University of Central Florida

College of Electrical and Computer Engineering

Signal Operated Lock And Security system

(SOLAS)

Senior Design II Final Project Documentation

Group 16

Devon Anselmo - Computer Engineering
Matthew Guevara - Electrical Engineering
Keanu Zeng - Computer Engineering

Table of Contents

List of Figures

List of Tables

1. Executive Summary

One of the daily hassles of life is trying to unlock and open the front door when already carrying your items from the day. It might be groceries from the store, laptop and books from work, school project, or some boxes while moving. No matter what the items are, opening the door while trying to manage all of the other items always results in dropping either the keys or the carried items; and after a long day, that just makes a frustrating task even more frustrating.

Our project, the Signal Operated Lock And Security system, or SOLAS, strives to eliminate this frustration. Arriving home should be the time of relaxation after a long day of hard work. The main goal of this project is to introduce an easier way to open the door. With regular house doors, as you walk up to the door carrying items in both hands, you have to shuffle everything to one hand while you fish keys out of your pocket, and you end up losing everything else in the pocket, then holding everything awkwardly while you use the key to unlock and open the door. The SOLAS system reduces the need for a physical key almost entirely, instead using RFID as its "key". Each user will wear a light bracelet embedded with an RFID tag. Whenever they come near enough to the door, the door automatically unlocks, then only requiring the user to open the door which can be done while carrying items in both hands. This reduces the need to unlock the door with a physical key, although the door lock will still have a keyhole in case the battery dies. This however will not happen after at least a year (target), so the user can replace the system's battery only occasionally and thus still never have to use a key.

A door lock system would not be complete without added security though, and the SOLAS system has several security features for the user. The first feature is the gesture controller to be used inside of the lock system. When the user approaches the door, the system will first check if any accepted RFID tag is in the vicinity. If there is, a gesture controller will read in hand movements from the user which act as a passcode to unlock the door. If accepted, the door will then unlock. Of course, having to input hand gestures to open the door goes against the accessibility objective of the system, and so this feature will be customizable by the user. If they prefer more security, they can program the gesture to controller to take a long series of complex gestures before unlocking the door. If the user is more concerned with accessibility, they can program the gesture controller to take a very simple password that can be input even while carrying many items. For example, the password could be moving a hand towards the left in front of the controller, then back to the right towards the handle which has to be opened anyways, hardly increasing any effort from the user. If desired, the user can disable this feature entirely, and the RFID system will still provide sufficient security.

Another feature the SOLAS system will have is a camera linked to a website to monitor the use of the lock and any activity in the porch area around the lock. The SOLAS system comes equipped with a short-range proximity sensor which will detect movement, notifying the rest of the system when it does. Whenever movement is detected, the camera will take a picture and post it to a website that the user can login to. Not only does this allow the user to monitor who is approaching their house, but also deters any package delivery theft, or provides evidence in case it does happen. In the event that someone approaches the door, an acceptable RFID tag is sensed, but the gesture controller reads in an incorrect password, a picture will be taken, and when posted to the website it will be red flagged. This allows the user to scrutinize this picture more carefully and ensure that it was only an accidental incorrect password and not perhaps a stolen bracelet with an intruder attempting to gain entry.

2. Project Description

The SOLAS system has many objectives to be fulfilled in order to meet its goals and features set by the team. The main objectives concern security, accessibility, and ease of use, which are covered in the next sections.

2.1 Motivation

The motivation for this project came mainly from the desire of easier accessibility to the house at night. Most days coming home, many items like lunchboxes, water bottles, and laptops are being carried, making unlocking the door difficult. Additionally, when trying to get keys out of the pocket, the other items like pens always come out with the keys without fail. This only makes the process of getting into the house harder, not easier.

2.2 Inspiration

The inspiration for how to solve the problem, and thus the basis for this project, came from two main areas. The first is the locking system on the newer generation of cars that senses when the keys are nearby. When the user puts their hand on the front door handles of the car, all of the doors unlock automatically. This completely eliminates the need for them to ever have to deal with using the keys to either manually unlock the doors one by one or use the key fob to electronically unlock the doors; when approached the car door or trunk unlocks automatically. While it is not 100% hands free, it is still much easier to use and makes life much easier.

The other area of inspiration comes from current electronic door lock systems. There are many door lock systems that require no physical key, but instead had a keypad to enter a numeric password. This keypad makes entering the house much easier than using a physical key. In comparison, using physical keys to open a door is much more inconvenient and almost outdated technology.

2.3 Objectives and Goals

Our project looks to encompass features from both areas of inspiration into an optimal door lock system featuring accessibility and security. The system will use RFID to detect authorized users, similar to the way newer cars do with the keys that the owners carry. It will also require a passcode of sorts like other electronic door locks but using gestures rather than numbers.

2.3.1 Accessibility

The main objective of the SOLAS system is increased accessibility to authorized users to unlock a door. This will be achieved via the RFID system set up between the user's bracelet and the door lock system. Inside of the bracelet will be a passive RFID tag that will communicate with an RFID reader inside of the door lock system. Once the RFID reader reads in data sent from the bracelet, it will send it to be analyzed to the microcontroller, which will determine whether the tag is one

used by an authorized user. All of this is done with no effort from the user, making access through the door easier.

2.3.2 Security

Another focus of SOLAS is to create a door lock system that is more secure than other similar products. This is made possible through 3 separate security features. The SOLAS system will include an integrated camera, an RFID bracelet, and a gesture-based password. The strongest layer of security in the system will be the RFID system that only accepts the specific tag in the bracelet. Without the bracelet, entry through the door is not possible. The second layer is the optional gesture-based password. Even if somehow the bracelet is brought to the door by someone with ill intentions, if they do not know the gesture-based password by the rightful user, they will be denied entry. The final layer of security is the camera, which allows the user to monitor all usage of the door lock and immediate area.

Combining all three of these layers of security, this project is designed to be as accessible but more secure than other similar products on the market. The door is more secure due to the fact that with the pin pad-based door locks, it is still possible for children of the user to tell their friends the code in confidence, but then the code is used for intruders to get in, unbeknownst to the users. Even without knowing the code, it is possible (albeit very unlikely) for an intruder to guess the password and gain entry. With RFID and gesture controls, no intruder can gain entry through the door.

2.3.3 Website

Another objective for the project is to have an easy-to-use and intuitive website for the user to be able to see the results of the SOLAS system. The camera integrated into the door lock will take pictures of any relevant movement in the surrounding area as well as whenever an incorrect password is read in by the lock system. Since the SOLAS system is designed with accessibility as well as security in mind, a website will be developed for the user to be able to view these pictures of possible suspicious activity so that they can determine whether they need to take any action. Not only will these pictures allow the user to see possible intruders, but it will allow them to view package thieves, an ever-present issue.

The website will be designed with simplicity in mind for the user to easily access all pictures posted by the camera, requiring only a logon, name, email, and an identifier associated with the camera. Since multiple members of a family might be interested in seeing daily activity from the camera, multiple accounts for each camera will be allowed; the camera will post all pictures to all associated accounts.

2.3.4 Comfort

Since gaining entry through the door requires the user to always wear a bracelet, that bracelet should be comfortable for the user to wear. If the bracelet is uncomfortable or bulky, that will reduce user satisfaction and accessibility to some degree. Therefore, the bracelet will be made as light as possible and as small as possible. The bracelet should not be any heavier or any larger than an average watch and will have no features other than a small RFID tag in order to meet this objective.

2.4 Requirement Specifications

This section establishes the constraints that will need to be met to create an acceptable product and help create focus on the scope of the project and establish design goals.

2.4.1 Hardware Specifications

When designing SOLAS a few hardware specifications must be met to achieve the basics of what this project will accomplish. These specifications are shown in table 1 below:

Table 1: Hardware Specifications

Hardware Specification	Units
How long it takes to open the door (from movement detection)	10 seconds
The RFID reader will detect the tag at a specified distance	18 cm
The proximity sensor will detect motion and wake up the rest of the system at specific distance	1.22m
An adjustable bracelet to house RFID tag, low weight	14 - 20.3 cm, < 4oz
The SOLAS system will have long battery life	1+ year
The system will enter low power mode after inactivity of no RFID tag read or no gesture read	10 seconds
Lock dimensions	4 x 6.6 x 12.2 cm
Power drawn	10-15 watts

2.4.2 Software Requirements

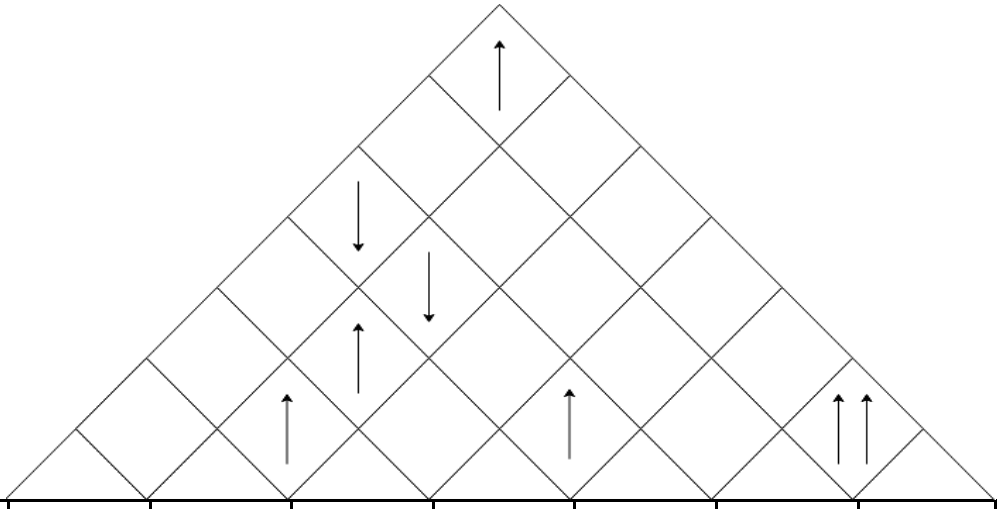
Similarly, requirements must also be established for the software side of SOLAS. These requirement are summarized in table 2 below:

Table 2: Software Requirements

Subsystem	Requirement
Website	A web application will be available for users to login, view, and manage pictures captured by the camera.
Website	The web application will have a forgot password feature with a security question and email verification to restore user access to their account.
Website	The user's password will be hashed into the database for security.
Lock	A gesture-based password in combination with the RFID tag will unlock the door.
Lock	Use of LEDs to indicate incorrect passwords, correct password, and system is powered on and active, or inactive.
Lock	The system should be in low-power mode until motion sensors detect movement and then it becomes active and ready to detect the RFID bracelet.
Lock	If motion is detected but no bracelet detected within another 10 seconds, the system will return to standby mode to save power.
Lock	After motion and bracelet are detected, if no password is read in within 10 seconds, the system will return to standby mode waiting for motion to save power.

2.5 Marketing and Engineering Requirements

The house of quality shown in figure 1 below outlines the user-based objectives of the project versus the technical requirements to meet those objectives. The main objective is accessibility, which only relates to the overall cost of the project; adding more sensors or other features to make it more accessible will increase the overall cost. The second focus is the security of the system. Making the system more secure will most likely increase the cost, increase the lock dimensions, increase how long it would take to install and setup the system, and increase the power drawn by it which reduces the battery life.



		Cost	Lock Dimensions	Installation Time	Power Drawn	Battery Life	Bracelet Size	Bracelet Weight
		-	-	-	-	+	-	-
Accessibility	+	↓						
Security	+	↑	↓	↓	↓	↓		
Bracelet Comfort	+						↑↑	↑↑
Cost	-	↑↑	↓		↑	↓↓		
Picture quality	+	↓↓			↓↓	↓↓		

Installation ease	+		↑	↑				
Targets for Engineering Requirements		<$300	1.6 x 2.6 x 4.8 inches, < 1lb	<30 min	10-15 watts	≥ 1 year	<5mm	<4oz

Figure 1: House of Quality

3. Research Related to Project Definition

In order to make an electronic door lock system comparable to those already on the market, the team researched said products on the market and how they worked. Research was done into how the current products are designed, what parts are needed, and what features they advertise for their product. A lot of focus was put into the RFID subsystem for the door lock, since that is going to be one of the main aspects of the SOLAS system, as well as the gesture controller.

Since the SOLAS system is largely hardware based, a lot of research was done for parts selection, making sure the parts were easily obtainable, will help the project meet requirements, and work together. The user website was decided to be simplistic and so it will not be difficult to design and develop the site and database required.

3.1 Existing Projects and Products

Products

There are many existing smart locks that saturate the market. Each offering the ability to unlock doors without having to use a physical key. In this section we discuss a few of them.

3.1.1 August Smart Lock

The base August smart lock starts off at a price of $149.99. It features easy installation without a need to remove the existing deadbolt on a door. Control and configuration of the smart lock is made possible through Bluetooth and a smart phone with the August home app installed. With this software installed, the smartphone is now the key to unlocking and locking the August smart lock automatically. Bluetooth connection, Wi-Fi, and GPS enables the software to track the user's location this lets it know when they are not home or at home. When the user is not home and comes back home a feature called Auto-Unlock is active. This feature automatically unlocks the door upon arrival home without having the user take out their phone and opening the app to unlock manually.

The downside to the August lock is that Wi-Fi connectivity is not possible with the base device. A separate device that plugs into an electrical outlet called the August Wi-Fi bridge must be purchased for $79 or bundled with the smart lock for $199.99, but an upgraded August lock model that includes built in Wi-Fi is also available for $249.99. Without the Wi-Fi bridge for the base-model notifications will not be received when you are not in Bluetooth range of the device and unlocking/locking is not possible when not in range also. It simply offers the convenience of managing the August lock while away from home. The reasoning for having a separate device for Wi-Fi could be to reduce footprint and save power as Wi-Fi consumes more power than Bluetooth.

Figure 2 below shows an August smart lock attached to the door. The August lock is installed only on the side of the door that faces the inside of the house. The existing cylinder case for the keyhole is not modified.

Figure 2: August Lock, permission to use granted

3.1.2 Level Bolt

The Level Bolt is a minimalistic smart lock design that replaces the deadbolt and hides it from view. This product features control of the lock through a smartphone app, auto locking after a slight delay, auto-unlocking upon returning home, guest access through email passes, and activity monitoring. Its design includes a hollow deadbolt for the battery to be inserted and advertises its durability with claims of 1,000,000 cycles of stress testing. At a price of $229 the Level Bolt combines a discreet, minimalistic design with convenience and security.

The Level Bolt includes Bluetooth connectivity, but not Wi-Fi. This is understandable to maintain its small footprint and 1-year battery life with a single CR2 battery. Wi-Fi is possible through a third-party device such as the Apple HomeKit software and hub. The HomeKit Hub allows for access to the lock without being in Bluetooth range, voice control to lock and unlock with a smart assistant and setting a schedule for when to unlock. Figure 3 below shows the Level bolt's design. It all fits inside the doorframe replacing the existing deadbolt and hiding the fact that it is a smart lock.

Figure 3: Level Bolt, permission to use granted

3.1.3 Wyze locks

The Wyze lock is a smart lock similar in design to the August lock, but for a cheaper price point of $99.99. Wyze also includes a device that enables Wi-Fi communication between the lock and the smart phone app. The lock features auto locking/unlocking, sensors to detect whether the door is open or closed, Sharing lock access, easy installation over an existing deadbolt, and voice support with a smart assistant. The figure below shows the Wyze lock and its blocky design like the August lock.

Figure 4: Wyze Lock, permission requested

Projects

The projects listed in the subsections below are past UCF senior design projects that are similar in design to what SOLAS is. The HOMES and SHES projects are compared to SOLAS to discuss any differences and similarities in their designs.

3.1.4 Home Observable Monitoring Entry System (HOMES)

HOMES is a senior design project from Spring 2015 by Colleen Caffey, Bruno Calabria, and Ricardo Georges. The goal of HOMES is to provide a home security system with the addition of a smart lock for the front door. It can monitor all entry points of a house such as windows, back doors, and front doors and send notifications to the smart phone app or web app whenever they are open. HOMES features many ways for users to enter their home from the front door. Unlike SOLAS which will have two methods for entry, HOMES has four. The smart lock system uses facial recognition, fingerprint scanner, Bluetooth wearable device, and smartphone application for entry. It also has an LCD touch display on the outside of the door, which displays various options for guests to select such as ringing the doorbell, leaving a message, or selecting a specific method to unlock the door.

HOMES is very similar to SOLAS with both having a motion sensor activated camera to take pictures of the front door, a wearable RF key, and a web application. The main difference between these two is that HOMES focuses more on security with small PCBs attached to all entry points of a house and a front smart lock, whereas SOLAS focuses only on the front door lock.

3.1.5 Security Hands Free Entry System (SHES)

SHES is a senior design project from Fall 2011-Spring 2012, Group 17, consisting of Anh Loan Nguyen, John E. Van Sickle, Jordan K. Acedera, and Christopher Spalding. Its primary goal is to design an affordable and convenient device. This system emphasizes a hands-free approach to unlocking a door. Instead of just unlocking the door it uses RFID technology and voice recognition to unlock and automatically open the door for a predetermined amount of time before closing again.

Compared to SOLAS, SHES is relatively simple. It is not connected to the internet or any other smart device. It is just a standalone, isolated system. Below is a figure of SHES[42] with the RFID reader at the top right of the door. On the other side of the door is the electronic door latch that will open the door automatically.

Figure 5: SHES Door

3.2 Relevant Technology

In order to gain more knowledge on the subject, research was done into the underlying technologies needed to construct the SOLAS system, including RFID and Bluetooth. This was so the team can select parts better suited for the goals of the project.

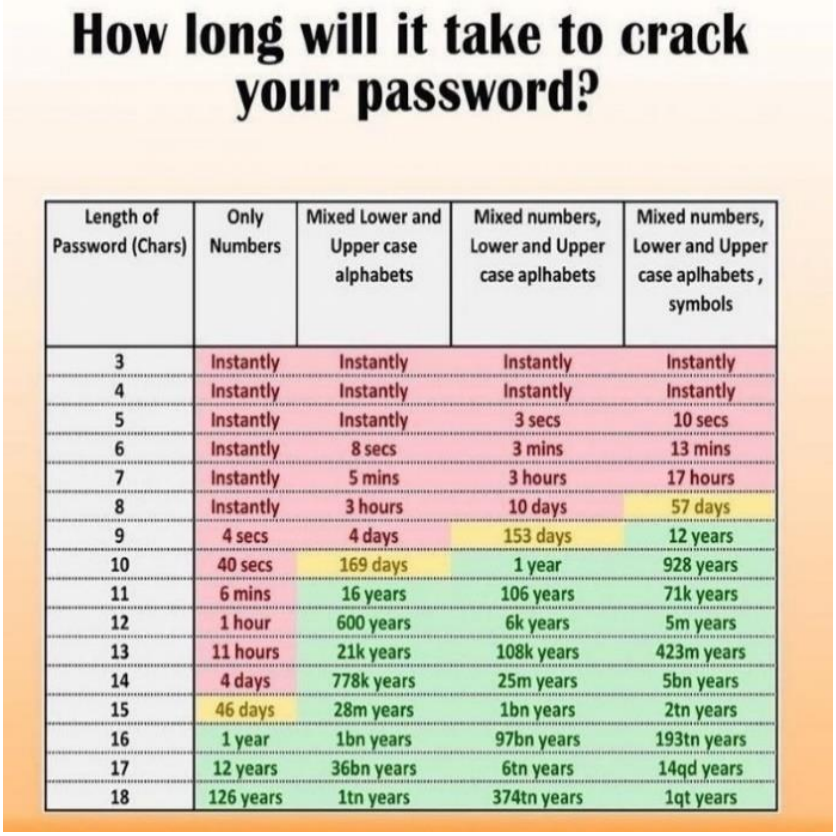
3.2.1 Password hashing

When the user first registers with the SOLAS website, they will be required to input an email, password, first name, last name, as well as the unique serial number for their camera. Once they login, the database will check their email and password versus all existing accounts. However, storing the plain text password directly in the database would not be very secure, thus in this project password hashing will be used. The password hashing algorithm to be used is bcrypt. This algorithm is very effective as it has two steps in hashing the password before it is ever stored anywhere. The first step is to take the user's input password and add a salt to it. The salt is a

predetermined number of bytes (set by the user) of randomly generated characters which are added to the plain-text password. Once this is done, that resulting string is then put through the hashing algorithm and then stored in the database. Not only does the salt add to the overall length and complexity of the password, but it makes it so that two users who use the same password will not have the same apparent password in the database[1].

This form of password hashing is very effective for additional reasons; the salt that is added to the password can be increased or decreased in length, increasing the complexity of a user's password without the user having to input a 'strong' password. Figure 6 below shows how each added character adds to the strength of a password exponentially. Since bcrypt uses the Mixed number, Lower-case and Upper-case alphabets, and symbols category, adding even a few salt characters will make the password of users strong and increase the security of the product.

The hashing algorithm is also only a one-way algorithm. The hashed passwords that are stored cannot be reverse hashed or broken in this way; rather when the user attempts a login, the password they type is hashed and then compared to the password already in the database, and comparisons must be done using another bcrypt function, and are not done using manual string comparisons.



Length of Password (Chars)	Only Numbers	Mixed Lower and Upper case alphabets	Mixed numbers, Lower and Upper case aplhabets	Mixed numbers, Lower and Upper case aplhabets , symbols
3	Instantly	Instantly	Instantly	Instantly
4	Instantly	Instantly	Instantly	Instantly
5	Instantly	Instantly	3 secs	10 secs
6	Instantly	8 secs	3 mins	13 mins
7	Instantly	5 mins	3 hours	17 hours
8	Instantly	3 hours	10 days	57 days
9	4 secs	4 days	153 days	12 years
10	40 secs	169 days	1 year	928 years
11	6 mins	16 years	106 years	71k years
12	1 hour	600 years	6k years	5m years
13	11 hours	21k years	108k years	423m years
14	4 days	778k years	25m years	5bn years
15	46 days	28m years	1bn years	2tn years
16	1 year	1bn years	97bn years	193tn years
17	12 years	36bn years	6tn years	14qd years
18	126 years	1tn years	374tn years	1qt years

Figure 6: Cyber Partners password security, permission requested

3.2.2 RFID

For the SOLAS system to be secure, it was decided to use RFID as one of the "keys" to the door. Each user of the door lock or resident of the house will wear a bracelet that has an RFID chip in it corresponding to that specific door lock. The RFID technology is an acronym for radio-frequency

identification and is part of a sector of technology known as Automatic Identification and Data Capture (AIDC)[3]. Using this technology, objects can be identified, analyzed, and in our case, either approved or denied without any effort from the user.

The RFID system consists of an RFID tag and antenna in the object to be identified, as well as the RFID reader. The RFID reader sends out radio waves requesting data from the RFID tag. The RFID tag sends identifying data back through radio waves which the reader interprets and stores. The data collected by the reader is then further analyzed by a host system which will either approve or reject the tag.

The RFID tag itself can be passive or active. The passive form has the chip, antenna, and substrate. The chip contains all of the necessary data and can be read-only, write-once and read-many, or read-write. The antenna is responsible for absorbing the radio waves from the RFID reader, and then sending its own data back. This antenna uses the energy from the reader's radio waves to send back its own data, thus not requiring any energy supply. Larger antennas can receive and send data from a longer range while the smaller antennae have a shorter range. The low-high frequency antennae have a coil shape due to these frequencies having magnetic properties, while ultrahigh-frequency antennae or more cylindrical in shape due to the electrical properties of those frequencies[4]. The low-high frequency spectrum is used with the passive tags while active tags and antennae use the UHF side of the spectrum, shown in figure 7 below. Finally, the substrate holds these pieces in place on the tag, popularly made of mylar or plastic.

Active RFID tags contain the same 3 components as passive tags as well as power supply and onboard electronics. The power supply enables the tag to constantly send out a signal, as well as allow it to reach a further range than the passive tags. The onboard electronics differ from product to product, mainly including sensors or processors to read in and analyze local variables depending on the purpose of the product.

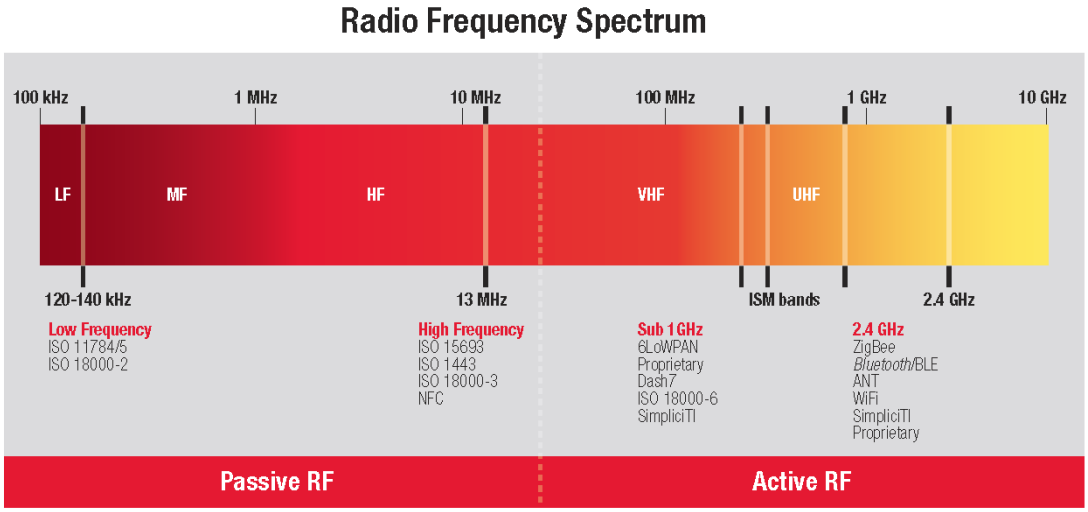


Figure 7: ABR Radio Frequency Spectrum, permission to use granted

3.2.3 Bluetooth

When considering the main entry method, Bluetooth versus RFID was a tough decision. It was decided that RFID would be used in the SOLAS system, focusing slightly more on security rather

than accessibility in the system. When considering RFID, the tag has a very specific identification number, normally either 32 or 64 bits, that are sent at a specific frequency to the RFID reader in order to be accepted. In order to be spoofed, where a malicious user steals a “password”, they would have to intercept the same signal while it is being sent and then send that same signal to the reader module at its frequency. If passive RFID tags are used, this would be very difficult to do with the short range required.

Bluetooth however can be a much less secure connection. If the SOLAS system was designed so that a mobile phone connected to the door lock during setup and/or during other times, then almost any mobile device would be able to connect to the system, requiring an additional layer of security for this feature that is supposed to be the main security feature. Additionally, if the SOLAS system relies on a Bluetooth connection from a smartphone to unlock, if the user’s phone has died on the way home, then additional measures would have to be made so that the user still has a way to enter, which would most likely reduce the security of the system further.

3.2.4 Pulse-width Modulation (PWM)

PWM is the technique of sending digital signals in bits and pieces to reduce the average power consumed. The digital signal it produces creates a wave form that can vary in its width depending on what is called the duty cycle[10]. A duty cycle is the amount of time the digital signal is on described as a percentage. For example, a duty cycle of 50% where the signal is on half of the time would create an ideal square wave form and a duty cycle of 100% would just have the signals on high the entire time. This continuous on and off cycle of the digital signal is what reduces the average power. In figure 8 below it showcases how the duty cycle affects the waveform of the digital signal.

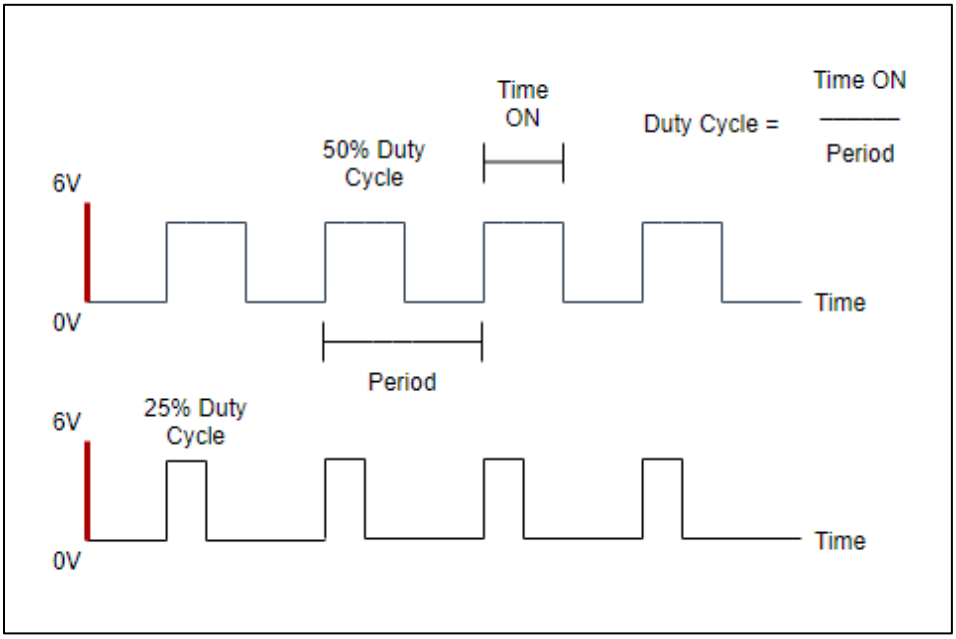


Figure 8: PWM Duty Cycles

PWM is not only used to save power, but also to mimic analog signals. By pulsing the signal on and off an average voltage equivalent to the analog voltage is produced. For example, a 6V signal

with a 50% duty cycle will simulate 3V. This technique allows microcontrollers to drive devices such as motors and LEDs. For example, an LED driven by PWM: one would expect the turning on and off of the LED would just result in flashing the LED. That is true if duty cycle is low enough, but with a high enough duty cycle and frequency the LED will flash so fast it is imperceptible to the human eye and will look as if it is just on.

