

K.E.E.S.
(Keyless Electronic Entry System)
Senior Design 1

Group 17

Chris Condella

Josh Baxter

Sam Demole

Jason Wagner



UCF DEPARTMENT
OF EECS

Table Contents

1.0 EXECUTIVE SUMMARY.....	1
2.0 PROJECT DESCRIPTION	2
2.1 MOTIVATION	2
2.2 GOALS AND OBJECTIVES.....	3
2.3 PROJECT REQUIREMENTS & SPECIFICATIONS	4
2.4 MILESTONES.....	7
3.0 RESEARCH	9
3.1 SIMILAR PROJECTS	9
3.1.1 <i>The Electronic Door Lock</i>	9
3.1.2 <i>The Better Door Viewer</i>	10
3.1.3 <i>The Smart Home System</i>	11
3.2 RELATED PRODUCTS.....	12
3.3 POWER SUPPLY	14
3.3.1 <i>Battery</i>	15
3.3.2 <i>AC Power Supply</i>	15
3.3.3 <i>DC Power Supply</i>	15
3.3.4 <i>AC/DC Adapter/Converter</i>	15
3.3.5 <i>Linear Voltage Regulator Supply</i>	16
3.3.6 <i>Uninterruptible Power Supply</i>	19
3.3.7 <i>Solar Energy/Power</i>	20
3.4 SENSORS.....	20
3.4.1 <i>Piezo</i>	20
3.4.2 <i>Photo Sensor</i>	22
3.4.3 <i>Motion</i>	23
3.4.4 <i>RGB</i>	26
3.5 RFID RESEARCH	26
3.5.1 <i>Parallax RFID Reader Module</i>	27
3.6 SINGLE-BOARD COMPUTERS/MICROCONTROLLERS.....	30
3.6.1 <i>Raspberry Pi</i>	30
3.6.2 <i>Beagleboard</i>	31
3.6.3 <i>Arduino</i>	31
3.6.4 <i>ARM Microcontroller</i>	35
3.6.5 <i>Sitara SoC</i>	36
3.7 LINUX PORTS	37
3.7.1 <i>Arch Linux ARM</i>	37
3.7.2 <i>Ubuntu ARM</i>	38
3.7.3 <i>Debian ARM</i>	38
3.7.4 <i>Fedora ARM</i>	38
3.8 CAMERAS.....	39
3.8.1 <i>CMU CAMv4</i>	39
3.8.2 <i>Raspberry Pi Camera Module</i>	40
3.8.3 <i>HD Webcam HD 2300</i>	41
3.8.4 <i>Logitech Quickcam Pro 9000</i>	41
3.8.5 <i>Logitech C270</i>	41
3.8.6 <i>Logitech C300</i>	42
3.9 LOCK.....	43
3.9.1 <i>Servo</i>	43

3.9.2	<i>Electric Strike</i>	43
3.10	SOFTWARE RESEARCH	44
3.10.1	<i>Web Server & Database</i>	44
3.10.2	<i>App Development</i>	50
3.10.3	<i>- Wireless Communication</i>	53
3.10.4	<i>Video Image Processing</i>	54
3.10.5	<i>Video Streaming</i>	57
3.10.6	<i>Embedded Communication</i>	58
3.10.7	<i>Voice Recognition & TTS</i>	62
4.0	DESIGN DETAILS	68
4.1	RISK ASSESSMENT	68
4.2	HARDWARE ARCHITECTURE	70
4.2.1	<i>Power Supply</i>	70
4.2.2	<i>RFID Hardware Architecture</i>	77
4.2.3	<i>Sensor Array System</i>	79
4.2.4	<i>Lock</i>	80
4.3	SOFTWARE ARCHITECTURE	82
4.3.1	<i>App Software</i>	82
4.3.2	<i>Web Server & Database</i>	88
4.3.3	<i>RFID Software</i>	90
4.3.4	<i>Piezoelectric Sensor Software</i>	92
4.3.5	<i>Embedded Software</i>	95
4.3.6	<i>Image Processing/Video Streaming</i>	97
5.0	DESIGN SUMMARY OF HARDWARE AND SOFTWARE	100
5.1	HARDWARE SUMMARY	100
5.2	SOFTWARE SUMMARY	101
6.0	PROJECT PROTOTYPE CONSTRUCTION	102
6.1	PARTS ACQUISITION/FINANCIAL BUDGET	102
6.2	PCB VENDORS AND ASSEMBLY	104
6.3	PROTOTYPE CONSTRUCTION AND CONFIGURATION	104
6.3.1	<i>Software Build Plan</i>	104
6.3.2	<i>Hardware Build Plan</i>	105
7.0	PROJECT PROTOTYPE TEST PLAN	105
7.1	TEST ENVIRONMENT	105
7.2	UNIT/FUNCTIONAL TESTING	106
7.2.1	<i>RFID Testing</i>	106
7.2.2	<i>Piezo Testing</i>	108
7.2.3	<i>PIR Unit Testing</i>	110
7.2.4	<i>Strike Test(s)</i>	111
7.2.5	<i>Strike Test</i>	114
7.2.6	<i>Camera/OpenCV Unit Testing</i>	117
7.2.7	<i>Web Server, Web Services, & Database:</i>	119
7.2.8	<i>Video Streaming/Camera Test</i>	122
7.2.9	<i>Light Sensor Test(s)</i>	123
7.2.10	<i>Voice Recognition & TTS:</i>	123
7.3	INTEGRATION TESTING	124
7.3.1	<i>Embedded Integration Testing</i>	124
7.3.2	<i>OpenCV/Camera/PIR/Server/Video Stream Integration Test</i>	126
7.3.3	<i>Web Server, Web Services, & Database:</i>	127

7.3.4	<i>App/Webserver/Camera/OpenCV Integration Test</i>	130
7.3.5	<i>Voice Recognition & TTS:</i>	132
7.4	REGRESSION TESTING	133
8.0	SUMMARY	134
9.0	APPENDIX	136

Table of Figures

Figure 1	Milestones Timeline	7
Figure 2	Milestones Sections.....	7
Figure 3	Voltage Regulator Vout vs. Vin.....	17
Figure 4	Voltage Regulator Iout vs. Vin	18
Figure 5	Piezo Schematic.....	21
Figure 6	Parallax Reader Module	28
Figure 7	ID Innovations Reader Module	28
Figure 8	ID Innovations Reader Pinout.....	29
Figure 9	ID Innovations Schematic.....	30
Figure 10	Arduino Model Size Comparison	33
Figure 11	Sitara Architecture	37
Figure 12	Electric Strike.....	44
Figure 13	Google App Engine Flow	45
Figure 14	HTTP Server Memory Usage comparison.....	46
Figure 15	SpeakJet IC Synthesizer	63
Figure 16	Emic 2 TTS Synthesizer	64
Figure 17	Front and back view of the system mounted on the door	69
Figure 18	Front and back view of system mounted on the wall.....	69
Figure 19	LM 7508 constructed in EagleCad	70
Figure 20	LM 7508 Voltage Regulator tested in Multi-sim.....	71
Figure 21	7805 plotted results Vout vs Vin	72
Figure 22	7805 plotted results Iout vs Iin	72
Figure 23	Pin vs. Pout	73
Figure 24	7805 Efficiency (x100%).....	74
Figure 25	LM22679 Regulator Circuit from Webench	74
Figure 26	LM22679 First Graphical analysis	76
Figure 27	LM22679 Second Graphical analysis	76
Figure 28	Crystal Schematic.....	77
Figure 29	RFID Subsystem Schematic.....	78
Figure 30	Power Supply Regulator w/ Microcontroller and Sensors schematic ..	80
Figure 31	Strike Open Switch State.....	81
Figure 32	Strike Closed Switch State	82
Figure 33	KEES APP GUI	83
Figure 34	KEES App State Diagram: Login.....	85

Figure 35 KEES App State Diagram 1	86
Figure 36 KEES App State Diagram 2.....	86
Figure 37 KEES App Class Diagram	87
Figure 38 MVC Architecture Component Relationships	89
Figure 39 KEES Web Server Architecture	90
Figure 40 RFID Data Encoding	90
Figure 41 RFID State Diagram	91
Figure 42RFID Notification State Diagram	92
Figure 43 Piezoelectric State Diagram	94
Figure 44 Piezoelectric Notification State Diagram	95
Figure 45 Embedded Communication Script Commands.....	97
Figure 46 Image Processing/Video Streaming System Diagram.....	98
Figure 47 Image Processing Class Diagram	100
Figure 48 I2C Embedded Communications Pinout.....	101
Figure 49 Raspberry Pi Software Architecture and Android App.....	102

Table of Tables

Table 1 Embedded System/Sensor Array Requirements	5
Table 2 App/Webserver/Database/Image Processing Requirements	6
Table 3 Overall System Requirements	6
Table 4 Milestones Details.....	8
Table 5 LM7805C Electrical Characteristics Part 1	16
Table 6 LM7805C Electrical Characteristics Part 2	17
Table 7 LM22679 Voltage Regulator Electrical Specifications	19
Table 8 7BB-20-6L0 sensor.....	22
Table 9 PIR Sensor Comparison	24
Table 10 Arduino Chip Specifications.....	33
Table 11 HTTP Server features	47
Table 12 AVR Embedded Server I/O.....	48
Table 13 Python Web Framework features	49
Table 14 Android OS Distribution	51
Table 15 Android Screen Size and Densities Distribution	51
Table 16 Embedded Communications Protocols Part 1	61
Table 17 Embedded Communications Protocols Part 2	62
Table 18 SpeakJet IC Amplifier Specifications	63
Table 19 Voltage Regulator simulated results	71
Table 20 Power In/Power Out values	72
Table 21 Design inputs	74
Table 22 Operating Values	75
Table 23 Project Budget	103

1.0 Executive Summary

The purpose of this project is for senior undergraduate level Electrical and Computer Engineering students to create a design that will give them real world experience into the project design and implementation from its inception until its conclusion, but to do so in a manner that is challenging yet ultimately accomplishable; the Keyless Electronic Entry System completes and exceeds these goals. With the recent advances in technology there has been an increase in automation and additional functionality of everyday devices. There is a growing desire to have control over our devices with a focus on ease of use and accessibility. For our senior design project our team wanted to develop an embedded system that could provide this functionality and control as well as give our team members a challenging and rewarding design experienced. The Keyless Electronic Entry System (KEES) provides an innovative way to gain access through the entrance of a home or office as well as maximizing control and ease of use by.

The KEES project consists of a variety of subsystems that provide functionality of use, while offering the engineering group a challenging real world application of integration and design. One of the main systems implemented in the design of the KEES project is the RFID system. The RFID subsystem allows the user entry through the door by effortlessly swiping a valid identification card. This system will allow the door to unlock as well as be programmable by the user to enter in new valid cards as well as deleting cards from the system. Another element of the KEES project is the piezoelectric subsystem, this system allows the user to gain access through the door by entering a secret knock that is programmed into the embedded system. In addition to being able to gain access from a valid knock, the user of the system will also have the ability to program new knocks into the system. This provides maximum flexibility to the user, giving more options for how they would like to utilize the system. To add even more functionality for the user, the senior design group added the Image Processing subsystem to the KEES project. This system utilizes a camera mounted to the door that is capable of facial detection as well as facial recognition. The system is capable of recognizing whether a guest can access the domicile, or if access is restricted to the person. The user will also be able to utilize an application loaded on their phone that is integrated with the Image Processing subsystem. This will allow the user to know if and when guests arrive at their door as well as giving the ability to deny or allow access to the guest at the touch of their fingertips.

The multiple subsystems of this embedded system working together make up the KEES project. The Keyless Electronic Entry System (KEES) provides an innovative way to gain access through the entrance of a home or office as well as maximizing control and ease of use. This project provides a challenging real world application for the students that created it as well as allowing the engineers involved to express their creativity and ability to work on a project as a team.

2.0 Project Description

2.1 Motivation

Initially the group had many different ideas of potential projects. They decided that a mixture of both hardware and software was best since the primary goal was to have the project interact with many different sensors and have software that provided a cool factor. Each member individually came up with project ideas, and emailed such ideas to the entire group. A list of potential project ideas was created and narrowed down to the most desirable. Many different projects were considered, such as a Sonar based system that could identify fishes in a lake, a luggage carrying robot, an augmented reality helmet, an electronic lock system, a smart cup that could identify liquids, and a fret light guitar that aided the user in pulling the right strings in a song. The project ideas were diverse and were mainly inspired by personal interest.

To come to a decision on which project to choose, a meeting was scheduled in which all the group members rated each project to see which one was the best suited. The metrics that were used to rate each project was cool factor, cost, scalability, and design challenges. The projects that had the highest scores were the electronic lock system and the fret light guitar. The fret light guitar had a very high cool factor, but all of the group members agreed that its scalability was limited and that the amount of hardware that could be added to the project was limited. The electronic lock system was very scalable, practical, and had a cool factor. All members liked the idea of being able to remotely unlock your front door from anywhere. Its scalability was very high: at the meeting each member easily came up with many features in a few minutes such as face recognition, RFID based unlocking, and an app interface with the lock system. The project also satisfied the group's desire of having an app interface with an embedded system. The computer engineers of the group had previously created an Android app, and were very interested in using an app to control a system. Furthermore, the electronic lock system included the integration of many different sensors and hardware, such as a piezoelectric sensor, RFID sensor, a photoelectric sensor, and a camera. Such a system would require the use of microcontrollers as well as its own PCB board, which was desirable as all members were interested in embedded systems.

In addition, network communication was also a technical interest of the group, and communication was a crucial component of the electronic lock system. The system would need a method of communicating between the user's app, which would be sending commands to the system to unlock or lock. While some of the group members had experience in communicating between systems via Ethernet, none of the members had any experience in wireless communication between an app and an embedded system, and we all desired to gain such experience. Furthermore, communication between the various microcontrollers that make up the electronic lock system is also a crucial aspect.

All of these factors led to the conclusion that designing an electronic lock system that could be remotely unlocked from anywhere was the best project idea for senior design. The project was a simple, yet a very practical idea with a large amount of possibilities. All of the group members remembered times when we forgot our key at home and were unable to unlock our front doors. There was a unanimous desire to take something as simple as a lock, and augment it with technology to make life more convenient. In this technological age, one should not be limited to unlocking a door by using such an obsolete device as a key, when smart phones, microcontrollers, sensors, and powerful software development kits (SDKs) are at our fingertips.

2.2 Goals and Objectives

Key based door locks are such an old innovation, and almost all doors use this system. The obvious disadvantage is that if the key is lost, cutting a new key, picking the lock, or breaking the door down are the only options of opening the door. The purpose of the Keyless Electronic Entry System (KEES) is to use various technologies as the method of unlocking the door, as well as having sensors that add other capabilities that either create a cool factor or provide the owner with useful information.

