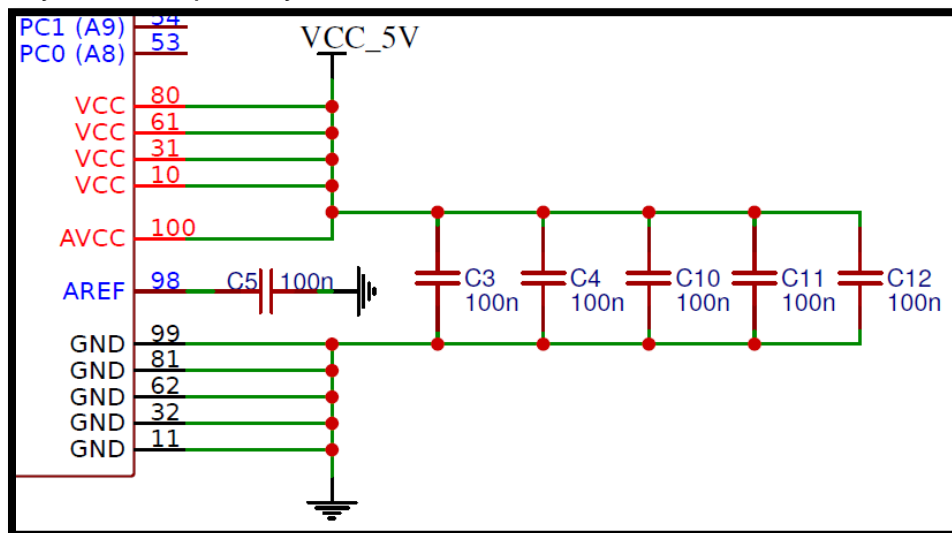


Figure 41 - VCC & Ground Schematic

Figure 41 demonstrates these smaller 100nF capacitors are used in our design for noise filtering. Atmel recommends a 100nF capacitor as close as possible to each VCC and GND pair on the board. Since we have five pairs, we therefore added five capacitors between pins 10 and 11, 31 and 32, 61 and 62, 80 and 81, and finally 99 and 100. Though some designs can get away with less filtering capacitors, it is important we take noise into account as our design utilizes wireless communication.

6.2.5 - Reset and Oscillator

Other required components for the ATmega2560 include the clock crystal and tying the RESET input to a logical high. These two subsystems are shown below in Figure 42.

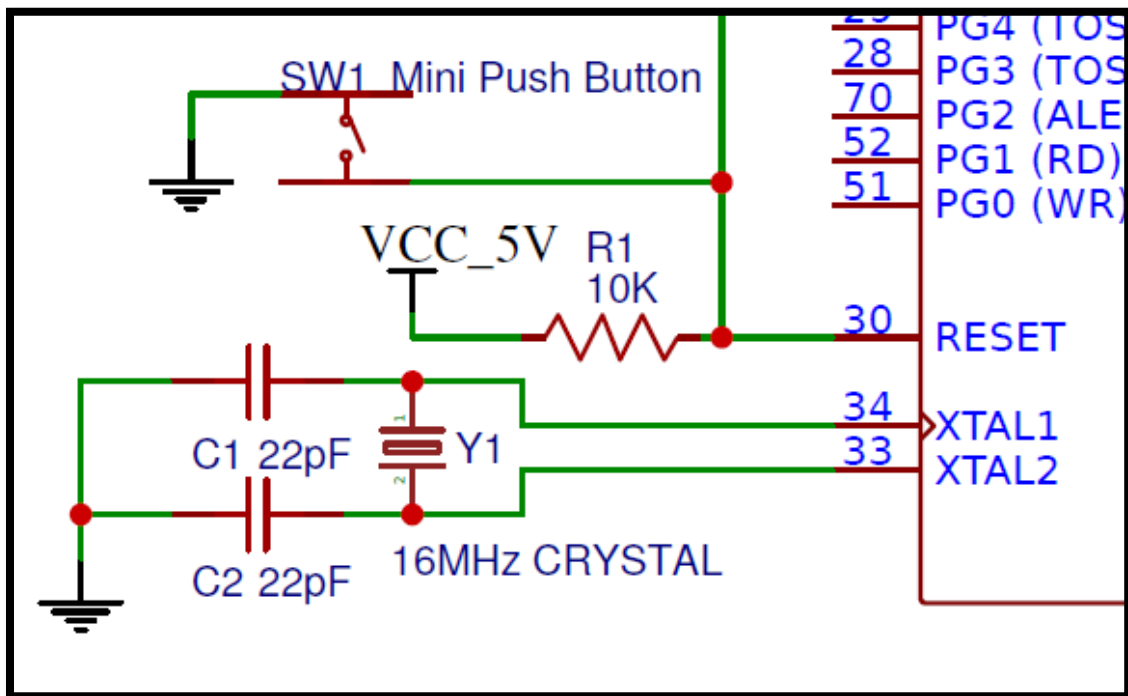


Figure 42 - Reset and Oscillation

For our board we had to implement a reset switch, for future problems we may run into. Pin 30 on the ATmega2560 is dedicated for the reset function, this requires a turn on voltage of 5V supplied from VCC along with a 10k pull up resistor. A reset button was implemented to help problems we may face while coding our board. The push button will also require a connection to ground in order to pull the pin low when pressed. A mini pin switch is needed to connect the ground together making a connected circuit. The RESET switch was also tied to our ICSP programming header in case it is needed while trying to program the MCU.

The crystal part of the schematic is required to generate a clock signal. For this schematic, we added a 16 MHz crystal as an electronic oscillator circuit that uses the resonance of a vibrating crystal. The 16 MHz crystal oscillator has to be

soldered on the PCB to provide a clock signal to the ATmega2560. This will provide the ATmega2560 a square wave signal to determine the time it is required for each state. The 16MHz frequency crystal will take 1/16 micro seconds to run 1 state. The 16 MHz crystal requires a 22 Pico farad capacitor to be connected to ground to help keep the oscillator stable.

6.2.6 - User and System Feedback

Our box relies on sensors and visual feedback from and to the user to accomplish its goals. This is facilitated by two different switches which can detect if the box is opened or closed and if the lock is locked or unlocked, the latter being included in the following section 6.2.7 - Locking Actuation. For this part of the schematic shown below in Figure 43, we have included inputs for a reed switch to detect the state of the lid of the box, a button input to detect when a user is trying to scan a package, and an LED to display to the user when the box is unlocked.

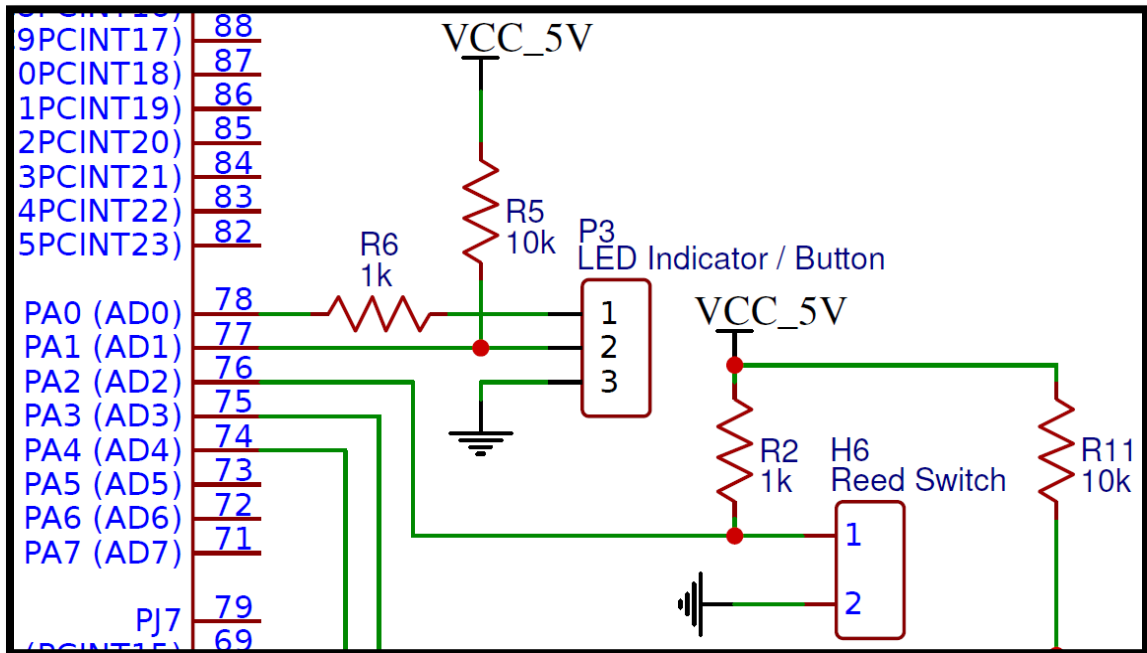


Figure 43 - User and System Feedback Schematics

The LED and button are simple components which will be mounted to the top or front of the box and connected to our main board through cables. Since they both will be located in the same relative area of the box and both require a ground connection, they can share the same header on our main board. The LED utilizes pin 1 on the header, which connects through a 1kΩ current limiting resistor to a general-purpose input and output register on the microcontroller. The button, however, connects directly to another GPIO pin on the board while being pulled normally high to VCC through a 10kΩ resistor. The reed switch works in the same manner as the user input button; however, it is actuated by a magnetic field triggered by a magnet on the lid of the lock. In the same manner as the button, it is pulled high with a 10kΩ resistor. This allows us to both know when the box is

closed and when it is locked so we can avoid unrecommended states such as the box being open, and the lock locked, or the box being closed and the lock unlocked.

6.2.7 - Locking Actuation

The arguably most complicated part of our schematic is the control to actuate our 12V inductive solenoid lock shown below in Figure 44. Since it has an internal limit switch, we have also included the feedback from that switch in this section, though it technically belongs in the above section 6.2.6 - User and System Feedback.

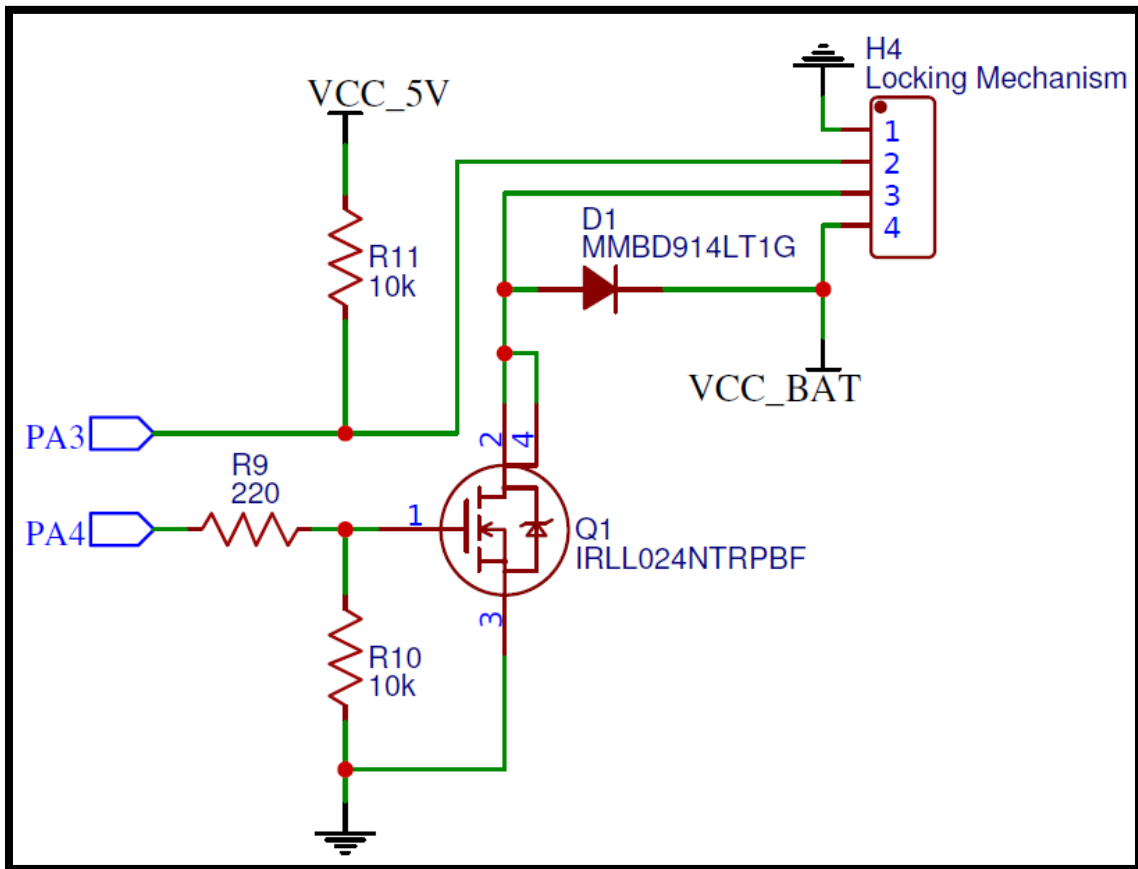


Figure 44 - Locking Mechanism Actuation

Since the feedback from the lock is a simple limit switch, we can pull the input to a GPIO pin on the ATmega2560 high through a 10kΩ resistor and let the limit switch pull it low when actuated. Since the lock is an inductive solenoid requiring 12V, we needed to actuate it through a MOSFET. The chosen MOSFET is capable of “logic level” actuation with a V_{gs} of 4V at $R_{ds(on)}$ of 100mΩ. The 4V is well above the 3V logic high level specified in the 5V logic standard but still below 5V enough that a current limiting input resistor of 220Ω can be utilized. An additional 10kΩ resistor is added between the gate and source of the MOSFET to help dissipate capacitance. The IRL024NTRPBF also contains an internal zener diode to help against reverse current from its self-inductance. Since the battery voltage is plenty