For SOLAS, PWM will be used to drive the LEDs to display different signals that will provide feedback to the user and it will also help drive the motor for turning the deadbolt. Using PWM with a DC motor works in a similar way to the LED. The motor pulsed with a constant on and off will not produce movement of slowing down and speeding up, it will only slow down by an imperceptible amount before the signal turns on again and it goes back to speed. By adjusting the average voltage, the speed and torque of the motor can be fine-tuned to the minimum necessary to turn the deadbolt and thus reducing power needed.

3.2.5 Gesture Controller

For increased security of SOLAS its design includes the capability to detect specific hand gestures made by a user. A specific set of hand gestures will act as a password for unlocking the door. This gesture system is achieved through IR-based sensors. The components required for this to work include an infrared light-emitting diode (IR LED) and four directional photodiodes.

The IR LED emits infrared light at a specific distance and when a hand is in range of it the infrared emitted will bounce back. This bouncing of the infrared is detected by the photodiodes and helps determine the direction of the gestures from the different intensities of the reflected infrared that is received by each photodiode. These photodiodes are positioned to detect up, down, left, and right directions.[2]

Figures 9 and 10 below how the gesture controller detects and interprets directional movement. The LED at the lower portion of the sensor system in figure 10 emits light which reflects back off of close objects towards the upper photodiodes. This only shows current location however, the chart in figure 9 illustrates how taking the location over time shows movement in a certain direction. As one photodiode receives more light than less light over time, it is interpreted that an object just passed over that photodiode. Overlaying all 4 diode time charts results in 1 gesture, since it can be seen which photodiode first “saw” the object, and which saw it last.

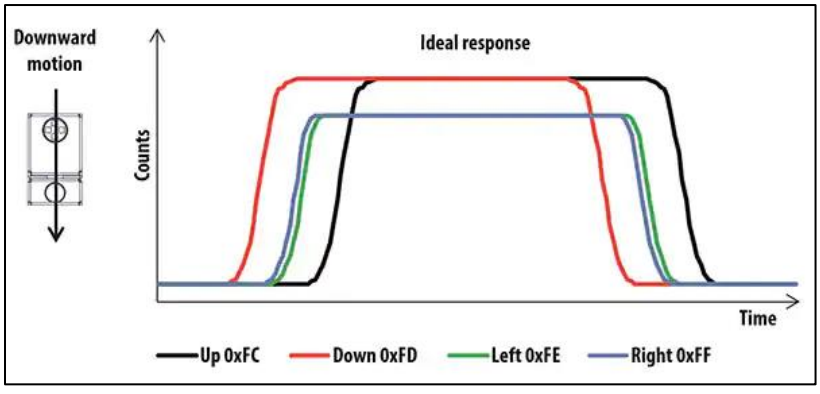


Figure 9: Broadcom gesture controller response graph, permission to use granted

The gesture controller has 4 built-in gestures, and with the interpretation of the charts, other custom gestures can be coded into the controller.

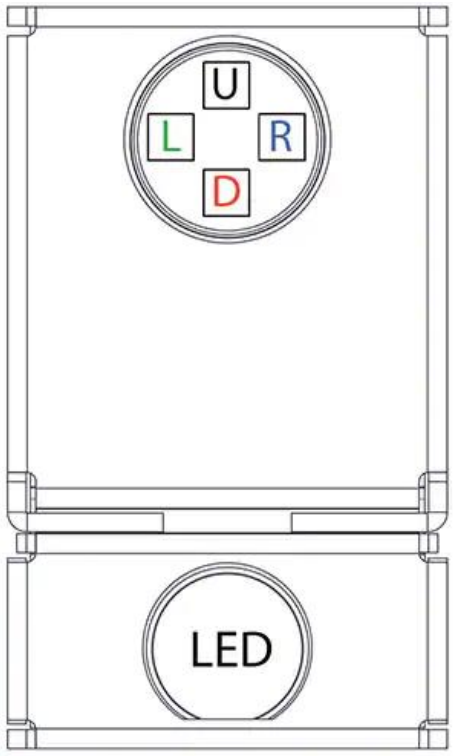


Figure 10: Broadcom gesture controller photodiode layout, permission to use granted

3.3 Strategic Components and Part Selections

The SOLAS system has a lot of hardware pieces that need to be ordered and put into their respective subsystems, which are covered here.

3.3.1 Motor

A motor is needed to turn the deadbolt to unlock the door. When the RFID bracelet is detected and gesture-based password successfully entered, the system will need to turn the deadbolt, therefore unlocking the door. There are two components required to drive the motor. The motor itself and a motor driver. The motor will be responsible for converting electrical energy into mechanical energy, which will turn the deadbolt. There are three common types of motors to consider for this function.

The first type is the DC motor, and it can be either brushed or brushless. The brushed motor is the cheapest and simplest motor out of the choices available. They are simple because all it needs is a DC source to start spinning and can also spin in the opposite direction. The downside to a brushed motor is the noise it produces. As the graphite brushes inside the motor make contact with the spinning commutator sparks can be produced due to friction. This can produce electromagnetic noise that can affect other parts of the system. This also means the brushes will wear down eventually and will require replacement of the motor more often.

A brushless motor does not use graphite brushes like a brushed motor does, instead the rotor of has a fixed ring of magnets that spin with stationary electromagnets on the inner edge of the motor. This design does not produce noise like a brushed motor does and simplifies the design, but tuning the motor is complicated by the need for a controller. The controller converts DC to AC to control the different aspects of the current such as phase. This allows for fine tuning of torque, speed, and energy usage. A brushless motor has the advantage of being longer lasting due to having no brush and being more efficient.[9]

The second type is the stepper motor. It is mechanically similar to brushless motors except the motor's rotation is divided into fixed angles hence the name stepper. This design allows for accurate positioning to be made. The major downside to this motor is the low energy efficiency, these motors are constantly drawing maximum current which can also cause issues with heat.[9]

The final type to discuss are servo motors. Like stepper motors, servo motors can rotate in precise angles, but only limited to 180 degrees. Servos are built with a DC motor brushless or brushed, a few gears, a controller, and a sensor for position.[9]

The motor that SOLAS will include in its design will be a brushed DC motor. Its operation is simple enough for the purpose SOLAS. The motor will only be operating when the deadbolt needs turning which should take under 1 second to perform. This eliminates any problems with motor noise and the wearing down of the graphite brushes.

Motor Driver

The purpose of a motor driver is to help the microcontroller driver the motor. Low-powered microcontrollers usually do not output a high enough voltage to drive the motor. This is where a motor driver comes in, taking a low input current and amplifying it for the motor. Instead of designing a driver circuit, SOLAS will use the ULN2003[27] motor driver. The ULN2003 will allow the system to supply the motor with enough power to turn the deadbolt.

3.3.2 LED

An LED will be necessary to give feedback to the user for actions such as an incorrect password being entered, or the device not being powered. SOLAS will use 3 different colors to indicate the specific states that the device is in. These colors will be red, green, and blue. Because including three different LEDs is not practical due to the space required to be able to display three separate LEDs for users to look at, for SOLAS, a single multi-color LED will be ideal. An RGB LED usually consists of three separate LEDs put together by plastic with 4 pins.

3.3.3 Camera

SOLAS will use a camera module to take images in front of the door activated by motion sensors. These images will then be uploaded to the SOLAS website for the user to look at. A camera module consists of four components, a lens, image sensor, a printed circuit board (PCB), and an interface.

An image sensor according to the AN5020 Application note it is an analog device that converts the light it receives into electronic signals to form an image and the two types of image sensors that can be used in digital cameras are charge coupled device (CCD) sensors and complementary metal oxide semiconductor (CMOS) sensors. A lens is necessary to focus the light onto the image

sensor to help capture whatever it is pointed at. The camera module interface will allow for a connection to the microcontroller. In the AN5020 Application note the main signals transferred between a camera and a microcontroller through parallel or serial interfaces are control signals, image data signals, power supply signals, and camera configuration signals. The control signals control the transfer of data through clock synchronization. Image data signals deal with the transfer of each image data bit to the microcontroller. Power supply signals will power the camera module provided by the microcontroller and the configuration signals set the resolution and other image features.

A camera for SOLAS that will be implemented in its design will be chosen based off its cost, size, and image resolution.

ESP32-CAM

This camera module is a low-cost Wi-Fi camera that also has Bluetooth. It is a microcontroller featuring a low-power 32-bit CPU and 9 GPIO pins, 520 KB SRAM, up to 160 MHz clock speed, and a small footprint of 27x40.5x4.5mm. With its Wi-Fi capabilities this camera features support for image upload over it.

The image sensors this device supports are the OV2640 and OV7670. The OV2640 is a low voltage CMOS image sensor and a 2 Megapixel camera with a resolution of 1632 x 1232 and outputs 8-bit images in various formats. The OV7670 is also a low voltage CMOS image sensor and produces 8-bit images with a lower resolution of 640 x 480 compared to the OV2640. Both are capable of outputting video.

ArduCAM Mini

The ArduCAM Mini is a high-definition camera that uses an SPI interface. It can use a 2 Megapixel OV2640 or 5 Megapixel OV5642 CMOS image sensors and it is compatible with any microcontroller as long they are capable of I2C and SPI [28]. The SPI interface is used for camera commands and data stream while I2C is used for the image sensor. It is a user-friendly camera that provides an open-source library for Arduino, STM32, and Raspberry pi. It features low-power modes, image sensor control, JPEG compression, and a board size of 34 x 24mm [28]. This module comes together through the ArduChip, the proprietary camera controller for this module and simplifies programming the camera sensor.[29]

Pixy CMUcam5

The Pixy CMUcam5 is a camera focused on robotic applications with computer vision [31]. It is a low-cost and easy to use device that supports the C and Python programming languages, and it is entirely open source with its hardware, firmware, and software. A unique feature of this module is the object detection algorithm it supports. The Pixy allows for an easy method to train this algorithm, simply by holding objects of interest. It will gather the different colors for the object and find those objects again based on color profile. The downside to this camera is that it only sends information of what it detected to the microcontroller and not the image itself, but it is possible to take the raw image it uses from the USB port.

OpenMV Cam

The OpenMV Cam is a low-powered microcontroller specialized for machine vision tasks with its 32-bit ARM CPU [30]. It comes with a OV7725 image sensor with image resolutions of 640x480. It has extensive support with its Python libraries and already available applications baked into the camera such as person detection, face detection, eye tracking, image capture, and much more. The table below features all the cameras discussed so far and their technical specifications.

Table 3: Camera technical specifications comparison

	ESP32-CAM	ArduCAM Mini	Pixy CMUcam5	OpenMV Cam
Size	27 x 40.5mm	34 x 24mm	53.34 x 50.8mm	35.56 x 44.45mm
Image resolution	1600x1200 or 640 x 480	1600x1200 or 640 x 480	1280x800	640 x 480
Power Consumption	~180 mA	20 ~ 70 mA	~140 mA	~150 mA
Price	$14.99	$25.99	$59.95	$65.00

Camera choice

The ideal camera chosen for SOLAS is the ESP32-CAM. It is the lowest priced camera among the ones discussed. At such a low price point, it offers many features such as a processor, Bluetooth, and Wi-Fi. Since power consumption is high, techniques to limit it will be utilized such as turning on the camera only when needed.

3.3.4 Microcontroller

There are few things to consider when choosing a microcontroller that meets the specifications for SOLAS. Mainly, power consumption should be kept low while still being able to support the variety of sensors, motor, and other modules. The microcontroller should be able to accomplish all these and be reasonably priced. These are the microcontrollers considered for SOLAS.

MSP430FR5994

This microcontroller from Texas Instruments offers ultra-low power and low cost. It features a 16-Bit RISC architecture, clock frequency of up to 16 MHz, 256kB FRAM, and Low-Energy Accelerator. This MCU provides increased performance with low power consumption compared to other MSP430 processors. One unique feature of this MCU is the Low-Energy Accelerator (LEA). It is a 32-bit hardware accelerator for signal processing and performs these operations for the CPU saving it time and energy. A plus to using the MSP430 microcontrollers is the extensive hardware and software support that aid with design.

STM32 L4+

This microcontroller from STMicroelectronics features an ultra-low power design based around a 32-bit ARM Cortex M4 CPU with a clock frequency of up to 120 MHz It has up to 256KB of FRAM, 110 μA/MHz Active mode, and 22 nA Off mode. This is a higher-performance microcontroller with the bonus of also being ultra-low power compared to the MSP430. It can render enhanced graphics for graphical user interfaces (GUI) using Chrom-ART Accelerator (DMA2D), but this is unnecessary for the design of this project.

ESP32-U4WDH

The ESP32 by Espressif Systems is an ultra-low power and low-cost System on a chip microcontroller designed for mobile and internet-of-things applications. It has a single core 32-bit processor with clock frequency of up to 160 MHz and 4 Mbytes of flash memory. It is a highly integrated microcontroller with modules integrated into the chip, such as power management modules, filters, and antenna switch. But the main draw of it is the integrated Wi-Fi and dual-mode Bluetooth. This eliminates the need for a separate Wi-Fi module to be added when compared to the MSP430 and STM32.

ESP-WROOM-32S

A close brother of the ESP U4WDH is the WROOM, offering a higher clock frequency of 240MHz and 4MB of flash memory as well, but with 2 cores. The chip also still comes with Wi-Fi and Bluetooth.

Microcontroller Choice

It was difficult making the tradeoffs to get as close to an ideal microcontroller for this project as possible. The ideal microcontroller choice for SOLAS is the ESP-WROOM-32. Although overall current consumption is high compared to MSP430 and STM32 L4+, this is a necessary trade off as the ESP-WROOM offers integrated Wi-Fi (802.11 b/g/n) required for web access. This integration of Wi-Fi and other peripherals increase overall power consumption for the device, but there are adjustments that can be made to reduce this. Overall, the ESP-WROOM packs many advanced features into a single powerful chip providing a solution that does not take up much space on a print circuit board.

Below in Table 4 is a comparison of each of the microcontroller's specifications that were discussed above.

Table 4: Comparison of Microcontrollers

	MSP430FR5994	STM32 L4+	ESP32-U4WDH	ESP32-WROOM
CPU	16-bit RISC architecture	32-bit ARM Cortex M4	Xtensa 32-bit LX6	Dual-core Xtensa®32-bit LX6 MCU
Memory	FRAM: 256 KB RAM: 8 KB	Flash: 1 MB SRAM: 320 KB	ROM: 448 KB SRAM: 520 KB	ROM: 448 KB

			RTC SRAM: 16 KB	SRAM: 520 KB RTC SRAM: 16 KB
Clock Frequency	Up to 16 MHz	Up to 120 MHz	Up to 160 MHz	Up to 240 MHz
GPIO pins (#)	68	136	34	34
I2C	4	4	2	2
SPI	8	3	4	4
UART	4	6 (USART)	3	3
Timers	Six 16-bit Timers	Two 32-bit Timers Nine 16-bit Timers	Four 64-bit Timers with 16-bit clock prescaler	Four 64-bit Timers with 16-bit clock prescaler
Power Modes	Active Mode: 118 µA/MHz Shutdown: 45 nA	Active Mode: 110 µA/MHz Shutdown: 22 nA	Active Mode: 95 ~ 240 mA Deep-Sleep: 10 ~ 150 µA	Active Mode: 95 ~ 240 mA Deep-Sleep: 10 ~ 150 µA
Supply Voltage	1.8 ~ 3.6 V	1.71 ~ 3.6 V	2.3 ~3.6 V	2.3 ~3.6 V
Price	$8	$11	$1.60	$4.50

3.3.5 Motion Sensor

One of the problems that needed to be addressed when designing the SOLAS was trying to reduce the power consumption of the system. One of the ways to achieve this is to have the system stay in a locked, standby state when it is not being used and activate when it is ready to be unlocked. To integrate this function into the system, the SOLAS will use a passive motion senor that will act as the switch for the system. When the sensor detects movement in front of the door, it will turn on the camera and then the entire system. There are three types of motion sensors that can be used, Passive Infrared (PIR), Microwave, and Dual Tech/Hybrid.

The PIR motion sensor uses a pyroelectric sensor to detect changes in infrared radiation in the detectable area which is comprehended as movement. When the sensor detects the “movement” it creates a pulse that allows the system to recognize the movement. These types of sensors are typically inexpensive, small, and low power.

The Microwave motion sensor uses Microwave Radiation to send high frequency radio waves to detect motion. The waves are transmitted out and reflect off object and are received. The sensor uses this method to read a frequency shift, which indicates there was movement. These sensors typically have a larger range but are more expensive and are susceptible to electric interference.

Finally, the Dual Tech/Hybrid motion sensor uses both the PIR and microwave motion sensors. Combining the two sensor types allows the motion sensor to reduce the number of false alarms that can occur in the other two sensors. Since this sensor has both types of motion detectors, the

component will be more expensive and be larger in size than the two other sensors. Table 5 below summarizes these differences.

Table 5: Pros and Cons of Motion Sensor Types

	Pros	Cons
PIR	• Low Cost • Low Power • Small size	• Sensitive to sudden temperature changes
Microwave	• Larger Range	• High Cost • Sensitive to electric interference
Dual Tech/Hybrid	• Fewer false alarms	• High Cost • Large size

Since the motion sensor is going to be used in the standby state of the SOLAS system, the sensor should use as minimal power as possible. With this condition, the best option to use is the PIR motion sensor. Since the system is going to use a camera to confirm the movement detection, the accuracy of the motion sensor does not need to be exact.

BL412 PIR motion sensor

For the specific type of PIR motion sensor, we looked for a component that was small enough to be integrated into the system but have a large enough detection range. A suitable motion sensor for this project would the BL412 found on Adafruit. This component has a detection range of approximately 5 meters and has a 9mm x 9mm size. With the voltage supply needing to be in range of 2.7V and 3.3V, this component would be the best choice for the SOLAS system.

3.3.6 RFID Tag

A major component of this project is going to be the RFID tag and system. The components required will be the RFID tag to go in the bracelet, and the RFID reader to go inside the lock system. The main consideration will be active or passive RFID tags to go inside of the bracelets.

Passive RFID tag – TI RF37S114HTFJB

Texas Instruments offers an inlay passive RFID tag for only $0.697. It operates in the high frequency spectrum at 13.56 MHz and is only 16mm square, which would fit perfectly inside of a wearable bracelet. It is compliant with ISO/IEC15693and ISO/IEC18000-3 standards and has an integrated antenna. Its operating temperature ranges from -25°C to 70°C (-13°F to 158°F), which has much more range than will be required for its use. It has 64 bits of UID read-only memory to protect its identification from unintentional writes, and can retain data for > 10 years, eliminating the hassle of replacement[5].

Passive RFID tag – Parallax 28445

From Parallax there is another passive RFID tag costing more at $2.49. It is circular in shape with a diameter of a12.4mm ± 0.2 mm, and around 2.0mm thickness. Its memory is 64 bits read only

and has an operating temperature of -25°C to 85°C (-13°F to 185°F). The RFID is encapsulated and has an operating frequency of 125kHz[34].

Passive RFID tag – Sparkfun RFID Button

Sparkfun offers a similar RFID tag for $3.95, operating at 125khz and 16mm in diameter. It has a 2.0mm thickness and 32-bit ID.

Active RFID tag – STM M24LR04E-RMC6T/2

STMicroelectronics offers several active tags, one of which costs only $1.10. It operates at the same 13.56 MHz frequency but can retain its data for over 40 years. The reader complies with ISO 15693 and ISO 18000-3 and uses the I^2C serial interface and takes a supply voltage ranging from 1.8V-5.5V. Its operating temperature ranges from -40°C to 85°C (-40°F to 185°F). The user memory is read-write, but password protected for both I^2C and RF. The size of the chip is 3 by 6.4mm[6].

RFID tag selection

Table 6 below shows the breakdown of the differences between the 3 tags, including size, frequency, and cost.

Table 6: RFID tag comparison

Tag:	**TI RF37S114HTFJB**	**Parallax 28445**	**STM M24LR04E-RMC6T/2**	**Sparkfun Button**
Cost:	$0.697	$2.49	$1.10	$3.95
Temperature range:	-25°C to 70°C	-25°C to 85°C	-40°C to 85°C	-
Top-down size:	16mm^2	20.1mm^2	19.2mm^2	20.1mm^2
Depth:	0.66mm	2.0mm	0.65mm	2.0mm
Active or passive:	Passive	Passive	Active	Passive
User Memory:	Read-Write	-	Password Read-Write	-
ID memory:	Read-only	Read-only	Read-only	Read-only
Voltage supply range:	-	-	1.8-5V	-
Data Retention:	10+ years	-	40+ years	-
Operating frequency:	13.56 MHz	125khz	13.56 MHz	125khz
Standards compliant:	Yes	-	Yes	Yes
Internal Communication interface:	-	-	I^2C	-

Comparing the RFID tags side by side, it is seen that the main differences to be concerned with involve whether an active or passive tag will be used, and the data retention time. The active tag has a much longer data retention time, but considering that it appears all tags can retain data more than 10 years, the extra 30+ years can be considered a diminishing return. The extra 30+ years that the active tag can retain data is also negligible due to the fact that depending on the battery used inside the bracelet, the battery will most likely have to be replaced before 10 years pass, much less 40. A minor goal of the bracelet is to reduce the amount of maintenance performed, so even if the tag can be read for 40+ years before needing to be replaced, having to replace the battery most likely every year renders the data retention moot, giving an advantage to the passive tags as far as maintenance is concerned.

That leaves whether to use an active or passive tag inside the wearable bracelet. Due to having their own power supply, active tags can be read from 100+ feet, while passive tags can only reach around 20 feet since they are powered by the signal, they receive from the reader[4]. However, considering that the expected necessary range for the tag is only 5 feet, even 20 feet gives plenty of buffer for the purpose of the project. Due to the fact that the passive tags do not require a power supply, which would reduce the size and weight of the bracelet; since the bracelet should be small and light if the user needs to put in a motion-based password, having reduced hardware will greatly aid in meeting these objectives.

Using a passive tag in the bracelet will not only make the bracelet lighter and smaller, require less maintenance for the bracelet, but does not require setting up a communication interface between the tag and a microcontroller. Since the passive tag allows for project specifications to be met and reduces development complexity, the RFID tag to be used in the bracelet will be the Texas instruments RF37S114HTFJB, Parallax 28445, or Sparkfun button. The difference between them being the operating frequency mainly, research into readers will have to be done before a decision can be made.

3.3.7 RFID Reader

As stated previously, a major part of the SOLAS system is the use of RFID signals to unlock the door. With the RFID somewhat tag selected, an RFID reader needs to be chosen so that the subsystem will be complete.

Texas Instruments RFID Reader – TRF7963ARHBT

An RFID reader is also needed inside the lock system to read in the data sent either passively or actively from the tag. Once again while there are several to choose from, Texas Instruments offers a reader for only $2.933. It operates in the same frequency as the RFID tag at 13.56 MHz and operates from -25°C to 85°C (-13°F to 185°F), well withstanding any expected temperature seen by the lock system. The size of the reader is 25mm square and thus easily integrated into the design of the lock. It can operate from 2.7VDC-5.5VDC, has an integrated voltage regulator to deliver to an MCU. The reader has multiple power saving modes allowing it to save power until our proximity sensor receives input and it can be woken up. In sleep mode the reader draws only 0.120mA of current and draws either 67 or 130mA of current at full power when using 3.3V or 5VDC, respectively. Its communication interface the MCU is Parallel or SPI and complies with ISO/IEC14443A and B standards[7].

STMicroelectronics RFID Reader – ST25R3911B-AQFT

An alternate RFID reader would be one made by STMicroelectronics for $5.75. This reader can operate at 13.56 or 27.12 MHz the reader can operate on 2.4-5.5V and from -40°C to 125°C (-40°F to 257°F). The size of the reader itself is 25mm square and has built in power supply regulators. An additional feature this reader has is low power detection of tag presence, thus not having to use the proximity sensor interrupt as an event to wake up the reader. The communication mode between a microcontroller and the reader is 4-wire SPI and is ISO 18092, ISO14443A, ISO14443B, and ISO15693 compliant. When in power saving mode, the reader draws 3.6μA of current.

NXP RFID Reader – HTRC11001T/03EE

For an RFID reader with a different operating frequency, NXP offers the HTRC11001T/03EE at 125khz. Costing $3.40 and requiring 4.5V ~ 5.5V, this reader is larger than the other 2, 8.75mm by 6.2mm, with a height of 1.75mm. It can operate in temperatures from −40 °C to 85°C (-40°F to 185°F). The reader communicates with the microcontroller via three or 2 wire interfaces. When in use the reader draws 137mA of current, and less than 20μA in power-down mode[35].

RFID reader selection

There are many factors to consider when choosing which RFID reader to use inside the lock system. Table 7 below shows a side-by-side comparison of the 2 discussed.

Table 7: RFID reader comparison

Reader:	**TI TRF7963ARHBT**	**STM ST25R3911B-AQFT**	**NXP HTRC11001T/03EE**
Cost:	$2.933	$5.75	$3.40
Temperature range:	-25°C to 85°C	-40°C to 125°C	−40 °C to 85°C
Top-down size:	$25mm^2$	$25mm^2$	$54.25mm^2$
Depth:	0.5mm	0.5mm	1.75mm
Voltage supply range:	2.7VDC-5.5VDC	2.4-5.5V	4.5-5V
Operating frequency:	13.56 MHz	13.56 or 27.12 MHz	125khz
Standards compliant:	Yes	Yes	Yes
Internal Communication interface:	Parallel or SPI	4-wire SPI	3 or 2 wire
Active current draw:	67 or 130mA	-	137mA
Sleep current draw:	0.120mA	3.6μA	20μA

For the selection of the RFID reader, all of the readers in table 7 have very similar characteristics as far as the project is concerned. The STM operates in a wider range of temperatures, and can operate on slightly lower voltage, but neither of those ranges are expected to be reached. Another difference is the current draw in sleep mode, which since the difference is only 0.1164mA between the STM and TI, is negligible, and the STM reader does not have an active current draw available

which could be higher than the TI. The main difference between the readers appears to be size and frequency.

Taking this into consideration, since the TI RFID reader is almost half the price of the STM and can be ordered from the same manufacturer as the tag, the optimal RFID reader for the project if 13.56Mhz is used is the Texas Instruments TRF7963ARHBT, and if 125khZ is used the NXP HTRC11001T/03EE will be used.

3.3.8 RFID Module

The RFID system can also be simplified by using a pre-built RFID module that would supplement the RFID reader and the circuitry that would be needed to utilize the reader. Included in the integrated circuit of the module is the antenna for the RFID signal.

SparkFun Electronics – ID-12LA

One option for the RFID module is from SparkFun Electronics, the ID12LA. Even though the module is priced higher than the RFID Reader, at $29.95, time and some money will be saved without then need to build a circuit and attach an antenna. The component operates at a frequency of 125kHz and has a voltage supply range of 2.8V – 5V. The dimensions of the module are 26.4mm x 25.3mm.

SparkFun Electronics – ID-3LA

Another option we could use for the RFID module is again from SparkFun, the ID-3LA. This RFID module is almost exactly similar to the ID-12LA. The Operation frequency of the module is 125kHz and the supply voltage range is 2.8V – 5V. The main differences between the two modules are the price and the dimensions. For the ID-3LA, the price is $25.95, and the dimensions of the module are 22mm x 20.5mm. Even though its size is smaller, the module does not have a built-in antenna.

SparkFun Electronics – ID-20LA

A third option from SparkFun is their ID-20LA. This reader module is much bigger than the other options, sizing at 40.3mm x 38.5mm. The voltage supply is ideally 3-4.5V, and the reader operates at 125kHz. It is more expensive than the previous 2, costing $34.95.

NXP USA – RFID-RC522

A fourth for the is an RFID module typically used with the Arduino Raspberry Pi, the RFID-RC522. This module is offered on amazon for a cheap price of $6.99. The RFID-RC522 also operates at much higher frequency of 13.56MHz and has a supply voltage of 3.3V. Even though this module less expensive and has a higher frequency, the RFID-RC522 is much larger is size at 3.7’’ x 3’’.

RFID Module Selection

Table 8 below covers the main differences we are concerned with when choosing an RFID module to use inside the door lock.

Table 8: RFID Module comparison

	ID-12LA	ID-3LA	ID-20LA	RFID-RC522
Price:	$29.95	$25.95	$34.95	$6.99
Frequency:	125kHz	125kHz	125kHz	13.56MHz
Voltage Supply:	2.8V – 5V	2.8V – 5V	3-4.5	3.3V
Dimensions:	26.4mm x 25.3mm	22mm x 20.5mm	40.3mm x 38.5mm	3.7'' x 3''
Antenna:	Yes	No	Yes	Yes
Read Range:	12cm – 18cm	-	18 – 25cm	0cm – 3.5cm

When choosing a possible RFID module to use for the SOLAS system, there are different factors to consider. The ID-20LA has suitable dimensions and has internal antenna with a good read range but is the most expensive of the four modules. The ID-3LA also has a good size for the project, but without an internal antenna, the size of the module becomes irrelevant. Finally, the RFID-RC522 is the best priced out of the three modules and has a much larger operational frequency, but the size of the module is much larger, and the read range of antenna is much shorter than the ID-20LA. Due to the modules large read range and smaller size, the better option would be the SparkFun Electronics ID-20LA. Since the operation frequency is 125kHz, the Parallax 28445 125kHz passive RFID tag will be used in the bracelet.

3.3.9 Gesture Controller

When researching possible gesture controllers that could accept a password viable for the SOLAS system, only the Broadcom APDS-9960 was found.

Broadcom APDS-9960

A module designed for gesture detection is the Broadcom APDS-9960. It uses the IR LED and photodiodes to convert physical motion into digital information. It features ambient light and ambient light subtraction which help improve detection of gestures, 8-bit data convertor, 32 dataset storage FIFO, and I2C compatible interface data rates up to 400 kHz with dedicated interrupt pin.