The first main objective of KEES is to use an app interface to remotely unlock and lock the door. The existence of smart phones is almost ubiquitous in this country and they are becoming more affordable. Smart phones are now very powerful devices: many of them have multiple processors and more than a GB of RAM. They are the future of computing, and are obviously a lot easier to carry around than a laptop. As a result, an app interface is very convenient. The commands that the KEES app sends to the system should be reached from anywhere, provided that the user is in an area of network connectivity. This requires the KEES to utilize an efficient and stable method of communication between the app and the system.

The second main objective KEES is to use image processing to recognize the faces of people who show up at the door. Many people desire to know who come at the front door when they are gone. Knowing that your friend "John" came to your front door at 2:38 p.m. enables the owner to have virtual eyes that are always watching the door. This objective has obvious security applications in which a person does not want a particular person to come to their home, and would like to be notified of such an event. The KEES will use a computer vision SDK and a camera to accomplish this task. A database of faces will also have to be stored on the system so that the face of the person who shows up at the door can be compared to faces in the database to determine if the person is a person of interest. The KEES will have to inform the user who is at the door via communicating with the app. This will have to be done within a few seconds

after the person arrives at the door so that the person can get an accurate indication of the time that the person arrived at the door.

The third main objective is to use an RFID scanner as an alternative method to unlock the door. Although many people do have smart phones, there are still a good portion of individuals who don't own one. Using RFID tags to unlock the door is another easy way to gain access. It is a technological step up from using a mechanical key. Multiple RFID tag frequencies can be stored in an RFID reader and the technology is a very reliable method that works.

The fourth main objective is to incorporate voice processing technology. The popularity of the Siri iPhone app has shown how voice commands are a desired and useful feature. People love the ability to speak commands to a system, and to have the system respond to those commands quickly. Voice recognition software can be used in KEES to process voice commands effectively.

The fifth main objective is to have a lock system, such as a servo or electric strike, that responds to the lock and unlock commands sent by the mobile phone app. The group did not want to use a normal door lock as doing so requires no innovation. Designing our own lock enables us to have more control over the overall system, as well as choose a design that is most beneficial to the performance of the KEES.

The sixth main objective is to efficiently utilize many sensors to enhance the capabilities of the KEES. Many sensors are very cheap and can easily be interfaced with a microcontroller. The KEES will include a pyro electric sensor for processing custom door locks that can be used to unlock the door. This is a very interesting application that further personalizes the KEES. The KEES will also include a photoelectric sensor, which will be used to turn on the light when it is dark. Finally, a passive infrared sensor will be used to detect motion, which will be used to notify the system that a person is in the vicinity.

2.3 Project Requirements & Specifications

The following table lists the requirements of the KEES. All of the KEES's requirements pertain to one of the following categories: embedded System/Sensor Array, App/Webserver/Database/Image Processing, and overall system.

Table 1 Embedded System/Sensor Array Requirements

Req. ID	Requirement Description
1	The KEES shall unlock within two seconds whenever the user selects the “unlock” command on the mobile app in a strong Wi-Fi/mobile data environment.
2	The KEES shall lock within two seconds whenever the user selects the “lock” command on the mobile app in a strong Wi-Fi/mobile data environment.
3	The KEES will unlock within three seconds upon hearing specific pre-defined knocking patterns.
4	The KEES will unlock within one second after an RFID tag is scanned.
5	The KEES shall operate off of a 12V standard power supply.
6	The KEES will turn on the door light when it is dark.
7	The KEES piezoelectric sensor shall recognize a custom lock within 3 seconds.
8	The KEES passive infrared sensor shall detect the presence of a person who is within 10 feet from the door.
9	The KEES embedded microcontrollers shall have sufficient processing power to handle all of the input from the various sensors.
10	A green LED on the KEES will flash for three seconds to indicate that the door has been unlocked.
11	A red LED on the KEES will flash for three seconds to indicate that a failed attempt has been made to gain access.

Table 2 App/Webserver/Database/Image Processing Requirements

1	The KEES shall recognize the face of a person who is already in the database within five seconds of the person coming into close proximity.
2	The KEES mobile app shall get information, which includes a picture of a person's face and the time that the doorbell was pressed, within ten seconds after the doorbell is pressed.
3	The KEES mobile app shall allow the user to take a picture of a person's face to add to the KEES database, and specify as a "friend".
4	The KEES mobile app shall display information, which includes a picture of a person's face, the time that the doorbell was pressed, and the person's name if the door is never answered within two minutes after a "friend" rings the doorbell.
5	The KEES mobile app shall enable the user to see what is in front of the door at any time.
6	The KEES mobile app will enable the user to query whether or not the door is locked or unlocked. The response will be received in less than five seconds.
7	The KEES mobile app shall enable the user to specify settings, such as automatically locking after a specified amount of seconds have passed after the door was unlocked.
8	The KEES face database will be able to hold up to 30 images.
9	The KEES camera subsystem must have enough processing power to quickly facilitate image processing.
10	The KEES will be able to record and process voice input.

Table 3 Overall System Requirements

1	The KEES shall be highly reliable and in the event the system goes down or loses connection the owner must be notified via the KEES mobile app.
2	The KEES will require an internet connection via Wi-Fi to communicate with the KEES mobile app.

2.4 Milestones

To be able to complete the KEES project in a timely manner, milestones must be created and loosely abided by to ensure the group stays on track. Such a chart will give approximations as to how long specific parts of the project will take to complete. The research phase should be emphasized and considered to be the most important part of the project. The design phase and all proceeding steps following will be based on the research conducted.

Milestones may need to be adjusted in the future based on time constraints within the group. Dependencies must be considered and certain items should be indicated as high priority if further progression of the project cannot continue without first completing the specified part. Some items may not be deemed to be as important and should be noted as low priority. Other items may even be noted as only to be considered if time permits. All kinds of considerations will need to be made going forward in the project and by setting milestones, a path is laid out for the group so they do not fall behind. See the figures below for the milestone timeline / percentages and the table below for a further description on the details for each section.

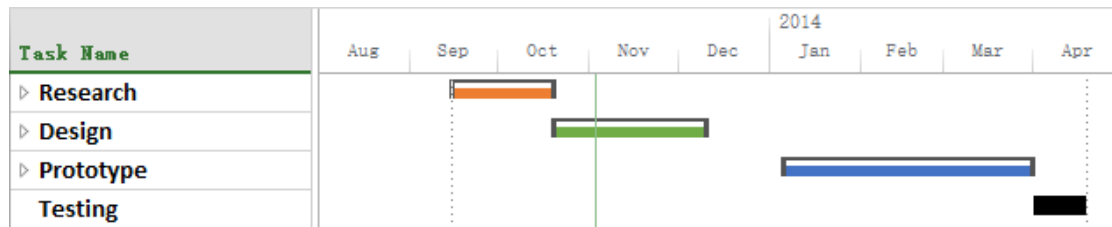


Figure 1 Milestones Timeline

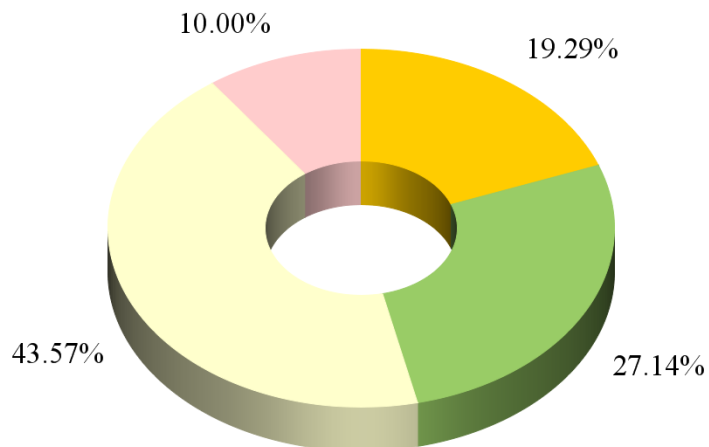


Figure 2 Milestones Sections

Table 4 Milestones Details

Task Name	Duration	Start	Finish
Research	27 days	Sun 9/15/13	Sat 10/19/13
Hardware	27 days	Sun 9/15/13	Sat 10/19/13
Arduino			Sat 10/19/13
Raspberry Pi			Sat 10/19/13
RFID			Sat 10/19/13
Sensors			Sat 10/19/13
Software	27 days	Sun 9/15/13	Sat 10/19/13
OpenCV SDK			Sat 10/19/13
Web server / Database			Sat 10/19/13
Design	38 days	Sun 10/20/13	Tue 12/10/13
Hardware	38 days	Sun 10/20/13	Tue 12/10/13
Lock System			Tue 12/10/13
Camera Subsystem			Tue 12/10/13
Software	38 days	Sun 10/20/13	Tue 12/10/13
Web server / Database			Tue 12/10/13
Image Processing			Tue 12/10/13
Mobile Application			Tue 12/10/13
Prototype	61 days	Mon 1/6/14	Mon 3/31/14
Hardware	61 days	Mon 1/6/14	Mon 3/31/14
Lock System			Mon 3/31/14
Camera Subsystem			Mon 3/31/14
Software	61 days	Mon 1/6/14	Mon 3/31/14
Web server / Database			Mon 3/31/14
Image Processing			Mon 3/31/14
Mobile Application			Mon 3/31/14
Testing	14 days	Tue 4/1/14	Fri 4/18/14

3.0 Research

3.1 Similar Projects

For the project research phase the group decided to investigate similar senior design products that used door unlocking systems so the team could learn from other groups design implementations as well as lessons learned. With this knowledge the team may be able to mitigate errors and lost time by understanding other projects strengths and weaknesses. In this section the team selected a few projects from a vast amount of groups to analyze and look at closer. Some of the additional advantages of studying other senior design projects that are similar to the Keyless Electronic Lock System are that the ways in which the teams tackled difficult problems can be used in the development strategies of the project. Seeing the process that a team takes to tackle a large problem, such as breaking the problem down into smaller more logical pieces, can aid the group and ensure that the prototype process is streamlined. This will be a great value going forward and will help keep the project into perspective as the group enters the actual design stage.

3.1.1 The Electronic Door Lock

The first project that was reviewed is The Electronic Door Lock designed by Joseph Petrovic, and Robert Sanford and the ECE University of Illinois. This locking system was designed to take in a 4-digit password from the user at a keypad on a 16-button standard keypad that is mounted on the door and allow the user access once the correct password is entered. Also once the correct password is entered the user has the ability to change the password for future entry through the door. To implement the logic for their keypad they used a motorola 68HC12 microcontroller to control all of the inputs and outputs of the system. The group incorporated a 555 Timer to control the time the door was unlocked so they could didn't have to take away processing time from their microcontroller.

The 16-Button keypad is a possible feature that may be integrated into the Keyless Electronic Entry System. One of the considerations of using the a keypad for the team was how does the system take in the inputs for the keys and check that the password is correct, as well as how will it be possible for the password to be changed for the implementation. The Electronic Door Lock project was able to capture the data that is entered into the keypad and use the keyboard encoder to decode the message. Once the message is decoded the keyboard encoder hand this to the microcontroller which controls whether the door lock will be initiated. The functionality of this project is similar to the KEES project in that both use a device that communicates with a micro controller to open a locking mechanism. The group will take this project into consideration

when implementing communications to unlock the door, as well as the timing that the door will be opened for.

Another one of the functions that the Electronic Door Lock utilized is a timer to interface with the locking mechanism. The interface with the locking mechanism allowed the micro controller to send a command to the lock for a specified amount of time. This will be important for the implementation of the KEES project. The group will have to come up with a timing scheme in which the door will be unlocked and stay unlocked. The logical amount of time that is allowed to pass that the door is unlocked should give enough time for the user to enter the residence without feeling rushed, but not so much time that someone can come into the residence behind the owner without their knowledge. The allotted amount of time that the Electronic Door Lock team chose was a 20 second interval where the door is in the unlocked state. The KEES team will have to make the determination on whether 20 seconds is needed or if a lesser time will make sense for the groups implementation.

3.1.2 The Better Door Viewer

The next project that the group took into consideration is The Better Door Viewer this project was designed by Roger Cotrina, Vike Francis, Chao-Hung Sun, and Nicholas Zynko from Florida Institute of Technology. The goal of their project was to create door monitoring technology with facial recognition, object tracking, that utilizes embedded technologies. The Better Door Viewer system uses a camera, beagleboard, automatic door lock, and a mobile application to provide facial recognition and tracking.

The Better Door Viewer project used the OpenCV library to implement their facial recognition algorithms. The group had the option to use 2D or 3D face recognition algorithms using the OpenCV platform. They opted to use 2D face recognition algorithms for their project due to the vast amount of support for it, and because 3D facial recognition algorithms are still actively being researched, and could pose a problem in implementation. The tradeoff for using the simpler facial algorithms is that the 2D recognition is limited in its accuracy. This also allowed the group to select a single lens camera, instead of a dual lens camera needed for 3D recognition.

The group that developed The Better Door Viewer started their implementation of the facial recognition algorithm one step at a time. From their documentation it can be seen that their plan was to first start by tracking simple objects such as balls and move onto color detection. The first steps taken to implement the facial recognition algorithms were first implemented on the laptop utilizing the embedded camera. From there they could take out a lot of the complexity of the embedded system and purely focus on the algorithms that they had to implement for the face detection as well as the facial tracking. The initial focus for the face detection and face recognition was to implement Eigenfaces as well as Fisher

faces. Once they completed this task they were able to move onto the more complicated step of pushing these algorithms onto the embedded system. Much can be learned from their process of face detection and face recognition by using simple implementations and algorithms and then stepping the process out with more complexity. As far as the OpenCV algorithms are concerned the KEES team will consider utilizing the process that the Better Door Viewer team used in their implementation of face detection and face recognition.

The project allows the user to interface with the devices using their cell phone. They utilize an Android app to allow the user to remotely unlock the door with a button push. As well as being able to remotely unlock the door they developed the system in such a way that alerts, images, and video can be streamed to the phone from the door camera. This is one of the possible functions that the Keyless Electronic Entry System may implement. The Android app is able to interface with the BeagleBoard xM through an internet connection. This is very similar to the KEES implementation because both aim to open a door from an application on a phone from wherever the user is at that moment in time.

3.1.3 The Smart Home System

The Smart Home System is a system designed by students at Georgia Tech. The project was implemented by David Myers, Phillip Robinson, Jared Santinelli, and Nazar Trilisky. Their vision for the project was to have a RFID accessed smart home with automated lighting. The design is meant for a multiple room setup to provide access for places such as an office building, and to supply controlled lighting for the dwelling as well. The Keyless Electronic Entry System implements an RFID access function similar to this project so the team thought that it may be beneficial to research this project to find out how they completed this project, as well as to see if any interesting functionality from this project could be added to the KEES system. Another function that is added into their design is motion detection so the lighting system within the dwelling can be automatically switched on once the occupant moves from room to room. Within the design specifications for the KEES project, the team will be using a motion detection sensor to detect whether or not there is someone at the door. Once someone is detected in front of the door then the camera will turn on, this serves as a way to control power of the system by not having the camera on and processing the video feeds when there is not anyone at the door at the moment.