for the locking mechanism, we simply just supply the lock with 12V through the header straight from the battery and back through the N-channel MOSFET on the low side. From our prototyping, we discovered our inductive lock does return reverse current when the MOSFET is opened. To mitigate this, we added a flyback diode to prevent the current from reaching our MOSFET or worse our MCU. Similar in characteristics to a 1N4148 diode which we used in our prototyping, the MMBD914LT1G has a forward current of 200mA and forward voltage of 1V, plenty to mitigate the low reverse current supplied by the breakdown of the solenoid's magnetic field. This circuit will allow us to actuate the locking mechanism of our box safely, easily, and fast.

6.3 - PCB Design

When using a good schematic editor, generating the printed circuit board design becomes relatively easy. Most software allows the user to not worry about properly wired components as they will keep track of the electrical networks in real time while the user re-orient or moves individual components. The challenge comes when determining trace widths and routes. Displayed in Figure 45 is the full design of both layers for our PCB. For our high-power routes, specifically VCC_BAT and GND for the locking mechanism and relevant circuitry, we used a trace width of 20 mils. This should be more than sufficient for 12V and 2A maximum. For the general power lines, we settled on 10 mils for most VCC_5V, VCC_3.3V, and GND lines. This will allow power to be distributed evenly to sub-traces. These sub-traces compose most of our design, not only are they used to provide final power to the MCU and other chips, they also compromise all our signal lines. For these smaller traces we settled on a width of 6 mils. All traces regardless of their width were set to a spacing of 6 mils as well.

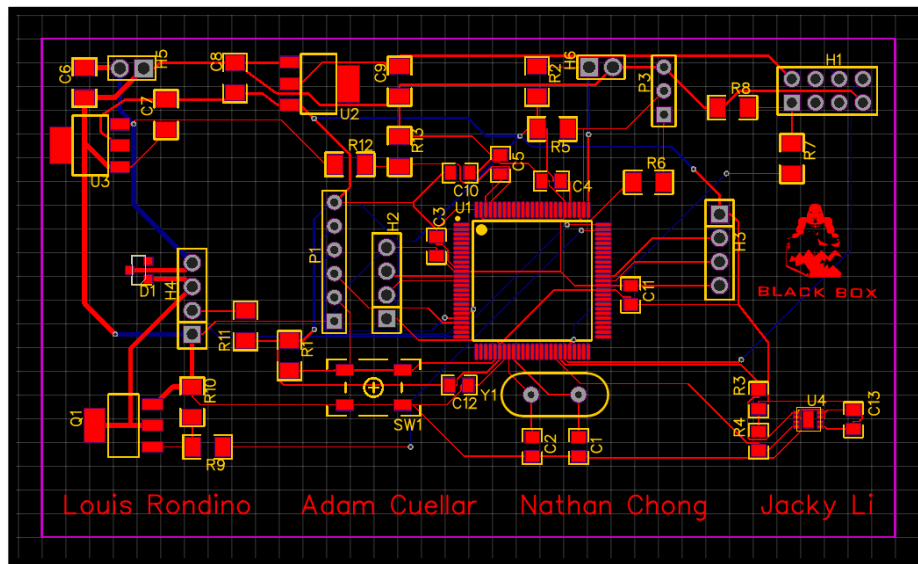


Figure 45 - PCB Traces

In revision two of this design, we need to consider adding mounting holes for where the board will sit in our box. We could also add some sort of components between the MCU, Wi-Fi module, and environmental sensor simply to help control how the heat dissipation of those components affect our temperature reading. If we were to do a major redesign of this board, it would be nice to have all the headers and connectors at the edge of the board. This would allow us to use 90-degree connectors if we needed to save vertical space in our final design. Since our Wi-Fi board rises vertically, however, we do not envision this being an issue.

For most basic components we ended up using either a 1206 size package or 0805 size. The 0805 was utilized for all small size capacitors, specifically the 100nF filters. This allowed for more space on the board to consolidate components, especially since most of the capacitors were required to be close to the MCU which is where the bulk of the traces and components terminate. Where there was space or for power components, we chose to use a 1206 package. When fabricating our design, we plan on at least attempting to assemble the board ourselves and feel that the larger surface mount components will help us practice our soldering skills before applying them to our future endeavors.

Another advantage of our combined schematic and PCB design software is the photo render it can create of our completed design. As shown in Figure 46, this is the end goal for our PCB. In this render, we can easily see the actual size of the copper pads for each component. It also allows us to determine if components might be too close to one another. Unfortunately, since this is a real render, we can only see the top layer of our board and therefore the top routes of traces. To stylize our board, we attempted to include a picture of our logo, but this render helps display that a complex image does not turn out great on the final product. If there is time in our design process, we may rework or remove this image.

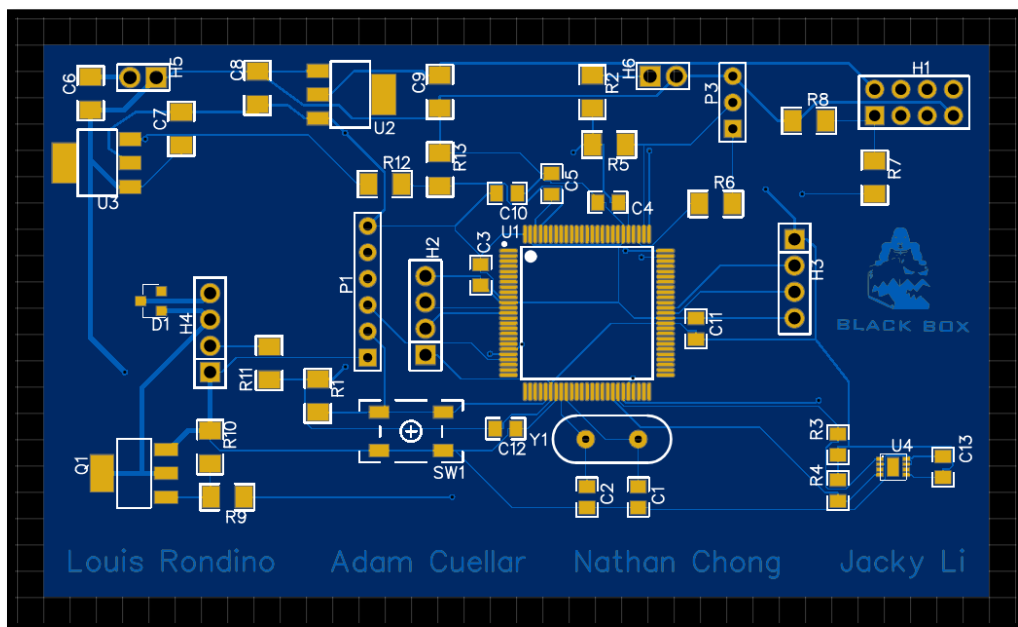


Figure 46 - PCB Render

In its current state, this schematic should work for our design. However, this effort was comprised in the final days of Senior Design 1 and will be continued and perfected in the beginning of Senior Design 2. In this time, we will be able to produce better routes, especially for grounding, and be sure our design will work come production.

Bill of Materials

Table 37 is the Bill of Materials we've obtained from the designed schematic.

Table 37 - Bill of Materials for Schematic

Name	Footprint	QTY	Part	Price
ATMEGA2560-16AU	TQFP-100_14X14X05P	1	ATMEGA2560-16AU	6.22
16MHz CRYSTAL	HC49US	1	HC-49US 16.000MHz	1.24
Mini Push Button	SMD-BUTTON	1	COM-08720	2.00
STS-30-DIS	DFN-8-1EP_2.5X2.5MM	1	STS-30-DIS	2.24
ICSP	HDR-6X1/2.54	1	Header-Male-2.54_1x6	0.021
Barcode Scanner	DIP-1X4P-2.54MM-M	1	210S-1*4P L=11.6MM	0.0261
Fingerprint Sensor	DIP-1X4P-2.54MM-M	1	210S-1*4P L=11.6MM	0.0261
Locking Mechanism	DIP-1X4P-2.54MM-M	1	210S-1*4P L=11.6MMGold-plated black	0.0261
WiFi Module	DIP-2X4P-2.54	1	220S-2*4P H=8.5MM	0.1045
Battery	HDR-2X1/2.54	1	826629-2	0.1941
LED Indicator / Button	HDR-3X1/2.54	1	Header2.54mm 1*3P	0.01
Reed Switch	HDR-2X1/2.54	1	826629-2	0.1941
NCP1117ST33T3G (Voltage Reg)	SOT-223	1	NCP1117ST33T3G	0.2577
NCP1117ST50T3G (Voltage Reg)	SOT-223	1	NCP1117ST50T3G	0.2043
IRLL024NTRPBF (Mosfet)	SOT-223	1	IRLL024NTRPBF	0.48
MMBD914LT1G (Mosfet)	SOT-23	1	MMBD914LT1G	0.03
Total		16		13.274

6.4 - PCB Vendor and Assembly

A major concern for constructing our own PCB design is choosing a vendor that will create our PCB reliably. The qualities of the vendor our group is looking for is a vendor known for reputable services, products made within a reasonable time range, and services that are cost effective. With that being said, OSH Park fits the criteria well and we were able to find out about them through many of our peers who have experienced their work.

OSH Park's standard services are summarized in Table 38 below:

Table 38 - OSH Park Standard Services

	Standard 2-Layer	Standard 4-Layer
Number of Copies	3 (can order in multiples of 3)	3 (can order in multiples of 3)
Shipment Days	9-12	9-12
Board Thickness	63 mills (1.6mm)	63 mills (1.6mm)
Copper Weight	1 oz	1 oz (outer) 0.5 oz (inner)
Price per square inch	\$5.00	\$10.00

The service also states that the boards ship with FR4 substrate, purple solder mask over bare copper (SMOBC), and an electroless nickel immersion gold (ENIG) finish. These qualities help make the PCB layers have excellent solderability, great environmental resistance, and are suitable for lead-free processes.

As for assembly purposes, our group is planning on buying bundles of boards to attempt to solder the components ourselves. This would help us gain experience in how to solder small components in the board. If we fail to solder the components correctly, we can always give the PCB and components to a company that is reputable in assembling PCBs.

Keeping that in mind, a great PCB assembler is QMS (Quality Manufacturing Services, INC.). This company excels in meeting performance needs and exceeding expectations. They have senior level engineers that have expertise ranging from assembling minimal layer PCBs with basic through-hole components to a more complex twenty-five layered PCB. Furthermore, they provide free services to students in senior design who need to have their PCBs assembled professionally, making this company an ideal assembler.