All the functionalities are driven by a state machine and each state can be modified and included or excluded from the cycle. The device begins in the Sleep state when the device is powered on and progresses to Idle, Proximity, Gesture, Wait, Color/ALS, and Sleep. The Proximity state will be ideal for detecting a hand and readying the system to read gestures then moving to the Gesture state. The Color/ALS state is used for detecting RGB intensity data which will not be used in SOLAS.

3.3.10 Power Supply

One challenge that need to be overcome is how the SOLAS system will be powered and how to give it a good life span. We want a source that has enough voltage to power the system without having too large of a voltage such that the efficiency of the source is lowered. With our highest voltage supply needed at 5-volts, the source needs to be higher than this value.

Coin Batteries

For these types of batteries, their biggest advantage is their size. The coin batteries are able to supply 1.5 volts (Alkaline) or 3 volts (lithium) which would mean the system would have to use 2-4 of these batteries. The main drawback is how long the batteries will last. Coin batteries are typically used in smaller electronics that do not need a large supply. Since the SOLAS system includes a motor, the coin batteries would most likely die quickly.

AA Batteries

Similarly, to the coin batteries, the AA batteries can supply 1.5 Volts individually. In order to power the system, there would have to be 4 batteries powering the SOLAS system. These batteries are used in a majority of the electronics that are built and are not too large of a size. AA batteries also would have a longer life span than the coin batteries.

9V batteries

As the name suggests, this battery would supply 9 volts to the system. With such a large voltage, the SOLAS system would only need one battery to be powered. Even though these batteries would have a high enough voltage, its life span is not as long as the AA battery and are much larger.

Power Source Selection

With enough batteries, each type of battery can give the necessary voltage needed for the SOLAS system. The coin batteries have a good advantage by being the smallest size out of the three, but their short life span would not work for this type of system. The 9-volt battery has the advantage of only needing one physical battery to power the whole system. However, since the AA batteries have the longest life span, they are the choice for this project.

Using these batteries as our power source, the SOLAS system will need voltage regulators to give power to the components that will require less voltage than the power source itself will give. To have the optimum voltage for each component, a 3.3-volt regulator and a 5-volt regulator will be used.

3.3.11 Bracelet

The SOLAS system includes a bracelet that the user will wear in order to gain entry through the door. The goal is to have a soft small bracelet that can fit most users' wrist and have a small pocket with an embedded RFID tag. The market for bracelets with a slot for inserting your own RFID tag was very small, but there are a few options.

Chuangxinjia "Insert RFID card silicone wristband"

This company based in China specializes in RFID products, mainly ones with RFID tags already embedded. For the purpose of this project, we looked at the single product they sell which allows you to insert your own RFID chip, the "Insert RFID card silicone wristband", shown in figure 11. It is adjustable in size up to 10.2 inches, and can be tightened much smaller, although that dimension is not available. The band itself is only 0.7 inches wide, except for where the chip is inserted, which is slightly wider, which fits our needs for comfort perfectly. The RFID tag slot can

hold a tag up to 25mm x 25mm, more than enough for our planned 12mm. It is waterproof and available in many colors.

Figure 11: Chuangxinjia "Insert RFID card silicone wristband", permission requested

ANQUEUE PocketBands

The only alternate bracelet found that meets the project's purpose is the ANQUEUE PocketBands on Amazon, sold by SHENZHEN ANQUEUE Technology Co. These are a solid silicone band available in different sizes that have a large pocket on the inside of the band. They are designed to carry spare bills or keys while running, but the description also lists "RFID chips" as a use of the pocket, as seen in figure 12 below. The band length for medium and large are 7.5 and 7.9 inches respectively, which are around the size most of our clientele should use. The band width is 1.18 inches however, larger than what might be comfortable for wearing all day every day. The pocket is 2.36x1.18 inches, more than large enough for the small RFID tag we plan to use. The band is listed to use while running or surfing, so it must be water-resistant to some extent to protect the RFID tag.

Figure 12: ANQUEUE PocketBand

RFID Bracelet Selection

Both bands would serve the overall desired purpose of the bracelet, however the Chuangxinjia "Insert RFID card silicone wristband" is slightly more suited to our needs. It is smaller in width than the ANQUEUE PocketBand, and thus less noticeable or obnoxious to the user throughout their day. It is also adjustable in size, thus easier to be sold to the user in one size, and easier for the user to order. The pocket is smaller but still plenty big enough to hold the RFID tag and is marketed as waterproof.

However, while the Chuangxinjia silicone wristband is best suited for the user's needs, for the purpose of the demonstration, the team will be using another product that was found, made by Yarontech as shown in figure 13. These bracelets come in small numbers and with 125kHz RFID tags already embedded, and large enough to add a logo or design to them, which we used to number them. Since almost any 125kHz passive RFID tag would work for the system, this product was used.

Figure 13: Yarontech RFID bracelets

3.4 Parts Selection Summary

All of the parts for this project were picked with intercommunication and the end objectives in mind. The motor for the door has a simple task and is simple itself, and so the motor attached to the deadbolt in an existing electric door lock system will be used, most likely out of the OrangeIOT system since it is the cheapest. The ULN2003 motor driver to power the motor was chosen for its ability to use low current to turn the deadbolt.

The SOLAS system utilizes different colors to signify which state the door lock is currently in. With the use of an RBG multi-color LED, space and time will be saved. When the system is in a standby or shut-down state, the LED will not be on, but during any other states, it will be different colors in order to communicate to the user what the current/next action they need to take is.

The camera is a major part of the lock system, if the pictures are not easily readable or are too fuzzy, then it and the website are useless. However, if the images are too large, then the camera will most likely use more battery, and use more space in the database. If items in the database get too large, the website load time may increase as well. The ESP32-CAM offers 2 different resolutions, a small size, and a low price. The resolution can be increased if picture quality is low or decreased if power drawn is too high. The ESP32-CAM also has built in Wi-Fi to enable the upload of its pictures to the website.

The microcontroller chosen is the ESP-WROOM-32S which also offers built-in Wi-Fi and Bluetooth as well as a very low price. Although it does not have as sophisticated communication abilities in other areas, they will not be necessary for this project and the other comparable microcontrollers did not offer integrated Wi-Fi and Bluetooth capability.

The motion sensor chosen is the BL412 PIR motion sensor for its ability to draw very low power in the standby state and its small size.

For the RFID tag in the user's bracelet, the Yarontech RFID bracelets with passive tags were chosen. The simplicity of a passive RFID tag allows for the bracelet subsystem to be simple and lightweight, and it is also cheaper than any active RFID tag. Although it was initially expected to be able to read an RFID tag from up to 20 feet away, it was discovered that the range depends a lot more on the reader than expected. The RFID reader chosen is the SparkFun ID-20LA, which advertises to be able to read the tag 18cm away, however it operates at around 3cm. While not the desired range of 18cm, 3cm should still allow communication without taking away from the desired accessibility aspect of the SOLAS system.

For the gesture controller, only 1 suitable at all was found, but with its various states including proximity state to reduce noise read in and idle/wait states to reduce power when not in use, it should be perfect for the project. The gestures are read in by the microcontroller as simple number which can be easily interpreted. If desired additional gestures beyond the 4 basic encoded gestures can be programmed into the controller to allow for further customizability of the system.

The power source chosen for the door lock system was AA batteries, since they are very common, standard in most systems, and can be easily converted to the necessary 3.3 and 5V. With 4 of the batteries, they will last a fair amount of time and supply enough voltage for the system.

The bracelet worn by the user is not the optimal choice due to the fact that they only come in 1 size and an adjustable bracelet was desired, they will serve the purpose of a product demonstration since they are rubber and stretch somewhat. They have embedded passive RFID tags to communicate with the RFID reader in the SOLAS lock.

4. Related Standards and Realistic Design Constraints

When planning the initial design for the SOLAS system, many factors need to be implemented so that the door lock can be used practically within the market and can physically be built. This section will discuss the standards SOLAS will choose to follow and the realistic design constraints that will influence the overall design of SOLAS.

4.1 Related Standards

The related standards that will be discussed below in this subsection will follow standards developed by the Institute of Electrical and Electronics Engineers (IEEE), International Organization for Standardization (ISO), and International Electrotechnical Commission (IEC).

4.1.1 Motor Controller Standards

The IEEE Std 3001.11™- 2017 covers many topics related to the motor and motor driver that is used in the SOLAS system to turn the deadbolt. The standard covers how if the DC motor has a full voltage start method, the motor is generally 2hp or less due to the high current. If the DC motor uses reduced voltage starting, then once the motor reaches full speed all resistance in the circuit should be removed.

Protection of the motor controller is also covered, including physical protection of the motor controller in a casing, and overload protection from high temperature resultant from runtime or high current. Since the motor will run for a short time and will not be powerful, neither of these will be expected. Protection from overloading is required normally via overload relays in the motor circuit[13].

4.1.2 LED Standards

The ANSI C78.377-2015 covers the allowed chromaticity and chromaticity tolerance of given LEDs. The measurement of the chromaticity must follow the methods given in the IES LM-79-08 standard[14].

4.1.3 Camera-related Standards

There are many standards related to the various measurements related to camera usage, which were covered extensively in a page by Society for Imaging Science and Technology[15] and are discussed in this section.

The measurement most widely known to users, “ISO 12233:2017 – Photography – Electronic still picture imaging – Resolution and spatial frequency responses” covers the methods on measuring the resolution detail and spatial frequency response (SFR) of a camera. The resolution of a camera is its ability to capture fine detail, and SFR is a “metric that measures contrast loss”[15]. For noise measurements “ISO 15739:2017 – Photography – Electronic still-picture imaging – Noise measurements” covers methods on measuring the various noises seen by cameras.

For the settings related to the sensitivity or exposure of a camera, “ISO 12232:2019 – Photography – Digital still cameras – Determination of exposure index, ISO speed ratings, standard output sensitivity, and recommended exposure index” covers how all of those metrics should be recorded. For exposure, “ISO 14524:2009 – Photography – Electronic still-picture cameras – Methods for measuring opto-electronic conversion functions (OECFs)” also specifies reporting focal plane exposures and output codes on the camera. Also related to the shutter speed, lag, or other time-related events, “ISO 15781:2019 – Photography – Digital still cameras – Measuring shooting time lag, shutter release time lag, shooting rate, and start-up time” specifies how to they should be measured.

Related to colors seen by the camera, “ISO 17957: 2015 – Photography – Digital cameras – Shading measurements” covers color, intensities, and other shading factors of images. Similar details including measurement of color fringe boundaries are covered by “ISO 19084:2015 – Photography – Digital cameras – Lateral chromatic displacement measurement”. Stray light seen in an image decreasing the correct contrast is covered in “ISO 18844:2017 – Photography – Digital cameras – Image flare measurement”. For low light-related measurements, “ISO 19093:2018 – Photography – Digital cameras – Measuring low light performance” is used. When correcting the color from poor conditions, “ISO 17321-1:2012 – Graphic technology and photography – Color characterization of digital still cameras (DSCs) – Part 1: Stimuli, metrology and test procedures” has many color charts and other helpful test methods.

Images can be distorted due to the magnification of the lens and measuring that distortion is specified by the “ISO 17850:2015 – Photography – Digital cameras – Geometric distortion (GD) measurements”. For effects related to restabilized a picture due to movement, “ISO 20954-1:2019 – Digital cameras – Measurement method for image stabilization performance – Part 1: Optical systems” is used.

Finally, when testing a camera performance against all of these factors and effects, the “ISO/TR 19247:2016 – Photography – Guidelines for reliable testing of digital still cameras” is used.

4.1.4 Microcontroller standards

With microcontrollers being used in so many millions of devices, it is important that they are safe and reliable. They are even used in cars and medical equipment, both of which need to be as close to 100% reliable as possible. In the case of system failures or other events related to system and user safety, many standards were written on preventing and correcting various errors. A broad standard that covers hardware and software is IEC 61508 – Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems, which attempts to reduce and manage system failures.

More specific to microcontrollers are the IEC 60730 and IEC 60335 – standards for safety. These allow manufactures 3 options regarding the safety of devices made. The first option is for the MCU to have dual-channel architecture, one channel performing tasks while the other checks it. The 2nd is single-channel architecture, but the functionality is tested before it is sent from the factory. The 3rd is single-channel architecture that periodically checks its own critical functions[16].

4.1.5 Motion sensor standards

Many sensors, including the proximity sensor, and their performance terminology are discussed in IEEE 2700-2017 – IEEE Standard for Sensor Performance Parameter Definitions[18].

4.1.6 RFID standards

Regarding the RFID system which will be used by the SOLAS system, ISO/IEC 15693-2:2006 covers several limits and constraints on the Radio waves sent and received by the tag and its reader. It covers the communication between the tag and reader, commands to be accepted by the tag, and dimensions of the tag. The communication frequencies are also set by this standard, as well as limits on how far the reader is allowed to propagate its own radio waves. Specific communication between the devices and anti-collision procedures are also set by this standard. This is the standard the used RFID tag operates with, which has since then been superseded by ISO/IEC 15693-2:2019[18].

Another standard covering the use of the RFID system is ISO/IEC 18000-3:2010. This standard covers 3 MODES of operation, each of which are for different uses of the RFID system. It also further covers the collision management of the RFID system at 13.56MHz[19]. The ISO/IEC 14443-4:2018 standard also covers more transmission protocols, including activation and deactivation sequences[20].

4.1.7 Gesture controller standards

Gesture controllers have standards in ISO/IEC 30113 that are not related to standardizing the technology used in sensing and analyzing gestures, but the physical gestures themselves. Most of the sections appear to focus on describing the gestures that the technology will be able to read and standardize what each gesture does consistently across platforms, such as opening a menu[21]. ISO/IEC 30113-5:2019 even defines GIML (Gesture Interface Markup Language) and its syntax[22].

4.1.8 Power supply standards

For the power supply, batteries have numerous standards governing their manufacturing process, testing, performance, and many other aspects. IEC 60086-1 covers the overall design, dimensions, and various markings on a battery. It also goes into many performance features, including discharge performance, leakage, open-circuit voltage limits, safety, and proper testing[25]. To provide more depth and detail, IEC 60086-2 designates physical and electrical specifications of several categories of batteries[26]. For lithium batteries which are very common, ANSI C18.3M sets safety requirements including venting, temperature regulation, current regulation, and the battery enclosure. Additional conditions on lithium batteries were tested in this standard to include open and closed-circuit voltage, insulation resistance test, and discharging. Tests for intended use and misuse are covered such as thermal shock, vibration, incorrect installation, impact, and external short-circuit[27].

4.1.9 C standards

When bridging the gap between hardware and software to create one working system, the C programming language is often used, and is what we will use in the project to program our software.

The IEEE 1666-2011 – IEEE Standard for Standard System C Language Reference Manual defines and explains the C class library for any developers to use in an open source and free format[28].

4.2 Realistic Design Constraints

In this section, constraints that are imposed upon SOLAS system are discussed. These constraints are external factors that limit the design of SOLAS, which include economic, time, environmental, political, and other restraints. These constraints whether self-imposed or imposed by the stakeholder are necessary to help narrow down design choices that meet the needs of those invested in it.

4.2.1 Economic and Time Constraints

The economic and time constraints play a major role in the design of SOLAS. The funding for this project is provided entirely by the three members of this group. A contribution of 100 dollars from each member creates a budget of 300 dollars for designing SOLAS. With this budget in mind careful consideration must be taken in choosing parts that are inexpensive, but also of decent quality.

Time constraints dictate the amount of time a project must be completed by or have a working prototype ready. For SOLAS, these time constraints are those imposed by the University of Central Florida. From the start of Senior Design 1 till December of 2020, time is spent focusing on the research and design of SOLAS. The implementation and prototyping of SOLAS will be done in Senior Design 2. The entire period for the development of SOLAS will take approximately 7-8 months to complete. A timeframe such as this means that careful consideration must be made on accomplishing certain milestones to gauge the progress of the project development. The complexity of the design must also be considered.

4.2.2 Environmental, Social, and Political Constraints

Environmental constraints affect how the design could potentially impact the surrounding environment. One such consideration for SOLAS would be designing it to be low powered so it not only has long lasting battery life that satisfies the customer, but also reducing battery waste that could potentially impact the environment.

Social constraints relate to the effects a product will have on society and individual users. Currently there are no such social constraints to consider for SOLAS. As with social constraints, political constraints will not affect the development of SOLAS. Political constraints are those that deal with Government regulations and other Government processes.

4.2.3 Ethical, Health, and Safety Constraints

An ethical constraint deals with the morality of certain behaviors and actions. For SOLAS, the ethical constraints to consider would relate to the handling of the website with questions such as “Would SOLAS collect any data on its users?” and “Are photos collected on the website accessible by the company?”. These are the serious ethical issues that must be considered when designing SOLAS.

For the design of SOLAS the health of the user must also be considered. The sensors and other devices that use electromagnetic radiation in this project must meet the standards for safe levels emitted.

Safety constraints considers the safety of the user. SOLAS is a system designed to be connected to the internet for its camera to function. The threat of remotely accessing the camera is a possibility that may compromise the identity and location of the user and will be taken seriously in the design of SOLAS. Another consideration is the overall security of SOLAS, the possibility of spoofing the RFID tag exists. This puts the user in danger of unauthorized access. This is why SOLAS will have two factors in order to unlock the system, the RFID tag and gesture-based password.

4.2.4 Manufacturability and Sustainability Constraints

The manufacturability constraints affect the constructability of SOLAS system. To improve the manufacturability of SOLAS, parts and materials that are easily sourced will be used and construction of custom parts will be limited. SOLAS will be an all-in-one system where all features will fit into two housing cases, one for the front of the lock and one for the back of the lock.

Sustainability constraints considers the maintenance of a system and reusability of its components. SOLAS will be made easy to maintain, the housing of the system will be easily removeable to access important components. There will also be some level of modularity on certain components for easy replacement, future upgrades to the system, and recycling those components for use in other projects. These components include the motor, camera module, and RFID reader.

5. Project Hardware and Software Design Details

With the main research done for the SOLAS smart lock, the designing phase can begin. The construction of the system will be separated into two main sections, the hardware, and the software. The hardware section will explain the construction of each electrical subsystem and the design of the full system schematic. The software section will explain how the SOLAS system will operate the components in each state and how the website will connect to the smart lock.

5.1 Hardware Design of Subsystems

As the designing process of the SOLAS system begins, the chosen components will need to be built into the multiple subsystems of the smart lock. By using the typical applications for each component, the subsystems will be designed accordingly and will be adjusted to fit the full system schematic.

5.1.1 Camera

For the SOLAS system, a camera will be used to take pictures when the system senses movement from the PIR sensor. To accomplish this task, the SOLAS system will use an ESP32-CAM module and an image sensor. As stated in section 3.3.3, the camera module can function by using either the OV2640 image sensor or the OV7670 image sensor. For the SOLAS system, the OV2640 image sensor will be used. Since the image sensor connects straight to the module, there is no design needed for the connection. The camera module itself however will have to be designed into the system.

First, the ESP32-CAM module requires a 3.3-volt power supply in order to operate. To achieve this, the power supply will have a voltage regulator that will be wired to the camera module. Next, the microcontroller board will be connected to the ESP32-CAM module using pin 25 connected to the camera's pin 12 and pin 26 to pin 14. These pins will be used with interrupts to signal the necessary changes that need to occur. Pin 26 will signal to the camera to go into deep sleep and pin 25 will receive a signal from the camera to wake up the microcontroller from its deep sleep. Finally, the camera module will have the IO0 pin and the ground connected to pin headers so they can be connected when the code is uploaded.

With the pins wired to their designated pins, the camera module needs to be attached to the system. The ESP32-CAM module comes built with sixteen male pins, eight on each side. In order to connect the module to the PCB board, the system will be built with two 8-pin female headers to allow the camera module to be attached easily to the SOLAS system.

5.1.2 RFID Module

The RFID Module that we plan to incorporate into the SOLAS system will be used as one of the steps to allow the user to unlock the deadlock. To achieve this, the ID-20LA RFID module will be the component we plan to use for this function. Similarly, to the ESP32-CAM module, the RFID module is built with male pins to connect to other components, 5 pins on one side and 7 pins on the other. To connect the component, we will use the ID-20LA footprint obtained from SnapEDA.

With the footprint in place, the connections to the system need to be made. First, the module requires a 3V – 4.5V power supply. This pin can be connected to the power supply using a 3.3V regulator. The power supply pin on the RFID module will then be strapped to the reset bar on the module. Then, the ID-20LA module will be connected from the data pins to GPIO pins on the microcontroller. Finally, to select the format the information from the RFID module will be in, format selection pin will be tied to ground.

5.1.3 LED and Sensors

In the SOLAS lock, the different states the device is in will be displayed with an RGB LED: Red for an incorrect password, Green for a correct password, and Blue for the standby state. The RGB LED will connect straight to three GPIO pins on the Microcontroller and will be set in series with a resistor, as seen below in figure 14. For each state that SOLAS is in, the LEDs will turn on and/or flash.

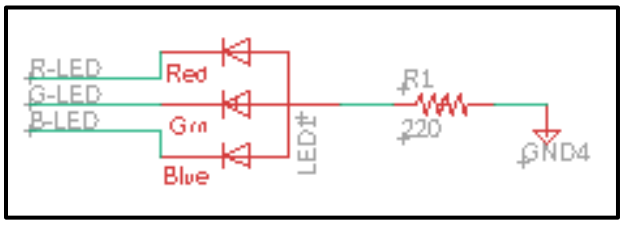


Figure 14: RGB LED

The SOLAS system will also be utilizing a PIR motion sensor which will allow the lock to turn sense someone approaching and turn on. Similar to the LEDs, the connection of the motion sensor to the Microcontroller is fairly simple. The PIR motion sensor requires a supply voltage in order to operate, which has a typical value of 3.3V as shown in figure 15. Then, the output voltage of the motion sensor will connect to one of the ESP32-CAM GPIO pins to allow the system to receive information when movement is detected.

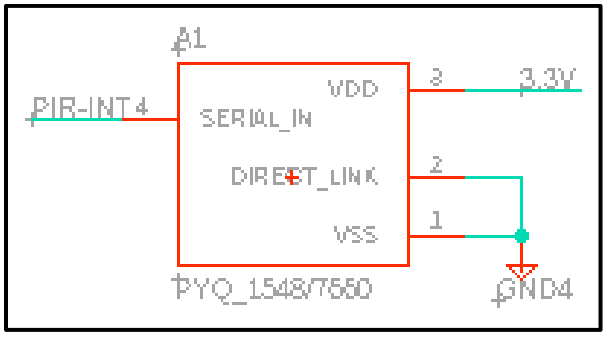


Figure 15: BL412 Proximity sensor

5.1.4 Smart Lock

For the physical lock for the SOLAS system, a previously built smart lock and motor will be utilized to minimize the mechanical engineering needed to construct the smart lock. To operate the motor from this lock, a motor driver will be used, the ULN2003 motor driver. The motor Driver is a 16-pin component which requires a supply voltage large enough to power the physical motor.

To achieve this voltage, the components power supply will be wired to the 5-volt regulator. Then, the input will be connected to one of the GPIO pin on the microcontroller. Finally, the DC motor will be placed in parallel with a resistor and connected from the output of the motor driver to the 5-volt input from the voltage regulator.

5.1.5 Power Supply

With all the subsystems integrated into the SOLAS system, a power supply needs to be designed which will be able to work efficiently and have a good life span. To power the system, we plan to use a 4 AA battery holder. With the 4 batteries, the system should have a 6-volt power supply which should be enough to power the system. Since multiple components in the SOLAS system have a maximum voltage range under 6-volts, the power supply will also utilize two voltage regulators, a 5-volt regulator and a 3.3-volt regulator, as shown in figure 16. Since the power supply needed to operate the motor of the smart lock is estimated to be a larger voltage, the component will be connected directly to the battery holder.

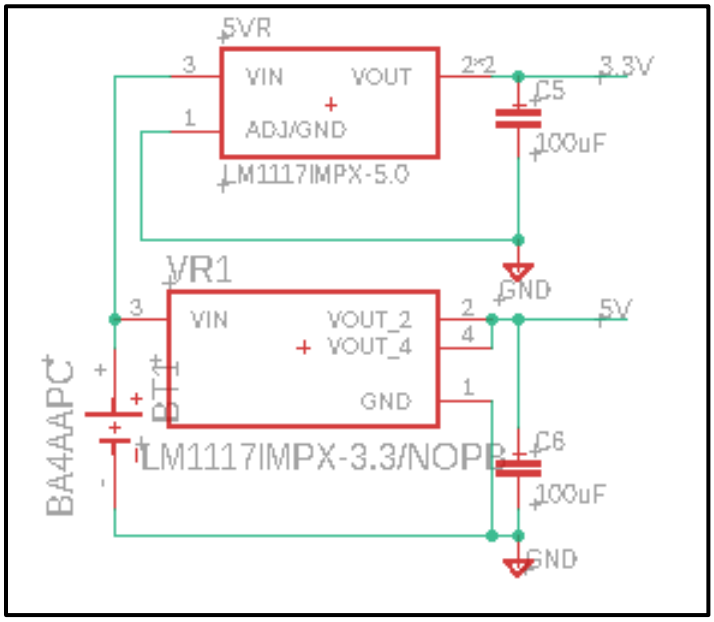


Figure 16: Power supply subsystem

5.1.6 Gesture Controller

The APDS-9960 Gesture Controller is an 8-pin surface mount component which can translate interpreted gestures into a digital signal. To access the connections on the gesture controller, an Adafruit breakout board will be utilized. The component requires a voltage supply at the Vin pin within the range of 2.4 volts and 3.6 volts. To achieve this, the 3.3 voltage regulator in the power supply subsystem will be utilized. Finally, the Microcontroller I2C pins will be connected using the INT, SCL, and SDA pins on the gesture controller.

5.1.7 Bracelet

The bracelet subsystem will be very simple. All it will incorporate is the Yarontech RFID bracelets. They come in sets of 5, which will allow demonstration of different users being able to open the

door, as well as the fact that some will not be able to open the door. The size of the bracelet where the RFID tag is embedded will allow the attachment of a logo or decal to differentiate the bracelets.

5.2 Hardware Design of Full System

The SOLAS lock system uses multiple sensors and readers to complete the functions that are specified. In order to connect all the subsystems together and to understand the data transmitted from those subsystems, the SOLAS system will utilize a microcontroller. The ESP32-WROOM microcontroller is a 48-pin component which includes thirty-four GPIO pins. As seen in figure 17, the microcontroller will connect to virtually every component in the system, excluding the smaller and wireless components. Some of these subsystems will connect to the microcontroller by using the basic GPIO pins, such as the motion sensor and LEDs. Other subsystems will need more specific GPIO pins to transfer more sophisticated data, such as the gesture controller and the RFID module.

When connecting the subsystems to the microcontroller, the design also needs to include the connections needed to supply power to the system. The Power Supply subsystem will be used to supply the necessary voltage to each component. As stated in section 5.1.5, there are some components that require higher supply voltages than other components are able to use. For the microcontroller, the typical power supply range is 2.3 volts to 3.6 volts. To achieve this voltage, the microcontroller will be connected to the 3.3-voltage regulator in the power supply subsystem. From this, other components that have this voltage within its power supply range will also be connected to this pin. As for the components that require a larger voltage, such as the smart lock motor driver, they will be connected to either the 5-voltage regulator or directly to the four AA battery holder.

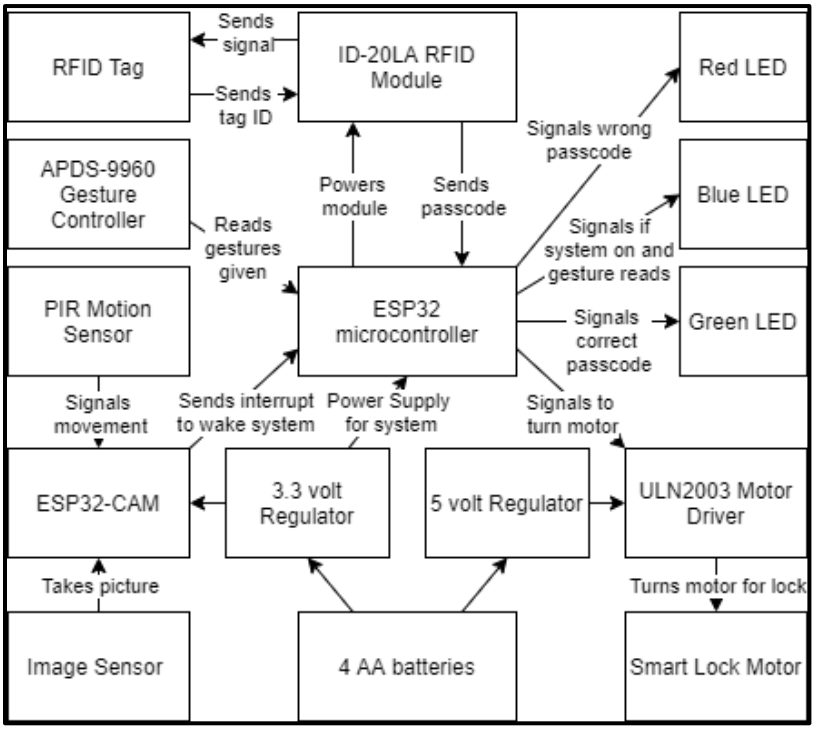


Figure 17: Hardware design of full system

Another hardware design that will need to be implemented into the SOLAS system is the casing in which the components will sit in. Since many of the components for the system will have to be design on the outside of the door, a casing is needed to protect the device from weather and possible break-ins. The material of this casing will need to be non-conductive as to not interfere with the RFID waves. With this in mind, we plan to use a hard-plastic electrical junction box as our outside casing. The junction box should allow the SOLAS system to fit all of its components.