For the design of the Smart Home System the team decided to use a Cypress SRN060 Microcontroller for the logic and functionality of the system. The Cypress microcontroller drives the logic to open or close the door lock which is a Enforcer Strike door lock. Using a transistor the Cypress is able to switch the door strike to the unlocked position and unlock the door with 12V of power. The Cypress Microcontroller also drives the automation for the hallway lights. The hallway lights can be turned on or off by the microcontroller if the Motion detector is tripped by someone walking into the room. Once the motion detector is tripped

the lights will be turned on for an allotted amount of time which is controlled by the 555 timer. If there are no occupants in the room, or the motion sensor isn't tripped within the allotted amount of time than the hallway lights will be signaled to turn off until there is motion detected again.

The Cypress Microcontroller is connected to an eBox 2300 JumpStart. The communication that is initiated between these devices is RS-232. The eBox is a mini computer that is used to process the information from the RFID reader. Once the RFID reader is activated by a correct RFID device the RFID code for the ID information is sent down a USB line to be processed inside of the eBox System. Once it is recognized that the RFID code is valid then the eBox sends this information to the Cypress Microcontroller which unlocks the door strike.

This system is very similar to the Keyless Electronic door strike. Both systems integrate the RFID reader to restrict and allow entry through the door. The Door lock is controlled by microcontroller as well in both of the systems. The Smart Home system utilizes a motion detection sensor to drive the hallway as well as the room lighting. The systems differ on the use of the motion detection sensor, in that the motion detection sensor for the KEES project is primarily used for the camera control and whether or not to turn the camera on and process images if there is someone that trips the motion detection. Despite of the different implementations of the motion detector, they are still controlled and integrated in much the same way. Since this project is so similar to the Keyless Electronic Entry System, the documentation and block diagrams for the Smart Home system will be useful during the design stage for the system.

3.2 Related Products

One of the first key developments of this idea and design was to find other related products coming soon or already out in the market. Researching similar products and potential competition is key for any product development and strategically advantageous for any start up or existing business, idea or project. The products and projects found were seen as good points of reference, general design scheme sources, development ideas and guides to start from. The intent is to gather information and create new, different and extended ideas from each product and project that is found. Then the most effective and liked ideas will be pulled together to be incorporated into the design for a senior design worthy project.

One of the first products discovered was from a previous UCF electrical engineering graduate, Phil Dumas. For his senior design project he developed a system that could open a car door simply using a cell phone. "You called your car, entered in your password and the doors unlocked." Upon further research it was discovered that Phil launched a company called Unikey Technologies. The company produces smart security technology for a residence. The "Kevo" which essentially is a smart locking deadbolt device can be managed from a mobile

application. The app allows a user to set up the lock as well as send, disable or delete e-keys and manage “fobs”, which are simply wireless keys. Also the app has a log of the lock activity. The product uses Bluetooth 4.0, has intelligent detection technology, is pick resistant, has security validation, uses four AA batteries and offers simple installation. All of these factors make for a well-rounded product and spark some inspiration.

Another product that caught some attention was AT&T’s “Digital Life-Home Security and Automation” product. The “Home Security” is comprised of two different packages, simple security and smart security. The “Simple security” offers remote security, which includes door and window sensors, a wireless keypad, indoor siren and a keychain remote. The “Smart security” offers monitoring, accessing and managing a home’s security system from a computer, smartphone and/or tablet. The system allows one to receive texts and video alerts of events from the home and allows one to turn on or off the alarm as well as be able to monitor doors and windows and tripped sensors. The “Home Automations” are add-ons to the smart security package. These add-ons allow one to remotely manage the entire house i.e. the security, the thermostat, doors, lights, and small electronic devices and other appliances. The camera add-on package allows one to monitor the home at all times. The door package allows one to lock and unlock doors remotely. The energy package allows one to remotely adjust temperature, lights and other appliances remotely as well as program them to turn on or off at certain times. The water detection package allows the detection of any leaks, which can prevent major water damage. The water control package allows the homeowner to detect water leaks and then if needed turn on and off the main water source remotely. AT&T seems to offer a wide selection of home monitoring and security technologies that promote competition in the smart home security industry.

Lockitron is another keyless entry system using an application from a smartphone. However a smartphone is not necessarily needed the lock system can be locked or unlocked using text message commands enabling the use of virtually any phone from anywhere. This product is not a door lock but an electronic system that fits over any deadbolt frame from the inside of a home or apartment etc. The system can be set to sense when someone is walking up to the door and automatically unlock. It can be made completely customizable too. The system runs off batteries and contains intelligent power management with alerts and has built in wifi.

Kwikset has a line of electronic door locks out on the market as well. These locks contain a keypad on them so all that’s needed to unlock the door is a sequence of numbers. They offer a keyless entry deadbolt lock and an electronic lever lock. With the lock also is offered “Home Connect” technology that allows the door lock to wirelessly talk to other systems in the house, such as the security system, lighting, thermostat, and entertainment system.

Not only were products found but there were a few inspirational projects found floating around as well. From “labs.laan.com” an Arduino based electronic door lock/unlock project was discovered. This project uses an Arduino, a WiFly shield, a servomotor, a piezo element, some software and a few household items to create an e-lock. With the system one can lock/unlock a door with a voice using Siri, an iPhone application, an SMS message or a mobile webpage. One can monitor whether the lock is locked or not. One can receive SMS messages when someone knocks on the door and also the door lock can be unlocked when a secret knock is applied. The interesting thing is that the servo-motor does not directly drive the physical deadbolt lock itself, the servo is used to drive two makeshift arm levers. One lever is attached to the deadbolt and the other attached to the servo, they are linked together and therefore when the servomotor is triggered depending on which position it is set in it will move its own arm which moves the arm on the deadbolt either locking or unlocking the door. This project also uses a piezo element to detect a preprogrammed knocking sequence.

Another similar project was found from “grathio.com”. This project is called the “Secret Knock Detecting Door Lock” and is an Arduino based project. A servomotor is used in this project as well and is attached to the deadbolt via a makeshift clamp. The interesting part of this project is that not only does it require a secret knock to allow the servo-motor to unlock the door but it also has a program button in which a new secret knock can be programmed directly into the Arduino at any time therefore one can change the knock sequence in case the previous secret knock was heard or it’s just time to change the sequence. Any other knock besides the secret knock will result in the door staying locked. Another cool thing is that the knock is programmed to rely on the absolute timing of the knocks in the chain of the secret knock. This means that the tempo of the knock must be correct however it can vary with speed allowing the secret knock be performed slower or quicker relative to somebody else’s knocks, meaning that ultimately knowing the sequence itself is important.

A project similar to the previous ones but this one involving a RFID tag system and an Arduino was found as well, it is called the “Arduino RFID Lock”. The Arduino RFID lock system is a secure lock driven by a servomotor however this lock can easily be opened when presented with a valid RFID card. The lock works with a server database that can register card codes from different RFID tags. Attached to the door is an RFID reader that is connected to the lock. If the RFID reader reads a registered valid card then the door will unlock.

This goes to show that there are many other similar products out there already. The main objective is to create a better, more efficient, easy to use low cost design that works.

3.3 Power Supply

A power supply is simply a device that supplies electric power to an electrical load and or a machine. A regulated power supply will control the output voltage

or current to a specific value, this maintained voltage will ensure that any electrical load or machine connected will receive a constant steady voltage and will be damaged.

The integrated design for the project will require a constant supply of power. There are several different factors such as cost, performance, efficiency and size that will be taken into consideration when choosing and/or designing a power supply that will suit the needs of the design.

3.3.1 Battery

A battery is a device that converts stored chemical energy to electrical energy. A battery is probably one of the most simplest and effective power supplies. Batteries are used everywhere and for many electrical applications as energy sources and come in many sizes, from very tiny used for watches for small devices to large battery banks the size of a room or bigger. There are generally two types of batteries disposable batteries (primary) and rechargeable batteries (secondary). Batteries provide DC voltage.

3.3.2 AC Power Supply

An AC power supply will just take power from the wall outlet and will step down/step up the voltage to a desired value or invert the voltage and pass it through a filter if needed depending on the specific application. Many household items that plug into the wall use stepped down AC power, i.e. a cell phone charger.

3.3.3 DC Power Supply

If DC signal is desired for a particular electronic device than a DC signal will be needed from the AC wall power outlet. Diodes are used in a rectifier circuit (half wave or full wave) to convert alternating voltage to a pulsating rippled direct voltage. Usually a filter is attached to the rectifying circuit, containing capacitors and resistors to filter out most of the pulsation and thereby smoothing the signal and producing a generally approximate flat DC voltage signal (Figure 1). A computer power supply is a switch-mode power supply that converts AC power to several different DC voltages generally. This can be used for applications that require different DC voltages.

3.3.4 AC/DC Adapter/Converter

An AC adapter can be used in series with a voltage regulator to power the design providing 12vDC before the regulator and a desired regulated DC voltage from the output of the regulator. Some AC/DC adapters are linear power supplies. They normally contain a small step-down transformer, a full wave bridge and a filter to smooth the AC waveform to DC. The output voltage of these adapters

usually varies with load. Most adapters will contain a linear voltage regulator for a stable output voltage.

3.3.5 Linear Voltage Regulator Supply

Constant and specific voltages will be needed for this project to maintain a stable system. Certain components of the project will need certain voltages to function properly i.e. The electric door strike will need approximately 12vDC while the microcontroller will need approximately 5vDC to operate. Voltage fluctuations can damage the system and the sensors on the system. A voltage regulator generates a fixed output voltage of a preset magnitude that remains constant regardless of changes to its input voltage or load conditions. Voltage regulators often use a regulator chip such as the LM78XX. The load regulation parameter gives the change in V_{out} as the regulator's output current is varied from a minimum to a maximum current. Also the regulator parameter gives minimum allowable voltage across the regulator for the regulator to maintain a constant output voltage. The maximum ratings that must be in consideration, such as the maximum input voltage, the maximum allowable load current, the maximum power dissipation allowed and the maximum operating temperature. Any of these items can potentially destroy the regulator. The wrong polarity voltage applied to the input, having V_{out} be greater than V_{in} , which can occur if the energy storage in the output is greater than the energy storage at the input.

Linear voltage regulators power dissipation is directly proportional to its output current for a given input and output voltage, so typical efficiencies can be around 50% or even lower, therefore at low levels of power, linear regulators are cheaper and occupy less printed circuit board space.

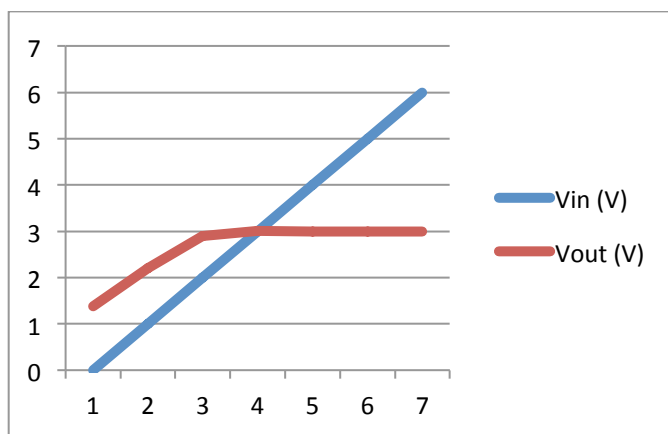
Table 5 LM7805C Electrical Characteristics Part 1

	Parameter	Conditions	Min	Type	Max
V_{in} (V)	Voltage In		4.5	5	43
V_{FB} (V)	Feedback Voltage	$V_{IN} = 8V$ to $42V$	4.925/ 4.9	5	5.075/5.1
I_Q (mA)	Quiescent Current	$V_{FB} = 5V$		3.4	6
V_{ADJ} (V)	Current Limit Adjust V		0.65	0.8	0.9
I_{CL} (A)	Current Limit		6.0/5. 75	7.1	8.4/8.75
I_L (μA)	Output Leakage Current	$V_{IN}=42V, S_{SPin}=0V,$ $V_{SW} = 0V$		32	60
		$V_{SW} = -1V$		31	75

Table 6 LM7805C Electrical Characteristics Part 2

RDS(ON) Ω	Switch On-Resistance			0.1	0.14/0.2
fO (kHz)	Oscillator Frequency		400	500	600
TOFFMIN (ns)	Minimum Off-time		100	200	300
TONMIN (ns)	Minimum On-time			100	
IBIAS (nA)	Feedback Bias Current	VFB = 1.3V (ADJ Version Only)		230	
ISS (μ A)	Soft-start Current		30	50	70

The figures below show a three volt dropout voltage (constant voltage) for voltage regulator tested in laboratory experiment. However the dropout voltage for any voltage regulator can be designed to have a dropout at any voltage desired. The LM7805C characteristics from Fairchild Semiconductor are below

**Figure 3 Voltage Regulator Vout vs. Vin**

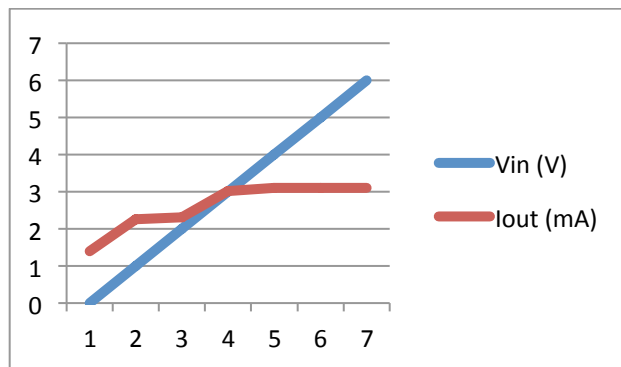


Figure 4 Voltage Regulator Iout vs. Vin

Switched Mode Power Supply

A switched mode power supply integrates a switching regulator. Ideally this power supply will not dissipate any power. The switching regulator uses an internal electrical switch that regulates the energy that is transferred from the input to the output. Regulation is achieved by switching the device rapidly on and off. This transfer/conversion of the electrical power creates less loss, which results in a more efficient transfer of energy. Whereas a linear power supply that regulates the output voltage by continually dissipating power. Switched-mode power supplies can also be and usually are smaller and lighter dependent on its application than a linear supply because of the smaller transformer size.

The LM22679 is switching regulator that provides all of the functions necessary to implement a voltage step-down regulator using the least amount of external components and is capable of taking in a variety of input voltages providing up to 5A of load current and a regulated 5 volt output voltage adjustable as low as 1.285. An Internal thermal shutdown circuit protects the LM22679 should the maximum junction temperature be exceeded. The soft-start feature (pin) will allow the regulator to gradually reach steady-state which reduces start-up stresses and has internal loop compensation designed to provide a stable regulator over a wide range of external power stage components. Voltage mode control offers short minimum on time, allowing the widest ratio between input and output voltages. * Data from Texas Instruments LM22679 42V, 5A Simple Switcher Step-down Voltage Regulator PDF.