With that in mind with the recommendation of our senior design professor Dr. Richie, providing us the information that populating the PCB was free from Quality Manufacturing Services. Quality Manufacturing Services is operated and owned

by a fellow UCF alumni. When provided the information and identification that we are a UCF senior design student, Quality Manufacturing Services will populate the PCB for free. The only rules for populating the PCB is that all components will have to be ready and also a silk screen on the PCB will be present due to confusion of population process. A duration of two to three days turns around is to be expected during the population process. It is very helpful to have a company that is nearby to help us fellow students every semester,

6.5 - Final Coding Plan

This section will describe how the software will communicate with hardware by using the ATmega2560 microprocessor. Details of how and where pins should connect between each module will be reviewed and stated to ensure that each component will function correctly. Additionally, by verifying the pins' functions and pin location, the user can be allowed to initialize and define them in the Arduino IDE. Objectively, the code will have many tasks to carry out. The coding plan has to indeed follow the entity relationship diagram we have planned as a group. A mobile application will also have to be implemented for the use of this Black box project. These tasks are summarized in the bullet points below:

- Signal peripherals such as temperature sensors, barcode scanner, etc. to turn on when needed to
- Communicate the server and the box by sending frequent requests for notifications on status
- Grant/deny access to fingerprints and barcodes that are match/mismatch
- Engage in low power modes for battery life efficiency when idle

There are many peripherals that need to be coded properly. This is why we must think carefully and proceed with caution as we construct the overall function of the code.

6.5.1 - Boot Loader

A Bootloader is used to load a program into the blank chip in any of the ATmega2560 microprocessor we will decide to use. Because we cannot use the development board in our senior design project, all the ATmega microprocessors will be shipped with no environment and it comes in as a blank chip with no functions until we boot load it. The Bootloader in the Uno will communicate with our ATmega2560 to wait for the software on our computer to program the new board we have printed to load in the code. The Bootloader allows us to load programs into our microprocessor without the use of a USB cable, for the first boot up on the printed PCB it is required upon startup we have to use an Arduino Uno to boot load it. All the ATmega2560 MCU does not come from factory with bootloader preinstalled, we have to go through a process using an Arduino Uno to

install it on our own. Figure 47 above shows the boot loading process the aforementioned boards.

6.5.2 - The Boot Loading Process

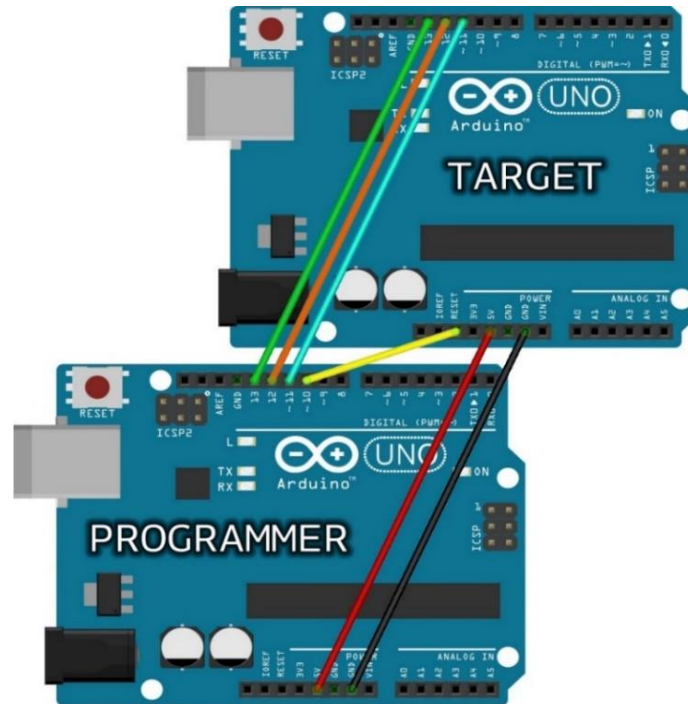


Figure 47 - Boot loading Visual (Permission to reproduce submitted)

The ATmega2560 will need a couple of components to get working in order to install the bootloader onto the microprocessor. A crystal is needed along with 2 capacitors to make a functional clock for our Microprocessor. This gives an overall idea how to wire the bootloader to a board that needs to be boot loaded. The Target board has no functions until the bootloader on the programmer board loads in bootloader. Many Development boards have an all-in system programming header within the development board. Atmel has introduced a circuit serial programming header for many self-made projects a breeze.

Table 39 - Pin Summary

AVR Programmer	Arduino as an ISP	2 by 3 ICSP header	ATMega2560	ATMega 328
5V	VCC or 5V	Pin 2	VCC	VCC
GND	GND	Pin 6	GND	GND
MOSI	GND	D11	D51	D16
MISO	MOSI or D11	D12	D16	D14
SCK	MISO or D12	D13	D50	D15
Reset	D10	Pin 5	Reset	Reset

Table 39 provides a detailed overview of what pins are needed for the SPI serial communication interface. For the ATmega2560 we have chosen for our project to first load the bootloader is to place the ATmega2560 pinned out correctly, there is an indentation on the ATmega2560 indicating the top position of the chip. VCC needs to be connected to a dedicated 5v source, on our PCB design we can use the 5v as we dedicated that as VCC. The Ground ISP needs to be also be connected to our ground on wire on the PCB. To designate and program our clock to working order, it is required to connect the 16MHZ crystal to pins 9 and 10 on our board. The capacitors we have chosen which was 22 pF to be connected to the crystal to the ground. To visualize the schematic, we will have to connect of the 22pF capacitor to pin 10 and to ground, with pin 9 we will have to connect that also with the 22pF capacitor and also to ground.

7.0 - Project Prototype Testing Plan

In order to verify the device works entirely as expected, a test procedure will need to be implemented. Both the hardware design and software design will need to be verified prior to synthesis into a final schematic for implementation into the deliverable product. Making sure the design is robust involves testing both defined and undefined input, however, for the sake of prototype testing it is only necessary to verify functionality of each individual subsystem.

7.1 - Hardware Testing

Before finalizing our total design, we must test each individual subsystem. This allows us to verify each smaller design so if there are any problems we know exactly where they are before they are lost in the full system. These tests, detailed in the following section, also help give us a greater understanding of the technology and how to best integrate them into our total final product.

7.1.1 - Locking Subsystem

The first and arguably most important part we received was the locking mechanism. To fit our desired design of affordable and top loading, we went with an Atoplee drop bolt lock with DC induction solenoid latching. This lock was also ideal as it contains an internal limit switch to tell when the locking mechanism is latched shut or open. Despite being a great choice for our project, this lock unfortunately came with very little information as to its operation. The only info listed is that it works on 12 volts DC, requires 2 amps to operate, and should not be left energized for more than half of a second.

We knew there was some testing to be done before integrating this lock into our design. The first thing we wanted to know is what the locks brown out voltage would be to see if we could power it directly from the fluctuating battery voltage or if we need a regulator. The lock reliably worked from 15 volts all the way down to 6 volts before no longer actuating. As our battery would probably be declared dead anywhere below 10 or 11 volts this was perfect since we knew voltage regulation, or a higher supply voltage would not be necessary.

Looking at our second piece of given information, a current draw of 2 amps, told us we immediately needed to design a switching circuit in order for the low current GPIO of a microcontroller to be able to control this locking mechanism. Since this is an inductive load like a DC motor, we weighed the possibility of using an off the shelf motor controller chip. When given the voltage at no load at 11.96 volts the voltage under load for many of the test was around high 9 Volts, giving us an average voltage under load when it was in its on position was 10.63 Volts. This gave us a very clear idea that underload we would want to stay at a consistent 10v given to the lock underload, or problems may occur where the lock may not open due to the voltage drop. By conducting this test, it was clear that the data sheet provided by the manufacture was no totally accurate with its operation voltage and current. When tested even at 6v the lock still was in working condition, however we did also see some slow down when it opened, this was provided with a constant 2 amps from the power supply. After testing, we have decided to give the lock its full potential power at 12v at 2 amps providing a dc to dc power convertor for the ATmega2560 with 5 volts and also one for the Wi-Fi module providing it with 3.3 volts

However, this seemed unnecessary as the motor controllers were multi-channel and we only required the one lock. The debate then became between a high power MOSFET or transistor or small relay. Almost immediately, the relay idea was discarded as either a MOSFET or transistor would be the most prevalent on hand technology to test with in the lab. Given the lack of data sheet information, we did our own working current tests see what kind of MOSFET or transistor we will need. In Table 40, we were pleasantly surprised to find that the lock only requires on average 1.02A making the given 2A value a maximum more than nominal.

Table 40 - Voltage/Current Load Test Summary

Voltage with no load (V)	Voltage under load (V)	Current under load (A)
11.96	9.61	1.12
11.96	10.63	0.84
11.96	11.89	1.11
11.96	10.10	0.93
11.96	9.60	1.09
Average	10.36	1.02

Before finishing our transistor design, we needed to test one last characteristic for the inductive load of the solenoid mechanism. A common problem with these devices is that when the circuit switches to open, the collapse of the magnetic field causes a negative current to be sent back through the device. This is mitigated by a diode which allows the negative current to form a complete circuit with the inductor. To test that we even have this problem, we measured the voltage when switching the lock on to see if there is a negative voltage and thus a reverse current. Shown below in Figure 48 are our findings.

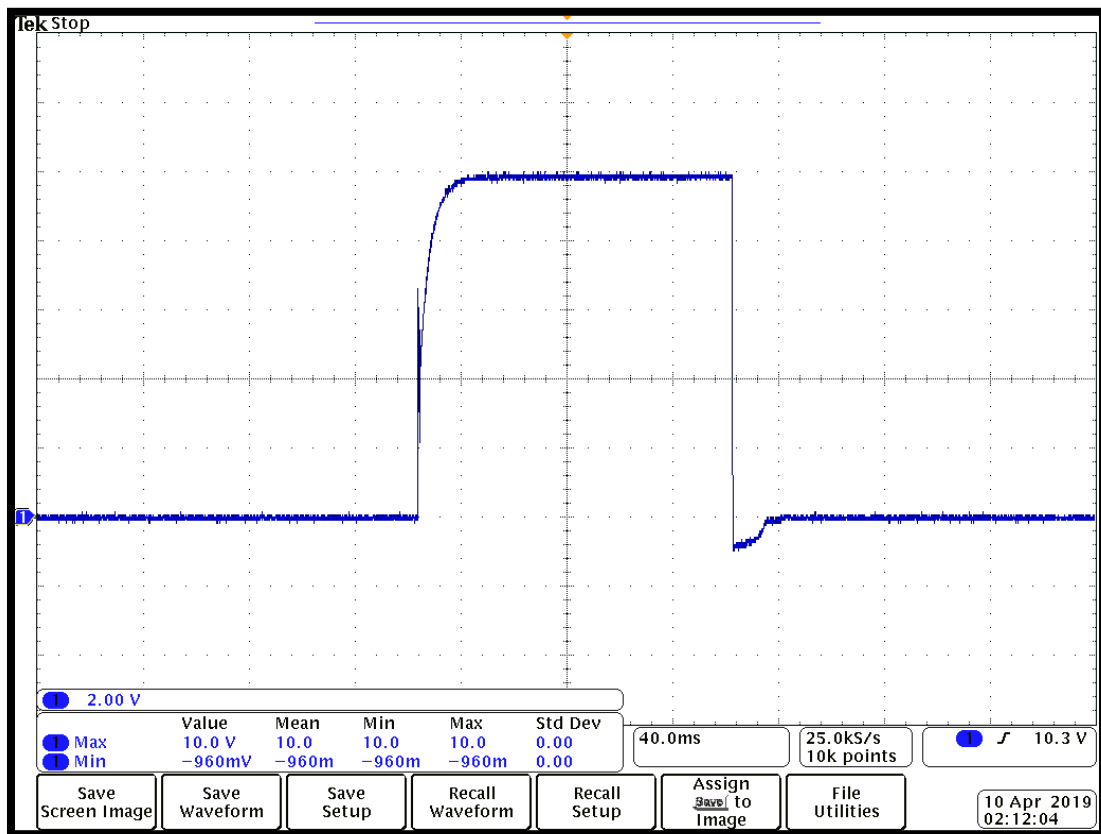


Figure 48 - Reverse Current Lock Step Response

This test concluded that we do indeed need a fly back diode, as almost -1 volt is produced when switching off the solenoid. Given that the measured internal resistance of the solenoid is $\sim 5.9\Omega$, this translates to $\sim 170\text{mA}$ of reverse current. For breadboard testing, the only MOSFET we had on hand was the IRF630B. Thankfully, with a drain current of 9A and drain-to-source voltage of 200V max, this MOSFET has enough voltage and current capacity for us to drive our lock for these tests. Since the gate-to-source voltage is so high, however, we will not use an input resistor for this test in order to maximize the signal current from the GPIO of our microcontroller. Though due to our low frequency on time of less than one quarter of a second, this resistor should not be needed to protect the microcontroller. To dissipate capacitance and act as a pull-down resistor, we attached a $10\text{k}\Omega$ resistor between the source, which is grounded, and drain of the MOSFET. And as a result of our previous tests and measurement of the reverse current, we added a standard 1N4148 diode with the cathode at 12V across the inductive load that is the lock. The resulting waveforms shown in Figure 49 concludes that our design for the lock switching mechanism works with the ATmega2560. Set to pulse the lock for 100ms, less than the half second maximum specified by the manufacturer, the lock accurately executed an unlock and our MCU was able to read the status of the lock from its internal limit switch.