5.3 Software Design

With the subsystems of the SOLAS smart lock designed and a schematic of the full system created, the next step in the design process can start. In this section, the process of how the software states will be designed and behave, as well as the selection and design of the website which will be used.

5.3.1 Software states of door lock

In order to incorporate the different layers of security, the software based in the door lock will be orchestrated in different states. The main states will be standby, waiting for bracelet communication, and waiting for correct password. Each of the states and various traversals between them will be accompanied by different colors of the LED in order to convey to the user what the current state is.

Before the user approaches the door, for example as they pull into the driveway, the SOLAS system will be in the standby mode to saving power. As the user walks to the porch area, the proximity sensor will be activated, at around a planned 4 feet from the door. Once the proximity sensor detects movement, it will wake up more of the system, starting its search for the bracelet, and the camera takes a picture of the area which is uploaded to the website. The RFID reader will send out its request for the RFID tag to send its identification and then wait for a response. This state will continue for 15 seconds, the RFID reader requesting a response once a second. These 15 seconds allows for a discrepancy between the time the proximity sensor senses movement and when the RFID tag comes within range, as well as preventing the website from being sent countless pictures. If after 15 seconds the RFID tag is not detected and the proximity sensor does not detect any movement, the whole system goes back to its standby mode to save battery.

Once the proximity sensor detects movement and bracelet is in range, the gesture controller subsystem is activated and attempts to read a password. The gesture controller is based upon receiving back light bouncing off of a nearby surface, much like the RFID reader, which should help eliminate “noise” in this part of the software state, since the only light received back will be off of the user’s hand. From here the software state can go in three ways. If the gesture controller does not read in anything that could be interpreted as a password attempt, it will time out and the whole system will go back to standby mode. If the gesture controller reads in an incorrect password, the esp32 will attach a red flag in the database to the photo taken. Once a password is successfully put in, the deadbolt will unlock, and the user can enter.

Some users will have their gesture password “turned off”, and so when they scan the RFID, the system will check to see if they have a gesture password and automatically unlock if they don’t.

All of these various states and responses of the system are illustrated in a visual manner below in figure 18, demonstrating which actions can lead to which states and how all of the states are connected to each other.

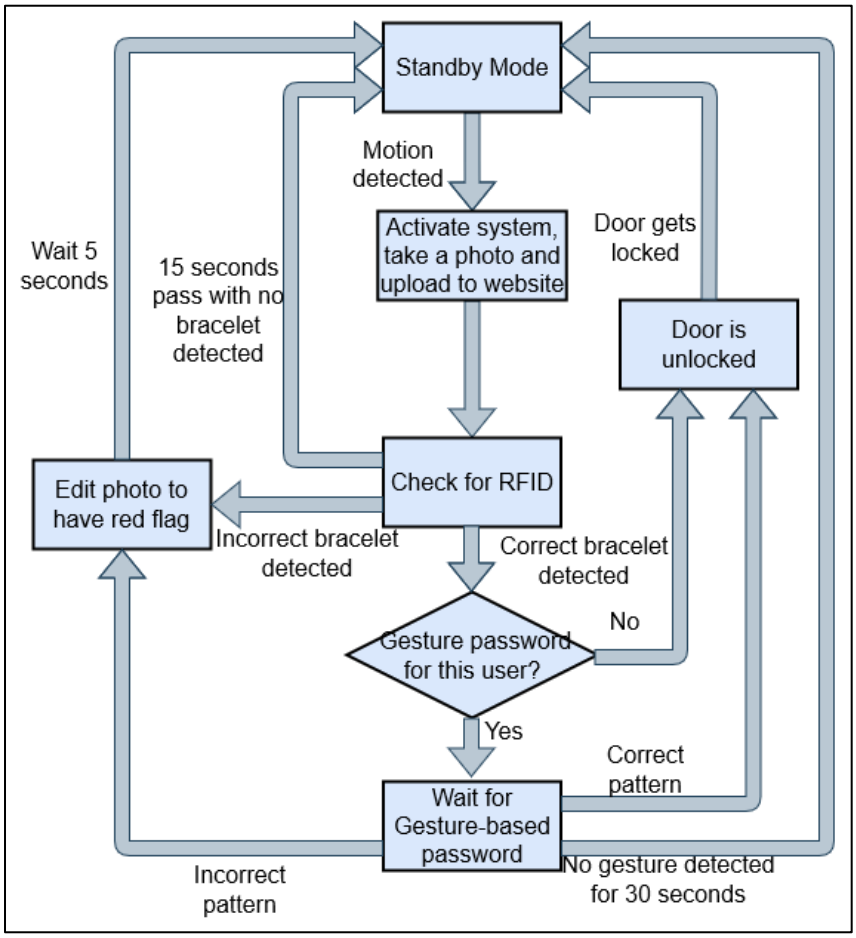


Figure 18: State diagram

5.3.2 Website

This section covers the specifics for the user website, including the stack design, interface, and database setup.

Website software

The only software component of this project for the user to utilize and interact with is the website to support the use of the camera. The most important feature of the website is the ability to view pictures sent from the camera. When movement is detected on the porch area by the proximity sensor, the camera takes a picture and automatically uploads it to the website. The user can then login to the website and view or delete these pictures. When choosing how to develop the website, the largest consideration will then be the feasibility of uploading files and their organization in the database.

One product for managing photos and videos on a website is Cloudinary. Cloudinary has free and priced options for media APIs to manipulate and manage multimedia on a developer's website. However, the products seem to be designed more for video-heavy websites selling products or photography pages. Our website does not need to advertise any products or commercials, and so a much simpler design will be sufficient. That leaves creating a full-stack website and using the built-in libraries of the various stack components.

While there are many web-stacks available to utilize, only 2 will be considered for this project as the software development team has experience with them. The first stack to consider is the MEAN stack, unique in the fact that it uses only 1 programming language, which would simplify the developing of the website. The stack uses the no-SQL database MongoDB, which compared to other databases will be very beneficial considering the use of images, and GridFS can be used if the image files from the camera end up being more than 16MB each. The web server that would be used would be Express.js, AngularJS, React.js, or Vue for frontend framework, and Node.js for runtime. If this stack is chosen, a possible product to use to implement the website with minimal development would be Meteor.

The other web stack to consider is the LAMP stack, comprised of the Linux operating system, Apache webserver, MYSQL database, and PHP script interpreter. While the more popular choice, the LAMP stack was much more difficult to implement for previous projects. The web-server interface is not as user friendly as with the MEAN stack, the latter even including a log file in which to debug error. The LAMP stack is also SQL communication based, which slowed down development time with its precise syntax, as opposed to the flexibility of MongoDB. All of the components also use different languages, further increasing development complexity. Considering the numerous downsides of the LAMP stack in the scope of this project, the MEAN, MERN, or MEVN stack will be used to develop a website for the user to view camera photos.

The only other main consideration for the website is how the email functionality will work. EmailJS will allow us to send emails to target emails using templates without having to keep an email username and password alongside the code on the server or local machine like nodemailer does. Using API keys, EmailJS is a more secure option to enable us to send emails to users when they register and when they forget their password. Not only is it more secure, but it also has a much easier visual option to customize the look and text of emails sent, as well as several other options and features including auto-reply, shown in figure 19 below.

EmailJS also allows the saving of various email templates. When configuring the website's backend, whenever an email needs to be sent, the function requires a template ID parameter, which is connected to the EmailJS account. With this method, we can make an email template for a forgot password email, welcome email, or any other emails from the website that becomes necessary, rather than having to hard-code the text of the email into the code.

The free version of EmailJS only allows for 200 emails to be sent from the account, but for the purposes of the demonstration, 200 should be more than enough, and paid plans would allow for a larger limit in the case of marketing the SOLAS system. EmailJS allows sending from most major email providers, including Gmail, Yahoo and Outlook. The team will be creating a Gmail account specifically for the demonstration so that when emails are sent, they will be coming from what looks the staff of the SOLAS system, rather than one of the team members personal email accounts. The only limitation of the EmailJS framework is that it only allows up to 500 emails per day, which the team does not expect to use.

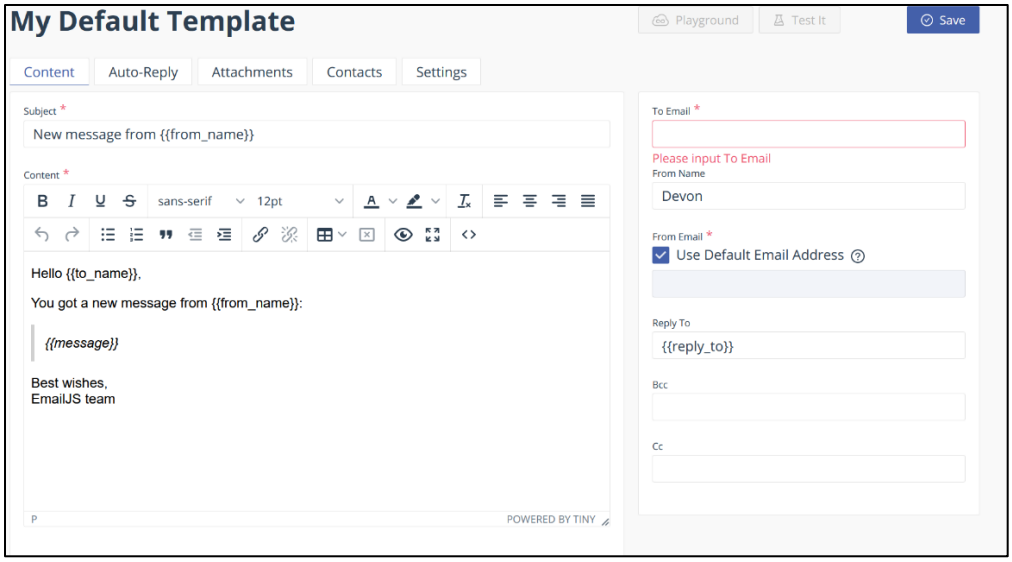


Figure 19: EmailJS template

However when the team went to configure EmailJS for automatic emails, it was discovered that EmailJS is only used for sending confirmation emails when users sign up for subscriptions or virtual newsletters on a website and could not be used for the desired purposes. Thus, SendGrid was used successfully in order to send emails to user emails from an email created for this project.

Website interface

When a user first visits the website, they will be brought to a homepage much like the one in figure 20, with a simple login or register option. The register page will look very similar except that the user will enter their first and last name, email and password, and a serial number or other identifying number associated with the camera on their door lock.

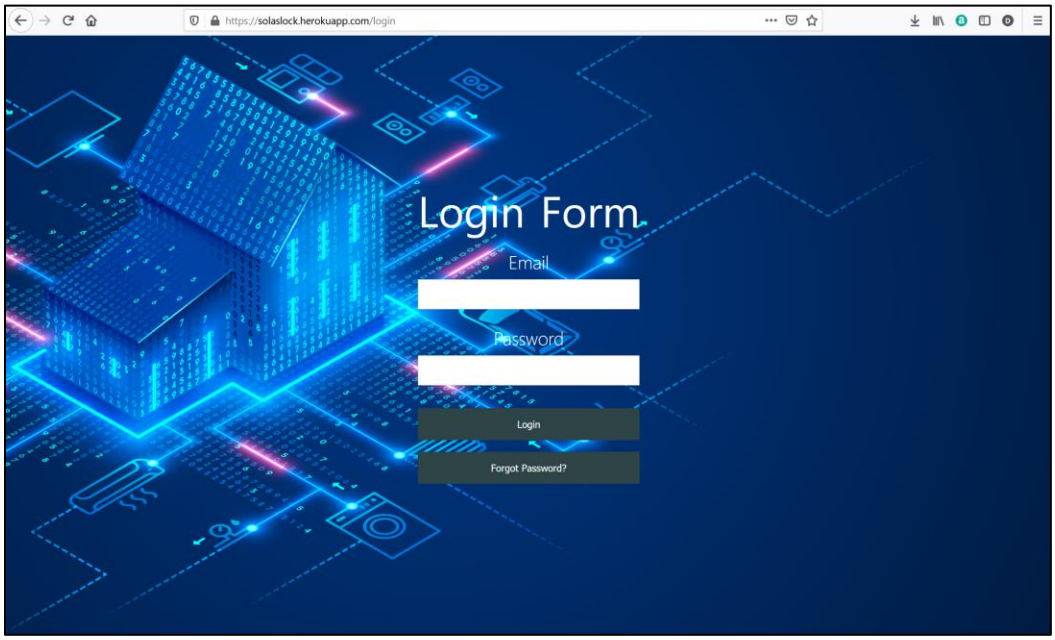


Figure 20: Website login page

For accessibility purposes, the website will also include a forgot password feature. When the user registers with their email, if they ever forget their password, they can recover it with the forgot password feature. All they have to do is type in the email associated with their account like shown in figure 21, and if there is one an email will be sent with a link to reset their password.

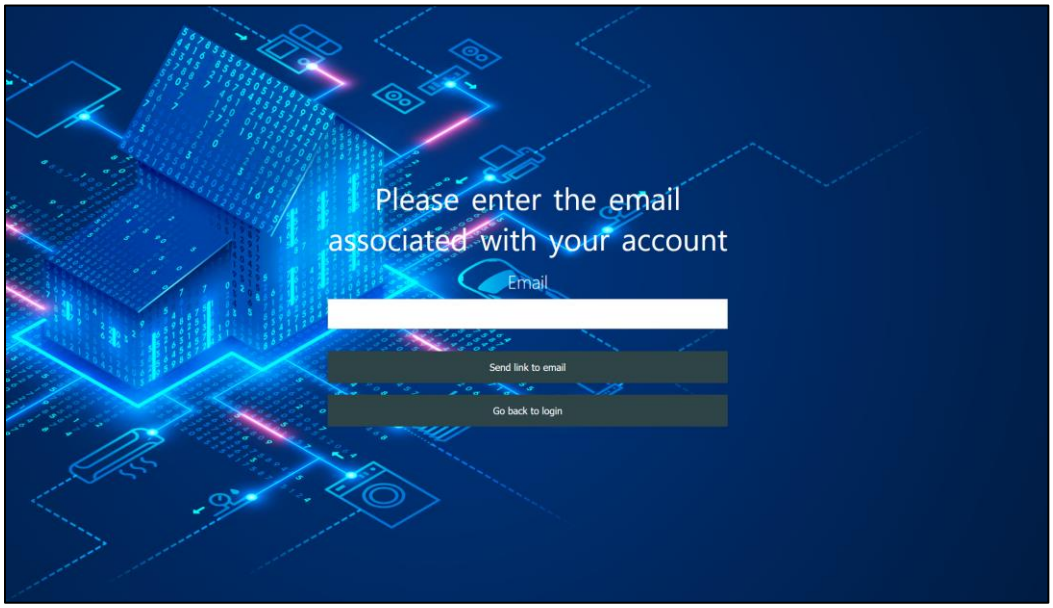


Figure 21: Website forgot password screen

Once logged in, the user will see a screen showing all photos their camera has taken, sorted by what time and date they were taken at. Each of the photos can be clicked on to view it at a larger size. The user will also have an option to delete the pictures individually or delete multiple all at once. The top of the interface will show which user is currently logged in, the option to log out of the session, as well as settings for the gesture controller, shown in figure 22.

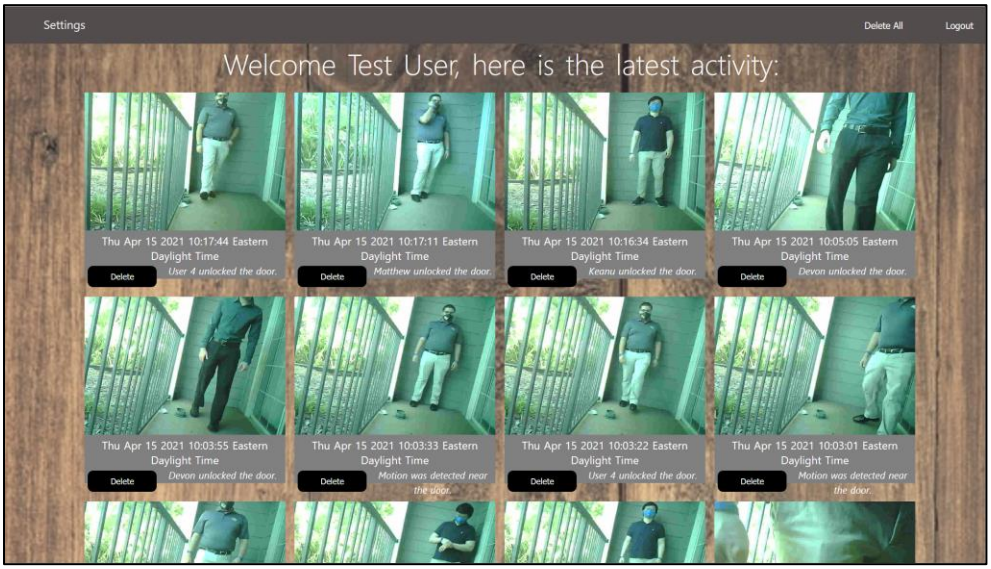


Figure 22: Website home screen

When the SOLAS system reads in an incorrect password and the camera posts a picture with a red flag, this will be clearly visible to the user. Any pictures that are sent with a red flag from the camera will be easily discernible to the user, and the text below the image with information on it will be in red, shown in figure 23.

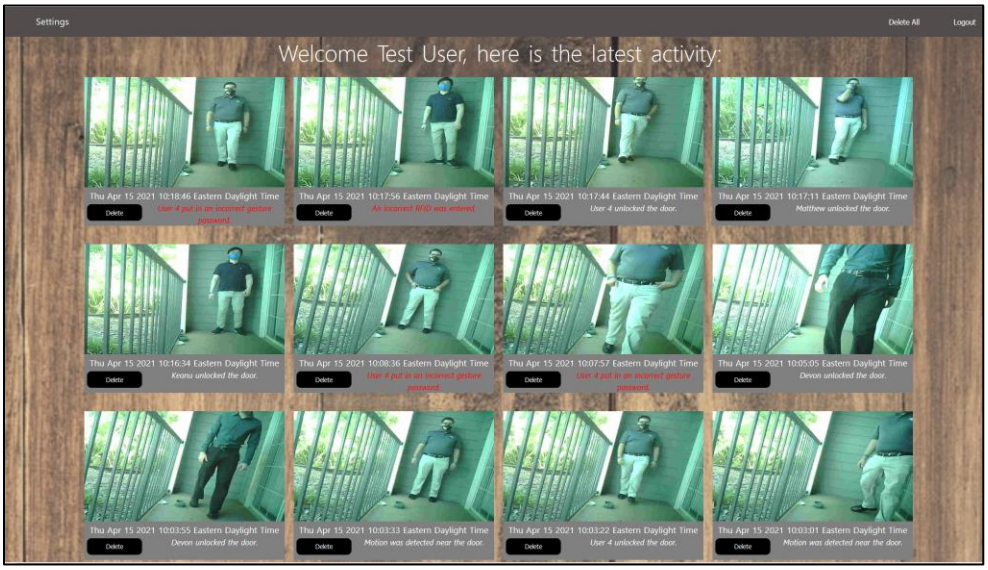


Figure 23: Website home screen with red-flagged images

A setting that the user will be able to edit through the website is the gesture controller and its password. As stated before, the gesture controller can be set to take a very complex password, a simple password, or be disabled entirely. A separate screen will be used for these settings, where the user will be able to choose from all forms of a single gesture and string several together to form their password. The choices for each gesture will be shown as well as the current password as they construct, shown in figure 24 below.

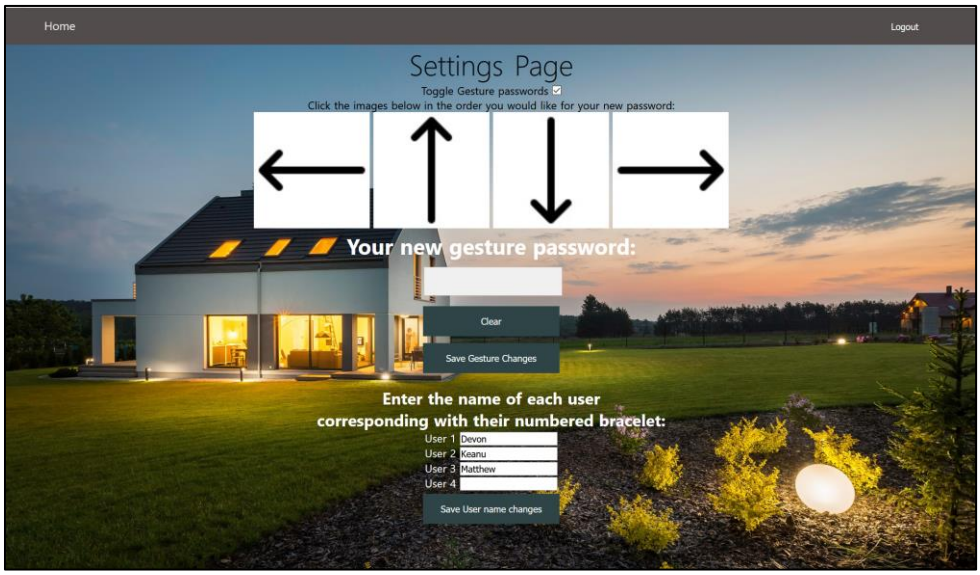


Figure 24: Website settings screen

On this same page, an additional feature is implemented to associate a name/user to each bracelet. Since the bracelets are numbered, the user can type in who is using each one so that their name appears with their respective event image.

Website database and camera communication

The SOLAS system comes equipped with a camera, which takes pictures of the surrounding area when it takes movement, then uploads the picture to the SOLAS website for the user to view. The specific camera must be connected to only specific accounts on the SOLAS website (it is possible for multiple accounts to be connected to one camera). Rather than attempting to have the camera login to users' accounts in some way using email and password, the camera will be connected to each user via a unique serial number only the camera owner can access. When the camera attempts to post the picture to the SOLAS website, this serial number will be sent as well. When the login page receives a POST with such a number, the database checks the camera serial numbers of all users and posts only to accounts that match. This is the same serial number that the user will input when registering their account which they will have access to when setting up their SOLAS system. Figure 25 below shows how the website database will be constructed to connect the users to images sent by the camera.

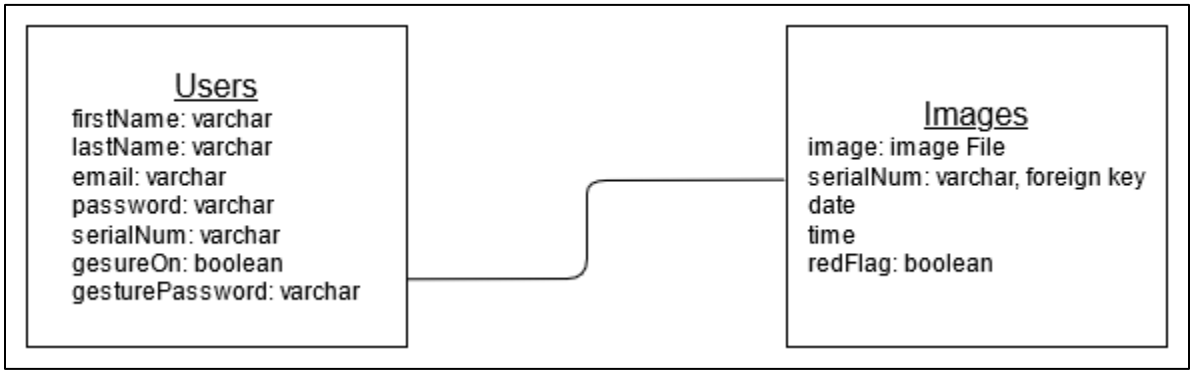


Figure 25: SOLAS website database structure

Website Use Case Diagram

The website will have the basic user functionalities expected of any products website. The user will be able to register with name, email, password, and serial number. They then login with that email and password combination. From there the user can view the pictures and delete the pictures the camera has uploaded. On that main screen they can also access a page to edit their gesture password. If they cannot login to their account, they can use the forgot password feature, which will send them an email if there is an associated account. The website will also have a logout button on the dashboard.

The only functionality/access the camera will have to the website is the ability to upload the pictures it takes as well as edit images to have a red flag, which is done by using the serial number and comparing to the number users have put in for their account.

These main cases are outlined in the use case diagram in figure 26.

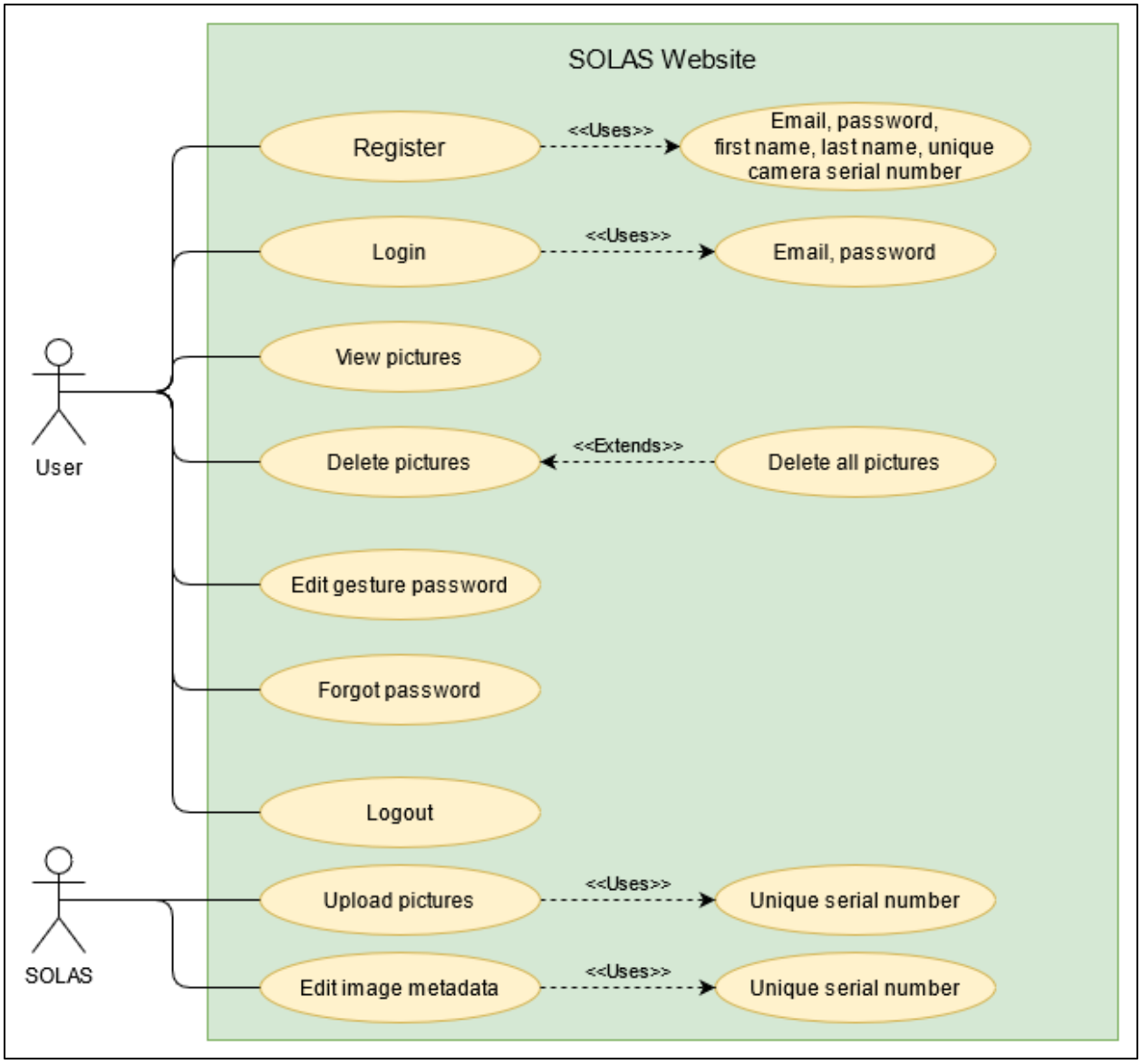


Figure 26: Website use case diagram

5.4 Summary of Design and Overall Schematics

The SOLAS system integrates multiple design aspects from both software and hardware. The following sections will summarize the design process that was used for the full system.

5.4.1 Hardware

The hardware in the SOLAS system integrates multiple different components, both active and passive. Despite the number of components in the system, the design of the hardware is a reasonable task. Some of the components in the SOLAS system are devices which require simple circuits to operate. However, most of the components, like the RFID module and ESP32-CAM module, are multi-pin components which need to be properly connected to the system. These

devices usually include typical application schematics in their respective datasheets that illustrates the number of components needed and where the wires will connect to each component. Using these schematics, the subsystems for the multi-pin components can be built easily and then integrated into the system.

The hardware design portion of the SOLAS system in theory is fairly simple. Realistically, to ensure that the components work properly and that the connections made are correct. Then, the subsystems need to be integrated together into the full system and tested to observe how each component reacts to one another. With this in mind, the testing phase is planned to take more time than the designing phase of the system hardware.

5.4.2 Software

The software of the SOLAS system will be simple on both the development side of the design as well as the user side to make it intuitive. The project will focus more on functionality rather than having many settings that make it seem complicated and fancy. For the development side for the door lock, the programming will be done mainly in C, which the software team has experience in. The door lock software will be based upon waiting for various inputs, first waiting for movement, then for RFID signal, then waiting for a password. The largest challenge is expected to be connecting the lock subsystem to internet and uploading pictures to the website.

The website will also be mainly functionality based as opposed to aesthetic with many features and settings, having very few screens for the user to have to navigate through. It will have login and register functionality, setup including a password reset and camera connection, a page to set a password for the gesture controller, and the homepage where the user can view taken photos. The website will use MongoDB, Express, Vue, and Node.