Table 7 LM22679 Voltage Regulator Electrical Specifications

	Parameter	Conditions	Min	Typ	Max
V _{in} (V)	Voltage In		4.5	5	43
V _{F_B} (V)	Feedback Voltage	V _{IN} = 8V to 42V	4.925/4.9	5	5.075/5.1
I _Q (mA)	Quiescent Current	V _{F_B} = 5V		3.4	6
V _{ADJ} (V)	Current Limit Adjust Voltage		0.65	0.8	0.9
I _{CL} (A)	Current Limit		6.0/5.75	7.1	8.4/8.75
I _L (μA)	Output Leakage Current	V _{IN} =42V, S _{SPin} =0V, V _{SW} =0V		32	60
		V _{SW} = -1V		31	75
R _{DS(ON)} Ω	Switch On-Resistance			0.1	0.14/0.2
f _O (kHz)	Oscillator Frequency		400	500	600
T _{OFFMIN} (ns)	Minimum Off-time		100	200	300
T _{ONMIN} (ns)	Minimum On-time			100	
I _{BIAS} (nA)	Feedback Bias Current	V _{F_B} = 1.3V (ADJ Version Only)		230	
I _{SS} (μA)	Soft-start Current		30	50	70

3.3.6 Uninterruptible Power Supply

An uninterruptible power supply (UPS) is a power supply that can provide emergency or extra needed power to a load when the input power source power fails, stops supplying. Usually used for computers or any other sensitive electronic device a UPS provides power protection from input power outages, faults etc. by almost instantly supplying electrical energy stored in battery within the supply system to the load originally being supplied. The case of the battery

triggered to supply the load, the power supply runtime is relatively short, minutes usually. This gives enough time to save any files needed and safely power down the device or time to find an alternate power source.

3.3.7 Solar Energy/Power

Solar Power is energy in the form of radiant light and heat that is harnessed from the sun to produce electrical energy. Often the energy is absorbed via a solar cell aka photovoltaic cell. A photovoltaic cell is a device converts light into electric current using the photoelectric effect. These cells require three basic attributes: 1) able to absorb light, generating either electron-hole pairs. 2) The separation of charge carriers of opposite types. 3) The separate extraction of those carriers to an external circuit. Many times these cells are grouped together in a module to create solar panel, which generates and supplies electricity to any application desired.

3.4 Sensors

3.4.1 Piezo

A piezo sensor will contain some sort of piezoelectric material. Various piezo sensors include: Integrated circuit piezoelectric sensors, piezoelectric accelerometers, piezometers and piezoelectric sensors. Often piezo elements are used as Force Sensing Resistors (FSR) in various applications.

For this particular application the piezoelectric material will be used to measure small vibrations given from somebody performing a “secret knock” to the system. These vibrations will be converted to voltages and processed and then will carry out a specific action. To understand the sensor first piezoelectricity and the piezoelectric effect must be understood first.

Piezoelectricity is electricity resulting from external pressure. An electric charge can build up in certain materials such as crystals, ceramics, bone, DNA and proteins. The electric charge accumulated is a response due to an applied mechanical force to the piezo-element.

The piezo-electric effect is a linear electromechanical interaction between the mechanical and the electrical state in crystalline materials with no inversion symmetry, meaning that in a given three dimensional geometric point group for a crystalline structure for every point (x, y, z) in the unit cell there is not an exact symmetrical point $(-x, -y, -z)$. These types of crystalline structures are said to lack a center of symmetry aka a non-centrosymmetric structure. When the material is placed under a mechanical stress, the atomic structure of the piezo-material changes. The ions in the structure become separated and a dipole moment is formed which creates energy. The dipole formed must not be cancelled out by other dipoles in the unit cell. In order for ions not to cancel with each other, the piezoelectric atomic structure must be non-centrosymmetric. This action allows certain properties such as the piezoelectric effect to occur within structures.

From induced mechanical stress a voltage will be produced from the surface of the piezo-material. A common and widely used compound is lead zirconate titanate (PZT). This crystal will generate measurable piezoelectricity when its static structure is deformed by only about 0.1% of the original dimension. PZT is used in sensors, actuators and ceramic capacitors.

The piezoelectric effect also can occur in the reverse process in that when a voltage is applied to a piezo-material a mechanical output or vibration will occur. This opposite piezoelectric effect is used in producing ultrasonic sound waves commonly used for transducers.

A piezoelectric sensor is a device that uses the piezoelectric effect to measure pressure, acceleration, strain or force by converting the effect on the material to an electrical charge. Piezoelectricity is found in many useful applications such as the production and detection of sound, high voltage generation, electronic frequency generation etc. It is also the basis of a number of scientific instrumental techniques with atomic resolution, the scanning probe microscopies and everyday uses such as the ignition source for cigarette lighters and push-start propane barbecues. A piezo sensor cannot be used for static measurements. Also increasing temperatures and pressure can negatively affect the sensitivity of the element.

For the use of the application the “7BB-20-6L0” sensor (Figure 4) will be used. This is an external drive type of sensor, meaning that an external force must be applied to the sensor in order for the material to function. The sensor is thin, lightweight, and durable and also has lower power consumption. It is made of Brass with lead connector wires (AWG32). This sensor is often used in clocks, medical equipment, toys and PDA’s (personal digital assistant) to manage data and information.

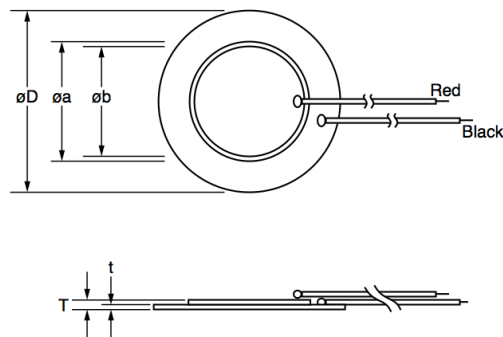


Figure 5 Piezo Schematic

Table 8 7BB-20-6L0 sensor

Resonant Frequency kHz	Resonant Impedance Ω	Capacitance (nF)	Plate Size Diameter (mm)	Electrode Size Diameter (mm)	Thickness (mm)	Plate Thickness (mm)
6.3±0.6kHz	1k (max)	10±30% [1 kHz]	20.0	12.8	0.42	0.20

3.4.2 Photo Sensor

A photoelectric sensor is a device that can detect the distance, absence, or presence of an object. There are a few different types of these sensors. A through beam, a retro-reflective beam, and a proximity sensing beam. An opposed through beam type consists of a receiver with a line-of-sight transmitter. When the light is blocked or diverted the receiver does not receive the light from the transmitter and therefore an object is detected. A retro-reflective type has a transmitter and receiver at the same location and uses a reflector to bounce the light beam back from the transmitter to the receiver. Just like before the object is detected when the beam is blocked or diverted and does not reach the receiver. Both of the above sensors are commonly used in sorting facilities and conveyor systems and in homes as garage door detectors. The proximity-sensing type is one where the receiver must receive reflected transmitted radiation from the object in order to detect it. This is the opposite type of detection from the other two types of sensors because in order to detect an object the receiver must see the transmitted light rather than doesn't see the transmitted beam. The proximity sensors are used in mobile devices, conveyor systems, wireless systems, fluid level sensing etc.

A photo-resistor is a resistor whose resistance varies as a function of the intensity of light it is exposed to. They can be programmed to be proportional to the amount of received light or inversely proportional to the amount of light received. A photo-resistor is made of a high resistance semiconductor. If light falling on the device is of high enough frequency, the photons absorbed by the semiconductor can give bound electrons enough energy to jump into the conduction band. The resulting free electron will conduct electricity and this will lower the resistance of the semiconductor. The photoelectric device can be either intrinsic or extrinsic. An intrinsic device has its own charge carriers and is not an efficient semiconductor. The intrinsic device only has electrons available in the valence band. The photons must contain enough energy to make the electron jump across the energy band-gap. Extrinsic devices already have added doped impurities. The impurities energies are closer to the conduction band. The electrons do not have to jump as far, and therefore lower energy photons will trigger the device. Photo-resistors are often made from cadmium sulphide (CdS) cells and are used in such applications as camera light meters, streetlights, clock radios, alarms and almost any solar application.

3.4.3 Motion

The ability to detect the presence of a person will be absolutely crucial for the camera subsystem of the KEES. In order to facilitate efficiency, it is best that a snapshot only be captured for processing when a person is detected to be in the vicinity of the KEES. Without a sensor to detect a person's presence, the camera subsystem will continuously process images captured from the camera to determine if a person whose face is in the KEES database is at the door. This increases overhead and would also require the camera to be on at all times, which is a waste of power. As a result, it is imperative to use an accurate sensor to detect a person's presence. Two types of motion sensors were considered: an area reflective sensor and a passive infrared sensor.

Area Reflective: Area reflective sensors work by using an LED for emitting infrared rays. The sensor uses the reflection of those rays to determine how far a person is from the device, and whether or not it exists within a specified range. Panasonic offers area reflective sensors that have a detection distance ranging from 1 cm – 200 cm. Therefore, the sensor is meant for applications in which a person needs to be detected in a very short range. While 200cm is a decent detection range, for the KEES a range that is a little larger is desired to optimize the chance of the camera obtaining a decent photo of the person's face. The sensor is recommended to be used in applications where non-moving people and objects without a temperature difference must be detected. For the KEES, people will be moving towards the door so the capability to detect non-moving people is not necessary. The person will most likely only stop moving once he or she has reached the door, and at this range the person might be too close to the camera to obtain a good shot due to being too tall or too short. The strength of area reflective sensors is that they are not easily affected by external sources of radiation, such as direct exposure to sunlight. However, many front doors of a house are shaded and are not directly exposed to sunlight, so this capability is not needed. As a result, area reflective sensors are not necessary for the KEES and a sensor that supports a larger detection range is necessary in order to compensate for latency in the system so that the camera can obtain a good shot of the person's face. For this reason, another type of motion sensor called a passive infrared sensor (PIR) will be used since they have larger detection ranges.

Passive Infrared: A PIR sensor can be used to prompt the camera subsystem when to begin processing images in a relationship analogous to master-slave. PIR sensors can sense motion and are often used in burglar alarms. The fact that they can detect when a person has moved in or out of the sensor's range can be utilized by the KEES. PIR sensors detect levels of infrared radiation using its built in pyroelectric sensor. They are called "passive" because they do not emit any radiation of their own. Humans and animals emit infrared radiation, and PIR sensors measure this emission against previous or standard

levels of infrared radiation to determine if someone is in the vicinity of the sensor.

There are two halves in the sensor, and both halves are setup so that they cancel each other out unless one half measures more radiation than the other. When a person or animal enters the vicinity it causes a positive change in the infrared radiation measured, triggering the sensor. When the warm body leaves the vicinity of the sensor, a negative change in infrared radiation is measured, which also triggers the sensor. They are very cheap, require little power, and are small. Many of them have a built in integrated circuit that automatically converts the analog output of the sensor to digital, outputting a digital high if motion is detected. This is ideal as it makes interfacing with microcontrollers, which will compose the camera subsystem, very simple.

PIR sensors have different ranges, and for the purposes of the KEES one that has a medium range will be necessary. Since its purpose is to detect when a person is approaching the front door, the range can't be too wide as this will increase the chance of detecting false positives, such as people walking on the sidewalk or squirrels running in the front yard. The detection range of the sensors can be reduced if desired by positioning it at a downward angle. A sensor that automatically converts the output to digital, instead of analog, is also desired to easily allow it to connect to a microcontroller without having to buy an analog-to-digital converter. A list of PIR sensors under consideration is shown below in the following table.

Table 9 PIR Sensor Comparison

Part Number	Voltage	Range	Price
VUPN5943	5V-9V	up to 20 feet	\$13.89
SEN-08630	3.3V-12V	not specified	\$9.95
Parallax 555-28027	3V-6V	two modes: 15 feet and 30 feet	\$12.95
Panasonic AMN32111	3V-6V	2m (~6.56 feet)	\$16.51
Panasonic AMN33111	3V-6V	5m (~16.4 feet)	\$16.51

Parallax 555-28027, VUPN5943, and SEN-08630: The Parallax 555-28027 has a reasonable voltage range. It also supports selecting the range via a jumper on the sensor. While this option of selecting the range is a nice feature, it is unlikely that a range of 30 feet will be needed since the goal is to simply detect when a person moves into the vicinity of the front door. However, this feature and its cheaper price makes it more practical than the VUPN5943 despite the fact that the VUPN5943 supports a larger voltage range. It also has a built in LED that

lights up when it detects motion. This would allow for us to visually observe when the sensor detects motion, which can aid during testing. The drawback of VUPN5943 is that the datasheet specifies that it was designed particularly for indoor use, and that using it outside may affect the stability of the sensor. Since the PIR sensor will be outside, this limitation rules out using the VUPN5943. Contrastingly, the Parallax sensor can operate in a temperature range of 32 F - 122 F which implies that the sensor can be effectively used outside. Furthermore, the sensor has a lot of examples and documentations for using it with an Atmega, which is the microcontroller that the group will be using to control the sensor. The SEN-08630 is cheaper, but its range of detection is not specified in the datasheet. While most PIR sensors have a range of approximately 15 feet, the lack of information in the datasheet concerning its range rules out using the sensor as such uncertainty is not desirable.

Panasonic Sensors: The most expensive sensors are the Panasonic models. The AMN32111 has a much smaller range of 6.56 feet. This smaller range is more applicable for the KEES, and would decrease the chance of the sensor reacting to other things beside people approaching the door. However, there is a downside to such a small range. The smaller range means that the camera will be informed when the person is very close to the door. While this is desirable, it also means that the camera has a smaller window of obtaining face pictures.

Furthermore, the AMN32111 also detects small temperature differences between the target and its surroundings: the smallest temperature difference it can detect is 7.2 F. Since PIR sensors operate by noticing differences in radiation, it could be more difficult to detect people in the summer since the radiation difference would be much smaller than in the winter. However, the ability of the sensor to detect such a small temperature difference makes the chances of it successfully detecting a person in the summer much higher. The sensor is also very small as it is only 14.5mm in length. The AMN33111 has the same price and specifications as the AMN32111, except it has a larger range of 16.4 feet. While this range does increase the possibility of detecting false positives, it does provide a larger window of detecting face pictures, increasing the chance that the camera will obtain a good shot of the person's face. As a result, the AMN33111 model is the most capable out of all these sensors. Its higher price is offset by its ability to operate effectively in both summer and winter as it is designed to work efficiently outdoors.