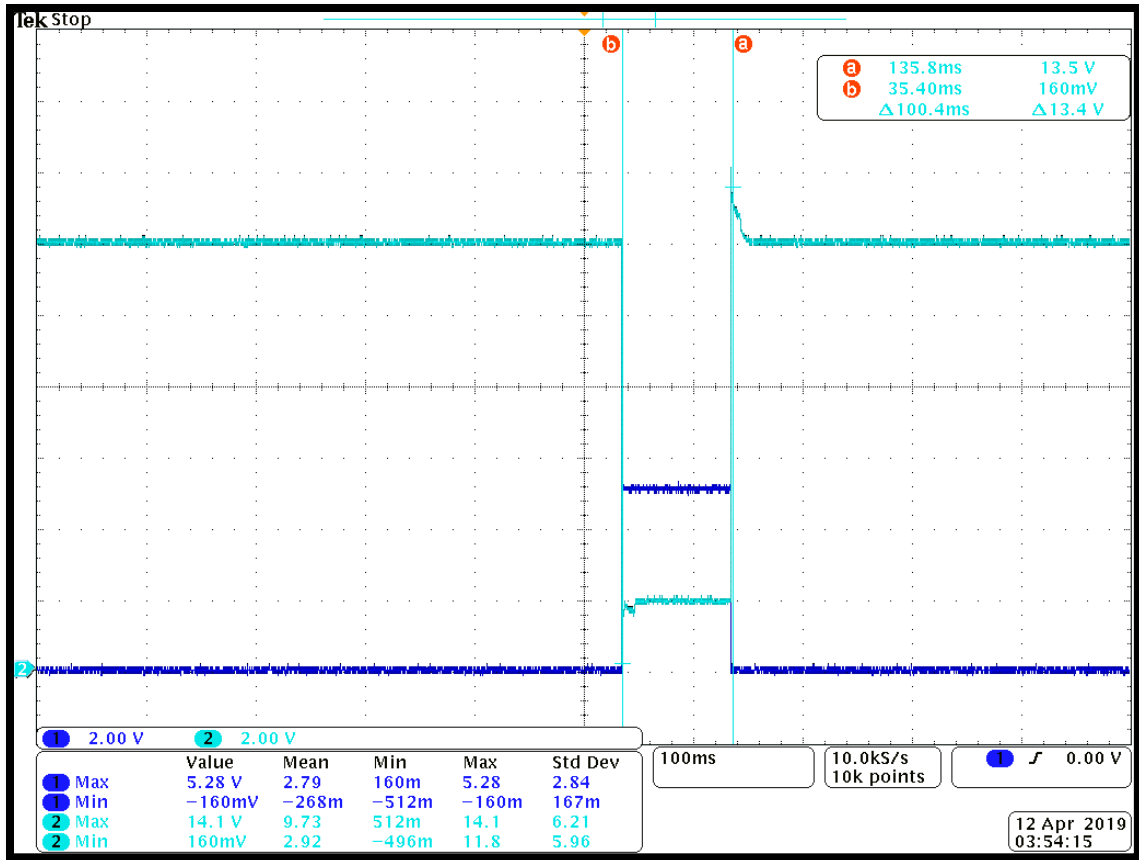


Figure 49 - Lock Response to Microcontroller Impulse

7.1.2 - Feedback Indicators Subsystem

When introducing ourselves to the ATmega2560 programming and hardware development, one of the easiest learning tools was to program with buttons and LEDs. Luckily, this is one of our subsystems. Our project will require at least an external button for the user to start barcode scanning and an LED to alert the user either when they can start scanning or when the box has unlocked itself. To prototype this originally, we simply used the included button and LED on the ATmega2560 development board. To further test our hardware design, we included our own button and LED on the breadboard. In doing our own hardware design, we determined that the LED required a 1k Ω current limiting resistor in series with the diode. Additionally, when adding an external button, we were required to pull the input high through a 10k Ω resistor when the button was open to avoid floating voltages and therefore undesired false inputs.

7.1.3 - Wi-Fi Subsystem

When researching examples for the ESP8266 online, many tutorials would lead someone to believe that it is a plug and play component. However, if you are an electrical engineer and decided to pick up the datasheet, you would easily realize it is not. The major issue is seen when considering that our microcontroller runs at

a 5V logic level and the ESP8266 runs at 3.3V. Though the ESP8266 may work most of the time when 5V is input to it, the datasheet does not specify that it is a 5V tolerant device. Therefore, in order to safely communicate using UART between these two devices, we must convert the logic level. There are integrated circuits available that can do this for use, however, since we are working with regulated voltages, there is a simpler way to convert these signals. When sending data from the ESP8266 to our MCU, the 3.3V signal is lower than 5V and safe for the device. It is also above the 3V logic high threshold for 5V logic and will correctly register a high versus low logic signal. Since these voltages are regulated, we can safely trust that the 3.3V signal will trigger a high signal successfully. On the receiving end, or going from the 5V MCU to the 3.3V ESP8266, this higher voltage can possibly damage the Wi-Fi module. To remedy this, we can simply use a voltage divider to step down the 5V from the MCU to a safe 3.3V for the ESP8266. When testing in the lab, we utilized resistor values of 1.8k Ω and 3.2k Ω to generate a 3.2V signal into the ESP8266. The 3.2V as opposed to 3.3V allowed a good margin of error to not risk damaging the ESP8266 by going above 3.4V. And since the logic level is 3.3V, a 2V signal is considered high by this device. Testing this voltage divider system as a logic level converter helped us understand better the different communication requirements used in modern systems. By prototyping our design, we were able to verify that a logic level converter chip was unnecessary for our design.

7.2 - Software Testing

To successfully determine the Black Box's software as fully functioning, it is necessary to test each division of software involved with the Black Box. These include the software on the MCU, the server's software, and the iOS application. While involving ourselves more with the scope of the Black Box and its functions, the following snippets show the specific functions we have tested and continue to test to ensure the functionality of the finished product.

Unit testing is a certain level of software testing where the individual units or components of a software is tested. Methodology approach to building a product by the means of testing and revising. This encourages many software engineers to modify their source code without many concerns on how much it changes may affect the function of other units or program as a whole. Unit testing has a very steep learning curve, the team really has to learn what a unit test is, also how-to unit test, and lastly how to use the software tools to automate the process through an ongoing basis.

The objective is to pass all the errors and have an efficient program that can be evaluated and that can be done by integration testing. The benefits are to use unit testing to detect the problem early before jumping deeper into the project. This can boost the confidence in writing code and also changing it, the good unit tests are when modified it shall work flawlessly every time. The code is reusable to make the code unit test possible. Lastly this increase the development process of the project. Making the test does take time however once written it is very easy to test

the code, therefore when project is finished say every week it can be tested to check for possible errors and problems down the road.

Unit Testing Tasks – These are the steps Software engineers perform step by step using Unit Testing

Unit test plan (this step is where engineers will plan what they are going to do)

- Preparation
- Review
- Rework
- Baseline

Test cases (establish many test cases to ensure the code is working properly)

- Preparation
- Review
- Baseline
- Unit Testing (Testing Phase)
- Performing the test

Software Test Timeline

Table 41 includes the software timeline it ensures our software developers on our team will conduct the following tests and also ensures we will make the deadlines. The software team will first develop a experience with writing common codes for the ATmega2560 on the developers board, this is done due to the time line and during the duration where the electrical engineers will be working on the schematic design.

Table 41 – Software Testing Time Line

Date	Job	Status
2/25/2019	Write the C code for the Microprocessor on the developer's board	Completed
3/1/2019	Development of the mobile application on the iPhone	Completed
3/25/2019	WIFI Connection compatibility	Completed
4/5/2019	Finger print testing	Completed
4/6/2019	Communication with the lock and triggering the lock	Completed

The next mile stone will be during the March testing which consist of the development testing of the mobile application, as many common projects are done on android devices our software developers insisted that we will do the mobile application on the iPhone. The next following mile stone was to connect the WIFI module to our developer's board to ensure the WIFI module is communicating with our board. Our Computer engineer, and software developer Adam Cuellar had a very brilliant idea to test it with toggling the L.E.D on a website that was hosted by host gator. When the button on our future website of this project is toggled the WIFI

module will also toggle an L.E.D output showing that the proper package was sent. Finally, for April mile stone the test was for the compatibility of the finger print testing and also communicating with our solenoid lock, both proving successful connections.

7.2.1 - Embedded Software Testing

Testing the functionality of the MCU and its integration with components through software required a specific set up. To read or write over UART, or any serial communication, it was necessary to set the baud rate to 9600 with eight data bits, no parity bit and one stop bit. Shown below is the exact configuration we used within Tera Term. This allowed for the most effective communication between the ATmega2560 MCU and the components we've purchased that involve serial communication.

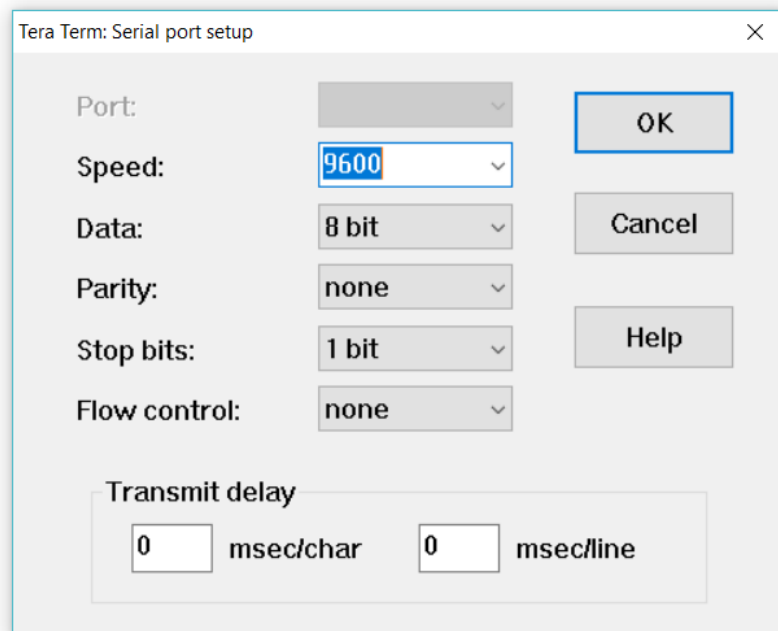


Figure 50 - Tera Term Serial Port Setup

Barcode Scanner Software Test

Using serial communication between the Wave Share Barcode Scanner and the ATmega2560 development board, we were able to scan a barcode and light up an LED if that barcode was considered valid. The Wave share barcode scanner had a very sophisticated because it had a built-in microprocessor which can compute many different barcodes. The barcode when it came in the mail was reading as a keyboard when connected to our computer through via U.S.B. When a text editor was opened, we scanned a common barcode it would read it and type it psychically on the text editor. We had to dive deep into the instructions to change it to a UART output for our ATmega2560 developers board. When this was done, we connected the bar code scanner to our board. While connected we had to test if the ATmega 2560 can run and output the proper scanned bar code. While conducting this test we had to run the barcode to many different variation and patterns of barcode to

ensure we can use this barcode scanner. When testing we also found out that there are 2 power modes for this device. One power mode is toggled when the button is pushed to engage the barcode scanner to an on position, this will ensure to reduce the amount of power draw.



Figure 51 - Valid Barcode Testing

Figure 51 above shows the barcode we used for testing, the ATmega2560 development board, and the Waveshare Barcode Scanner. The barcode was scanned with the barcode scanner and then processed by the MCU. After the MCU processed the barcode, the yellow LED was set to high to indicate that the provided barcode is valid. In Figure 52, shown below, the output of the COM5 serial port shows that the barcode scanner successfully picked up the barcode as well.