The various inputs and data flow are shown in figure 27 below. The user will provide the login, camera serial number, and gesture password to the door lock, and will be able to view images from the website. The website provides the information from the user to the database, which the microcontroller will communicate with, asking for the password (and possibly permanently storing) and uploading images as they are taken. When the camera uploads the images it also sends its serial number so that images can be sorted or queried by user/system. When the user provides gestures to the door lock, it will either compare it to a password it stored from the database or make a new request for the most current gesture password for that user.

For the RFID tag comparison, in the prototyping stage, the team will test the RFID tag by reading the data it sends to the RFID reader and displaying it on a console screen. Once the exact serial number for the acquired tag is known, the team will hardcode it as a variable into the microcontroller. Then once the entire system is constructed, whenever the RFID tag sends its serial number, the microcontroller can make a direct comparison to that local variable.

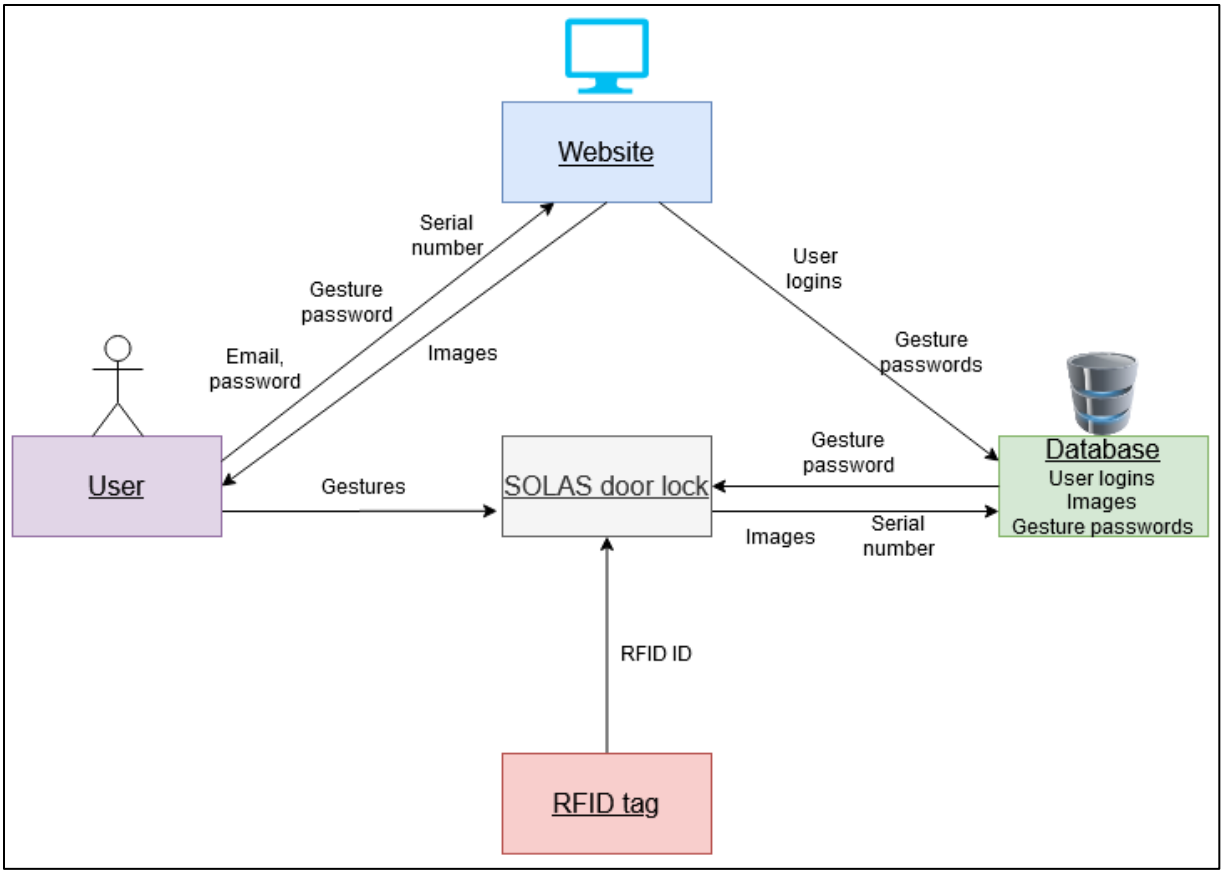


Figure 27: SOLAS system Information flow

6. Project Prototype Construction and Coding

This section covers the various prototype details that will be needed to construct the SOLAS system. The PCB schematics have been designed and routed, as well as much of the coding plan. The section also covers what parts have been bought for the project, what parts need to be bought, and what parts were already in possession in order to construct the prototype.

6.1 Integrated Schematics

To save the space needed to contain the circuitry for the SOLAS system, the Door lock will consist of two PCB boards, one to be place in the inside of the house, and the other on the outside. The PCB board that will be placed on the outside will have the RFID Module, the LEDs, the PIR Motion Sensor, and the Gesture Controller. On the inside, the PCB will have the motor driver, the ESP32 microcontroller development board, the ESP32-CAM module, and the power source for the system. To connect the two PCB boards, a female pin header will be added to each board and wires will be fed through the hole in the door to reach both sides. Since longer wires are going to be used, capacitors will be added to the voltage regulators to reduce the noise. The full schematic of the SOLAS system is shown below in Figure 28 and Figure 29.

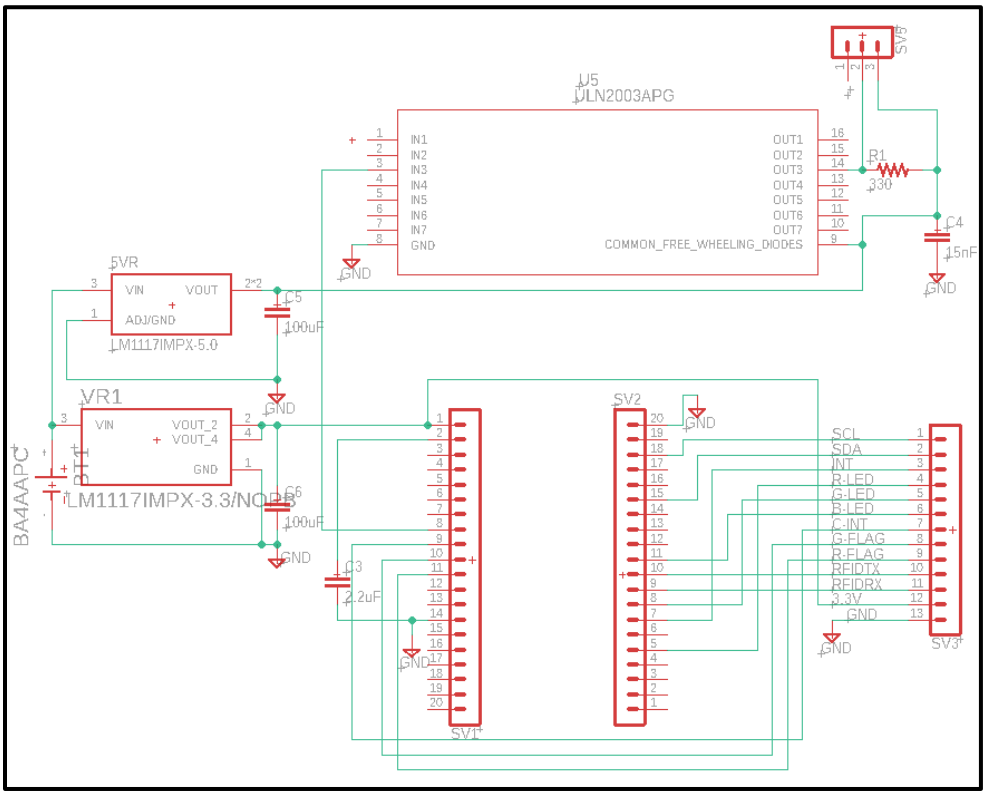


Figure 28: Schematic of SOLAS system main PCB

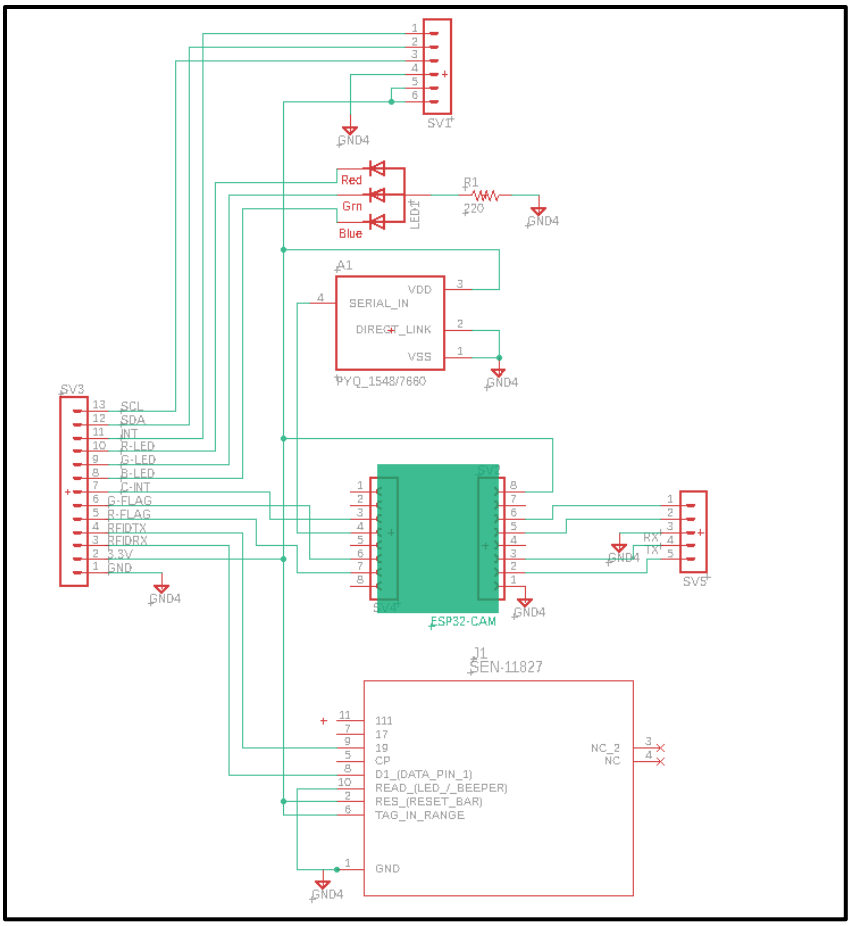


Figure 29: Schematic of SOLAS system sensor PCB

6.2 Project Parts Acquisition

The parts required from this project had to be ordered from several different vendors, and some were ordered from vendors that were not the manufacturer so that less orders could be placed, and costs of shipping could be reduced. Table 9 below shows which items were ordered from amazon. The ESP-WROOM-32 was ordered with a development board since ordering the $1.60 chip with $8 separate shipping was deemed unnecessary, and the board should aid in the final solution. The ANQUEUE PocketBands are much more expensive than the cheaper silicone bracelets from Chuangxinjia, but only one band is needed for demonstration.

Table 9: Items ordered from Amazon

Item	Cost
OrangeIOT Electronic door lock	$39.98
ESP-WROOM-32 development board	$10.99
ANQUEUE PocketBands	$9.99
ESP-32Cam Module	$13.99
Total with shipping and tax: $74.95	

Although initially an RFID tag from parallax was going to be used, it was changed to Sparkfun since an almost identical tag was sold by them and then it could be shipped with the RFID reader module. This order is shown in table 10 below.

Table 10: Items ordered from Sparkfun

Item	**Cost**
Sparkfun RFID Passive 125kHz Button	$3.95
Sparkfun ID-20LA	$34.95
Total with shipping and tax: $45.99	

After ordering the parts for the SOLAS system, they were observed and taken apart to understand their functions. The components that were bought so far are shown in figure 30 and their descriptions are labeled in table 11. From the Orange IOT smart door lock, the keyhole, deadlock, and motor were removed to be used in the SOLAS lock since these parts are a mechanical portion of the project. The RGB LED was obtained from an electronics kit from a previous course. Finally, the remaining parts were bought from various vendors such as Amazon and SparkFun.

Table 11: Part that have been acquired

Reference	Part Name
P1	ESP32-WROOM-32D Development Board
P2	ID-20LA RFID Module
P3	Passive 125kHz RFID Tag
P4	ESP32-CAM camera module
P5	OV2640 Image Sensor
P6	FTDI232 Programmer
P7	RGB LED, supplied from the team
P8	GM2215FD-0001 Motor from Orange IOT lock
P9	Keyhole
P10	Deadbolt
P11	Bracelet to hold RFID Tag

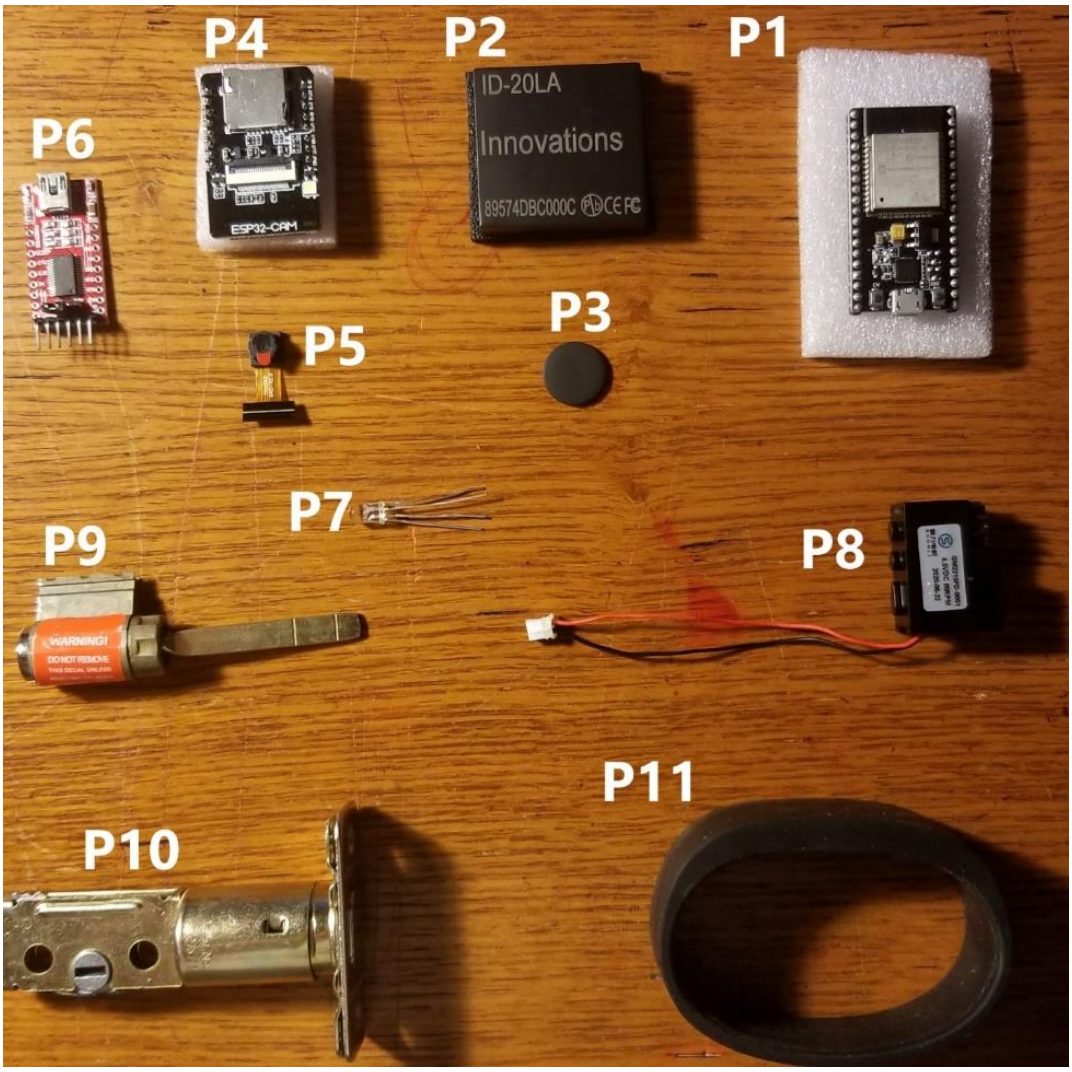


Figure 30: Components that have been purchased at this point

Tables 10 and 11 cover parts that were ordered for early testing of various subsystems, the tables 12 through 18 below cover all of the remaining various expenses that are expected to be occurred for a functional SOLAS system, including the website and domain. In these tables an estimate for tax and shipping are added in, using the previous orders for a base price on average shipping.

Table 12: Items to be ordered from Mouser

Item	Cost
Gesture Controller	$2.07
Estimated total with shipping and tax: $8	

Table 13: Items to be ordered from Texas Instruments

Item	Cost
Motor driver	$1.568
Estimated total with shipping and tax: $8	

Table 14: Items to be ordered from Digikey

Item	Cost
PIR motion sensor	$1.95
Estimated total with shipping and tax: $8	

Table 15: Items to be ordered from OSH Park

Item	Cost
PCB manufacturing	$60
Estimated total with shipping and tax: $70	

Table 16: Items to be ordered from Amazon

Item	Cost
Door handle	$26.95
Pinfox box enclosure	$8.99
Estimated total with shipping and tax: $40	

Table 17: Items to be ordered from Lowes

Item	Cost
Lumber	$13.26
Estimated total with shipping and tax: $20	

Table 18: Website domain and hosting

Item	Cost
Heroku	$7/month
Estimated total for 5 months: $35	

For all of the necessary additional parts such as lumber, software such as domain hosting, and shipping costs, the total estimated parts acquisition is $309.94, which is $9.04 above the team's original desired budget. If possible, during the prototyping phase of the project, the team will attempt to find similar parts at different vendors in order to reduce the number of orders and shipping costs.

These are the actual and expected costs from the first prototypes of the SOLAS system in December of 2020, the realized costs and expenses are summarized below in Table 23.

6.3 PCB Vendor and Assembly

There are many vendors with many pricing options for the PCB, which are discussed in this section. As well as the construction of the PCB board.

6.3.1 Vendor

For prototyping with a PCB, a good PCB vendor must be considered. An ideal PCB vendor would be one that fits within the project budget and will deliver in a timely manner. Two reputable PCB vendors that will be considered for SOLAS are discussed below.

Advanced Circuits

This vendor offers many PCB features and a free PCB design software called PCB Artist along with a PCB design checker through a file upload called FreeDFM. FreeDFM will check the Gerber file for any errors that could delay the PCB manufacturing and it will also correct some errors automatically. Advanced Circuits also offers special deals for 2-layer and 4-layer PCBs with specific restrictions.

2 – Layer Special Pricing Information:

- $33 Each with Minimum order Quantity of 3
- 3 Day turnaround time with shipping as quick as 1 Day
- FR-4 .062" Material
- 1 Oz. Copper
- Maximum Board Size of 60 sq. in.

4 – Layer Special Pricing Information:

- $66 Each with Minimum order Quantity of 4
- 5 Day turnaround time with shipping as quick as 1 Day
- FR-4 .062" Material
- 1 Oz. Copper
- Maximum Board Size of 30 sq. in.

OSH Park

This vendor offers excellent pricing, but turnaround time and delivery is lengthy. The turnaround can be reduced through its Super Swift Service at an increased cost.

2 – Layer Pricing Information:

- 3 Copies for $5 per square inch / Super Swift Service: 3 Copies for $10 per square inch
- Ships within 9 - 12 days / Super Swift Service: Ships within 4 – 5 Days
- 175Tg FR-4 Material
- 1 Oz. Copper with 2 layers
- Thickness of 1.6mm

4 – Layer Pricing Information:

- 3 Copies for $10 per square inch
- Ships within 9 - 14 days
- 190Tg FR408-HR Material

- 1 Oz. Outer and ½ Oz. Inner Copper with 4 layers
- Thickness of 1.6mm

JLCPCB

JLCPCB is an overseas PCB manufacturer based in Hong Kong. It is an experienced company with 14 years of experience. They offer an excellent deal on 2-layer PCBs under 100x100mm, with 5 copies for only 2 dollars and quick turnaround time which is only held back by overseas shipping time and costs.

2 – Layer Pricing Information:

- 5 Copies for $2 for Size ≤ 100×100mm and $56/$m^2$
- Ships within 3 - 5 days using DHL International Express shipping method and has 24 hours turnaround time
- FR-4 Material, HASL (with Lead) Surface Finish
- 1 Oz. Copper with 2 layers
- Thickness of 1.6mm

4 & 6 – Layer Pricing Information:

- 5 Copies for $5 for Size ≤ 50×50mm and $83/$m^2$
- Ships within 3 - 5 days using DHL International Express shipping method and has 4-5 days turnaround time
- FR-4 Standard Tg 130-140C Material, HASL (with Lead) Surface Finish
- 1 Oz. Copper with 2 layers
- Thickness of 1.6mm

Comparing these three manufacturers, the best vendor for this project was JLCPCB. Although a shorter shipping time is ideal, it is more important to keep within the budget of this project and the shipping times are not long enough to consider the more expensive manufacturing and faster shipping from Advanced Circuits. OSHPark is also held back by its cost per square inch so larger PCBs would be much more expensive. Although professional assembly of components is offered by PCB manufacturers, this option will not be taken as it is cheaper for the team to assemble it and hand solder each component.

6.3.2 Assembly

For the Assembly of the PCB boards that will be used in the SOLAS system, the EAGLE program will be used. First, the specific components that are going to be used needed to be added to the EAGLE library. To obtain the schematics and footprints for these parts, the Ultra Librarian and SnapEDA websites will be used. The parts that were downloaded from these websites include the ULN2003 motor driver, the ESP32-WROOM-32D microcontroller development board, and the ID-20LA RFID module. Since the schematics and footprints for the ESP32-CAM module and the APDS-9960 Gesture Controller Breakout board could not be found, they will be replaced with pin headers.

Once the components were downloaded, an initial schematic was built including necessary resistor and capacitors. When the final schematic is approved, the generate board function in the EAGLE program will be used to build the physical board. Here, the size of the PCB boards will be decided, and the components will be placed onto the boards. Once they are designed, the auto-routing function will be used to set the connections for the PCB boards. While setting up the connections, the space behind the ID-20LA RFID module and WFI antennas needs to be cleared of wires to allow the signals to be sent without any interference. The final design for the PCB boards is shown below in figure 31 and figure 32.

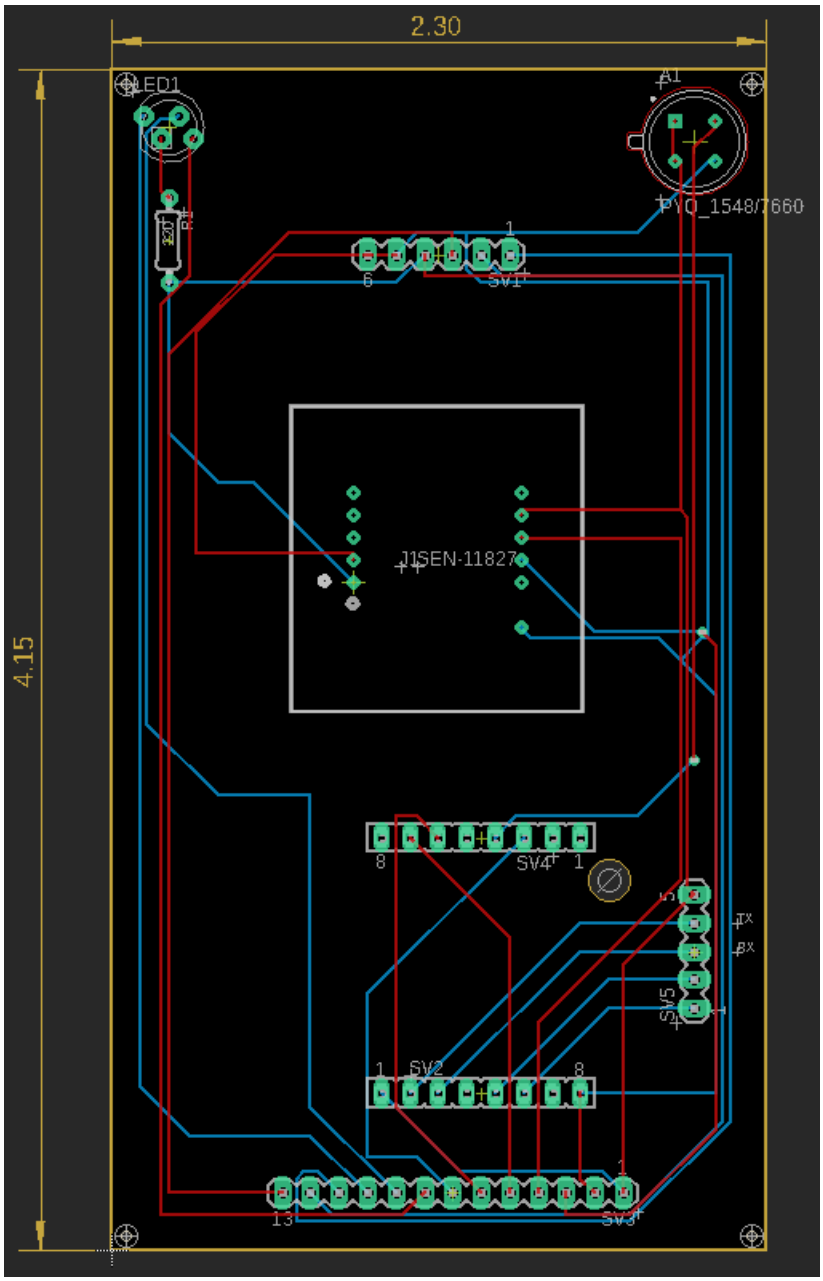


Figure 31: PCB board that will be connected to the Outside Portion of the SOLAS system (in inches)

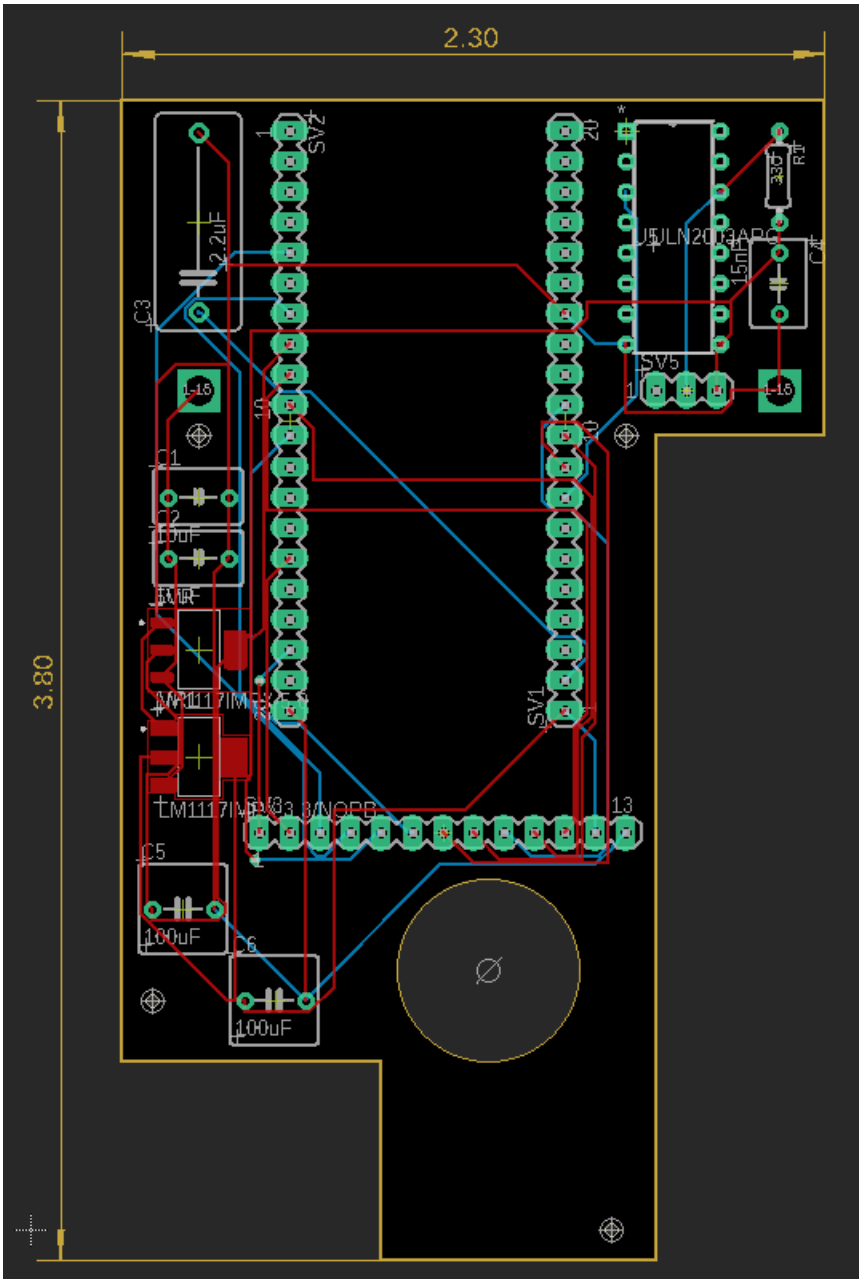


Figure 32: PCB board that will be connected to the Inside Portion of the SOLAS system (in inches)

6.4 Final Coding Plan

In order to program the ESP32 board, several pieces of software will have to be used, all of which are outlined in an Espressif programming guide specifically for ESP32 boards[37]. Since this guide is made by the manufacturer of the board, the team will use the suggestions and steps outlined in the document when programming the board.

All of the source code and libraries containing many of the functions that will be used are in ESP-IDF, which is obtainable in a git repository. ESP-IDF also supplies the framework to configure the

project and scripts to operate Toolchain. Toolchain is the software that will be used to compile the code for the ESP32. Once the code is compiled, CMake and Ninja finish building the application that is uploaded to the device. The IDE that the team plans to use in conjunction with CMake is the Arduino IDE. The interaction and connection between the software, computer, and ESP32 are visually shown in figure 33 below.