Conclusion: The two sensors that are deemed the best options are the AMN33111 and the Parallax 555-28027. The AMN33111 has a very detailed datasheet that clearly specifies its capabilities but is a little more expensive. The Parallax 555-28027 has more examples and support of its use and should also operate fine outside. It also has the ability to adjust the distance, which is good for scalability purposes in case the group decides to provide an additional security feature for the KEES that depends on detection over a larger range. As

a result, the Parallax sensor will be used due to its cheaper price, ability to debug easier via its onboard LED, documented use with Atmega, and its ability to support two different ranges. The sensor has three pins: one for ground, one for voltage input, and a third for the output. The output is active high meaning that a digital one is transmitted if motion is detected.

3.4.4 RGB

As it is known, mixing differently frequency (colored) light can form white light. One of the most if not the most common method is to use the colors red, green, and blue (RGB). An RGB LED is a single LED that is able to produce many different colors including red, green and blue do to the diffusion of the RGB light. Though these LEDs are rarely used to produce white light, RGB's can be pulse width modulated to obtain a variety of different colors or another common practical use is to use an RGB LED for a status indicator LED. Assigning certain colors to indicate specific tasks creates a more user-friendly design. For example a green LED may tell the user to perform a certain function while changing the green to a red may tell the user to stop performing a certain function.

In this design the RGB will be used with the RFID sensor to tell the user the status of the RFID and the door. The RGB LED will initially appear blue to indicate it is ready to read an RFID card. When the RFID reader reads the card and decides whether it is a valid card or not, if the card is valid the blue LED will turn green indicating an accepted card and will trigger the TIP31A transistor to unlock door. After a certain amount of time the strike will re-locks and the LED will turn back to blue to indicate the system is waiting/ready to read another card. If, however the card is invalid then the LED will change to RED to indicate to the user the card is invalid and after a certain amount of time the LED will change back to blue again.

Using an RGB LED for a status indicator can reduce the number of LEDs needed which frees up space on PCB design, may free up energy use and will lower the overall cost of the design.

3.5 RFID Research

The group decided that implementation of a Radio Frequency Identification (RFID) keyless entry system would be a desired function for the Keyless Electronic Entry System. RFID is a technology that fall under the Automatic Identification and Data Capture (AIDC) category. AIDC devices generally will identify a product or object, transmit data associated with the product and process the data onsite or from a remote server.

RFID accomplishes this task by utilizing radio waves to send and receive identification and data. A RFID system generally consists of an RFID reader as well as an RFID tag. Inside of the RFID tag or, card is a radio transmitter and a radio receiver. The radio transmitter and receiver can be powered by a battery in an active tag, but typically passive tags are used that is powered from the reader.

The card and reader are able to communicate through a specified radio frequency, the RFID reader demodulates the radio signals from the tag. The demodulation process slows down the incoming signals so the reader can decipher the signals. Once the radio signal has been sufficiently slowed down the reader decodes the signal into words that can be processed and interpreted by a microcontroller. One of the major benefits of the RFID system is that the reader and the tag circuits can fit into any shape or size the engineer's specification requires. This scalability and cheap cost have led to widespread adoption of this system.

The function that the RFID system will serve for the project is to gain access without the need of the traditional lock and key system. The user will have a RFID identification card, and hold the card up to the reader and unlock the door for approximately 5 seconds. This is the basic function that is needed for the system in which is being designed, with that in mind the group will enter the research phase to find applicable devices that will fit the need for this project. Some of the devices that group will be researching and selecting will be the RFID reader, RFID tags, and other circuit components needed to complete this system.

Some of the desired specifications of the RFID reader is that the reader must be able to scan an RFID card at a distance within the range of 2 to 5 inches, the reader must also be able to integrate with the microcontroller the team has selected for the project, and be compact and low cost. As the group researches possible readers that would integrate into our project properly, the group narrowed the selection down to 2 selections, a RFID Reader Module designed by Parallax, as well as Low Voltage Series Reader Modules designed by ID innovations. These two readers fit our basic specifications so researched both further to see which would best fit into our project design.

3.5.1 Parallax RFID Reader Module

The Parallax RFID Reader module was designed to be a simple to use low cost RFID reader solution. The card is approximately 62 X 32mm which fits the design specification requirements as well. Although the circuit fits within the design this is the largest of the modules that were researched. The module reads passive RFID EM4100 family transponder tags which fit the requirement that project uses passive RFID cards. The reader is capable of reading cards in a range from 1.75" to 3", under the best conditions this fits into the specification of range being 3 to 5 inches away. The card has 4 pins VCC, Enable, Sout, and GND. The Sout is a 1-wire, 2400 baud Serial TTL interface, this communication protocol could possibly be supported by the design depending on the microcontroller that will be chosen for the project. Also the 0.100" pin spacing for easy prototyping and integration, that is a good design attribute as it will aid the development and integration at later stages in the project which will prove to be beneficial later on. The card requires a VCC of 5volts to operate

Pin	Pin Name	Type	Function
1	VCC	P	System power, +5V DC input.
2	/ENABLE	I	Module enable pin. Active LOW digital input. Bring this pin LOW to enable the RFID reader and activate the antenna.
3	SOUT	O	Serial Out. TTL-level interface, 2400bps, 8 data bits, no parity, 1 stop bit.
4	GND	G	System ground. Connect to power supply's ground (GND) terminal.

Note: Type: I = Input, O = Output, P = Power, G = Ground

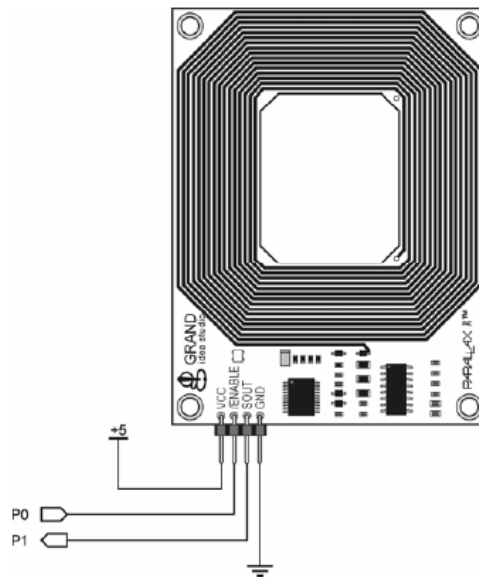
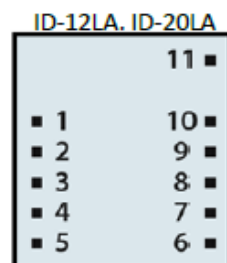


Figure 6 Parallax Reader Module

The reader modules in the series include the ID2-LA, ID12-LA, and the ID20-LA. The ID2-LA design does not include an antenna with the circuit and requires an external antenna to operate so the ID12-LA and the ID20-LA is what was decided to focus on for the project research. The modules support ASCII, Wigand 26 and Magnetic ABA Track2 data formats and require a VCC of 2.8-5volts. Both the D12-LA and the ID20-LA also have a beeper embedded in the circuit which is a nice feature that isn't included on the other module that was researched.



Bottom View

1. GND
2. RES (Reset Bar)
3. NC
4. NC
5. CP
6. Tag in Range
7. Format Selector
8. D1 (Data Pin 1)
9. D0 (Data Pin 0)
10. Read (LED / Beeper)
11. +2.8V thru +5.0V



Figure 7 ID Innovations Reader Module

All of the readers support 125kHz nominal frequency and are able to read cards in the EM 4001 64-bit format. Once a card is sensed the module sends a serial

string output containing the unique ID of the card. The dimensions for the reader is 26 X 25 mm, this is optimal size for the specifications needed for the project. The ID-20LA can read cards from a range of approximately 20cm and the ID-12LA can read cards at approximately 12 cm. This is the only notable difference between these two modules, as the architecture, design and pinouts are almost identical. The ID-12LA seems to be a better fit for the design specifications because the longer range of the ID-20LA isn't necessary for the functionality of the RFID system. The approximate distance that needed the design is about 5 inches and the ID-12LA fits that specification precisely.

Parameter	ID-2LA, ID-12LA, ID-20LA
Frequency	125 kHz nominal
Card Format	EM 4001 or compatible
Read Range ID3	Up to 30 using suitable antenna using ID-Innovations clamshell card @5v
Read Range ID13	Up to 12cm using ISO card, up to 18cm using ID-Innovations clamshell card @5v
Read Range ID23	Up to 18cm using ISO card, up to 25cm using ID-Innovations clamshell card @5v
Encoding	Manchester 64-bit, modulus 64
Power Requirement	+2.8 VDC thru +5 VDC @ 35mA ID-12LA, 45mA ID-20LA
RF I/O Output Current	+/- 200mA PKPK

Pin #	Description	ASCII	Magnet Emulation	Wiegand26
Pin 1	Zero Volts	GND 0V	GND 0V	GND 0V
Pin 2	Strap to Pin11	Reset Bar	Reset Bar	Reset Bar
Pin 3	To External Antenna ID-2LA only	Antenna	Antenna	Antenna
Pin 4	To External Antenna ID-2LA only	Antenna	Antenna	Antenna
Pin 5	Card Present	No function	Card Present*	No function
Pin 6	Tag in Range (Future)	Tag in Range	Tag in Range	Tag in Range
Pin 7	Format Selector (+/-)	Strap to GND	Strap to Pin 10	Strap to +5V
Pin 8	Data 1	CMOS	Clock*	One Output*
Pin 9	Data 0	TTL Data (inverted)	Data*	Zero Output*
Pin 10	3.1 kHz Logic	Beeper / LED	Beeper / LED	Beeper / LED
Pin 11	DC Voltage Supply	+2.8 thru 5V	+2.8V thru 5V	+2.8V thru 5V

* Requires 4K7 Pull-up resistor to +5V

Figure 8 ID Innovations Reader Pinout

The reader module that we selected that best fit the design specifications is the ID-12LA. The dimensions for the reader is 26 X 25 mm in contrast to the 62 X 32mm size of the Parallax circuit. While the dimensions both would integrate in the design, in general smaller is better if you can get the same functionality. Another one of the major concerns with the Parallax card is that the range was 1.75 inches to 3 inches which would could work but the preference is a distance of up to 5 inches which the ID-12LA delivered on. Although the Parallax module would be easier to prototype with because of the 0.100" pin spacing for easy

prototyping and integration, the ID-12LA is a superior device in many other regards, and ultimately is a better fit for the project

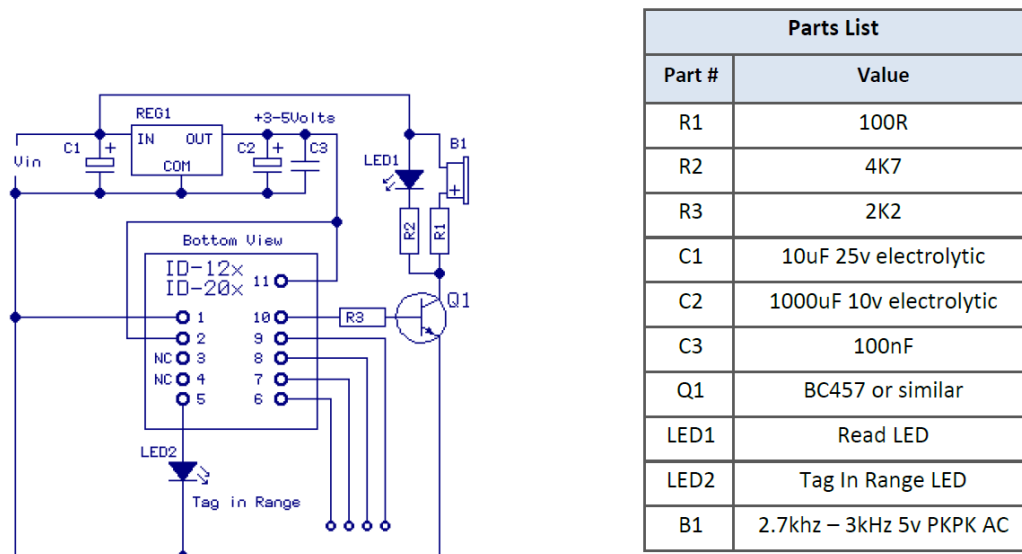


Figure 9 ID Innovations Schematic

3.6 Single-board Computers/Microcontrollers

3.6.1 Raspberry Pi

The Raspberry Pi has a Broadcom BCM2835 system on chip (SoC) that has an ARM1176JZFS CPU that supports floating point and has a clock speed of 700 MHz. The Pi also has 512MB of RAM. The Pi's processing power and amount of RAM make it a good candidate for handling face recognition, voice recognition, and hosting a web server if desired. Since it has an ARM processor, OpenCV algorithms for face recognition can be cross compiled to run on the ARM architecture using CMake. The floating point capabilities will be valuable for the face recognition algorithms as some of them heavily utilize floating point calculations. If desired, the Pi can be set to overclock to 800 MHz by simply using the "raspi-config" command on the Raspian Linux distribution that the Pi comes with. This can be used to fully utilize the processing power of the Pi. The Pi also has two USB ports and an HDMI port. These will be useful for debugging software by using a mouse, keyboard, and LCD screen to view output of the software. The USB port also provides the option of using a cheap Wi-Fi adapter to give the Pi Wi-Fi capabilities. The Pi also provides the ability to connect to the internet via Ethernet. Furthermore, a webcam can easily be connected to the Pi. The HDMI port also provides opportunities for scalability for the project: in the future, an LCD screen can be connected to the Pi to provide more features to the system, such as video chat between a person outside the door and a person who owns the KEES app. The Pi's GPU supports Blu-ray quality video playback and includes OpenGL ES 2.0, which can be potentially be used to augment a video

chat feature in the future. The Pi also supports Linux via its SD card slot: many Linux distributions such as Arch, Raspian, and Fedora can be installed on the system by loading the OS image onto a SD card. The SD card slot on the Pi also enables the database on the Pi to include many people, further adding to the scalability of the system. There is also a big open source community for the Pi, which will be useful for resolving any issues that are encountered along the way. The Pi also includes 26 GPIO pins, which can be used to connect to external hardware such as sensors or microcontrollers. It also requires only 5V to power, and a micro USB phone charger can be easily used to power the device during the software integration and prototyping phase of development. All of these features for only \$35 make the Pi a good choice for handling all of the high level software capabilities of the KEES.

3.6.2 Beagleboard

The BeagleBone is linux computer that is the size of a credit card. For the Keyless Electronic Entry Systems camera and image processing subsystem, the group requires a processor, or microcontroller solution that has a minimum processing power of 400 Mhz. The BeagleBone meets and exceeds this requirement with a 1GHZ Sitara XAM3359 processor. If this board is selected the image processing for the project will be able to capture and process many more frames per second which could be a huge value. Since the BeagleBoard is capable of running different Linux platforms on the ARM processor this will prove to simplify the prototyping and integration of the final project as well as making the simplifying the compilation and execution of the OpenCV libraries.

The BeagleBoard also has 65 General Pins for Input Output (GPIO) this will be more than enough for the project. Also the board has a 2G embedded Multi-Media Controller (eMMC). The eMMC is a package of both flash memory and a flash memory controller that is integrated on the same silicon die. The board had 512MB DDR3L as well as multiple connection types such as USB, UART, HDMI, and Video Out. This communication feature such as the USB and UART will allow the board to communicate with other microcontrollers if the team decides to use anything else.