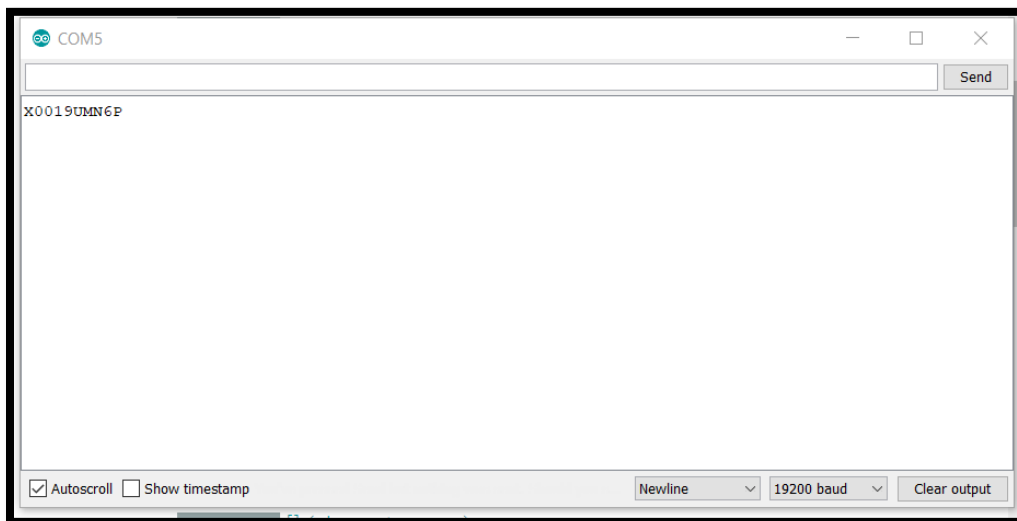


Figure 52 - COM5 Serial Port Response

Mechanical Lock Software Test

As shown in Figure 53 below, we were also able to test the ATmega2560's ability to detect whether the locking mechanism was locked or unlocked along with unlocking the lock as necessary.

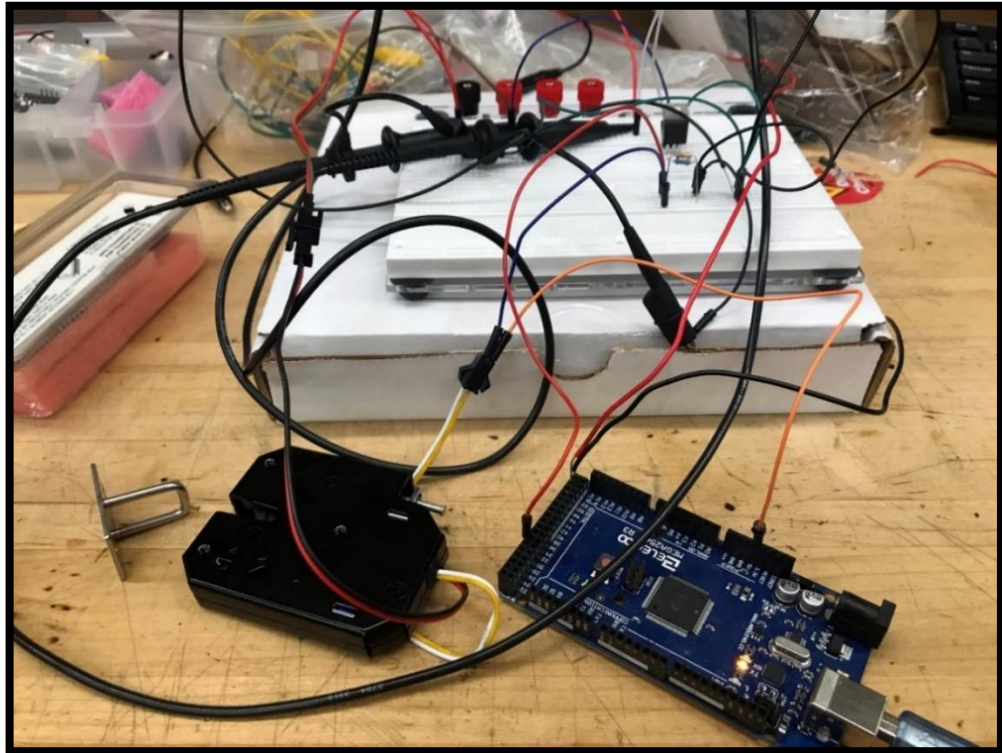


Figure 53 - Locking/Unlocking via Software

Figure 53 shows the locking mechanism in an unlocked state. This photo was taken after software uploaded to the ATmega2560 sent a signal to the lock to release the lock. To accomplish this, the software initiates the yellow LED as a output as well as the necessary pins to be able to receive and transmit to the locking mechanism. Pin 53 is used to transmit a high signal to the lock whereas Pin 31 is used to receive its status. The yellow LED is used to indicate the status of the box. The yellow LED set to high indicates that the lock is unlocked. The yellow LED set to low indicates that the lock is locked. As shown above, the lock is unlocked; therefore, the yellow LED has been set to high and is lit up. After the photo was taken and the lock was placed back into the mechanism, the software successfully detected its state and set the yellow LED to low.

ESP8266EX Software Testing

Testing the ESP8266 Wi-Fi module and its functionalities is shown below. Figure 54 shows a temporary local server created to test the functionality of the Wi-Fi module in conjunction with the ATmega2560. With 2 computer engineers in the group it was very helpful for their knowledge in having existing experience with this

chipset and also HTML background. A button was made to test if the Wi-Fi module was working and also sending out and receiving the packets from the ATmega2560. We ordered it to toggle a L.E.D to check if the testing has been correctly coded, and also giving us a clear sign that the Wi-Fi module is communicating with our ATmega2560. The server contains a simple button, Toggle LED, which sends an HTTP request which is processed by the ESP8266EX which then performs the action of toggling the on-board LED.

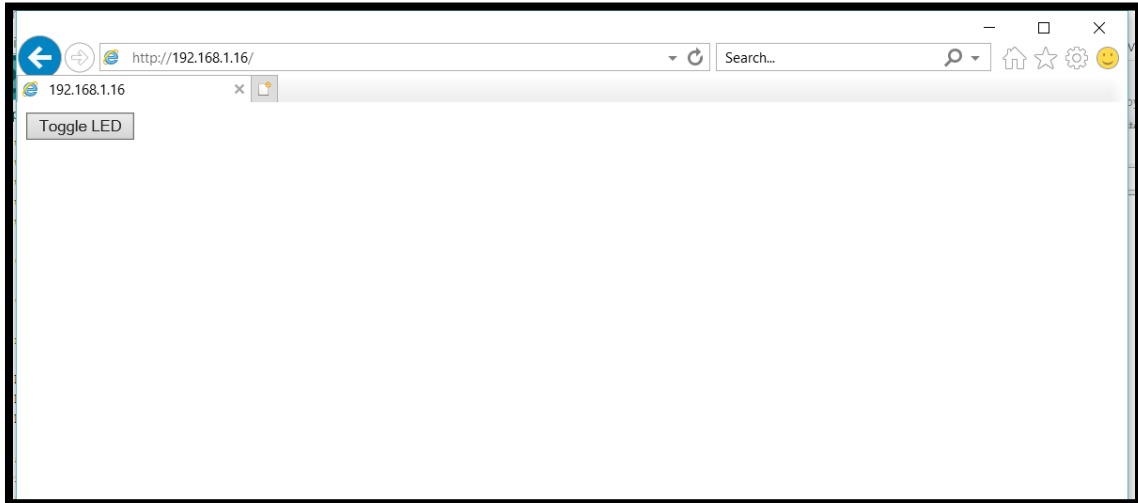


Figure 54 - Local Server

Shown in the terminal image, Figure 55, the Wi-Fi module was able to connect to a local Wi-Fi network, Alpha, and also host the HTTP server necessary for testing. This allowed for smooth communication from the browser and the on-board LED.

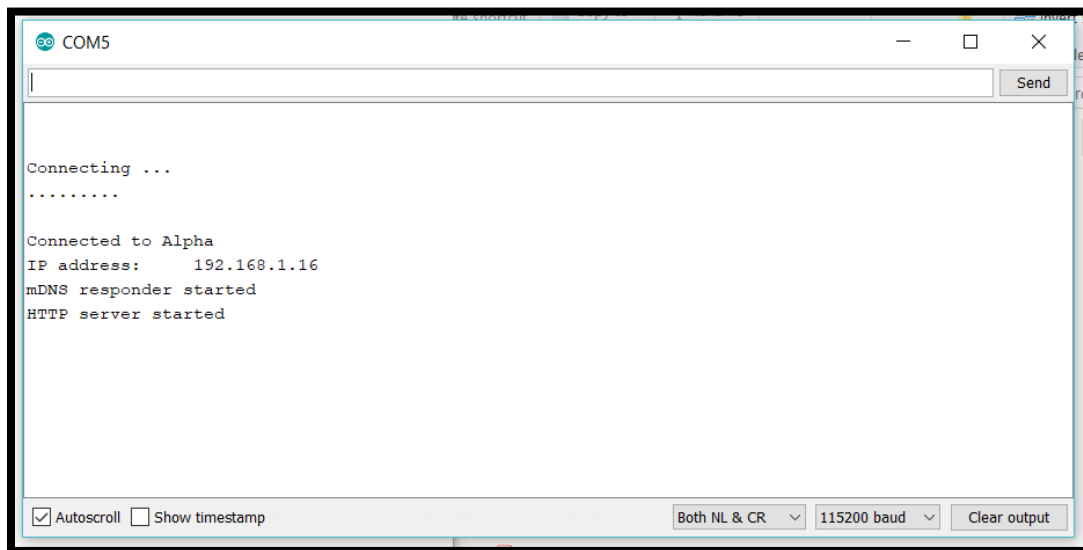


Figure 55 - Wi-Fi Module Test Terminal

For future implementation, we'd like the Wi-Fi module to connect to a server that is hosted online and respond to any command provided to it. Therefore, we can use the server as a medium between the Black Box and the iOS application by using HTTP POST or GET requests as previously mentioned.

Fingerprint Scanner Software Testing

As shown in Figure 56, we were able to test the functionality of the fingerprint scanner.

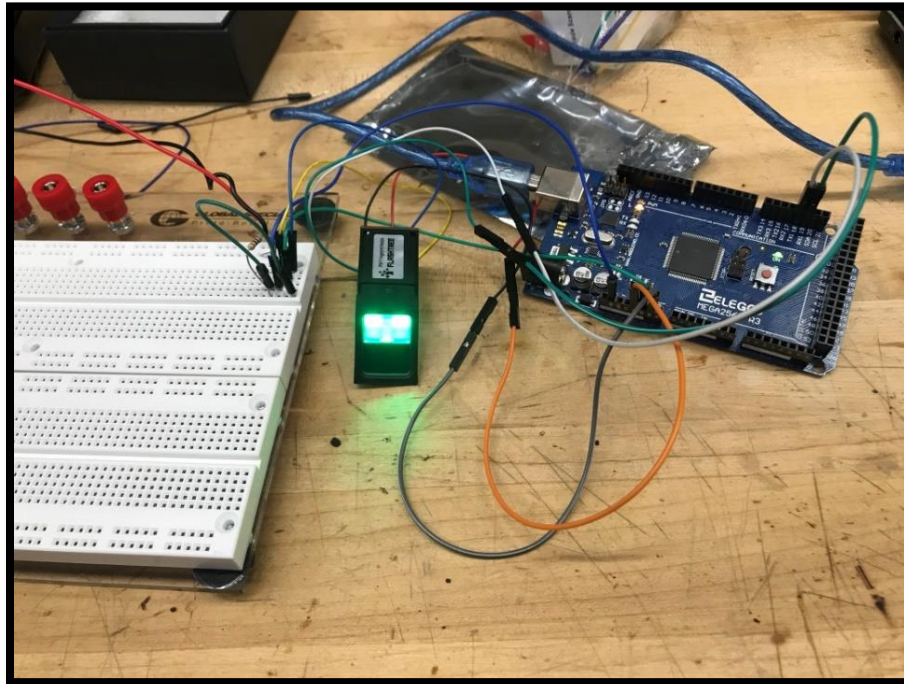


Figure 56 - Fingerprint Scanner Proof of Concept

Provided an input of one of the team member's pointer finger, specifically Jacky Li's, the fingerprint scanner was able to decode and save the fingerprint. The scanner was then used in conjunction with the ATmega2560 to determine whether the next scanned finger aligns with the fingerprint saved previously. We were able to detect Jacky's finger if it was placed on the scanner with accuracy levels provided in Figure 56 below. When testing out another team member's finger, specifically Adam Cuellar's, the scanner rejected the scan as it should have. We used this in conjunction with the mechanical lock to simulate granting or denying access to the Black Box's contents. This was done through the software by determining whether the fingerprint scanned has access to allow the mechanical lock to open. Using the same test subjects previously mentioned, the lock was only unlocked given Jacky Li's finger; whereas, the lock remained lock if Adam Cuellar's finger was scanned. When testing this finger print scanner, we had to figure out the hardest part was why it was not turning on. We figured out that we had to import the example code to turn on the finger print sensor. Once that was imported and ran a green L.E.D turned on and was ready to accept a new finger print. This also displayed the proper confidence level that is provided output below.