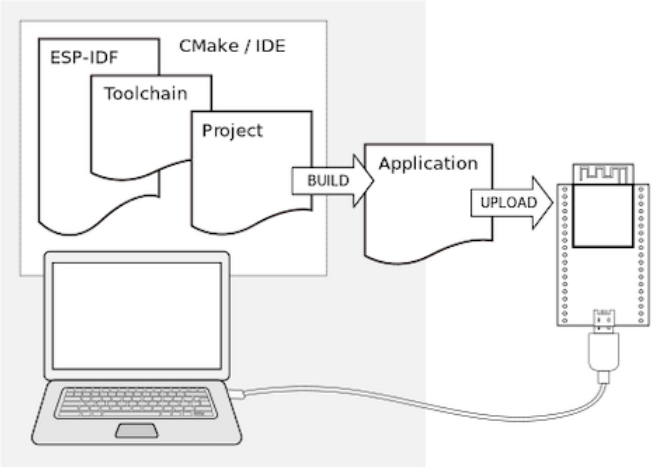


Figure 33: Espressif "Development of applications for ESP32", permission requested

Using the software outlined above, applications/projects will be able to be built and flashed onto the ESP32 board. The next step is programming the specific responses to inputs the board receives as the user requests entry.

A very useful and integral part of the lock software is going to be the use of interrupts[38]. Inside of the Arduino IDE, calling the attachInterrupt() function will allow interrupts from the proximity sensor to be read and specific functions to be called. AttachInterrupt() takes as arguments which Pin to monitor, the monitoring mode, and what code block to execute. The monitoring mode will be RISING, which means that as the input from the pin the proximity sensor is connected to goes from LOW to HIGH, motion has been detected and the next section of code can be called.

When motion is first detected, the next actions taken by the SOLAS system are to take a picture with the camera module and turn on the blue LED by setting its respective pin high. Among lots of other setups to be done, the photo is taken with esp_camera_fb_get() from inside the Arduino IDE[39]. Once it is confirmed that the camera was able to successfully able to take a photo, it is uploaded over Wi-Fi to the website automatically. The camera module has a Wi-Fi port and normally the WiFi.begin function would be called at the beginning of the camera's source code in order to be connected to the Wi-Fi using pre-coded SSID and password. However, for this project, having a pre-coded SSID and password would work, but not for a market user or once the lock system is moved.

Alternatives to using the pre-coded Wi-Fi credentials were discussed. The obvious solution to the problem was to create a mobile app for the project, and during initial setup, have the phone connect locally to the door lock system via Bluetooth and provide the Wi-Fi credentials to the ESP32 board to connect to Wi-Fi. However, as discussed before, the goal for the project is to not use Bluetooth in the system, unnecessarily adding more steps and security measures to the system while not

actually making the system any more secure. Additionally, the SOLAS system is designed to not require a mobile app for the user to have to download in order to configure their system. All the settings and configurations for the SOLAS system can be done straight from any internet browser, mobile or computer. If Wi-Fi credentials were provided via Bluetooth and a mobile app, a Bluetooth connection and setup would have to be made in the source code for the sole purpose of initial Wi-Fi connection, the same being true for the mobile app.

Since creating a mobile app and making an undesired connection via Bluetooth for the one time use of connecting the SOLAS system to Wi-Fi appeared to be a poor solution, another alternative was researched. An Arduino library called AutoConnect[40] solves exactly this problem. In order to eliminate the need of hard-coding Wi-Fi credentials into the code, it allows the user to enter them at setup time via a mobile phone. Rather than using Bluetooth and a full mobile application, the phone connects to the ESP32 or esp8266 board from available Wi-Fi connections with a password that could be provided to the user in an instruction manual. Once connected, the AutoConnect library provides a very simple screen to input the SSID and password of the desired Wi-Fi network, shown in figure 34.

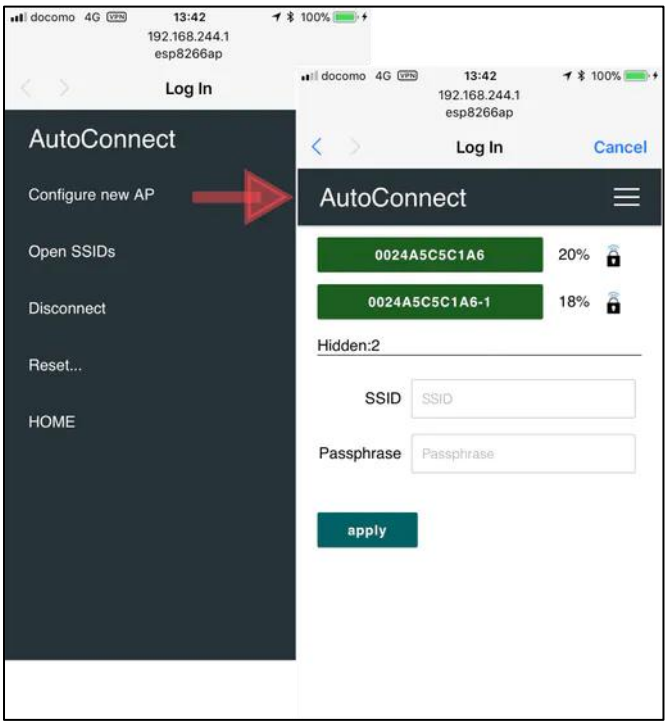


Figure 34: Hieromon-ikasamo AutoConnect screenshot, permission requested

AutoConnect provides the perfect solution to the Wi-Fi credentials issue. A full mobile app is not necessary just to enter Wi-Fi credentials, and it allows the system to connect to Wi-Fi networks anywhere, also allowing connection wherever the demonstration of the SOLAS project is. It also does not use the Bluetooth connection, but rather a password-based Wi-Fi connection. This software will allow the team to focus on the purpose of the project rather than small connection details.

Once the ESP32 board is connected to WIFI and an interrupt is called due to motion, a connection to the SOLAS website will be made. The client.connect() function will be used here, with a hard-

coded server name that will be input once the website domain is setup, as well as the serverPort, which should be 80. The photo is then formatted over several steps and sent to the website's database. This will be done by sending the photos to a URL of the website that reroutes the images and uploads them to the database. Since the ESP32-Cam has built-in WIFI and processor, it is possible that it will send the images to the website rather than sending them to the ESP32 board to send to the website, and then notify the ESP32 board once complete. This will depend on whether the AutoConnect can supply login credentials to the ESP32-Cam as well.

Once the picture is successfully taken, the next action of the SOLAS system is to identify whether the RFID bracelet is in range. In the ESP32 code, library software will be included or imported to aid in the setup of communication between the RFID reader module and the ESP32 board, to include the SoftwareSerial and SerialRFID libraries[41]. A Serial connection will be made between the board and reader module on specific pins with a 9600 baud rate which determines how fast the data is transmitted between the two devices; this will be done via the .begin() function from the libraries mentioned. Once the connection is started, the .readTag() allows a nearby RFID tag's ID to be read and analyzed. When the system is first tested, the tag used for demonstration will be read this way and printed out to a console so that it can be hard coded into a variable. Once hard-coded, any RFID IDs read in can be compared to this variable and accepted or denied. The comparison will be made using the SerialRFID::isEqualTag() function.

The output of the isEqualTag function determines what the software will do next. If after attempting to read in an RFID tag for around 10 seconds and none is detected, then the ESP32 board and RFID reader will go back to a power saving state. This will require the use of the millis() or delay() function[38]. The delay function merely blocks for the set amount of time before continuing, allowing for an easy loop to be constructed checking a parameter at set intervals. The advantage of the millis function is that it returns the time since program start without blocking, and if checked at the beginning of a loop iteration, the parameter can be checked at a set interval without blocking the rest of the program. Since at this stage the ESP32 may be still formatting and sending the image taken before, the millis function will be used; an interval of 1 second will be used, and every time it passes the RFID module will be called and attempt to read in a tag ID. If after 10 attempts no tag is detected, the software will have the hardware return to power-saving mode.

Assuming that an RFID tag is read in and matches the ID in the system, then the gesture controller will be awakened from its power-saving mode. The gesture controller works off very basic functions using 4 different diodes[2]. The main gestures that will be used are movement in an upward direction, downward direction, as well as left and right. If these gestures are seen, the gesture controller outputs 0-4, with 0 being no gesture seen, and 1-4 referring to distinct gestures. These calls to read a gesture are done from the ESP32 board by calling the. gesture() function included in libraries that will be used in this section of the code. The gesture controller will communicate with the ESP32 board via an I^2C interface. If desired, using the 4 diodes, many directional gestures can be added to be read in. When the ESP32 board receives a gesture code/codes, it will refer to the sequence set by the user on the website to decide if the password is correct. To do so a connection will once again be made to the website and read a file in a predetermined directory, which is updated by the user when they set and change their gesture password.

If when reading in gestures, a sequence is read that does not match what is in the system, then the SOLAS system will do 3 things. For the picture sent earlier, a Boolean will also be sent indicating

that the picture sent has a “red flag” associated with it. When displaying pictures to the user, the website will reference the pictures as well as the Boolean associated with them and either display normally or with a red border. The second will be to wait 5 seconds. This 5 second delay will act as a deterrent as well as prevent an incorrect password from being read multiple times in succession and sending many pictures to the website. The third is to turn on the red LED to show that an incorrect input was received.

Once a correct gesture password is read in, the ESP32 board will signal the motor driver to turn the lock that will allow the door to be opened and turn on the green LED. Figure 35 below visualizes how the components within the door lock subsystem will interact with each other and the information they hold, assuming that the ESP32-Cam is able to use AutoConnect to upload the pictures as well as its serial number over the Wi-Fi connection.

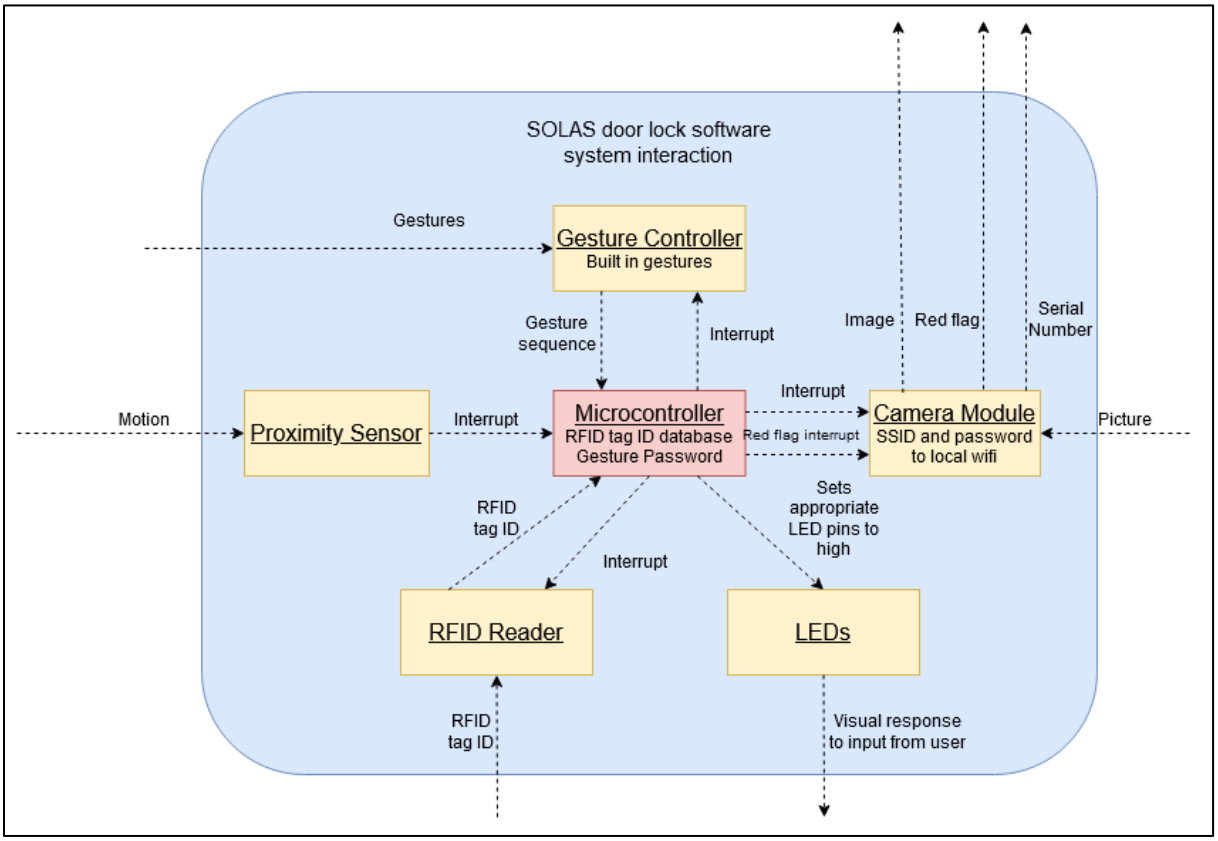


Figure 35: Components Interaction SOLAS Lock

6.5 Door Lock Casing

In order to prevent the electronic components from exterior damage and to secure the lock, a casing seeds to be added to the SOLAS system. Since the SOLAS system as multiple components in the front section that would normally be too large for other prebuild door lock cases, this door lock will need to have a custom casing. Since the casing is going to be built by the team, a material needs to be chosen. To ensure security, a metal or hard plastic would be necessary to protect the

lock. Due to the SOLAS system using WIFI and RFID signals, the casing will be built with hard plastic.

Now that the material has been decided, the casing size and shape needs to be designed. To simplify the construction of the casing, an electrical junction box will be used as the base and will be modified to fit the needs of the SOLAS system. Since the casing as to fit the entirety of the lock base and the electronics, the junction box needs to be at least 2 ½ inches in width and 5½ inches in height. Once we have a junction box with the appropriate dimensions, the casing will be built. First, several sections from the casing will be removed to fit the base of the lock and the multiple components including: the LED, the OV2640 image sensor, the PIR sensor, the ID-20LA RFID module, and the APDS-9960 gesture controller. With the casing constructed, the lock base will be attached using a brace to ensure the casing stays connected to the SOLAS system. A sketch of what the SOLAS system is to resemble is shown in figure 36.

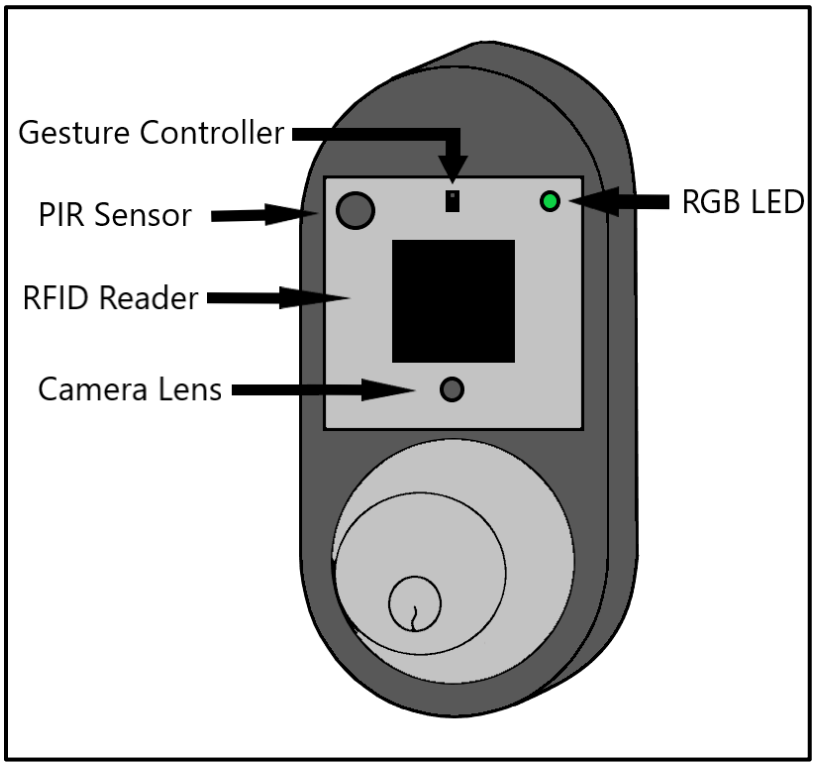


Figure 36: Rough Sketch of the prototype for the SOLAS system

7. Project Prototype Testing Plan

Once the prototype is constructed, many tests will have to be performed encompassing all of its functionality. These tests include the hardware, microcontroller software, and the user website.

7.1 Hardware Test Environment

To ensure that the SOLAS system can work properly and efficiently, the following test will be conducted. All the following tests will be performed on a bread board. Each of the subsystems will be tested separately with the ESP32 microcontroller to confirm that the connections made are correct. More detailed description of each test will be expressed in the next section. To measure the input voltage and output voltage of each subsection, the Analog Discovery 2 Oscilloscope and Instrumentation System provided, that was provided by Dr. Lei Wei and Dr. Samuel Richie, will be used. Once all the subsections are working properly, they will be connected to test the full SOLAS system.

7.2 Hardware Specific Testing

These section covers all tests that will performed on the hardware to verify it meets the requirements set in section 2.4.

7.2.1 Power Supply

Objective:

This test will be used to ensure that the power supply is delivering the correct Output Voltages.

Supplies:

- 4 AA battery Holder
- 5V Voltage Regulator
- 3.3V Voltage Regulator
- Capacitors
- Bread Board
- Analog Discovery 2 Oscilloscope and Instrumentation System

Procedure:

1. Connect the 5 V regulator input to the 4 AA battery holder output and connect the ground.
2. Repeat previous step with the 3.3 V regulator.
3. Insert the 4 AA batteries into the battery holder
4. Using the Analog Discovery 2 Oscilloscope and Instrumentation System, read the output voltages and currents of the 5V regulator and the 3.3V regulator.
5. Confirm that the output of the two regulators match their expected output voltage.
6. Add a capacitor in parallel to the battery holder and observe if the noise is reduced.

Result:

The result from this test showed that the voltage regulators work with the 4 AA batteries and will supply enough power for the system. The 5-volt regulator gave an output voltage of about 4.98V and the 3.3-volt regulator gave an output voltage of about 3.32V.

7.2.2 Motor

Objective:

This test will be used to ensure the motor driver will power the motor correctly and to confirm that the provided voltage will power the motor.

Supplies:

- ULN2003 Motor Driver
- GM2215FD-0001 Motor
- ESP32-WROOM-32D Microcontroller
- Capacitors
- Resistors
- Bread Board
- Analog Discovery 2 Oscilloscope and Instrumentation System

Procedure:

1. Connect the ESP32 Microcontroller Board to a computer using a USB connection.
2. Connect the IO33 pin of the ESP32 Microcontroller to one of the inputs of the motor driver.
3. Connect a 220Ω resistor from the output of the motor driver to 5V.
4. Connect the input power of the motor driver to 5V.
5. Connect a 10µF Capacitor in parallel with the motor driver from 5V to ground.
6. Measure the output voltage of the motor driver using the Analog 2 Discovery.
7. Connect the motor in parallel to the resistor from the output of the motor driver to 5V.
8. Program the ESP32 microcontroller to turn on the motor.
9. Test the torque of the motor.
10. Program the ESP32 microcontroller to be in stand-by mode.
11. Measure the output voltage of the motor driver.

Result:

The result from this test showed that the output of the motor driver will turn the motor with enough torque to unlock the deadbolt.

7.2.3 Motion Sensor

Objective:

This test will be used to ensure that the PIR Motion Sensor works correctly and to test the distance that the sensor can detect.

Supplies:

- BL412 PIR motion sensor
- ESP32 Microcontroller
- Bread Board
- Analog Discovery 2 Oscilloscope and Instrumentation System

Procedure:

1. Connect the ESP32 microcontroller to a computer using USB connection.
2. Connect the Vcc pin of the PIR motion sensor to the 3V3 pin of the ESP32 microcontroller.
3. Connect the ground pins of the ESP32 microcontroller and the PIR motion sensor together.
4. Measure the output voltage of the PIR motion detector when it does not detect movement.
5. Measure the output voltage of the PIR motion sensor when it detects movement.
6. Connect a GPIO pin of the ESP32 microcontroller to the output of the PIR motion sensor.
7. Connect the ONTIME pin of the PIR motion sensor to ground.
8. Program the ESP32 Microcontroller to print to the serial monitor when the motion sensor detects movement.
9. Test to see if the Motion Sensor is can properly detect movement and notify the ESP32 microcontroller.
10. Test the detection distance of the Motion Sensor.

Result:

The result of this test showed that the PIR motion sensor works and that the output voltage of the motion sensor is read well by the microcontroller. The measure distance that the PIR motion sensor can read is 5m.

7.2.4 Camera

Objective:

This test will be used to learn how to connect the ESP32-CAM module and Image Sensor and ensure that both components work properly and can send the image over Wi-Fi to the database. The connections that will be made are shown below in figure 37.

Supplies:

- ESP32-CAM module
- OV2640 Image Sensor
- Capacitors
- Resistors
- Bread Board
- Analog Discovery 2 Oscilloscope and Instrumentation System

Procedure:

1. Connect the ESP32-CAM to a computer using USB connection.
2. Measure the 5V pin on the ESP32-CAM to ensure that the voltage is correct.
3. Strap together the IO0 pin and the GND pin of the ESP32-CAM module.
4. Connect a capacitor in parallel to the ESP32-CAM module from the 5V pin to Ground to eliminate noise.

5. Connect the OV2640 image sensor to the ESP32-CAM module.
6. Program the ESP32-CAM module via the USB connection.
7. Take a picture in different light intensities to observe the resolution of the image sensor.
8. Measure the output voltage of the ESP32-CAM module.
9. Check the database to see that the images were sent and received properly.

Result:

The result of this test should show that the ESP32-CAM module works properly and can sent the taken image to the database. It should also show that the image sensor that will be used can take acceptable images for the SOLAS system. If this is not the result, the connections need to be checked.

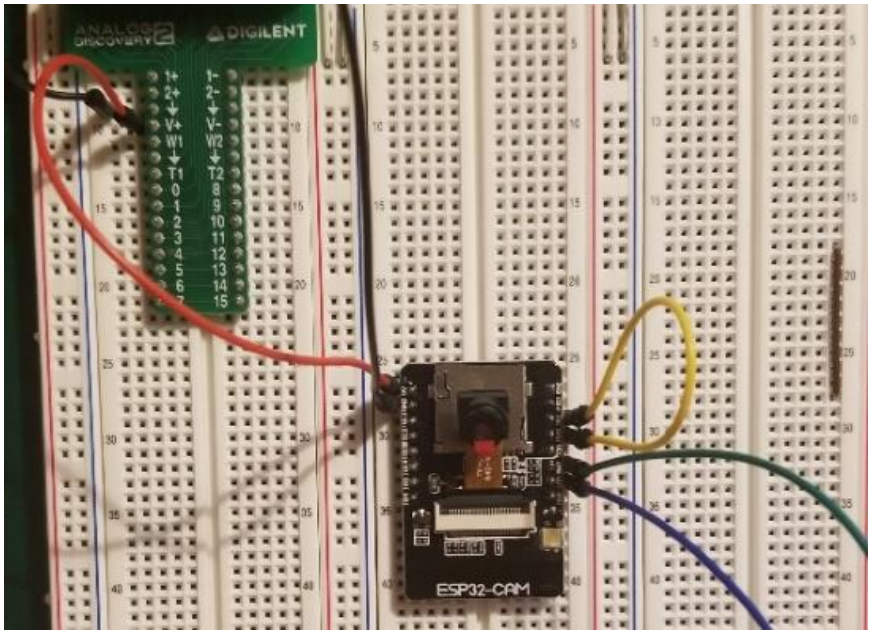


Figure 37: Breadboard image of ESP32-CAM subsystem

7.2.5 Gesture Controller

Objective:

This test will be used to learn how to connect the Gesture Controller to the ESP32 microcontroller and ensure that it works properly.

Supplies:

- APDS-9960 Gesture Controller
- ESP32-WROOM-32D Microcontroller
- Bread Board
- Analog Discovery 2 Oscilloscope and Instrumentation System

Procedure:

1. Connect the ESP32 microcontroller to a computer using USB connection.

2. Measure the 3V3 pin on the ESP32 microcontroller to ensure that the voltage is correct.
3. Connect the 3v3 pin on the ESP32 microcontroller to the Vin and 3Vo pins on the gesture controller.
4. Connect the INT pin of the gesture controller to the IO0 pin of the ESP32 microcontroller.
5. Connect the SCL pin of the gesture controller to the IO22 pin of the ESP32 microcontroller.
6. Connect the SDA pin of the gesture controller to the IO21 pin of the ESP32 microcontroller.
7. Connect the ground pins of the two components.
8. Program the ESP32 microcontroller to use the gesture controller.
9. Test to see if the gesture controller can detect movement.
10. Program a specific gesture to be used.
11. Test the programed gesture to see if the gesture controller can recognize a specific gesture.
12. Program the ESP32 microcontroller to be in stand-by mode.
13. Measure the output voltage of the gesture controller.

Result:

The result of this test showed that the gesture controller works properly and can recognize the gesture. The gesture controller was able to read up, down, left, and right gestures.

7.2.6 RFID Module

Objective:

This test will be used to learn how to connect the RFID module and to ensure that the module can read the RFID tag. The connections that will be made are shown below in figure 38.

Supplies:

- ID-20LA RFID module
- RFID tag
- ESP32-WROOM-32D Microcontroller
- Capacitors
- Resistors
- LED
- Bread Board
- Analog Discovery 2 Oscilloscope and Instrumentation System

Procedure:

1. Connect the ESP32 microcontroller to a computer using USB connection.
2. Measure the 3V3 pin on the ESP32 microcontroller to ensure that the voltage is correct.
3. Connect the 3v3 pin on the ESP32 microcontroller to the voltage supply pin on the RFID module.
4. Strap together the voltage supply pin and the Reset pin of the RFID module.
5. Connect the Data0 and Data1 pins of the RFID module to the IO17 and IO16 pins of the ESP32 microcontroller respectively.
6. Connect the ground pin and the format selector pin of the RFID module to ground.
7. Program the ESP32 microcontroller to use the RFID module.
8. Test to see if the RFID module can detect the RFID tag.

9. Using the ESP32 microcontroller, see if the data on the RFID tag transfers over.
10. Test the distance that the RFID module can read the RID tag.
11. Program the ESP32 microcontroller to be in stand-by mode.
12. Measure the output voltage of the RFID module.

Result:

The results of this test showed that the RFID module can properly read the RFID tag.

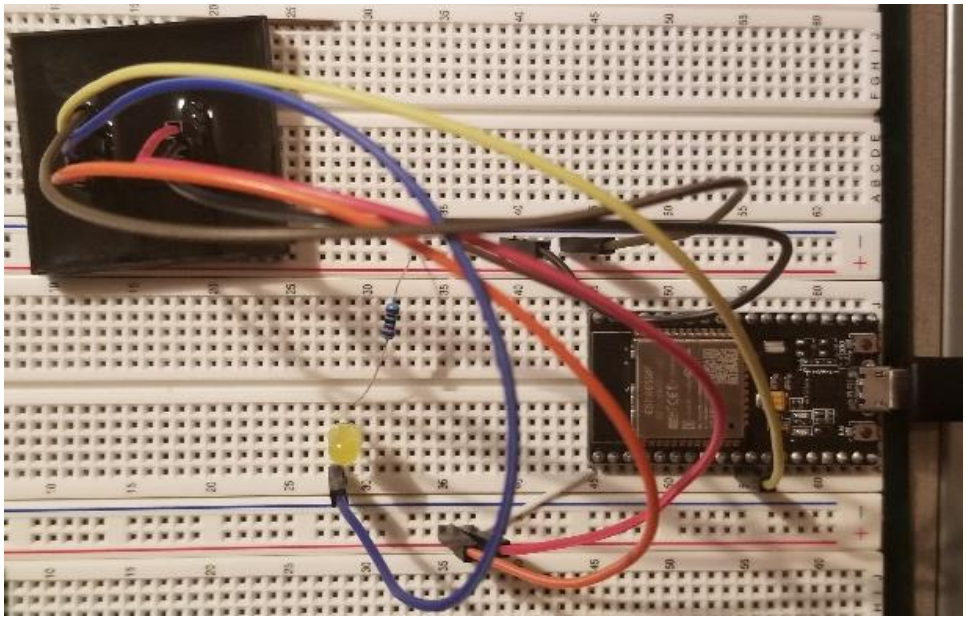


Figure 38: Breadboard Image of RFID module Subsystem

7.2.7 RFID Bracelet

Objective:

This test will be used to confirm that the RFID tag can be read by the RFID module through the bracelet material.

Supplies:

- ID-20LA RFID module
- RFID tag
- Bracelet
- ESP32-WROOM-32D Microcontroller
- Capacitors
- Resistors
- LED
- Bread Board
- Analog Discovery 2 Oscilloscope and Instrumentation System

Procedure:

1. Use the RFID module subsystem used in the previous test.
2. Insert the RFID tag into the bracelet.
3. Turn on the RFID module subsystem using the ESP32 microcontroller.
4. Test if the RFID tag can be read by the RFID module.
5. Measure the max distance the RFID tag can be read.

Result:

The result of this test showed that the RFID tag can be read at an acceptable distance. The test also concluded that the tag can be read through different non-conductive materials.

7.2.8 LED

Objective:

This Test will be used to confirm that the RGB LED works properly and the ESP32 microcontroller activates the correct color for each stage. The connections that will be made are shown below in figure 39.