3.6.3 Arduino

The Arduino is an AVR based microcontroller open-source electronics prototyping platform. It provides a platform for flexible, easy-to-use hardware and software. It's intended for designers and hobbyists interested in creating interactive objects or environments. Most simply put, you can program it to read sensors, perform actions based on inputs from buttons, control motors, and accept shields to further expand its capabilities. Possibilities are endless for the Arduino open-source hardware platform.

All Arduino boards have one thing in common: they are programmed through the Arduino IDE. This Arduino IDE is the software that allows a person to write and

upload code to the device. Beyond that, there can be a lot of differences. The number of inputs and outputs (how many sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, and form factor are just a few of the variables. Some boards are designed to be embedded and have no programming interface (hardware) which you would be required to be purchased separately. Some can run directly from a 3.7V battery, others need at least 5V. The input voltage for the board may be rated for a slightly higher maximum voltage but this is the safe operating range. Something to keep in mind is that many of the Li-Po batteries are 3.7V meaning that any board with an input voltage including 3.7V can be powered directly from Li-Po battery packs.

Most ATmega microcontrollers running at 3V will be clocked at 8MHz whereas most running at 5V will be clocked at 16MHz. Analog pins are labeled "A" followed by their number, they allow you to read analog values using the analog-to-digital converter (ADC) in the ATmega chip. Analog inputs can also be configured as more digital I/O if needed. Most Arduino boards, digital I/O pins 0&1 double as your serial send and receive pins and are shared with the serial programming port. Some Arduino boards have multiple UARTs and can support multiple serial ports at once. All Arduino boards have at least one UART for programming.

The Arduino UNO is the most famous out of the Arduino family. It includes 6 analog inputs, 14 digital outputs (6 PWM supported), and runs on an ATmega328 processor at 16 MHz clock speed with 32kB flash memory. There are near infinite amounts of shields available to expand its functionality. The Arduino MEGA is a more powerful version of the UNO and includes more pins, and a greater amount of storage. The Arduino Nano is praised for its compact and breadboard-friendly design which could be put on a custom PCB. It has relatively the same specs as the Arduino UNO and is in fact slightly cheaper. The Arduino community is huge and there are tons of guides, and tutorials available to get a project started within minutes. See the following table below for a comparison of the available Arduino devices.

Table 10 Arduino Chip Specifications

	Processor	Processor Voltage	Supply Voltage	Flash	SRAM	Digital I/O Pins	PWM Pins	Analog Inputs	Hardware Serial Ports	Dimensions	Shield Compatibility	Notes and Special Features
Uno	16MHz Atmega 328	5v	7-12v	32Kb	2Kb	14	6	6	1	2.1"x2.7" 53x75mm	Excellent (most will work)	
Uno Ethernet	16MHz Atmega 328	5v	7-12v	32Kb	2Kb	14	6	6	1	2.1"x2.7" 53x75mm	Very Good (some pin conflicts)	Has Ethernet Port. Requires FTDI cable to program.
Mega	16MHz Atmega 2560	5v	7-12v	256Kb	8Kb	54	14	16	4	2.1"x4" 53x102mm	Good (some pinout differences)	
Mega ADK	16MHz Atmega 2560	5v	7-12v	256Kb	8Kb	54	14	16	4	2.1"x4" 53x102mm	Good (some pinout differences)	Works with Android Development Kit.
Leonardo	16MHz Atmega 32U4	5v	7-12v	32Kb	2.5Kb	20*	7	12*	1	2.1"x2.7" 53x75mm	Fair (many Pinout Differences)	Native USB capabilities. USB Micro B programming port.
Due	84MHz ARM SAM3X8E	3.3v	7-12v	512Kb	96Kb	54	12	12	4	2.1"x4" 53x102mm	Poor (voltage and pinout differences)	Fastest processor. Most memory. 2-channel DAC. USB micro B programming port. Native micro AB port.
Micro	16MHz Atmega 32U4	5v	5v	32Kb	2.5Kb	20*	7	12*	1	0.7"x1.9" 18x49mm	N/A	Smallest board size. Native USB capabilities
Flora	8MHz Atmega 32U4	3.3v	3.5-16v	32Kb	2.5Kb	8*	4	4*	1	1.75" dia 44.5mm dia	N/A	Sewable Pads. Fabric-friendly design. Native USB Capabilities
DC Boarduino	16MHz Atmega 328	5v	7-12v	32Kb	2Kb	14	6	6	1	0.8"x3" 20.5x76mm	N/A	Can build without headers or sockets for smaller size. Requires FTDI cable for programming
USB Boarduino	16MHz Atmega 328	5v	5v (USB)	32Kb	2Kb	14	6	6	1	0.8"x3" 20.5x76mm	N/A	Can build without headers or sockets for smaller size. USB Mini B programming port.
Menta	16MHz Atmega 328	5v	7-12v	32Kb	2Kb	14	6	6	1	0.8"x3" 20.5x76mm	Excellent (most will work)	Mint-Tin Size and Prototyping Area. Requires FTDI cable for programming.

The Arduino is the perfect device to interface to all the external sensors of the KEES. It can act as a hub for the motion, Piezo, and photo sensors using digital interfaces called I2C or SPI. It would also very well be capable of powering the RFID system and controlling the Electric Strike. See figure below for a form factor comparison of some of the available Arduino devices. (Middle: Arduino Nano, Right: Arduino UNO).

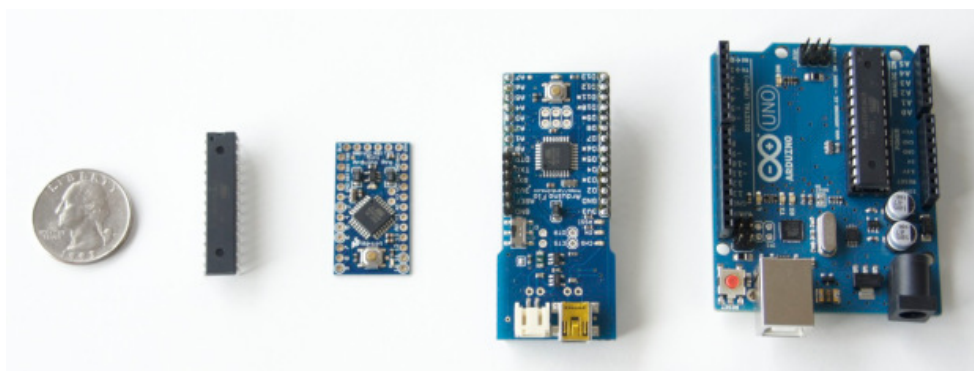


Figure 10 Arduino Model Size Comparison

Any of the Arduino models above require a clock source. This is used to drive the microcontroller and allow it to process instructions at every cycle. Clocks for the Arduino can be a basic crystal, external oscillator, or resonator. Oscillators have an advantage over simple crystals because they are more stable, and do not become unstable from hardware failures. An oscillator requires load capacitors for stability in which the types of load capacitors are specific based on the crystal, and electromagnetic interference in the environment. Internal oscillators are less common due to startup problems requiring extra capacitors known to off balance the circuit. Internal oscillators are also prone to failures caused by environmental factors such as dirt, humidity, or stray capacitances and impedances.

External oscillators are the most common clock sources available and do not suffer from the same problems noted previously. This is due to the fact that all components are sealed inside the can and safe from external conditions. The output signal is squared to ensure the signal is clean, strong, and in the VCC/GND level. Inside the can is a crystal made of quartz that has been cut to vibrate at a specific frequency. The tolerance of the internal crystal is around +/- 20ppm while the internal oscillator is +/-5% tolerance. A single oscillator is able to supply a steady clock to several chips at the same time. Common oscillation frequencies include 20MHz, 16MHz, 10MHz, and 4MHz. Less common frequencies such as 9.216MHz are also available which improve the accuracy of serial communications. Professional microcontrollers prefer to use an external clock for safety, reliability, and less chance of failure for the user. There are also downfalls for using external oscillators as they may sometimes require up to 15mA and consume lots of power. Depending on the available real-estate on a PCB, the size of the oscillator may also be an issue.

A resonator has less tolerance than external oscillators and is usually cheaper. A resonator is made from ceramic which has been precisely manufactured to achieve a given frequency. Unlike oscillators, resonators and crystals require additional circuitry to be used as a clock. Resonators are more accurate than internal oscillators but less accurate when compared to crystals. Resonators also are usually found running at lower frequencies, and can be made extremely small. The rated tolerance of the resonator is important when being used with RF signals. Also for serial communications, the tolerance of +/-5% found in external oscillators is usually too high. By using a resonator, the tolerance can be reduced to around +/-0.5%.

All oscillators suffer from aging which causes the frequency to fluctuate slightly over time, and may sometimes also vary with temperature. Running the Arduino at higher frequencies will require more power but will also allow faster processing. Considerations must be made as to which clock generator would be best suited for the microcontroller being used for the KEES project.

3.6.4 ARM Microcontroller

An ARM microcontroller was also considered as a hardware solution for handling image processing such as facial recognition as well as voice recognition. ARM is a widely used RISC architecture, and thus experience with ARM would be invaluable. Texas Instruments (TI) offers many ARM microcontrollers, ranging from 16 bit to 32 bit. Their 16 bit microcontrollers have a clock speed up to 25MHz which is obviously too slow to handle the intensity of facial recognition and voice recognition. To handle such tasks, a clock speed of at least 300 MHz is desired. TI' 32 bit microcontrollers have clock speeds up to 300 MHz, have up to 1 MB in Flash, and up to 512 KB of RAM. The 1 MB in flash should be enough space to handle code for both the facial recognition and voice recognition. It's clock speed of 300 MHz is fast for a microcontroller, and while it probably would handle the software necessary for facial recognition, it's processing time may not be fast enough to satisfy the real time requirement for the facial recognition and voice recognition. The KEES must inform the user as fast as possible upon receiving voice or a face image input. Furthermore, the cost of TI's 300 MHz microcontrollers, such as the TMS320C28346ZFEQ, is almost as much as a Raspberry Pi as it costs approximately \$30 from Arrow Electronics. As a result, it's inferior CPU speed, memory limitations, and no indication of support for a Linux operating system make TI's 32 bit ARM microcontrollers an impractical hardware solution.

Another option considered was an Atmega ARM microcontroller. Atmega offers more prestigious ARM microcontrollers than TI as it offers microcontrollers such as the ATSAMA5D31A-CU with an ARM Cortex A5 CPU with a clock speed of 536 MHz that support floating point, Linux, three USB ports and an ethernet port, and is even stated to be useful for Machine Vision applications. Such a microcontroller would provide more real time processing time that is desired. Furthermore, the ability to support Linux is invaluable: an operating system would enable the use of a file structure for saving data as well as multithreading due to the kernel's process scheduler. There is an embedded Linux port called uCLinux that is designed specifically for microcontrollers, which would probably work on Atmega ARM microcontrollers. The microcontroller also has an interface for CMOS cameras, and such a camera could be used to capture images.

The ATSAMA5D31A-CU has a very reasonable choice of \$14.94 on DigiKey. However, as the ATSAMA5D31A-CU is a microcontroller, it is greatly limited by the amount of RAM and program memory that it has: 160 KB of program memory and 168 KB of RAM. However, the microcontroller has an SD slot so the code could be potentially saved onto an SD card instead to overcome the small size of program memory. However, if it is desired to also use the microcontroller as a web server, 168 KB of RAM would be too small as the web server would be too small. While the microcontroller supports adding a 512MB 8 bank DDR2/LPDDR/LPDDR2, purchasing a 256MB stick would cost

approximately \$7, bringing the total cost of the microcontroller to around \$13 less than the Raspberry Pi. However, it was decided that it was worth paying \$13 more for the Raspberry Pi since it is 154 MHz faster, can overclock, and has an HDMI port which can be used to connect an LCD screen. All of these factors make the Raspberry Pi the better choice in terms of scalability.

3.6.5 Sitara SoC

The Sitara AM335x ARM Cortex-A8 MPU is one of the processor's that the group has decided would be worth researching. If the team decides to create an embedded system for the video and image processing, as well as the webserver, this class of chip proves to have enough processing power for this task, with speeds of up to 1GHz. The chip is based off of a 32-Bit RISC architecture, with 32KB of L1 Instruction 32KB Data cache, and 256KB of L2 Cache with Error Correcting Code (ECC).

The communication that the chip supports includes 6 UARTs and three I2C ports which will help if the project calls for any other microcontrollers in future revisions. Included with the 1 GHz processor is a NEON SIMD Coprocessor which could prove to be a huge benefit for the video image processing component of the project. The general purpose SIMD engine accelerates signal processing and multimedia such as video, and image processing functionality. This will allow for even faster processing of the video input from the camera and allow for more future functionality, and ease of programming in later stages of the project.

Another one of the added benefits of using the Sitara AM335x ARM Cortex-A8 MPU is that it has a 24-bit LCD controller, so if the team chooses to go in the direction of adding an LCD screen to the project it will be possible. In addition to the LCD controller is a Touch screen controller, so the team won't be inhibited later on if more functionality is added to the project in later stages. The Sitara AM335x ARM Cortex-A8 MPU matches all the desired specifications if its decided to go the route of developing an embedded system solution for the image processing component of KEES.

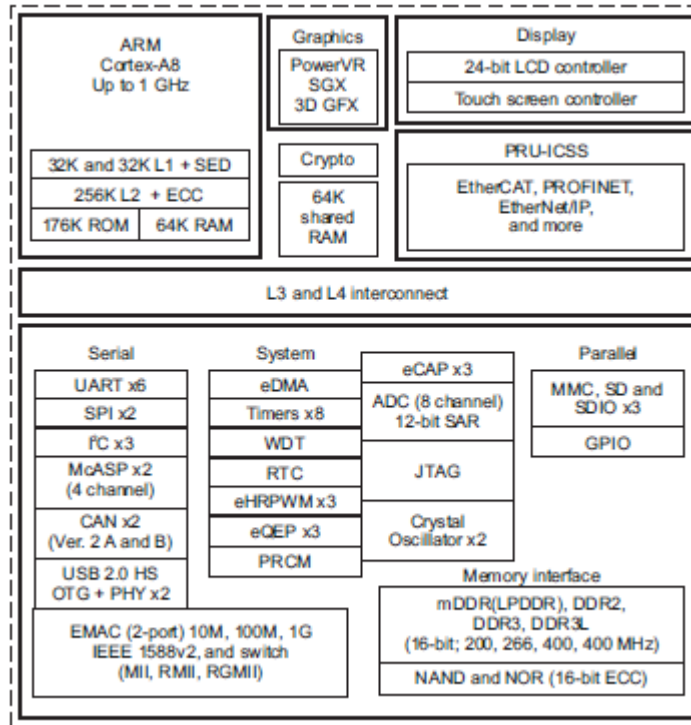


Figure 11 Sitara Architecture

3.7 Linux Ports

For the Keyless Electronic Entry System project camera and OpenCV subsystem board should have a stable and reliable environment to program in. The research for this platform consists of 4 different linux distributions Arch, Ubuntu, Debian, and Android. All of these platforms are able to run on the ARM architecture, and have been tested and proven on the BeagleBone and Raspberry Pi, in which the group is researching for viability for the project. The platform that is selected will need to be able to run and compile OpenCV, and should have good community support to be integrated into the project. Also it is required that the platform will be able to be used as the webserver, and access point to wifi, and database.