Table 42 - Fingerprint Scanner Testing Results

Test Number	Serial Response
Test 1	Found ID #1: Confidence level is 205
Test 2	Found ID #1: Confidence level is 77
Test 3	Found ID #1: Confidence level is 191
Test 4	Found ID #1: Confidence level is 221
Test 5	Found ID #1: Confidence level is 184
Test 6	Found ID #1: Confidence level is 162
Test 7	Found ID #1: Confidence level is 138
Test 8	Found ID #1: Confidence level is 201
Test 9	Found ID #1: Confidence level is 102
Test 10	Found ID #1: Confidence level is 102
Average Confidence	The Average confidence between the 10 test we conducted is 149.183

Table 42 shows the confidence levels of Jacky Li's finger being scanned. The input is coming from the fingerprint sensor to the ATmega2560's TX1 and RX1 pins. This signal is then processed by the software and retransmitted to the natural serial port of the ATmega2560 so the user can see what is going on. The confidence level for the device can range from 0 to 255. As depicted above, several of the attempts to scan Jacky's finger are well above the median confidence level; therefore, we can determine the device is reading the fingerprint accurately more consistently than not. For future implementation, we can use this information to determine what confidence level the software will consider a valid fingerprint to avoid fraudulent attempts at accessing the contents of the box. We can also set a limit on how many attempts a user will be able to complete before the box doesn't allow any more attempts for a certain period of time.

7.2.2 - iOS Application Testing

Testing the iOS Application and web server come hand in hand due to the nature of the application relying solely on the web server. The iOS application will be tested using the XCode IDE. The web server will be tested using Advanced Rest Client to ensure that our HTTP requests are implemented as intended. Both the MCU and the application will be used to communicate to the web server to ensure the efficiency of the handling of data.

To ensure that we're able to talk to the ESP8266, ATmega2560, and the sample server, we've created a sample iOS application that allows HTTP requests to be sent over the iOS device. The application sends HTTP requests just as Advanced Rest Client would do and can send/receive JSON packages. If a JSON packet is received, then it is properly decoded and processed by the application. If a JSON packet needs to be sent, then the necessary information is encoded accordingly. Using the app, we were able to tell the ESP8266 to communicate with the ATmega2560. The app was able to tell the board to set the built in LED to high or low depending on the user's preference.

We can use the information gathered by this testing to ensure that the aforementioned ideas for our project are in scope of what our iOS application will be capable of. Ideally, the application will be able to send and receive these HTTP requests to interact with Black Box using the web server as a medium. It will also be able to display all the necessary information being sent from the Black Box to the server.

An important factor to consider whilst using a web server as a medium is latency. The time it takes for the application or Black Box to communicate with the server or vice-versa could be a limiting factor; however, after observing the results from the testing described previously, we believe this will not be a major obstacle in development. Therefore, we persist in using the web server as a medium as long communication time remains insignificant in the grand scheme of the project. As well as observing, we plan on taking preventative measures in software to allow for quick searching and sorting of data within the web server's database. This will shorten the time it takes to transmit data between the Black Box and the iOS application.

The main application testing we will need to conduct is the user interface of the mobile application. We have to ensure that the average user will know where to navigate and how to use the application while ensuring the upmost detailed features. We plan to speed at least a couple weeks sending out to friends and family to test out the mobile application before submitting it for approval. By doing this it ensures that we can make any useful changes. Also, we have to make sure we do some testing on the actual feedback of the sensors and notifications it send making sure it is accurate and also in real-time.

8.0 - Administrative Content

Administrative planning is imperative to the development of any product. In order to create the most efficient Black Box certain administrative tasks must be drafted. Each of our group members has equal administrative privileges and responsibilities for the production of the Black Box.

This section includes:

- The division of labor between project members.
- Projected milestones and timeline for the project via a Gantt Chart.
- Overall finances and contributions by each group member.

8.1 - Milestone Discussion

This section will break down the milestones completed for the Black Box project design. The section is very helpful for our group to show where we should be at and the stages we should be during this time period. Due to the time constraints with the semester dates we decided to break down to a set of approachable milestones for this project. During the spring semester of 2019 we will conduct the design paper, researching similar designs on the market that is very similar to our project. We will also have to finalize our project command and conquer paper, which generally gives us an idea pitch of how we will attack this project full force as a team, with distribution of work.

Research the required parts will be needed for proper planning and also comparison charts with what will be the best components for the project. For the next step in senior design we have to finalize the parts selections with our team members, to make sure everyone agrees on the components that were selected. Once components are selected our team members can get started with the schematic design on eagle cad modeling software for scriptable electronic design, it will help us visualize how everything will be connected with each other. Then the schematic will be converted to a PCB design with computer aided features. Testing may occur during this process when the PCB design is also being worked on, it shows each segment of the schematic will give us an overview proof of design. Lastly once testing is complete, we can complete our design paper for senior design 1 wrapping up the 120 pages minimum required for this project.

The sequence will start from the initial idea thought processes stage which began in Senior Design 1 (Spring 2019 semester) to the final project presentation which will occur in the Summer 2019 semester in Senior Design 2. Where the PCB will be manufactured, and the main system will be setup, along with the applications design. The box will have to be planned for the placement of the various sensors and also the location of the PCB. A mobile application along with a website will also have to be implemented for the user to input the future tracking numbers for the box to read the bar codes. Finally, revisions of the PCB may occur due to some problems we may encounter while testing and loading in the code.

Table 43 - Senior Design 1 Milestone Table

Senior Design 1			
Number	Milestone	Completion date	Status
1	Project selection	Feb 1st	Completed
2	Divide & Conquer	Feb 1st 2019	Completed
3	Research requirements and preliminary components	February 19th 2019	Completed
4	Submit 60-page draft	March 29th 2019	Completed
5	Begin writing draft paper and early breadboard testing	March 29th 2019	Completed
6	Microcontroller	End of semester	Completed
7	PCB Layout	End of semester	Completed
8	Begin research app development/ build	End of semester	Completed
9	Recording & Data abstraction	End of semester	Completed
10	Submit 100-page paper, continue breadboarding	April 12th 2019	Completed
11	Revise components if needed, work on draft	End of semester	Completed
12	Completed project and paper	April 22nd 2019	Completed

Table 43 indicates our project schedule for Senior Design 1. Our goal is to adhere to this schedule as closely as possible. Completion dates include ample amount of time for both completion and revision of respective tasks. It is crucial that the aforementioned tasks are completed on time due to the time restriction in Senior

Design 2. Completing all of these tasks in Senior Design 1 will put us ahead in Senior Design 2. This will allow us to iterate on our product and make changes as we see fit as well as encounter and overcome any problems without a significant setback. Table 44 below shows our project schedule for Senior Design 2.

Table 44 - Senior Design 2 Milestone Table

Senior Design 2			
Number	Milestone	Completion date	Status
1	Test PCB design	TBA	In Progress
2	Mobile Application development	TBA	In Progress
3	Testing & Redesign	TBA	In Progress
4	Combine Work with people in the group	TBA	In Progress
5	Finalize prototype	TBA	Future
6	Peer presentation	TBA	Future
7	Final report	TBA	Future
8	Final Presentation	TBA	Future

Table 44 indicates an overall time line of what has to be done during the duration of Senior Design 2. First, we have to declare that our PCB is printed and tested in order to develop the next stage. If the PCB is not in working condition, due to the timing of summer semester we have limited time to brainstorm and also test what went wrong with our design if a failure were to happen. We also will add an extra week to the PCB Printing and mounting process, due to the turnaround time for the companies that will be providing the work for us.

Table 45 - Distribution of Labor

Senior Design 1			
#	Milestone	Primary	Secondary
1	Research components	Louis Rondino	Nathan Chong
2	Component gathering	Jacky Li	Adam Cuellar
3	Component testing and data logging	Louis Rondino	Jacky Li
4	Circuit design	Louis Rondino	Jacky Li
5	Early breadboard testing	Louis Rondino	Jacky Li
6	User interface design	Adam Cuellar	Nathan Chong
7	Mobile application	Adam Cuellar	Jacky Li
8	Begin research app development/ build	Jacky Li	Adam Cuellar
9	Recording & Data abstraction	Nathan Chong	Jacky Li
10	Schematic design finalized	Jacky Li	Louis Rondino
11	Database will need to be designed and also implemented	Jacky Li	Adam Cuellar
12	Html Website will have to be designed and implemented	Adam Cuellar	Jacky Li
13	Java script will also have to be designed and implemented	Jacky Li	Adam Cuellar
14	Mobile application will have to be published	Adam Cuellar	Jacky Li

Table 45 above indicates the distribution of labor for the development of the Black Box. Setting objectives and deadlines with both a primary and secondary contributor allow us to define responsibilities. Secondary contributors are provided to ensure the completion of the project under any circumstance in which a primary contributor can no longer provide effort.

8.2 - Project Management

Our team will make sure to keep in contact throughout the duration of this project. Adam Cuellar which is our computer engineer has volunteered to be our project manager for this project.

For the senior design paper, we had to share a document that we can all work on. It was thanks to our team members bringing up the idea to have a word share point collaboration document on word, this helped our project progress very successfully. We have learned during the divide and conquer that to have 4 separate documents and merging at the end when it is due is very difficult. For many of the research conducted we have uploaded many links and information for one another on a very well-known chat software called Discord and Google Drive. We held meetings once a week during the duration of senior design 1 and will be certainly meeting more during the duration of senior design 2. Many of these meetings were joined and collaborated over discord, we were on our microphones for hours at a time. We had to hold a meeting once a week to ensure the progress of the assignment, have discussions of our future plans and deadlines for each segment of milestones.

We also met with our two senior design professors through the semester ensure our progress of senior design. When meeting with them we had a list of problems we faced and hope for a very wise and helpful solution to our concern. When meeting with them we had to seek approvals for each checkpoints of our project. It was very vital for our group members to keep each other up to date when all four of us had very busy schedules, and also each of us had individual progress as well to remind the whole team each mile stone due date.

With the help of our computer engineering teammates have taken a class called Processes of Object-Oriented Programming, which is generally a software engineering course where they were taught how to generate a very sophisticated and detailed Gantt chart. With the help of the Gantt chart it helped the group and the project manager track its progress during the course of the project. During some parts of the project some of our team members did have some disagreements, however with the help of our project manager Adam Cuellar we solved the solution by having a presentation for both ideas and constructing a system for voting amongst the team. By doing this it shows that our project manager is very fair and also will give any great idea a chance. By doing all the things above, the project management during the duration of this project was very structured and it helped our group project to the next milestone.

8.3 - Budget and Finance Discussion

Below is the budget we've allotted as a team for the Black Box project. This is just an estimate and is subject to change throughout the completion of Senior Design 1 and 2.

Table 46 - Black Box Budget

Required Parts	Pricing (Market + \$5.00)
Metal Box	\$100.00
Keypad	\$30.00
Solenoid Lock	\$30.00
Thermal Sensor	\$8.00
Barcode Scanner	\$30.00
Wi-Fi Module	\$15.00
Battery	\$50.00
Siren	\$20.00
Power supply control/ AC to DC convertor	\$20.00
Backup battery	\$50.00
MCU	\$50.00
Paint	\$10.00
Waterproofing / packaging	\$50.00
Apple Dev software	\$100.00
PCB design, Final design	\$150.00
LED lights for viewing at night	\$25.00
Total	\$638.00 ± 200* = 838.00

*Total calculation added additional \$200 to ensure a higher budget for mistakes in PCB design and experimenting with components

Table 46 above depicts the planned allocated finances for the Black Box. Each member will contribute equally. In an incident in which more or less spending is required, we've incorporated a buffer of about \$200 to make sure we stay within a reasonable budget.