Supplies:

- RGB LED
- ESP32-WROOM-32D Microcontroller
- Resistors
- Bread Board
- Analog Discovery 2 Oscilloscope and Instrumentation System

Procedure:

1. Connect the ESP32 microcontroller to a computer using a USB connection.
2. Program the ESP32 microcontroller to deliver an output from the IO15 pin.
3. Measure the voltage of the pin.
4. Connect the Red pin, blue pin, and green pin of the RGB LED to the IO15, IO5, and IO4 pins of the ESP32 microcontroller, respectively.
5. Connect a 330 Ω resistor from the ground pin of the RGB LED to ground.
6. Program the ESP32 microcontroller to light the RGB LED using mock stages.
7. Observe if the RGB LED clearly shows the different stages of the ESP32 microcontroller.

Result:

This result from this test showed that the RGB LED can show what stage the SOLAS system is in.

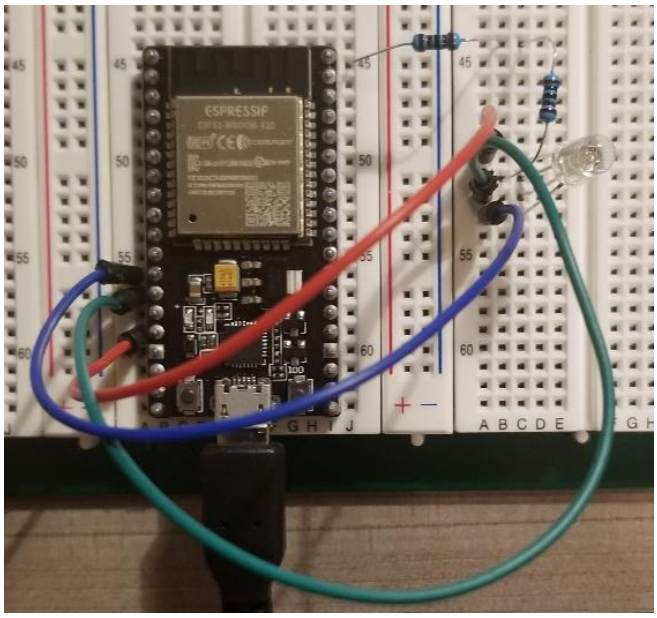


Figure 39: Breadboard Image of LED subsystem

7.3 Software Test Environment

To test the software of each of their functionalities an environment to perform these tests must be established. The environment chosen will ensure validity of the tests and minimize any external factors that may affect test results. The sections below will establish the test environments used for gesture controller testing and the web application testing.

7.3.1 Gesture Controller Test Environment

To properly test the gesture controller, a personal computer and an integrated development environment (IDE) will be used. The ESP32 microcontroller with the gesture controller connected to it will be connected to a PC with a USB cable. With an IDE open on the PC, the IDE will be used to monitor the outputs of the gesture controller.

7.3.2 Web Application Test Environment

Testing for the web application will be done entirely on a PC with internet connection. The database will be available to monitor its contents and verify that certain tests have been successfully performed. Software such as Postman or ARC will also be used to perform API testing.

7.4 Software Specific Testing

The major software features that need to be tested is the web application for SOLAS and the gesture reading on the gesture controller. These tests will ensure that the functionalities for each software feature is working properly. These sections below will describe the objective of the

specific test being performed, the procedures that will be carried out in order during testing, and the results that each test will achieve.

7.4.1 Gesture Tests

The main function of the gesture controller is to read gesture inputs from a hand and read them like a string of characters for a password. First the gesture controller code will be tested to see if the correct gestures are being recognized and then a test will be performed for recognizing a series of gestures as a password. Features that are only accessible through the website, but affect the gesture controller will also be tested, such as changing the password and turning off the gesture passwords.

Hand Direction Reading

Objective:

This test will verify whether the correct hand signals are read accurately by the gesture controller and to see if the gesture controller functional.

Procedure:

1. Connect microcontroller with gesture controller to a Computer via USB connection
2. Open the IDE used for the microcontroller
3. Open the serial terminal on the IDE
4. Click the run button on the IDE
5. Swipe up with your hand in front of the gesture controller
6. Check the serial terminal for an output reading
7. Swipe down with your hand
8. Check the serial terminal for an output reading
9. Swipe left with your hand
10. Check the serial terminal for an output reading
11. Swipe right with your hand
12. Check the serial terminal for an output reading

Results:

With the gesture controller communicating with the microcontroller using the I2C communication protocol the outputs of the gesture controller should be displayed on the IDE serial terminal for debugging. Each swipe of a hand over the gesture controller should generate a specific output on the serial terminal indicating the direction of the hand motion.

Password Recognition

Objective:

This test will verify if the correct gesture-based password is being recognized by the gesture controller and will successfully unlock the door or give positive feedback on a successful string of gestures.

Procedure:

1. Connect microcontroller with gesture controller to a Computer via USB connection
2. Open the IDE used for the microcontroller
3. Hard code a specific gesture password
4. Open the serial terminal on the IDE
5. Click the run button on the IDE
6. Input the gesture pattern hardcoded
7. Check the serial terminal for an output reading

Results:

If the correct gesture pattern is inputted over the gesture controller the terminal should output a string or value indicating that it is a success. The opposite should be true if an incorrect gesture is inputted with a different string or value outputted.

Gesture Password Feature Off

Objective:

This test will verify that with the gesture password feature off entry or unlocking the door will be possible only with the RFID bracelet and the gesture controller will not be used.

Procedure:

1. Open the web application
2. Enter login information and click login
3. Click on the password settings
4. Turn off the gesture password feature
5. With the RFID bracelet on approach the door lock
6. Flash the RFID bracelet over the RFID reader module
7. Verify that the door is unlocked.
8. Turn on the gesture password feature
9. Repeat steps 5-6.
10. Verify that the gesture password is working by entering the correct gestures

Results:

After turning off the gesture password on the website, the RFID bracelet should be the only feature that can grant entry. After testing the gesture password off, the test will be repeated to make sure turning the gesture passwords back on functions. The RFID bracelet should activate the gesture controller and entering a gesture password should unlock the door.

Changing Gesture Password

Objective:

This test will verify if the functionality of changing the gesture passwords through the website is working properly.

Procedure:

1. Open the web application
2. Enter login information and click login
3. Click on password settings
4. Click change password and enter the desired gesture pattern
5. Go to the door lock and enter the new gesture pattern
6. Verify that the door is unlocked

Results:

The new gesture password should overwrite the old password and unlock the door.

Gesture Reading under Ambient Sunlight

Objective:

This test will verify that the gesture controller is able to function properly under direct or indirect sunlight since the sensors will need to be on the outside portion of the door.

Procedure:

1. Place the SOLAS door lock subsystem outside in direct sunlight
2. Enter the gesture password
3. Verify that the door is unlocked

Results:

Depending on the settings of the gesture controller sunlight should not interfere with the IR LED and the four photodiodes. The gesture controller can reject ambient light, this test will verify if any adjustments need to be made to it.

7.4.2 RFID Bracelet Tests

While tests have to be done in order to ensure the hardware components of the RFID reader module and RFID tag work together, other tests have to be done to ensure that the action of reading an RFID tag produces the right results within the software in order to proceed to the next software state.

Reading a valid tag

Objective:

The purpose of this test is to verify that the software can successfully read the correct RFID tag.

Procedure:

1. Enter the RFID tag ID number into the section of the software where the comparison will be made.
2. When the system is in the ready/processing state, hold the RFID tag near the RFID reader module.
3. Ensure that the system recognizes the tag and turns on the gesture controller to start the next state if the password is enabled.

4. Ensure the door unlocks if the gesture password is disabled.

Results:

A reading of the correct RFID tag will either turn on the gesture controller if it is enabled or unlock the door if disabled.

Reading an invalid tag

Objective:

The purpose of this test is to verify that the software will respond accordingly if an RFID tag that is not registered in the system is read by the RFID reader. Since the team will only be purchasing one RFID tag for the demonstration, different numbers and comparisons will have to be hard coded for this test.

Procedure:

1. Enter an RFID tag ID number similar to the one of the given RFID tags into the section of the software where the comparison will be made.
2. When the system is in the ready/processing state, hold the RFID tag near the RFID reader module.
3. Ensure that the system no longer recognizes this RFID tag.
4. Ensure that the system does not turn on the gesture controller or unlock the door.

Results:

A reading of an incorrect RFID tag will not allow the system to proceed to the next state.

7.4.3 LED tests

In addition to testing the RGB LED to verify that it works with given power supply and other hardware, the LED must be tested to ensure that it displays the right color for respective states.

Changing the color of the LED

Objective:

The purpose of this test is to verify that the LED changes colors at the right times to display information to the user.

Procedure:

1. With the system in standby mode, ensure the LED is turned off.
2. Once the system detects motion, verify that the LED has turned on with a blue color.
3. Put the RFID tag near the RFID reader module.
4. Once the system is ready for the gesture password, input an incorrect password.
5. Ensure the LED has turned red.
6. Wait 30 seconds, repeat steps 1-3.
7. Input the correct gesture password.
8. Verify that the LED has turned green.

Results:

The LED properly conveys to the user what state the system is in.

7.4.4 Web Application Tests

The Web application contains only a few features with the main feature of the site is to be able to display the images that are captured from the camera. The minor functions to test are login, registering a user, and password recovery. These features will be tested both with valid tests as well as invalid attempts/tests in order to ensure preventative coding and quality assurance were done correctly.

Registering a valid Account

Objective:

The purpose of this test is to verify that a new user can be successfully created.

Procedure:

5. Enter the SOLAS web address which leads directly to the login page
6. In the login page, click the register button to redirect to the register page.
7. Enter a first name, last name, email, password, security question and answer, and serial code that is associated with the SOLAS device.
8. Click the register button which will redirect to the login page.
9. Open the website database and verify if the new account information has been successfully stored.

Results:

A successful account creation will result in the account information being stored in the database. Successful storage of the account information means logging in with this info should now be possible.

Registering an invalid Account

Objective:

The purpose of this test is to verify that when a user attempts to register with invalid data, the website will notify them to fix any such errors and try again. This is important since if the user is allowed to register with invalid data, it could cause runtime errors later. These tests also cover other edge cases that need to be accounted for.

Procedure:

1. Enter the SOLAS web address which leads directly to the login page
2. In the login page, click the register button to redirect to the register page.
3. Omit any one of first name, last name, email, password, security question and answer, and serial code when registering.
4. Click the register button.
5. Ensure that the register page will not attempt to insert this partial data form and will require all fields to be complete.

6. Go to the current users' database and find a current valid user in the database.
7. Go back to the Register page and enter a first name, last name, email, password, security question and answer, and serial code, but attempt to register with the same email as a user already in the database.
8. Ensure that it is checked that the email entered is unique from all emails of users currently in the database.
9. On the register page, enter a first name, last name, invalid email, password, security question and answer, and serial code.
10. Ensure that the user is not allowed to register with an invalid email.

Results:

All of these tests should provide useful feedback to the user, such as "Please ensure all fields are complete." or "That email is already in use, if you already have an account try logging in or using the forgot password feature.". None of these feedbacks should overlap, such as if a user attempts to login in without a first name and an email already in use, only one feedback is given to the user. Additionally, none of these attempts should affect the data already in the database or add any new user data.

Login with valid credentials

Objective:

To verify that a user can successfully login into the website and retrieve images associated with the account from the database.

Procedure:

1. Enter the SOLAS web address which leads directly to the login page
2. In the login page, enter an email and password that is associated with a current valid account
3. Click the Login button

Results:

A successful login should redirect the user to the camera images page. In this page the user will see images captured from their SOLAS lock.

Login with invalid credentials

Objective:

To verify that a user cannot login into the website and retrieve any images or other information associated with accounts from the database.

Procedure:

1. Enter the SOLAS web address which leads directly to the login page
2. In the login page, enter an email that is associated with a current valid account, but with a random and incorrect password
3. Click the Login button
4. Verify that the user is not redirected from the login page to any other page

5. In the login page, enter an email that is associated with a current valid account, but with a password very similar to the user's correct password. For example, enter "puppies" instead of "Puppies"
6. Click the Login button
7. Verify that the user is not redirected from the login page to any other page
8. In the login page, enter an email that is not associated with a current valid account, and with a random and incorrect password not associated with any account
9. Click the Login button
10. Verify that the user is not redirected from the login page to any other page
11. In the login page, enter an email that is not associated with a current valid account, but with a password associated with a valid account
12. Click the Login button
13. Verify that the user is not redirected from the login page to any other page

Results:

An unsuccessful login should not redirect the user to the camera images page. With any invalid attempts, the website should provide feedback to the user such as "Invalid credentials" or "Invalid email and password combination". The feedback should not specify whether it is only the password that is incorrect or both. With this information, once a malicious user finds a valid email, they can use the email to sign up in various webforms or try different methods to guess the password (such as rainbow or dictionary).

Forgot Password with valid email and security question answer

Objective:

To verify that account access can be restored if the user forgets their password.

Procedure:

1. Enter the SOLAS web address which leads directly to the login page.
2. Click on the Forgot Password button which will redirect to the forgot password page.
3. Enter a valid email from the database and hit submit.
4. Once redirected to the Security question page enter an answer for the security question.
5. Hit submit and check email for a reset password link.
6. Click on the reset password link and enter a new password
7. Click submit

Results:

After going through the password reset process, if the password is successfully reset, login should be possible with the new password. This test will also test the security question feature, the website should be able to display the correct security question for the correct user.

Forgot Password with invalid credentials

Objective:

To verify that account access cannot be restored if the user does not know what email they signed up with or do not know the answer to their security question.

Procedure:

1. Enter the SOLAS web address which leads directly to the login page.
2. Click on the Forgot Password button which will redirect to the forgot password page.
3. Enter a valid email from the database and hit submit.
4. Once redirected to the Security question page enter an answer for the security question that is not the correct answer to that user's Security question.
5. Hit submit and check that no email was sent.
6. Go back to the root SOLAS web address and click on Forgot Password.
7. Enter an invalid email not associated with any registered user.
8. Verify that the user is not redirected to the security question page.

Results:

After going through the password reset process, if the email or security question answer are not entered correctly, the user is not able to gain access to the account.

Logout

Objective:

To verify that when a user clicks the logout button on their dashboard that the session ends completely.

Procedure:

1. Enter the SOLAS web address which leads directly to the login page.
2. Login with a valid account.
3. Click the logout button.
4. Re-enter the SOLAS web address and ensure the user is not automatically taken to the dashboard or any other page other than the login screen.

Results:

When the user clicks the logout button, the website should end the session entirely.

Camera Image Upload with motion detection

Objective:

This test will verify if the camera can successfully upload the images it captures and send it using Wi-Fi to a computer or database and website.

Procedure:

1. Turn on power for the camera
2. Walk in front of the motion sensor to trigger the camera
3. Check the cloud database to see if the image is there
4. Enter the SOLAS website
5. Enter login information associated with the camera
6. Check if the image uploaded from the camera is displayed

Results:

If the camera is successfully triggered by the motion sensor, the image upload is successful if the image is visible on the database and on the camera image web page in the account associated with the camera.

Camera Image Upload with incorrect gesture password

Objective:

This test will verify if the camera can successfully upload the images it captures and send it using Wi-Fi to a computer or database and website.

Procedure:

1. Turn on power for the camera.
2. Walk in front of the motion sensor to trigger the camera
3. Have RFID tag near enough to RFID reader to be read.
4. Enter an incorrect gesture password associated with the tag.
5. Check the cloud database to see if the image is there.
6. Enter the SOLAS website.
7. Enter login information associated with the camera.
8. Check if the image uploaded from the camera is displayed along with a red border or other indicator.

Results:

If an incorrect gesture password is input, the camera should take an additional picture and be viewable from the website.

Delete Image(s)

Objective:

To verify that when a user deletes an image it is deleted completely from the database.

Procedure:

5. Enter the SOLAS web address which leads directly to the login page.
6. Login with a valid account that already has images on the dashboard.
7. Click the delete button for one of the images.
8. Verify the image is no longer in the database.
9. Logout of the website and log back in with the same credentials.
10. Verify that the image is still not on the dashboard.
11. Click the delete all button.
12. Repeat steps 8-10.

Results:

When deleting an image, the image should be deleted from the database immediately. The screen may not refresh right away, so this test also covers when the user logs back in that the image is no longer in their account.

Miscellaneous website functionality

Objective:

To verify that various parts of the website are in sync with each other.

Procedure:

1. Enter a valid login on the login page.
2. Verify that the user is redirected to the main dashboard.
3. Verify that the name displayed on the top of the screen is the one associated with the account.
4. Ensure the buttons display correctly on different screen sizes and browsers.
5. Ensure no images are overlapping each other or the buttons on the screen.
6. Go to edit gesture password screen.
7. Ensure that the website displays all options for gestures to add to the sequence, which should be displayed on the bottom of the screen, refreshing as the password is changed.

Results:

The website’s small features are all in sync and display correctly.

8. Administrative Content

This section discusses the schedule we undertook to achieve a finalized design for this project and the budgetary information. The information listed here is subject to change due to any specific circumstances that may occur.

8.1 Milestones

The milestones set for this project help track progress through Senior design I to Senior design II. These estimated dates will help give an idea on where the project should be and if the group is behind on a task. These dates are not strict and are only guidelines set for the group. Table 19 below shows a rough guideline of the design process for senior design 1 for this project, as well as generally who is responsible for the task.

Table 19: Senior Design 1 Project Milestones

Senior Design I		
Documentation	**Task/milestone Lead**	**Date**
Discuss Project Ideas	Team	8/27/2020 - 9/10/2020
Divide and Conquer Initial Documentation version 1	Team	9/11/2020 - 9/22/2020
Research different sensor technologies	Team	9/22/2020 - 10/2/2020
Research additional features	Team	9/22/2020 - 10/2/2020
Divide and Conquer Initial Documentation version 2	Team	9/27/2020 - 10/2/2020
45-page Draft of Final Documentation	Team	10/26/2020 - 11/13/2020
75-page Draft of Final Documentation	Team	11/13/2020-11/27/2020
90-page Final Documentation	Team	11/27/2020 - 12/8/2020
Research		**10/26/2020 – 11/13/2020**

Microcontroller	Matthew	10/26/2020 – 11/13/2020
Gesture Controller	Keanu	10/26/2020 – 11/13/2020
RFID Technology	Devon	10/26/2020 – 11/13/2020
Camera technology	Matthew + Keanu	10/26/2020 – 11/13/2020
Motion Sensor Technology	Keanu	10/26/2020 – 11/13/2020
Microcontroller coding	Devon	11/13/2020 – 11/27/2020
Website design	Devon	11/13/2020 – 11/27/2020
PCB Design	Matthew	11/13/2020 – 11/27/2020
Purchase Parts	**Team**	**11/13/2020 - 1/11/2021**

Senior design 2 is going to be much stricter with team-set deadlines and milestones to reach. Shown below in tables 20-22 are the individual tasks that need to be completed, as well as who will complete them, and by when. The schedule is also summed up visually in figure 40. As long as the team follows this schedule, we should be able to finish on time with plenty and retesting/debugging/finalization time to utilize. Table 20 covers all the hardware related tasks.

Table 20: Senior Design 2 Hardware Milestones

Hardware		
Task	**Task/milestone Lead**	**Date**
Build Individual Subsystems	Matthew	12/1/2020 – 12/20/2020
Test Individual Subsystems	Matthew	12/21/2020 – 1/10/2021
Build Prototype	Matthew	1/11/2021 – 1/30/2021
Test Prototype	Team	2/1/2021 - 2/20/2021
Debug/fix Prototype	Matthew	2/22/2021 – 3/1/2021

Build Final Design	Matthew	3/2/2021 – 3/20/2021
Test Final Design	Team	3/21/2021 – 4/1/2021
Finalize Project	Team	4/2/2021 – 4/17/2021
Final Presentation	Team	4/18/2021 – 4/30/2021

Table 21 below covers all the tasks that will need to be completed for the website. Since this can be done remotely, it is planned for much of its design to be completed during the break between Senior Design 1 and 2 so that the team can focus on the larger and more important aspects of the project. Once more of the hardware design of the project is complete, then the more complex aspects of the website can be worked on.

Table 21: Senior Design 2 Website Milestones

Website		
Task	**Task/milestone Lead**	**Date**
Set up database	Devon	12/9/2020 – 12/11/2020
Design Mockup	Devon	12/12/2020 – 12/24/2020
Basic User API calls functional	Devon	12/26/2020 - 1/5/2021
Front-End Design/Add aesthetics	Devon	1/6/2021 – 1/10/2021
Deploy to Heroku	Devon	1/11/2021
Connection to camera via serial number	Devon	1/11/2021 – 1/30/2021
Image uploads from motion detection	Devon	1/11/2021 – 1/30/2021
Ability to change gesture controller password	Devon	2/1/2021 - 2/20/2021

Debugging/finalization	Devon	2/21/2021 – 4/30/2021

The main software focus of the project will be the software inside the door lock subsystem. Since this requires a connection to the hardware, it will be done during the Senior Design 2 semester when the team will meet more frequently, shown in Table 22 below.

Table 22: Senior Design 2 Microcontroller Milestones

Microcontroller Software		
Task	**Task/milestone Lead**	**Date**
Connect microcontroller/ camera to WIFI	Devon	1/11/2021
Insert RFID tag into database, test recognition	Devon	1/11/2021 – 1/30/2021
Upload image to website with basic call	Keanu	1/11/2021 – 1/30/2021
Use interrupts from proximity sensor to upload images automatically	Keanu	2/1/2021 - 2/20/2021
Read in gesture controller password	Keanu	2/21/2021 - 3/14/2021
Image upload with red flag	Keanu	3/15/2021 – 3/22/2021
Attach LED colors to specific software states	Keanu	3/23/2021 – 4/1/2021
Debugging/finalization	Keanu	4/2/2021 – 4/30/2021

The schedule for Senior Design 1 is based upon first researching the major components and features of the project to encompass the 45-page draft of the final documentation. Once this is done, the team will move on to the 75-page draft and the final documentation, as well as ordering parts. These next 2 documents will include the basic prototyping and software plan for the project, such as specific software testing environments and hardware connections. If the team is on

schedule, basic testing may be included in the final documentation, such as communication between the RFID reader and tag, being able to activate the automatic deadbolt with the software, and a very basic framework of the website.

Figure 40: Visual Timeline of SOLAS testing and building

The Senior Design 2 schedule is based upon already having a working plan for a large-scale prototype from Senior Design 1, as well as much of the website developed. The first step will be constructing the hardware prototype to see how well all the components work together, and then the testing stage where the software for the prototype is written and tested to see how well the hardware can respond to the software. Although the schedule only reflects 1 prototype build and testing, with the 2nd construction stage the final one, it is expected to go through several major and minor construction and testing stages. The first prototype and test run will give the most feedback as to what changes to make, with a steep learning curve, which is why the schedule reflects 2 major builds; one to experience how well the designed prototype works, and then a final one after the prototype is redesigned, perhaps with some different components.

The two software-based sections also have a long period of time for "debugging" due to the nature of programming. The plan for these two sections is to implement the ability to use the features one by one into the software. As more features are added, the communication and interaction between them gets more complicated, which is why lots of time is scheduled in order to clean up and address any such issues.

8.2 Budget and Financing

To finance this project the cost of all the necessary parts will be split amongst the three group members. A contribution of $100 from each group member was expected to be more than enough to cover the parts necessary for this project. Since this project is not sponsored and is self-funded, it is ideal for it to be as low-cost as possible, but still meet the design requirements. Parts selected are listed below in the budget Table 23, designed to show what a market price for the SOLAS system would be. The team did go over budget on total cost, but the single unit cost of the SOLAS system appears low enough.

As it can be seen from the table, the market price is expected to be very low for this project, just over $120. This table reflects base prices, since with large orders shipping would become negligible. Such charges were covered in section 6.2, the realistic parts acquisition. This table covers the optimal price for a demonstration if the parts were ordered from their cheapest vendors with negligible shipping cost. The price could be slightly lower still, since many of the vendors offer discounts with bulk orders, averaging out at around $120 per SOLAS system, which is a very fair market value in the electronic door lock market.

Table 23: Bill of Materials

Part	Development Cost	Single Unit Cost
Anqueue bracelets	$9.90	$0
OrangeIOT door lock	$40.98	$20.49
ESP32cam + dev board x4	$42.95	$7.15
ESP32-WROOM x2	$21.98	$10.99
RFID tag	$7.50	$0
RFID module x2	$81.27	$34.95
SD card for camera	$14.15	$0

Resistors/capacitors	$15.87	~$1.50
PIR motion sensor x 4	$25.11	$1.56
Gesture controller x4	$36.41	$6.00
PCBs generation 1	$23.40	$0
Voltage regulators	$18.78	$1.65
Pin headers	$7.99	$0.13
PCBs generation 2	$24.30	$24.30
RFID bracelets	$5.89	$5.89
Heroku server	$25.09	$0
Enclosure box	$17.98	$8.99
Totals	**$419.55**	**$123.60**

8.3 Personnel

In this section, the team members involved in the research, designing, and construction are introduced, and their contributions are described in Table 24.

Table 24: Team Member Info and Contribution

Team Member	Email	Contribution
Matthew Guevara	**mattguevara98@knights.ucf.edu**	• Electrical Engineer • Hardware design • Schematics • Motion sensor Research • Power Supply Research • RFID module Research • PCB design
Devon Anselmo	**anselmo.devon@knights.ucf.edu**	• Website Design • Software states of door lock • RFID research • Bracelet design • Related Standards • Budget/finance research
Keanu Zeng	**kzeng1@knights.ucf.edu**	• Gesture controller Research • PWM Research • Camera Research • MCU Research • Existing products and projects • Realistic design constraints • Related Standards

9. Project Summary

The SOLAS system is a combination of hardware and software working in sync in order to deliver a product that offers security and accessibility to the household. This senior design project was motivated by the team's desire to make household access an easier task after a long day. As working students, the team almost always had books and electrical components and laptops and lunchboxes to carry home and deal with at the front door. Electric automatic door locks on doors and newer generation cars inspired the team to design a product combining features of both of these to increase user accessibility and satisfaction with their door lock security system.

The foundation of the SOLAS door lock system is the OrangeIOT automatic door locking system. The team took apart the system and removed all its features and parts except for the basic keyhole and deadbolt/motor system. As these are very standard, they were used as the base to connect the components chosen by the team to, just with the automatic deadbolt connected to, so that the team could focus on the software and hardware feature side of the project, rather than hardware manufacturing. The SOLAS system was mounted on a small demo door with no handle, only showcasing the result of the deadbolt turning to show unlocking. With mechanical design and structure out of the way, the team added the various sensors and features that make the SOLAS system unique, secure, and accessible.

The main security feature of the SOLAS system door lock is its RFID capabilities. Very few of the popular door locks on the market use an RFID system, instead marketing Bluetooth and a mobile app, making the SOLAS system unique and secure. The RFID system used in the lock is not only accurate, but fast, automatic, and is not able to be guessed or easily "spoofed" like regular passwords or pin numbers can be. While it is possible that the radio frequency sent from the RFID tag could be spoofed and resent maliciously at a later time, it would take a lot of setup time and effort, and the "spoofer" would have to be very close since the range of a passive tag is severely limited. If Bluetooth were implemented, almost any mobile device would be able to connect to the door lock system, instantly having limited access to it. Thus, having RFID in the door lock is more secure than using Bluetooth or a simple 4-number pin which could be overheard, guessed, or found if written down anywhere.

If the RFID security is spoofed in some way, then the Broadcom APDS-9960 gesture controller adds further security to the SOLAS system. Using lights reflecting off the user's hand or other objects, it adds an even more unique feature to door lock security, being able to use a gesture-based password to unlock the door, not unlike gestures read by smartphones. This controller is another layer of security that cannot be easily guessed or bypassed. This password can be complex, simple, or disabled, all based upon the user's needs.

The SOLAS system is also made to be accessible, achieved through the bracelet carried by the user. The bracelet is a simple design, a silicone wristband with an embedded RFID tag. It does not require any power source, processing power, or extra features, making the bracelet light and comfortable for the user to wear. It is also waterproof, allowing the user to wear it while running or walking in the rain.

The RFID tag utilized is a very cheap and small passive tag. This kind of tag allows the exclusion of an extra power supply and processor and other unnecessary weight in the bracelet. Without all these extra parts, maintenance of the bracelet is kept to a bare minimum, to a degree that the user should not ever have to worry about maintaining the bracelet. The only downside of using a passive

tag is the decreased range since it does not propagate its own signals, instead essentially using those sent by the reader. The expected range of the RFID system was around 18cm; while the team had hoped for several feet, 3cm should still provide the same result. The goal is for the user to be able to use RFID security without any extra effort; in this case the user wearing the bracelet will have it within 3cm of the system in order to put in the gesture password and open the door. The range of the RFID system could be increased to the desired several feet, but it would require a much larger and much more expensive RFID reader, costing several hundred dollars, which the $35 reader should be able to mimic closely enough with the given circumstances and cost constraints.

To provide more security and accessibility to the user, the SOLAS system also includes a website that the user can login to. An integrated camera and proximity sensor in the door lock monitor the surrounding area and upload pictures of any movement to this website for the user to review. This is achieved with the chosen microcontroller, the Espressif systems ESP32-CAM, which has Wi-Fi capability. In addition to any movement detected, pictures will also be taken whenever an incorrect gesture password is input, and a red flag sent along with the picture to the website for the user to look at any time.