3.7.1 Arch Linux ARM

The first Linux port that will be evaluated for integration into the project is Arch Linux ARM. The Arch Linux distribution is aimed for maximum simplicity and full control to the user, it is a light-weight distribution that is highly customizable. The design philosophy for the Arch distribution focuses on simplicity and minimalism, and also the environment doesn't have an officially released GUI, but there may be support for this feature from the community. The architecture has optimized packages for soft-float ARMv5te, and also for hard-float ARMv6 and ARMv7. The software is actively developed with new versions being packaged and posted daily. This distribution has a large amount of documentation which is critical for a

student design project. Arch Linux is easily ported to BeagleBone as well as Raspberry Pi, and has been proven to run OpenCV libraries.

3.7.2 Ubuntu ARM

The Ubuntu distribution linux port is a widely adopted platform that is widely supported by developers as well as supported by the programming community. Like Arch this distribution started as a linux iteration for the i686/x86-64 and branched out to support the ARM architecture. There is a lot of support for this platform as well as thousands of packages for the desktop and the server. The Ubuntu distribution also provides a GUI for user which could prove to be beneficial for ease of use and navigation of the file structure. The current Ubuntu ARM port is compatible with the ARMv7 and newer iterations of the ARM architecture, and uses the Thumb-2 instruction set for ARM. The benefits of using the Ubuntu linux port is that this distribution has a high level of support, as well as a GUI interface that will make the process a lot easier.

3.7.3 Debian ARM

The Debian Linux port is widely supported and a very popular Linux distribution. Debian has a lot of community support and supports the most amount of hardware architectures than all of the other distributions of Linux. The most current distribution is the armhf, which supports the ARMv7 architecture. Another iteration of the Debian port is Raspian which is optimized for the Raspberry Pi. This could be a viable option depending on whether or not the team decides to go with the Raspberry Pi or the BeagleBone. The Raspian operating system pulls directly from the Debian framework and supports many of the original Debian packages. The Raspian operating system supports hardware floating point for ARMv6, the architecture of the Raspberry Pi, which will greatly speed up floating point operations. Raspian also now has support for the Java SDK, and comes with Python support. It has a large repository, which includes OpenCV.

3.7.4 Fedora ARM

The Fedora ARM port doesn't seem to have as much support or as large of a community as the previous ports discussed, but they are still a major player when it comes to Linux products. The newest release for the ARM architecture is Fedora 19 which has been tested on the Beagleboard xM. There is also another release of Fedora designed specifically for the Raspberry Pi which is called Pidora. This Fedora remix is optimized for the Raspberry Pi by the Seneca College. The Fedora distribution is compiled specifically for the ARMv6 architecture. Depending on whether the group decides to use the BeagleBone or the Raspberry Pi, these are two viable distributions that we could implement into the project.

3.8 Cameras

The camera that is used by KEES will be used to capture images of people who approach the door. It will serve as the input device for the facial recognition process. The camera needs to have a reasonable frame rate to increase the chance of capturing a good image of the person's face. Since people will be moving toward the door at variable speeds, the frame rate has to be fast enough to capture a clear picture of the person. Also, the camera must have software support for capturing its frame in a software application. Since the camera will be connected to the Raspberry Pi, it will need to be supported by either the Linux frame capturing library called `video4linux2`, the frame capturing method of OpenCV, or must supply its own driver for capturing frames. Also, since voice recognition is a potential feature of KEES, a camera that has a built-in microphone would also be very useful. The desired video resolution for the camera should be at least 640 x 480 in order to produce images of sufficient quality for face detection and face recognition as resolutions that are too low will decrease the effectiveness of the process. Many cameras support many different resolutions, which is useful as the resolution used will be experimented on to determine which resolution is best for the face recognition and face detection applications. A favorable balance will have to be achieved as higher resolution means longer processing time by OpenCV's algorithms, a potentially lower frame rate, but it also increases the accuracy of the face recognition process. Webcams will be considered as the Raspberry Pi has a USB port. One camera board system called CMU cam will also be considered.

3.8.1 CMU CAMv4

The CMU CAM is a low cost computer vision camera solution designed by Carnegie Mellon University. At a price of 99.95 the camera is considerably more expensive than some of the other cameras that the team is researching. The camera is mounted onto a proprietary board that handles limited computer vision algorithms as well as easily being integrated with a microcontroller through a serial port. The lightweight interface and small form factor is some of the main reasons why the group is considering integrating the CMU cam.

The CMU cam's main processor is a Parallax P8X32A operates at 80 MHz and is capable of processing the real-time data from the CMOS cam, and performing simple computer vision functions such as blob and color detection. The cam's processor is able to assist a microcontroller by performing the computer vision side, and then pipe the data to a controller with less processing power. The limited processing power of this camera doesn't seem to match the requirements for the Keyless Electronic Entry System implementation. The KEES project requires face detection as well as face recognition, so if the group chooses this camera the Parallax processor will most likely not be used for the functionality of the project.

The camera is capable of taking 640x480 pictures at a rate of 30 frames per second. The raw images can be dumped over the serial interface or sent to a flash card. Although the processor that is integrated with the CMU camera is lacking in speed and functionality for the projects need, the camera itself has a good frame rate and resolution that matches the design specifications. A possible use of the camera could just be to dump the images to a flash or process it in real time over the serial port using a processor with around 300 MHz or higher. Despite of the poor integrated processor power the CMU cam will still be considered by the team and contrasted with the other cameras researched for implementation into the design.

3.8.2 Raspberry Pi Camera Module

One option is the Raspberry Pi camera module, a camera module that is designed specifically to work with the Pi. The Raspberry Pi camera module connects via the camera serial interface. It is very small and light, and supports video in 720p and 1080p. The camera can take pictures that are 2592 x 1944, has a fixed focus lens, and can be purchased for \$29.99. The drawback of the camera module is that OpenCV and video4linux can't be used to grab a frame from the camera's output. Instead, another third party software such as raspivid or raspistill has to be used. Raspivid and raspistill control the camera via MMAL functions and are open source. There is a tutorial for modifying the source code of the camera software to use it to feed OpenCV the camera's buffer, so integration with OpenCV is definitely feasible and it would just require some modifications. People's experience with the camera suggests that with a 320 x 240 frame, 8 and 17 FPS is achievable, and with a 640 x 480 frame, 4-5 FPS is achievable with a small delay. Such a frame rate should suffice for the face detection algorithm. It also does not require external power as it is powered through the Pi. This fact would help simplify the voltage regulator circuit as no voltage will have to be delivered to the camera through the PCB. This camera module is a good option for the Raspberry Pi as it is designed specifically for the system. However, it does not have sound recording capabilities, and can't be used to provide input for voice recognition. Another drawback is that if the group decides to migrate to the Beagleboard due to its superior processing power, the camera would not be supported as the Beagleboard does not have a serial camera interface. However there is evidence that the camera supports mjpeg and can thus be used with software to stream video output to an IP address. MJPG-streamer, a command line tool, can be used to stream JPEG images an IP network. This potential ability is good for scalability purposes in case we decide to implement such a feature in the future. A script can potentially be created to pass the arguments to the MJPG-streamer to facilitate automation. Software that can be used to stream the camera's video to an IP address will be discussed more in section 3.10.5 Video Streaming.

3.8.3 HD Webcam HD 2300

Another alternative is the HP Webcam HD 2300. It doesn't require a USB powered hub to function properly when plugged into the Raspberry Pi, but only if there isn't another device in the other USB port. Since the other USB port will probably be used for a Wi-Fi adapter, a powered USB hub would have to be purchased. It also supports video up to 720p, with a frame rate of 30 fps. While the resolution of an image captured with the camera is not specified, it will probably be at least 1 MP. The camera costs \$30 from HP's website and also has an integrated microphone. However, there no documentation was found of this camera being used with OpenCV in terms of capturing frames. While it should be supported by OpenCV due to the fact that it is supported by the Linux driver, this uncertainty of support is not desirable as using an outside source besides OpenCv to capture the frames is not the preferred method.

3.8.4 Logitech Quickcam Pro 9000

The Logitech Quickcam Pro 9000 supports video resolution of 640 x 480 pixels, and a resolution up to 1600 x 1200 (2 MP) for pictures. It also has a built in microphone which could be used for a voice recognition application. The benefit of using this camera is that it does not require external power, and will be powered by simply plugging it into the USB port of the Raspberry Pi. It also has technology that automatically adjusts brightness so that good quality pictures can be taken. Its video frame rate is 15 fps, which is sufficient. However, its hardware specifications indicate that is designed to work on a CPU with a clock speed of at least 1 GHz, so it is hard to gauge how well the frame rate will be when it is used on the Raspberry Pi. It is supported by OpenCV, which makes integration with the face recognition application seamless. Also, since it is a USB webcam, it could be used with software to stream JPEG images to an IP network. The main drawback is the price: the camera can be purchased for \$60 on amazon.

3.8.5 Logitech C270

A much cheaper Logitech webcam is the Logitech C270 as it can be purchased from Best Buy for \$29.99. It is reportedly supported in Linux and it has been reported that OpenCV can successfully capture frames from the camera. The camera supports pictures with a resolution of 640 x 480, 1.2MP, and 3 MP. It has a maximum frame rate of 30 fps at a resolution of 640 x 480 on a processor with a speed of 1 Ghz. It has a built in microphone, and can capture video with a quality up to 720p. The drawback of the Logitech C270 is that it would always remain on, which would not be efficient in terms of power. Also, it is designed for computers that have a CPU speed of at least 1 GHz, and using it on the Raspberry Pi would probably result in a lower frame rate. However, the hardware requirements listed for the camera assume that Windows is the OS, and Linux is a much more lightweight OS. In addition, it has been reported that

this camera works fine with the Raspberry Pi. A drawback is that the camera needs external power to work properly with the Raspberry Pi. As a result, a powered USB hub would have to be purchased. However, since it supports mjpeg it can be used with software that can stream JPEG images to an IP network.

3.8.6 Logitech C300

The Logitech C300 is an even cheaper camera that can be purchased from Amazon for \$20. Like the C270 it has a built in microphone and can support video up to 1280 x 1024 (resolution that most laptops use) with a frame rate of 30 fps. It can take pictures with a resolution of up to 5 MP, and like the C270 it can be used with an mjpeg streamer to stream video. It has been reported that video resolution of 320 x 240, 640 x 480, and 1280 x 1024 works with the Raspberry Pi. It has also been reported that this camera works with the Raspberry Pi without a powered USB hub, which is very favorable. Like the C270, OpenCV natively supports the camera. The drawback is that the video quality isn't HD, but its quality should be well enough for use with KEES.

To conclude, the Logitech Quickcam Pro 9000 is sufficiently more expensive than the Raspberry Pi camera module and the Logitech C270 for it to be a viable option. The Logitech C300 is even cheaper, and does not require a powered USB hub to function in the Raspberry Pi and is thus the best option out of all the webcams. However, the Raspberry Pi camera module would probably deliver the fastest frame rate since it is designed specifically for the SoC. Acquiring a decent frame rate is very imperative for the KEES as it increases the chance of capturing a decent picture of the person approaching the door. The raspberry Pi camera does require using external software to feed frames from the camera's buffer to OpenCV, but this should not be a problem to implement due to the availability of tutorials for accomplishing such a task. While it does lack a microphone like the Logitech C300, the Raspberry Pi camera module's decent frame rate make it the best option for the KEES in terms of success for the face detection algorithm used. It is uncertain what the frame rate would be if the Logitech C300 was used, but given the Raspberry Pi's decent processing power the capture rate should be decent. Also, the camera module connects to the Raspberry Pi via a camera serial interface instead of a USB cable, and the connector is very short in length. This would require the camera module to be connected very close to the Raspberry Pi, placing a restriction on the logistics of how the Raspberry Pi is placed on the door frame. While a USB microphone can be purchased and used as input for voice recognition, the fact that the Logitech C300 is cheaper and provides a microphone makes it a more valuable product.

3.9 Lock

3.9.1 Servo

A servomotor is a small motor that is controlled with an electric signal which determines the amount of movement of the shaft of the motor. These motors are small in size but can deliver big punch. They are highly energy-efficient and power-efficient as well. The servo is made up of a small DC motor, a potentiometer and a control circuit. The small motor rotates and the potentiometer's resistance changes, this gives the control circuit the ability to regulate how much movement and the direction of the movement. The motor has proportional control, which is a linear feedback type control system. The motor's speed is proportional to the difference between its actual position and desired position ($P_{out} = K_p * e(t)$, $e(t) = SP(\text{set point}) - DPV(\text{desired point variable})$). If the motor is near the desired position, it turns slow, if it isn't near its desired position it turns fast. This is very efficient because the motor only runs at the necessary speed to achieve the desired output. The way the motor knows to run and at what speed is through pulse width modulation (PWM), which are simply just electrical pulses sent. The PWM sent to the motor determines position of the shaft, depending on the duration of the pulse sent the motor will turn the rotor to the desired position. There are two types of these motors, AC and DC. An AC servo can usually handle higher currents and is used more in industrial machines, where DC servos are usually used in smaller applications because they are not are not designed for large currents. DC motors are generally less expensive than AC motors as well. The motor could be modified and transformed into a mechanical locking mechanism for the door. The motor could either directly drive the deadbolt or a lever attached to the deadbolt.

3.9.2 Electric Strike

An even more ideal option is an electric strike. An electric strike is a low voltage access control device used many times on doors in replacement of traditional locks to provide added security and conveniences such as traffic control, specific and limited access as well as remote lock/unlock. Electric strikes generally have at most +/- 10% voltage tolerances and most electric strikes will require 12 DC volts to function, meaning to lock and/or unlock the hinge.



Figure 12 Electric Strike

Electric strikes can come in two types of configurations, “fail secure” and “fail safe”. A fail-secure also called normally opened (NO switch) function type is one in which applying an electric current to the strike will cause it to unlock. They can be powered by alternating current (AC) or direct current (DC). AC can cause a little buzz (noise) where DC is virtually noiseless. In case of a power failure the strike would remain locked.

A fail safe type also called normally closed (NC switch) is one in which applying an electric current to the strike will cause it to lock, meaning that it needs power to keep it locked. Fail safe locks always use direct current DC. In case of a power failure the door could be opened by being pushed or pulled.