Financing Plan

For the Financial Plan of this project, Jacky is in control of most of the purchases throughout the process of senior design. With everyone on the team having to agree to split the project into four ways, we will share any financial burden created by this project including the manufacturing and cost of tools to complete this project equally.

Each member will be self-financed, if sponsors will be able to help us out it will relieve our burden with financial hardships and being able to buy better sensor components for our project. The microcontroller we picked out was the ATmega2560 it is very affordable compared to the Texas Instrument microcontrollers. The difference in price for the developer's board was about \$50.28 where we can use the money for revisions in the PCB or obtain extra parts they may be needed in the future of senior design.

The PCB is also considered a very low-cost option due the simplicity of the design that will require to be in running condition with our sensor components. We will have a Main Outdoors box that will be built with materials that are designed to be weather proof. We also will consider a silicon beading around the container to provide extra protection from the elements. For the PCB design, we plan to outsource the work to China. We have considered and planned out the duration of overseas turn around manufacturing time and also overseas shipping time. If the shipping time turns out to be too long, the other plan is to have OSH park to vendor it for us. We expect around a budget of 75 dollars for the first revision of the first print of our PCB design. We will then Outsource and populate the board with our close manufacturing plant, which was a suggestion from our senior design professor to use QMS. For the components we can request samples from companies that are willing to send them for free. We have already contacted Atmel for their microprocessor the ATmega2560. During the course of two weeks of waiting we got a response they are willing to support our project by sending us a Atmel starter kit of six ATmega2560 chips. We also allotted a maximum of \$638 dollars as our budget, giving us a \$200 dollar added to our projected for Revisions in PCB design and possible sensor malfunctions. This gave us a grand total of a maximum allotted \$838 dollars, with that in mind our group members will have to be comfortable Spending the maximum projected budget before proceeding this venture. The maximum split with 4 people will contribute a max dollar value if we utilize 100% of our budget will be approximately \$209.50. If this product was to be massed produced, this prototyping would be much higher due to higher end parts will be heavily tested and implemented to the possible designs for choosing the best one we think that would withstand the elements of mother nature. After figuring out the project we will consider doing, we have to seek approval with our senior design professors. Once we have approvals from our professors, we have to create a first prototype of our system. Once the first prototype is finished, we can then order the parts under one excel spread sheet to keep track of the expenses that will be made during this project. Many of the purchases will be

authorized with the agreement with everyone before purchasing each and every component.

Table 47 – Budget Allocation

SD1 Purchases as of 4/15/19	Price	Buyer
Wave share Barcode Scanner Module	\$46.58	Jacky
CC3220SF Board & MCU	\$71.52	Adam
Atoplee Lock	\$10	Nathan
ATMega2560 R3 Board	\$14.99	Jacky
HostGator	\$32.85	Jacky
22 Gallon Resin Deck Box	\$41.54	Jacky
Finger Print reader	\$18.99	Jacky
ESP8266 4 chips	\$13.99	Jacky
Brown Spray Paint	\$4.99	Jacky
Caulk silicone	\$3.49	Jacky
Total	\$258.94	

In Table 47 this indicates the current status of where we are with the purchases as of mid-April of 2019. We have purchased this Wave share barcode for \$46.58 from amazon, it was shipped through amazon prime shipped within 2 days. One important point to take note of is the presence of two boards bought as shown in the table above. The purchase of the CC3220SF Board was made prematurely resulting in a waste of money because ultimately, we decided to use the ATMega2560. The total expenses so far show promising results, since we predicted to spend about \$800. However, the expenses only show current ones which does not account for the future expenses needed to be made. This will include the PCB vending service fees as well as possible replacements for components. Emergency supplies may be needed for cases such as damage to current equipment from testing.

Black Box

Company Name

Project Lead : Jacky

Project Start
Tue, 1/15/2019

Display Week
6

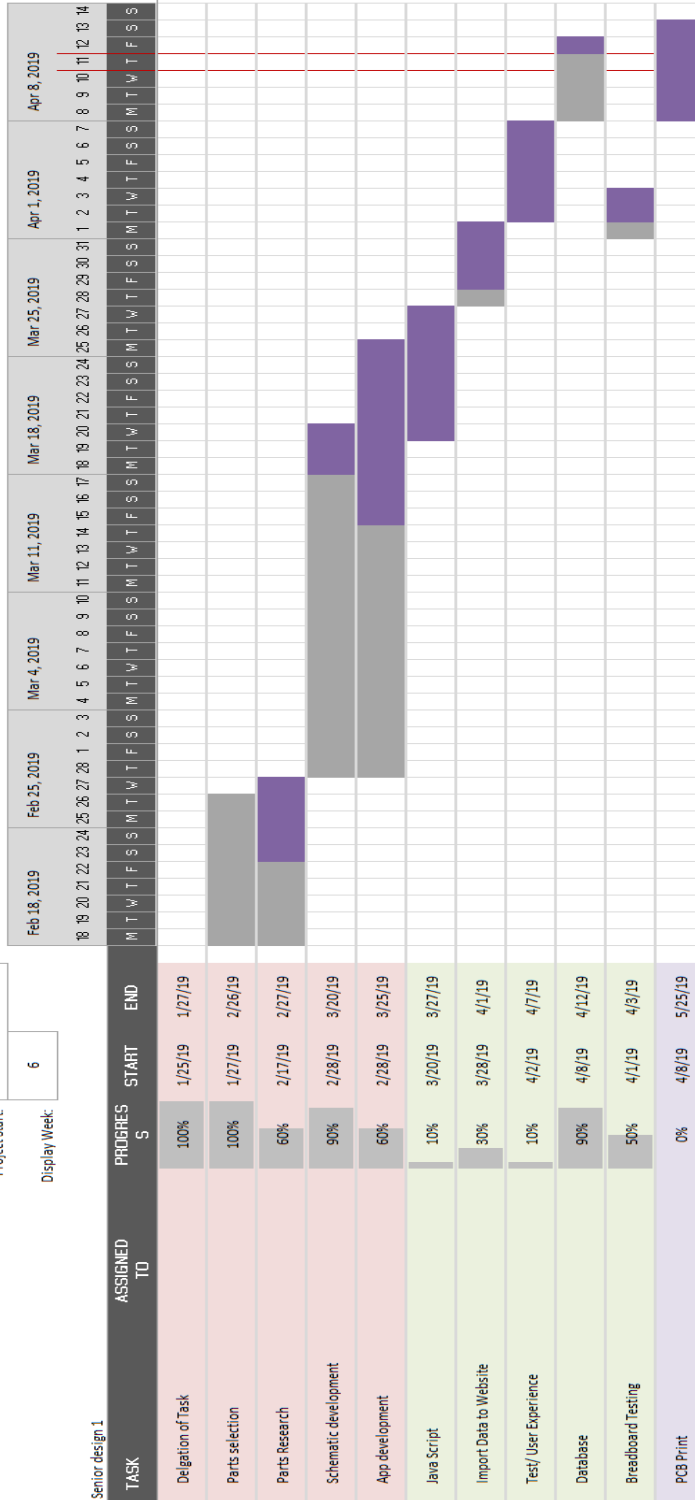


Figure 57 - Gantt Chart [Gantt]

Figure 57 displays the Gantt chart detailing the project milestones. This gives us a visual representation of what is to come, deadline after deadline.

8.4 - Stretch Goals

This section will describe the various features that could be a great idea to incorporate with Black Box project design. Each feature mentioned here are goals that can be considered or set, once the Black Box design has reached its' final production. If our team has managed to reach our goals during the duration of this project in our terms of completing all the sensors and peripherals in working order this is what we would implement more.

We are confident that we will have to try a couple of the listed stretch goals that we think as a team we can integrate with the timeline. When planning these goals, we had to see what the amazon locker and land shark had that we can implement.

Our goal of this project was to design a residential scale locker box for the average home user. With the RFID we wish to have an integration and partnership with vast delivery postal service companies where they all have RFID chips in their ID, therefore opening the box and notifying the owner of the package that this certain employee has successfully delivered their package. AC power was also a major brain storm in our group meetings, this provides the user to not worry about replacing the battery and also having a plug and play situation with the box if we implement a 3-prong plug.

RFID

Radio Frequency Identification (RFID) was a possibility for this technology to be incorporated with the Black Box in two possible ways. One way that our group thought of using RFID is by allowing the user to access the Black Box through RFID tags. The user can have a tag made just for him/her that will tell the MCU to send the required voltage to open the solenoid lock and thus, grant access for the user.

Another method in which RFID can be used is through giving access to mail delivery couriers such as USPS or FedEx. This would allow them to access the Black Box when they are delivering the package. They would have an RFID tag designed specifically for each courier.

RFID sounded like an intriguing idea; however, we did not want to include more means for the Black Box to be accessed. More accessibility methods can jeopardize security. Additionally, it is not practical to give courier services a unique RFID tag for each of them because if many people were to have this product, then there would be too many different tags that the couriers must carry.

AC Powered

Allowing the Black Box to use AC power is an idea worth considering. The current problem that the Black Box has is that if the user forgot to change the batteries, then how would the box open now, if the MCU will never be on to tell the lock to unlock? If the Black Box is powered through AC power, the user will rarely have to worry about powering the electronics. It is only in cases where a black out may happen causing an immediate shutdown to all plugged in electronics. This can be

avoided by using back-up batteries to continuously provide power to the MCU just in case.

However, if AC power is used, the user would have to find a way to plug the Black Box in an outlet without the plug being easily accessed from the outside. If the Black Box can be easily unplugged, then the power to it would be cut off, rendering it useless. Our group did not want to provide this type of inconvenience to the user; thus, the idea of using AC power was scrapped. Furthermore, it would make the power supply in the Black Box, a lot more complex and would be more expensive to implement.

Camera

Initially, the Black Box project idea had a camera sensor in mind; however, it was scrapped from the drawing board due to the complexity of how the hardware and software would be integrated.

The plan for the camera was to use it to take a picture when the box is opened by the deliveryman or if anyone else was trying to force himself/herself to open it. This would provide proof for the user to see if the package is delivered properly or provide proof of someone trying to steal a deliverable. For storing the picture, an SD card would be needed as well as a slot for it. It would be nice to have a camera functionality for increased security purposes; however, due to time constraints and the potential difficulty of integration, the idea was rejected.

Siren

A siren was another component that we thought of installing to the Black Box. If the siren were to go off, it would produce a loud audible sound that can be quite alarming and disturbing. The reason why our group wanted to implement this device is because we wanted to surprise and scare thieves who dared to tamper with the Black Box.

An accelerometer could be configured and be used to detect any rapid, violent motions that can occur when a thief is trying to force the enclosure to open. When these actions are detected, the sensor can communicate to the MCU and turn the siren on for a few seconds. It would be slightly complex for us to implement this device because we would not want the siren to go off during a delivery. To possibly avoid this situation, we can set a state where the accelerometer turns off when the correct barcode is scanned by the delivery man which would allow the Black Box to open. Once the Black Box is closed, then the accelerometer can continue measuring.

However, this would bring rise to a new problem. If the accelerometer was always active, how much power is being consumed? Would this sensor have a standby mode? Questions like these must be answered if this device were installed.

Amazon Alexa Support

The final idea we wanted to implement was having an Amazon Alexa support. As many people know, Alexa is a virtual assistant created by the largest e-commerce marketplace known as Amazon. The virtual assistant can be seen in popular

products such as Amazon Echo which is a smart speaker. Customers can ask their smart speaker by calling out Alexa's name and asking a question that a person would normally type in search engine.

With that being said, it would be a neat idea if Alexa can notify the user when a package has been delivered. Of course, this would be a special case in such a way the user is home, so he/she can communicate with their Amazon home product and if the user was unable to answer the door after the deliveryman rang the doorbell and thus, placed the package inside the Black Box. Other convenient uses can be Alexa notifying you about the current temperature and humidity that is inside the box. This would be convenient to know so the user can use this information to make a decision on how long they would want their package to be kept stored inside, if they are not present at the time. The idea of the Black Box having Amazon Alexa support works in conjunction with the Amazon delivery services they have. This can be a huge selling point if this product were to be fabricated professionally.