Appendix I. References

1. **https://medium.com/javascript-in-plain-english/how-bcryptjs-works-90ef4cb85bf4**
2. **https://www.digikey.com/en/articles/quickly-design-your-own-low-cost-3d-gesture-controller**
3. **https://www.abr.com/what-is-rfid-how-does-rfid-work/**
4. **https://www.abr.com/passive-rfid-tags-vs-active-rfid-tags/**
5. **https://www.digikey.com/en/products/detail/texas-instruments/RF37S114HTFJB/5798060**
6. **https://www.digikey.com/en/products/detail/stmicroelectronics/M24LR04E-RMC6T-2/4156630**
7. **https://www.digikey.com/en/products/detail/texas-instruments/TRF7963ARHBT/2798686**
8. **https://www.digikey.com/en/products/detail/stmicroelectronics/ST25R3911B-AQFT/5155655**
9. **https://www.seeedstudio.com/blog/2019/04/01/choosing-the-right-motor-for-your-project-dc-vs-stepper-vs-servo-motors/**
10. **https://www.analogictips.com/pulse-width-modulation-pwm/**
11. **https://www.nfctagfactory.com/products/RFID-silicone-wristband-with-card-slot-removable-replace-RFID-chip.html#.X6lbAlB7nic**
12. **https://www.amazon.com/ANQUEUE-Colorful-Silicone-invisible-wristband/dp/B07BFDMGMY/ref=sr_1_1?dchild=1&keywords=rubber%2Bbracelet%2Bwith%2Bpocket%2Banqueue&qid=1604935270&sr=8-1&th=1**
13. **https://mentor.ieee.org/3000-stds/dcn/18/stds-18-0002-00-PUBS-3001-11-2017.pdf**
14. **https://www.lisungroup.com/wp-content/uploads/2020/02/ANSI-C78.377-2015-Standard-Free-Download.pdf**
15. **https://www.imaging.org/site/IST/IST/Standards/Camera_Measurement_Standards.aspx#12233**
16. **https://www.digikey.com/en/articles/microcontrollers-for-safety-critical-applications**
17. **https://standards.ieee.org/standard/2700-2017.html**
18. **https://www.iso.org/standard/39695.html**
19. **https://www.iso.org/standard/53424.html**
20. **https://www.iso.org/standard/73599.html**
21. **https://www.iso.org/standard/74362.html**
22. **https://www.iso.org/standard/69059.html**
23. **https://www.sis.se/api/document/preview/568402/**
24. **https://www.sis.se/api/document/preview/568403/**
25. **https://www.nema.org/docs/default-source/standards-document-library/ansi-c18-3m-part-2-2019-contents-and-scope.pdf?sfvrsn=869c8ce_0**
26. **https://standards.ieee.org/standard/1666-2011.html**
27. **https://www.ti.com/product/ULN2003A**
28. **https://www.arducam.com/arducam-mini-released/**
29. **https://www.arducam.com/docs/spi-cameras-for-arduino/hardware/arducam-chip/**
30. **https://openmv.io/products/openmv-cam-h7**
31. **https://docs.pixycam.com/wiki/doku.php?id=wiki:v1:start**

32. **https://www.olimex.com/Products/IoT/ESP32/ESP32-CAM/#:~:text=ESP32%2DCAM%20is%20a%20low,external%20antenna%20can%20be%20connected.**
33. **https://www.digikey.com/en/products/detail/parallax-inc/28445/6009023**
34. **https://www.digikey.com/en/products/detail/nxp-usa-inc/HTRC11001T-03EE11/892934**
35. **https://cyberpartners.com.au/cyber-partners-magnificent-seven-guide-to-great-password-security-and-high-value-admin-password-security/**
36. **https://www.nfctagfactory.com/products/RFID-silicone-wristband-with-card-slot-removable-replace-RFID-chip.html#.X6lbAlB7nic**
37. **https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/index.html**
38. **https://randomnerdtutorials.com/esp32-pir-motion-sensor-interrupts-timers/**
39. **https://randomnerdtutorials.com/esp32-cam-post-image-photo-server/**
40. **https://www.hackster.io/hieromon-ikasamo/esp8266-esp32-connect-wifi-made-easy-d75f45#toc-step--2---let-s-taste-the-autoconnect-ability-1**
41. **https://www.techcoil.com/blog/how-to-use-an-esp32-development-board-to-read-rfid-tags-from-a-sparkfun-rfid-usb-reader/**
42. **http://www.eecs.ucf.edu/seniordesign/fa2011sp2012/g17/Media.htm**
43. **https://www.digikey.com/en/articles/how-to-drive-multicolor-leds**

Appendix II. Permissions

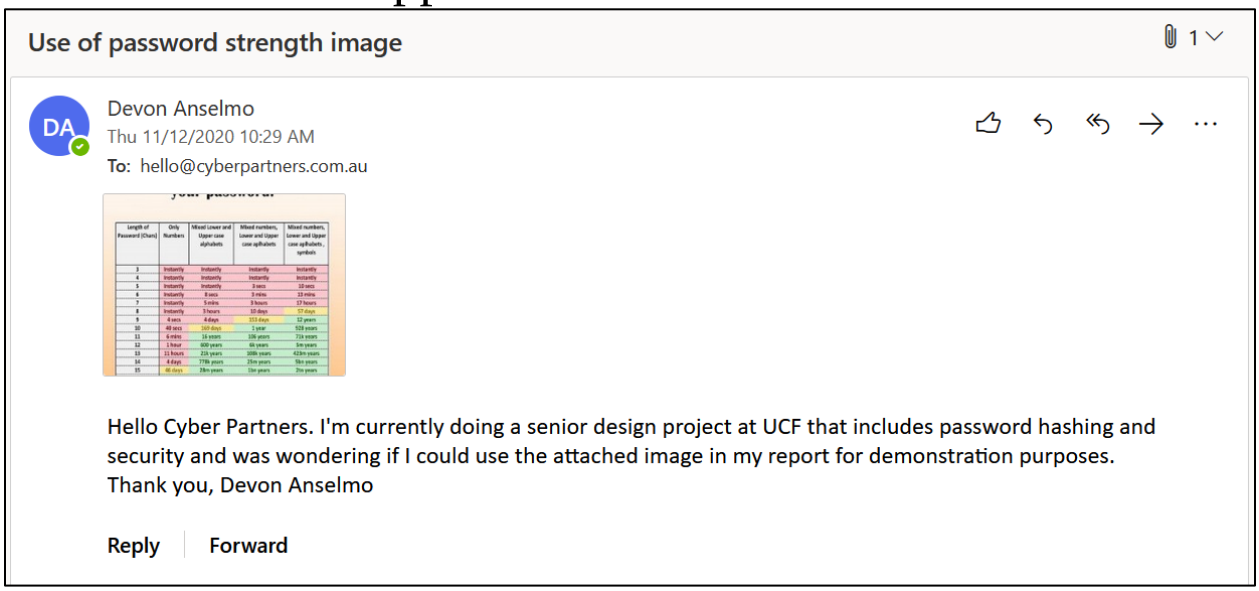


Figure 41: Request of use for password strength image

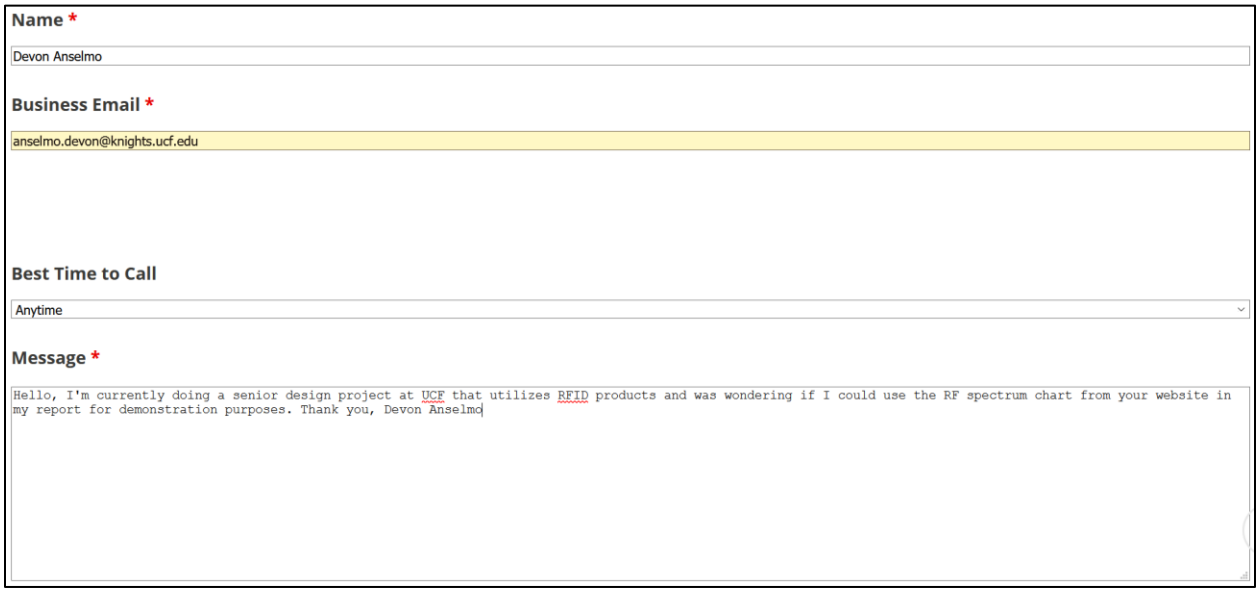
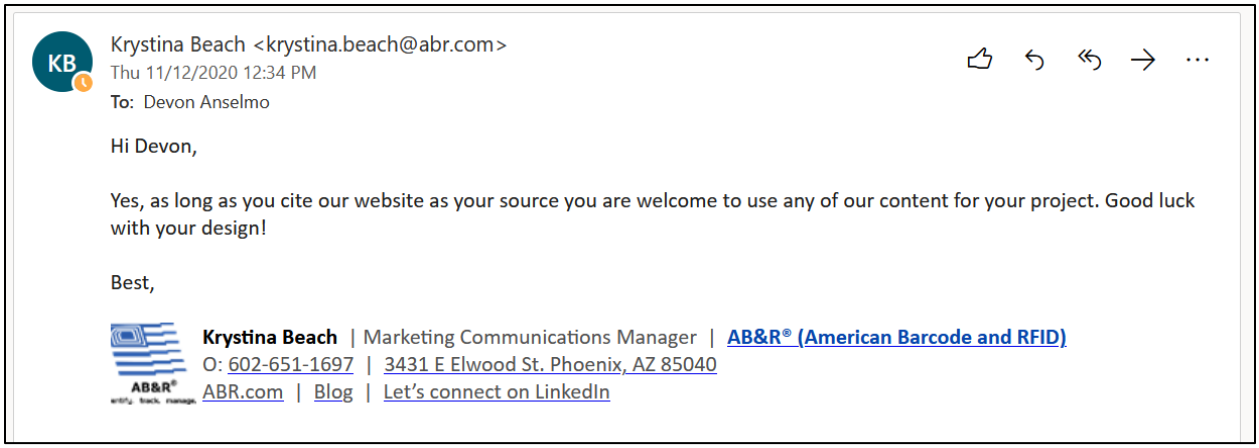


Figure 42: Request of use for RF spectrum chart



Krystina Beach <krystina.beach@abr.com>
Thu 11/12/2020 12:34 PM
To: Devon Anselmo

Hi Devon,

Yes, as long as you cite our website as your source you are welcome to use any of our content for your project. Good luck with your design!

Best,

Krystina Beach | Marketing Communications Manager | **AB&R® (American Barcode and RFID)**
O: 602-651-1697 | 3431 E Elwood St. Phoenix, AZ 85040
ABR.com | Blog | Let's connect on LinkedIn

Figure 43: Permission to use RF spectrum chart

Silicone wristband with insert RFID tag image use

Devon Anselmo
Thu 11/12/2020 10:47 AM
To: info@nfctagfactory.com

Hello, I am currently doing a senior design project that may use one of your products. I was wondering if I could use the attached picture for demonstration in my report.
Thank you, Devon Anselmo

Reply | **Forward**

Figure 44: Request of use for rfid bracelet image

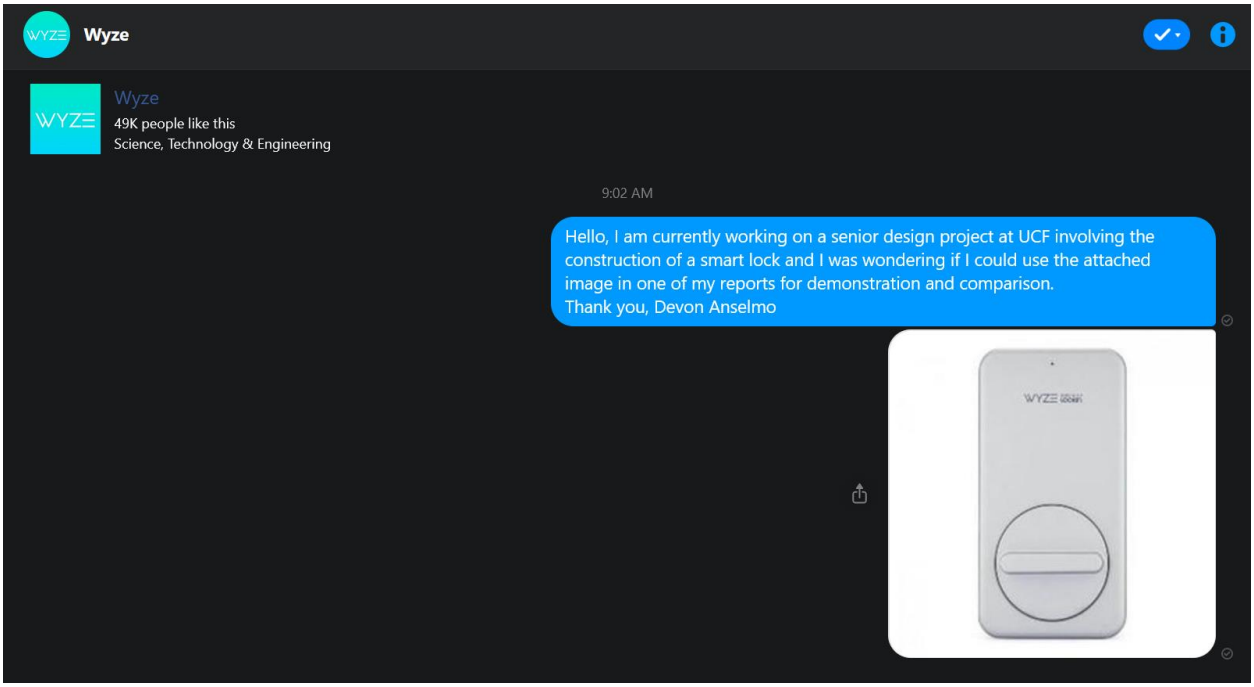


Figure 45: Request of use for wyze lock image

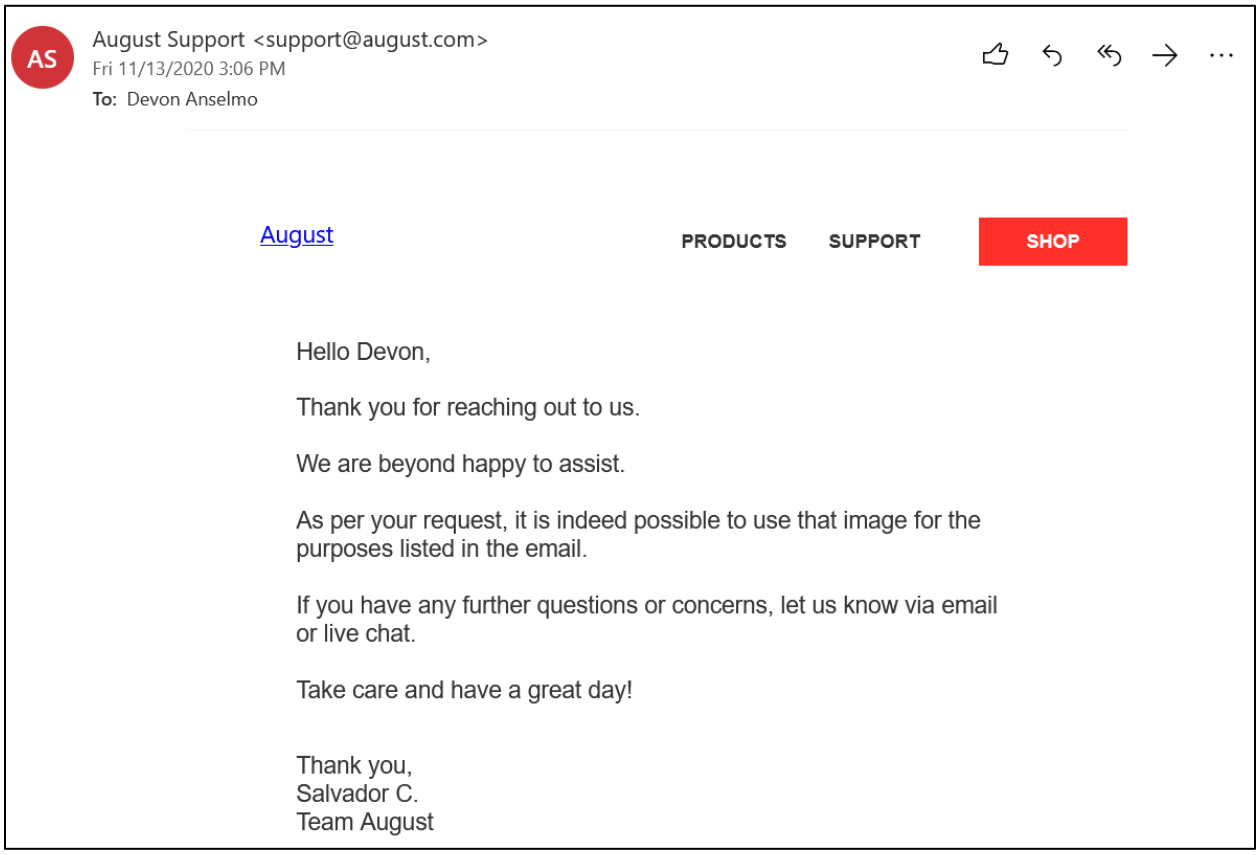


Figure 46: Permission to use august lock image

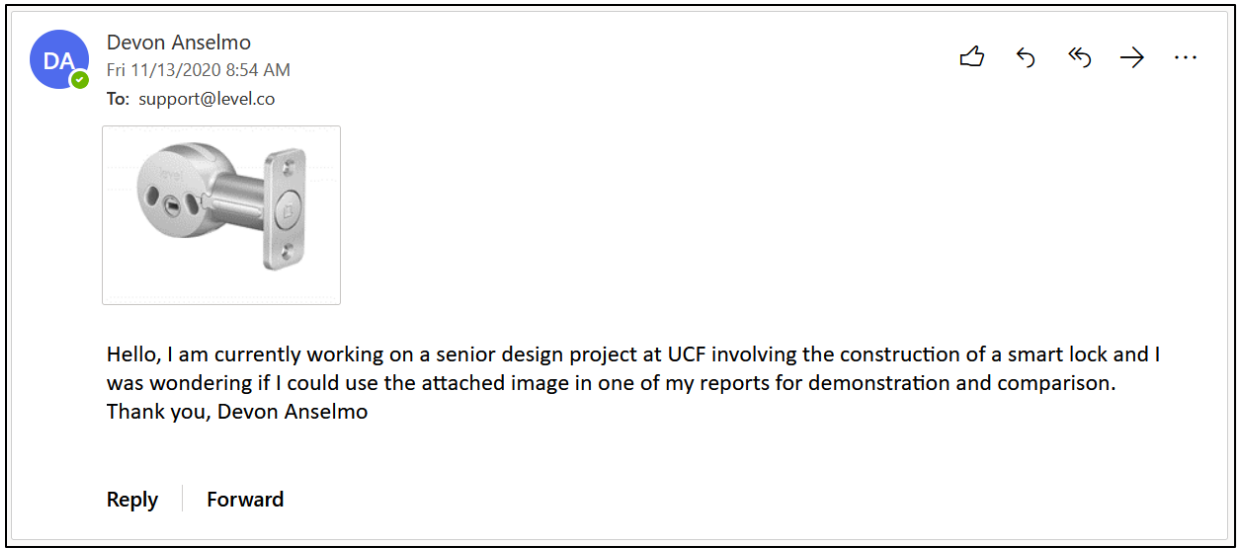


Figure 47: Request to use level bolt image

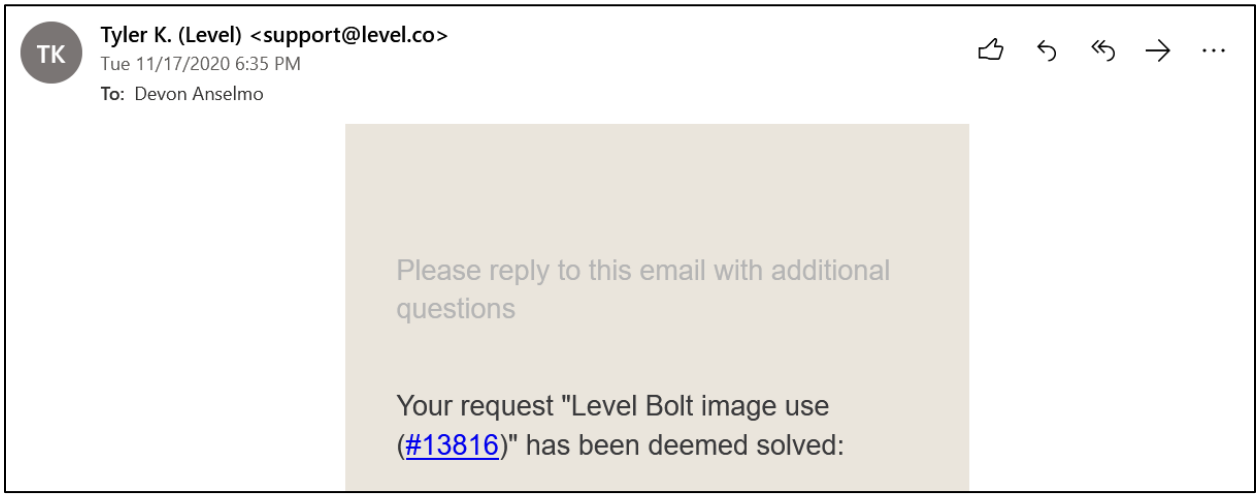


Figure 48: Permission to use level bolt image

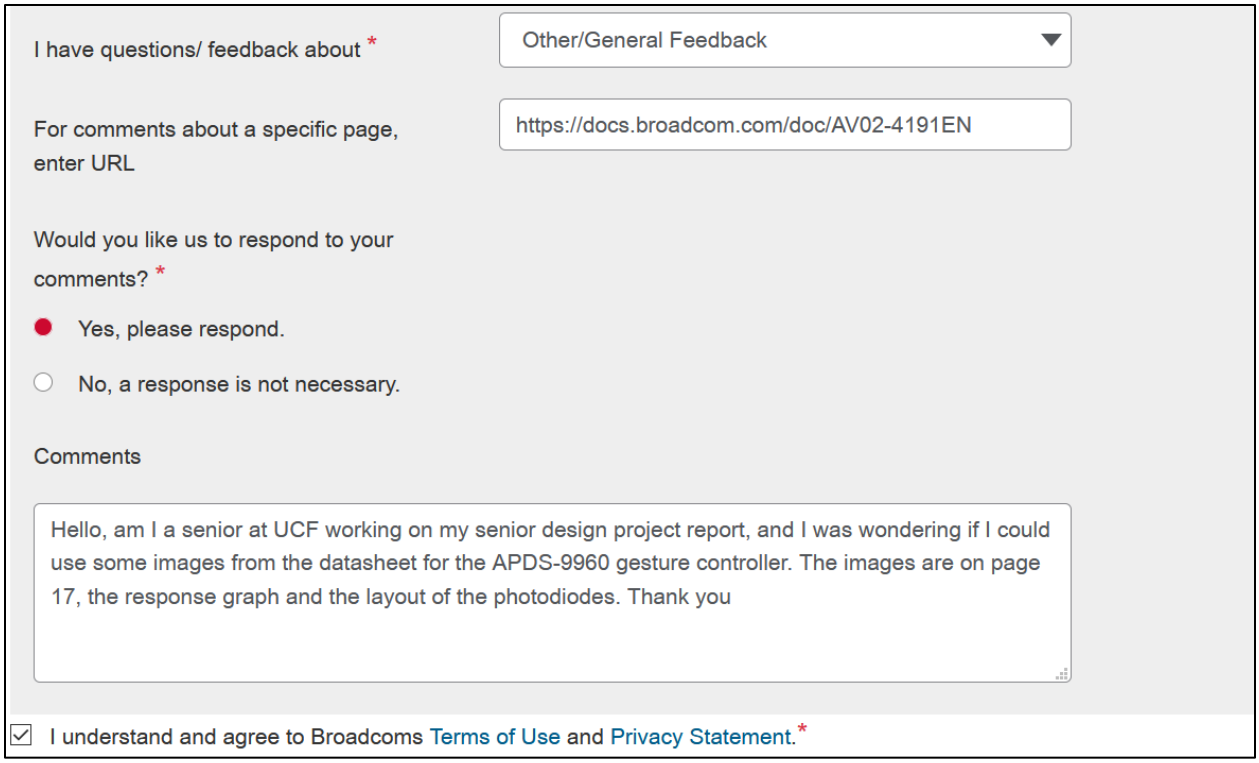


Figure 49: Request to use the 2 gesture controller useage images

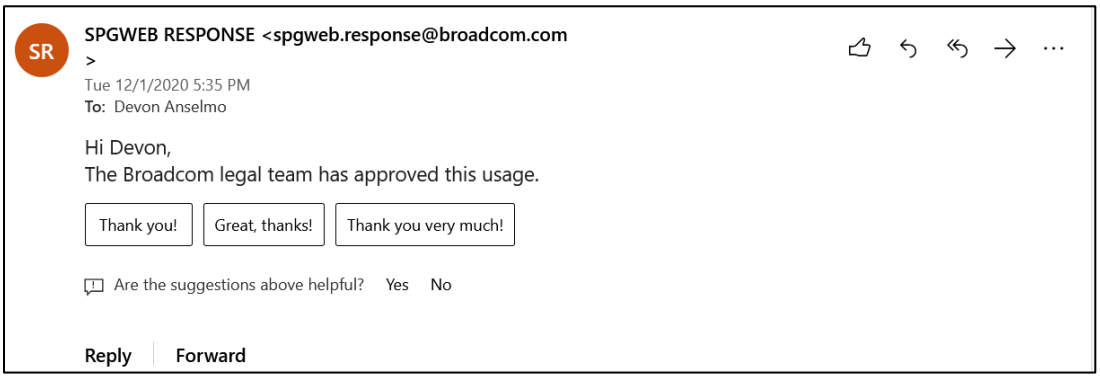


Figure 50: Permission to use the 2 gesture controller images

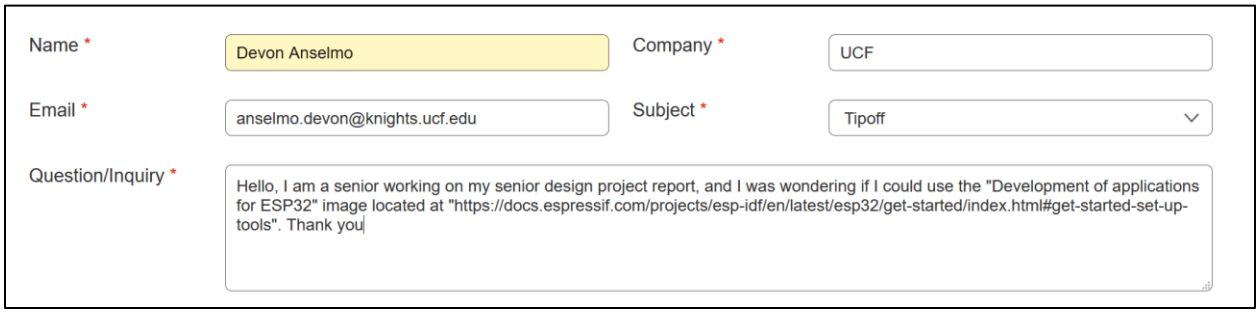


Figure 51: Request to use EspressIf development image

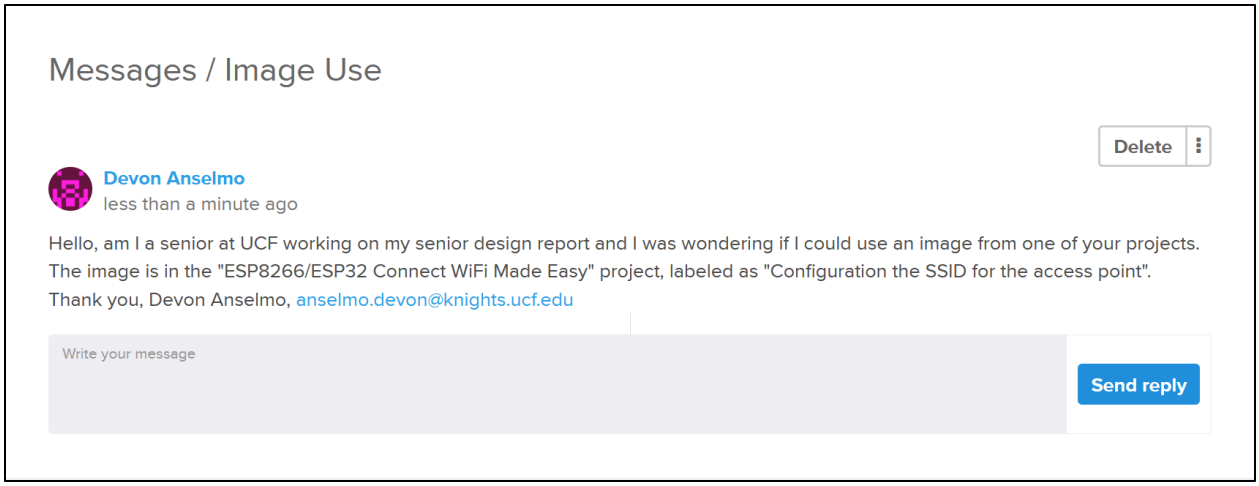


Figure 52: Request to use Autoconnect image