Another use can be asking Alexa to unlock the Black Box for the user instead of the user having to access the mobile app. The user can say, "Hey Alexa, unlock the Black Box", and the package protection system will be able to be accessed. Other uses can be asking the virtual assistant how much battery life the Black Box has or an estimated time that a package is going to be delivered. Of course, there are other smart home devices such as Google Mini which can also be neat if the Black Box has this type of support from Google's virtual assistant.

All these stretch goals have the possibility of being implemented to upgrade the Black Box into a more versatile package protecting machine. These ideas can always be put into action once the original plans for the project idea succeed. After all, we want to get the essential features that give life to what the Black Box represents and how it differs from the rest of the lockboxes. Once this has been implemented, then these ideas can be considered, and we will be looking for the best features that would largely benefit the Black Box

9.0 - Project Summary and Conclusion

With the motivation to design a device that will discourage and prevent thieves from stealing precious packages, the Black Box project idea was created. Throughout the semester, the four of us in this group have met continuously, engaging in work and ideas that would eventually come together to form the ideal design. We have reached an agreement that this project is our own unique design with special attributes. For other work/components that were pulled from online/external resources, their work has been appropriately documented as in the references page. Even though, in the past, there have been multiple designs of a lockbox in other Senior Design semesters, the Black Box has its' own unique features that separates itself from the rest.

It is reasonable to state that this paper has been well planned out and properly documented in such a way that details useful design procedures for a successful completion in creating the Black Box. Because of the state of this report, we know that it will provide a proper guideline in to successfully crafting our design to the way it needs to be and to the way it must operate.

All of the planning that went to finding necessary parts, creating a schematic that will integrate those said parts, and finding a vendor that can fabricate our PCB, have all been summed up in this paper as this will allow a smooth transition to Senior Design 2, which is when the Black Box will be produced. There will be obstacles and challenges that will make the production of the Black Box stressful; however, that is where the fun begins since it means that we will be challenging ourselves.

The four of us believe that the budget created for the Black Box is more than enough to make this project idea into reality. The physical build would be kind of tricky to implement, not because of the hardware, but because of where the hardware should be placed inside the box. It is undesirable for the electronics to get in the way of the package placement. This could risk possible disconnections of the hardware and MCU as well as potential components being damaged based on how the deliveryman would place it. Additionally, if a potential thief were to get violent with the box, it is important that the enclosure is durable enough to withstand the damage given, while also protecting the hardware from being damaged/disconnected.

As for the software perspective, it is going to be more time-demanding when compared to the physical labor. This is because of how complicated the code will be when it comes to connecting/communicating the hardware to Wi-Fi. Some issues that we thought of encountering are latency, frequent requests from server to update notifications, and incorporating low power modes to various features. On the topic of latency, we are concerned that the MCU will be stuck in a dead state when the Wi-Fi module experiences poor signal connection from a household router.

A server is used to communicate the Black Box with the mobile app and vice versa. Having poor connection or if the server is not responding well will leave the Black Box in a state waiting for a response. Another concern is the frequent requests from the server to update notifications. We want to use the mobile app to tell us information about a package delivery. To do that, the server would have to send a request from the MCU to know the occurrence. However, by sending frequent requests, this can potentially take up a lot of memory and power which is not ideal.

This problem branches off to the next concern which is enabling low power modes to various features. Enabling these modes can be quite difficult when trying to optimize battery life. We would have to be careful in choosing which power mode is needed. Our group would have to rely on what needs to be used for what needs

to get done, in order to successfully fulfill low power mode requirements and thus extend battery life.

Every invention has its' limits and the Black Box has no exceptions. This section of the conclusion will describe the overall design's limitations. First, the user MUST have a router that grants internet access. If the user does not have internet access, then the Black Box will not be able to communicate to the server and verify barcode numbers when a package is scanned by the deliveryman.

Additionally, to open the box, the user has two methods to grant access. He/she has the mobile app and the fingerprint scanner to open it. The deliveryman only has the barcode scanner to scan the package's barcode to verify access. This can be a problem if the user has a poor connection which can cause the verification to possibly take a long period of time. In worst cases, if the box somehow lost network connectivity or if the servers are not working, then the Black Box will continue to wait until the information is retrieved. This waiting time can take forever which will never cause the box to open, forcing the deliveryman to leave the package outside.

Another limitation that the Black Box has is the obstruction of opening the enclosure when battery life has been depleted. If the batteries are dead, how will the MCU communicate with any of the peripherals that signal the lock to unlock? Furthermore, the Black Box will be rendered inoperative permanently because if the user wanted to replace the dead batteries, he/she would have to open the Black Box which is impossible to do without powering the MCU. One way of how this situation can be avoided is by telling the MCU to open the lock if and only if the batteries are approaching depletion.

For example, the Black Box will open automatically when 5% of its battery life is left. Another idea that can work is by using rechargeable batteries that can be charged from an external port. A small section of the box can contain a USB charger input that the user can use to charge the batteries. These are possible solutions worth considering when our group begins to design the project.

Overall, our group believes that we have provided enough research, plans, and testing to ensure an accurate representation of the Black Box project idea. Pressing on towards the goal of creating this product will be continued in the next semester where all the parts will be put into demonstration. There will be obstacles and challenges ahead of us; however, if senior design has taught us anything, it has taught us to work together as a group to accomplish what is needed to be done which gives us a glimpse of how it will be like in the real world when working as engineers.

Appendix A Reference

- [ASM] ASM Educational Center, *CompTIA Network+ OSI Model*, Feb. 26, 2019. URL: <https://asmed.com/comptia-network-osi-model/>
- [Batt] Battery University, *Can the Lead-acid Battery Compete in Modern Times?*, Mar. 26, 2019. URL: https://batteryuniversity.com/learn/archive/can_the_lead_acid_battery_compete_inmodern_times
- [Batt2] Battery University, *BU-203: Nickel-Based Batteries*, Mar. 27, 2019. URL: https://batteryuniversity.com/learn/article/nickel_based_batteries
- [Batt3] Battery University, *Is Lithium-ion the Ideal Battery?*, Mar. 27, 2019. URL: https://batteryuniversity.com/learn/archive/is_lithium_ion_the_ideal_battery
- [Duck] George Duckett, *What's the Proper Soldering Iron Temperature for Standard .031" 60/40 solder?*, Mar. 28, 2019. URL: <https://electronics.stackexchange.com/questions/1980/what-s-the-proper-soldering-iron-temperature-for-standard-031-60-40-solder>
- [EEHe] EEHerald, *Online course on Embedded Systems*, Mar. 28, 2019. URL: <http://www.eeherald.com/section/design-guide/esmod12.html>
- [EL19] El-Pro-Cus, *How does Bluetooth Work?*, Mar. 3, 2019. URL: <https://www.elprocus.com/how-does-bluetooth-work/>
- [Gantt] Gantt, *Welcome to Gantt.com*, Apr. 11, 2019. URL: <https://www.gantt.com/>
- [Gib19] Kate Gibson, *"Porch Pirates" Steal Millions of Holiday Packages Each Year*, Apr. 11, 2019. URL: <https://www.cbsnews.com/news/porch-pirates-steal-millions-of-holiday-packages-each-year-how-to-not-have-yours-among-them/>
- [How19] How to Mechatronics, *How RFID Works and How to Make an Arduino based RFID Door Lock*, Feb. 14, 2019. URL: <https://howtomechatronics.com/tutorials/arduino/rfid-works-make-arduino-based-rfid-door-lock/>
- [iComp] icomputernotes, *C vs C++ in Hindi*, Apr. 3, 2019. URL: <https://www.icomputernotes.com/c-vs-cpp-in-hindi/>

- [Mous] Mouser Electronics, *Texas Instruments CC3220 SimpleLink Microcontrollers (MCUs)*, Mar. 18, 2019. URL: https://www.mouser.com/new/Texas-Instruments/ti-cc3220-MCU/?gclid=CjwKCAjw4LfkBRBDEiwAc2DSIKcljUjUy2E89LPAhVCxmgYnZf6hbnsQNQuwLedN2bjUz3g8UJlmchoCbD4QAvD_BwE
- [Powe] PowerStream, *Discharge tests of AA Batteries, Alkaline and NiMH*, Mar. 27, 2019. URL: <https://www.powerstream.com/AA-tests.htm>
- [RoHS] RoHS Guide, *RoHS Compliance FAQ*, Mar. 28, 2019. URL: <https://www.rohsguide.com/rohs-faq.htm>
- [Shoa] Wendy Shoa, *C++ Coding Standards and Style Guide*, Mar. 21, 2019. URL: <https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20080039927.pdf>
- [Stev] Steven Keeping, *Compare 2.4 GHz and 5 GHz Wireless LAN in Industrial Applications*, Apr. 3, 2019. URL: <https://www.digikey.com/en/articles/techzone/2017/jun/compare-24-ghz-5-ghz-wireless-lan-industrial-applications>
- [Ste19] Steve Sande, *Reminder: Keep Electronics Warm and Safe This Winter*, Mar. 21, 2019. URL: <https://blog.macsales.com/43145-reminder-keep-your-electronics-warm-and-safe-this-winter>
- [THSM19] Optimum, *Through-Hole vs. Surface Mount*, Mar. 28, 2019. URL: <http://blog.optimumdesign.com/through-hole-vs-surface-mount>
- [Trac] TrackingEx, *USPS Tracking*, Mar. 28, 2019. URL: <https://www.trackingex.com/usps-tracking.html>
- [Univ] University of Cambridge, *Soldering Safety*, Mar. 28, 2019. URL: https://safety.eng.cam.ac.uk/safe-working/copy_of_soldering-safety
- [Ver19] Verification Protocols, *SPI Protocol*, Mar. 28, 2019. URL: <http://verificationprotocols.blogspot.com/2017/05/spi-protocol.html>
- [Vis19] Peter J. Vis, *Soldering Temperature Chart*, Mar. 28, 2019. URL: https://www.petervis.com/Education/Soldering_Guide_for_Electronics_Students/Soldering_Temperature.html

Appendix B Permission To Reproduce

Cadex

Permission to use figures from site



Nathan Chong

Fri 4/19/2019 10:13 PM

To: answers@cadex.com ^



Reply all | v

Good evening,

I am an electrical engineering student enrolled in senior design at the University of Central Florida and I am writing to you to request permission to use some information from your paper "Can the Lead-acid Battery Compete in Modern Times?"

(URL: https://batteryuniversity.com/learn/archive/can_the_lead_acid_battery_compete_inmodern_times), "BU-203: Nickel-Based Batteries" (URL: https://batteryuniversity.com/learn/article/nickel_based_batteries)

, and "Is Lithium-ion the Ideal Battery?"

(URL: https://batteryuniversity.com/learn/archive/is_lithium_ion_the_ideal_battery) to our design paper that will not be published.

Thank you,

Nathan Chong

(407) 516-4940

nathanchong@knights.ucf.edu

Eeherald

To: editor@eeherald.com ^

Good evening,

I am an electrical engineering student enrolled in senior design at the University of Central Florida and I am writing to you to request permission to use some information from your paper, "Online course on Embedded Systems" (URL: <http://www.eeherald.com/section/design-guide/esmod12.html>) to use on our design paper that will not be published online.

Thank you,

Nathan Chong

(407) 516-4940

nathanchong@knights.ucf.edu

Elprocus



Nathan Chong

Fri 4/19/2019 10:32 PM

To: elprocus@gmail.com ↗



↻ Reply all | ▾

Good evening,

I am an electrical engineering student enrolled in senior design at the University of Central Florida and I am writing to you to request permission to use some information from your paper, "*How does Bluetooth Work?*" (URL: <https://www.elprocus.com/how-does-bluetooth-work/>), to use on our design paper that will not be published online.

Thank you,

Nathan Chong
(407) 516-4940
nathanchong@knights.ucf.edu

Mouser

To: sales@mouser.com ↗

Hello,

I am an electrical engineering student enrolled in senior design at the University of Central Florida and I am writing to you to request permission to use some information from your paper, "*Texas Instruments CC3220 SimpleLink Microcontrollers (MCUs)*" (URL: https://www.mouser.com/new/Texas-Instruments/ti-cc3220-MCU/?gclid=CjwKCAjw4LfkBRBDEiwAc2DSIKcljUjUy2E89LPAhVCxmgYnZf6hbnsQNQuwLedN2bjUz3g8UJlmchoCbD4QAvD_BwE) to use on our design paper that will not be published online.

Thank you,

Nathan Chong
(407) 516-4940
nathanchong@knights.ucf.edu