A fail secure (normally opened) type electric strike will most likely be used for this type of project because it is more ideal for security purposes.

3.10 Software Research

3.10.1 Web Server & Database

A web server will be required to access the KEES over the internet to ping its current status, check the visitors list, and receive notifications via the Android application. There are many options for the group to consider when choosing a web server. Should it be run locally or should it be a cloud-based server such as Google App Engine and hosted remotely? The Google App Engine eliminates much of the initial required setup time. It provides the backend for our application such as data storage, communications, and process management.

The data storage features of the Google App Engine include a Datastore which provides scalable storage system for data in a distributed NoSQL service. A second part of storage is also included called the Blobstore which allows storage

of large objects such as videos or image files. There is also a sophisticated caching system, logging, and search. The communications side of the Google App Engine includes a Channel feature which can create a persistent connection for real-time communications. The Process Management feature allows for a queue of tasks or for tasks to be scheduled to run at specific periods of time.

The Google App Engine includes a full SDK for Java and Python, and provides the redundancy, performance, scalability, and security well known in all of Google's services. However it may not be necessary to add a 3rd party to the KEES system that will only require a small amount of data transmission. It could also further complicate the architecture of the system because additional web services would be required for the KEES to send/receive data to the Google App Engine. Essentially, the cloud server would constantly be polling the status of the KEES and all data would need to be sent over the internet, rather than the local network where bandwidth is not an issue.

Considering this would be for home use and would most likely be on the same network as residents of the home, there can be no guarantee that the bandwidth available would be sufficient enough for transmission of data with no significant delays. The figure below is a representation of the communication links between the KEES and users of the system. The dotted lines represent the fact that an internet connection is required. Keep in mind, the frontend UI would be hosted through the Google App Engine and therefor would be heavily reliant on the connection between the KEES.

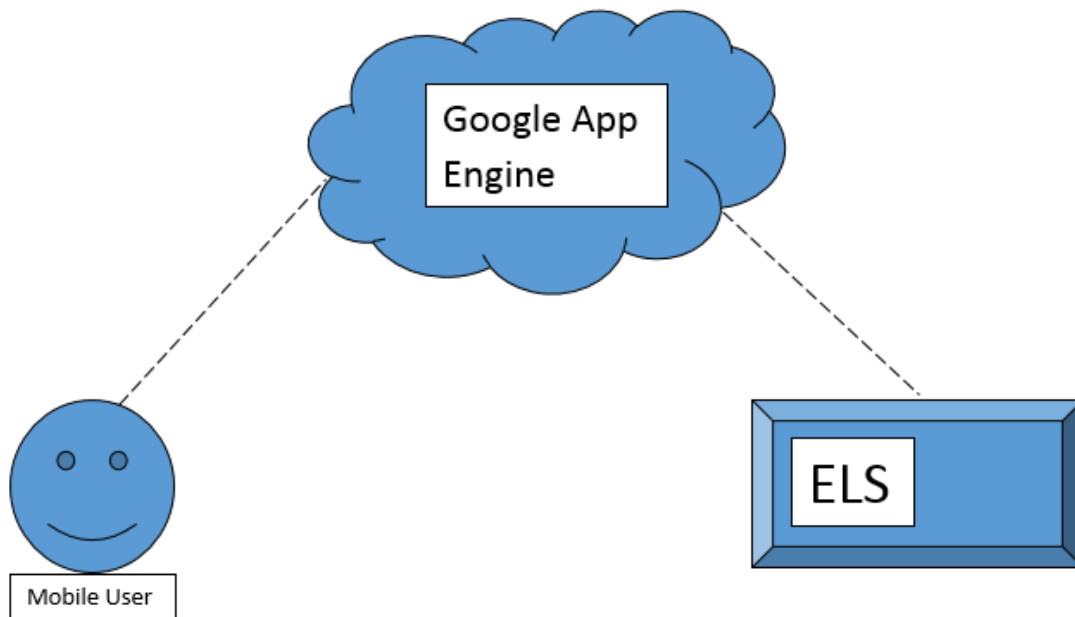


Figure 13 Google App Engine Flow

Consider a small example: a remote user wishes to check the current status of the KEES, locked or unlocked. A request is sent to the host (cloud server) which must then query the KEES for the status of the lock, or it already has the status

of the lock because it is constantly polling the KEES. Either way, once the status of the lock is returned from the KEES, it is available to the user. The required 3-way communication over the internet could suffer from delays, and is unnecessary. The always-on reliability provided by the Google App Engine is irrelevant if the KEES network is unstable. Thus, there isn't much difference if a user was to query the cloud server, or just query the KEES directly. This would also enable the administrator of the KEES to manage the system locally on their own home network even if they didn't have an internet connection.

Many considerations must be made if choosing not to use the Google App Engine as a web server for the KEES. If the local webserver route is chosen, first a decision must be made as to which web server to use. Since this is a small project with very little overhead, and if running locally it would be on a microprocessor, it only makes sense to use the most lightweight web server available. Apache would be the most popular choice today as it has a larger toolbox of features, but it also consumes over twice the amount of RAM out of the box when compared to more light-weight web servers such as: lighttpd or nginx. See in the next figure below for a comparison of the possible web servers to be chosen.

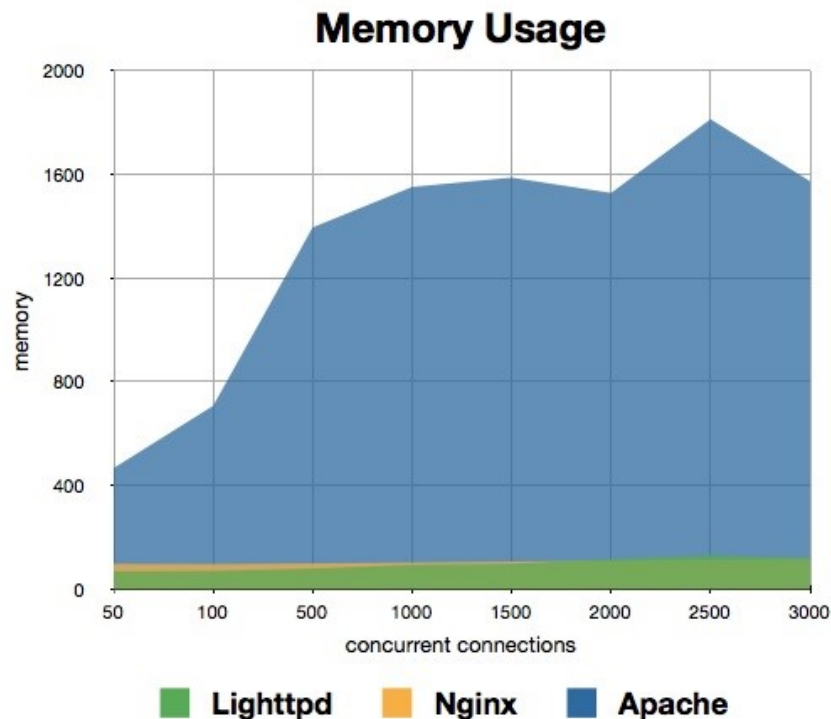


Figure 14 HTTP Server Memory Usage comparison

Table 11 HTTP Server features

	basic access authentication	SSL/TLS https	virtual hosting	CGI	Java Serv	SSI	ISAPI	Admin console	IPv6
Apache HTTP Server	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
lighttpd	Yes	Yes	Yes	Yes	No	Yes	No	No	Yes
nginx	Yes	Yes	Yes	No	Yes	Yes	No	Yes	Yes

With Apache being the resource hog that it is, and after the further investigating of nginx and lighttpd, nginx seems to be the web server of choice. Lighttpd apparently has a simpler initial setup than nginx but is prone to memory leakage. According to benchmarks around the web, nginx beats lighttpd in stability, CPU consumption, and ease of use by a long shot. As of October, 2013 – According to: http://w3techs.com/technologies/history_overview/web_server nginx is used nearly 15 times more around the web than lighttpd.

Rather than running a web server from a UNIX based micro PC, an embedded solution is also available. Embeddedmarket.com offers the AVR Embedded Web Server preloaded with the required TCP/IP stack for Ethernet communications. The AVR Embedded Web Server is an ATmega32 with an ENC28J60 Ethernet interface. It is capable of running web server software such as IIS or Apache. The AVR Embedded Web Server would be the most low-cost, smallest form factor solution. See the Table below for available I/O lines.

Table 12 AVR Embedded Server I/O

ATmega32 Port Pin	Interface Name & Pin
PB0	LCD – RS
PB1	LCD - RW
PB2	LCD - EN
PB4	ENC28J60 – CS
PB5	ENC28J60 – SI & ISP Port – MOSI
PB6	ENC28J60 – SO & ISP Port – MISO
PB7	ENC28J60 – SCK & ISP Port – SCK
PD2	ENC28J60 – INT
PD3	ENC28J60 – RESET
PD4	LCD – D4
PD5	LCD – D5
PD6	LCD – D6
PD7	LCD – D7
PC6	LED marked on the board as D2
PC7	LED marked on the board as D1
PA7	Switch marked on the board as SW2
PA6	Switch marked on the board as SW3
PA5	Switch marked on the board as SW4
PA4	Switch marked on the board as SW5
PA1	Temperature Sensor LM35
PA0	Variable resistance (Preset) located near the JTAG port on the board

Following the choice of a web server, the backend application framework must be chosen as well so that web services can be created and interactions can take place between the KEES and mobile users over the internet. Research for a type of web service begins with the two architectures behind serving web content, SOAP and REST (RESTful). RESTful API's allow for a more flexible data representation such as serializing data in JSON format and are completely stateless. RESTful API's are easier to understand because they add an element of using standardized URLs, and they are more light-weight by not including some of the additions in SOAP and by relying on HTTPS for a secure connection. So which REST based frameworks are available? The choice of a framework is mostly dependent on which language developers prefer. There are many popular web frameworks at the moment, some based on node.js (JavaScript), PHP, Ruby, Perl, and Python. High level frameworks in JavaScript and python are extremely powerful not because they can both be used to provide web services, but can act as web servers themselves. Some even include their own database storage mechanisms. Though they can still be paired with existing web servers (Apache, nginx, etc.) which improves performance based on benchmarks around the web. Expressjs would be the best choice if going for JavaScript based web services, and unfortunately is used everywhere. It would also be beneficial to developers of the KEES if the language was consistent across the entire project. Considering most of the project could be coded using

Python, this would be the logical choice. There are many full high-level Python web frameworks that would be a great fit for our application such as CherryPy, Django, TurboGears, and web2py. See the following table below for a comparison of these frameworks.

Table 13 Python Web Framework features

	Ajax	MVC	ORM	Testing	DB	Caching	Form validation
CherryPy	Yes	controller & URL dispatching	ORM agnostic	use stdlib's unittest and doctest	depends on ORM	Yes	Form validation engine agnostic
Django	Yes	Full stack	Django ORM	Yes	Provided by South	Cache Framework	Django Forms API
TurboGears	Toolkit-independent, provides support via JSON	Full stack, best-of-breed based	SQLAlchemy	nose	SQLAlchemy-Migrate	Support formemcached, and any WSGI compliant system	ToscaWidgets, utilizing FormEncode
web2py	Yes	Yes	Yes	Yes	Yes	Yes	Yes

As for a frontend UI to the webserver, just as with the backend framework, there are many options available. Web UI Frameworks such as JQuery Mobile, PhoneGap, Bootstrap, or HTML5 Boilerplate would be a good choice considering the KEES will mainly be accessed from a mobile device. With the combination of HTML5, CSS3, and JavaScript, the frontend to the web server can be created to provide an interface between the KEES and the user. Any of the UI Frameworks can be used in all browsers desktop or mobile, and can be ported to apps for iOS, Android, and Windows.

A database must also be considered for storing data related to guests in the system. SQL provides a table structure for storing data in columns which represent different pieces of data for each record (row) each having a unique primary key. Relationships between records can be formed using foreign keys if needed. There are many options to consider such as NoSQL, Hadoop, SQLite, and MySQL. SQLite stands out for being the most lightweight and is used in many home automation projects. It's simple to setup, compact, requires only a single database file, is server-less, and does not require any initial configuration. MySQL has better performance when handling large amounts of requests and dealing with big data, has better scalability, and more options for tuning the performance. SQLite has many recommendations around the web for being the best for small projects. Based on the requirements of data storage for the KEES, SQLite would be a good fit.

Dynamic DNS is a necessity if hosting the web server locally with the KEES. DDNS allows for a remote site to be accessed by its domain name even if the IP of the host changes. Static IP's are normally assigned by an ISP for business use only. Considering the KEES is a home solution, most users will have a dynamic IP address that changes frequently. Depending on the service (some free), a specific hostname can be chosen which gets updated every time the IP of the host changes. A request for update of the IP address in dynamic DNS is sent upon any change of the internet interface's IP address (including switching between primary and secondary internet connections). This keeps the DNS record for the particular IP address up-to-date and mapped services may be accessed by the corresponding host name. Many modern routers also include support for DDNS and can link to most providers directly without the need for additional services. A domain name for home connections can be created with services provided by DynDNS, No-IP, and changelIP. DynDNS is a popular solution but requires a credit card just to start a trial account. The main difference between free DDNS and paid DDNS services is that with free services the user can choose from second level domains such as 'els.no-ip.com', whereas with a paid service the user has their own domain name 'els.com'. Considering the restrictions DynDNS has put on their DDNS services over the years (expires in 30 days of inactivity, hostname expires every 30 days, etc.), a service such as No-IP would be a better choice. No-IP provides 3 free domain names and unlimited subdomains which will never expire. No-IP works the same way as DynDNS but has better reviews around the web.

3.10.2 App Development

A major component of the KEES involves using an app as a user interface to the KEES. The app's user interface has to be clean, easy to understand, and responsive. It will be a very simple UI that allows the user to make queries to the KEES database, unlock and lock the KEES, and receive notifications from the KEES web server. The first option considered was creating an Android app using the Java Android SDK. Three of the group members have experience making Android apps, and it would be very easy for the group to quickly create a very lightweight app. None of the group members have experience making iPhone apps, so this option was ruled out. The app will have a basic user interface that can be created using XML. A main goal of the app is to work on as many devices as possible. As a result, the app will need to be compatible with the Android OS versions that are most widely used. The Table below displays a list of the various Android OS and their distribution.

Table 14 Android OS Distribution

Version	Codename	API	Distribution
2.2	Froyo	8	2.2%
2.3.3-2.3.7	Gingerbread	10	28.5%
3.2	Honeycomb	13	0.1%
4.0.3-4.0.4	Ice Cream Sandwich	15	20.6%
4.1.x	Jelly Bean	16	36.5%
4.2.x	Jelly Bean	17	10.6%
4.3	Jelly Bean	18	1.5%

As the table indicates, almost 50% of smartphones use the older Android distributions Gingerbread and Ice Cream Sandwich. As a result, the Android app will be designed so that it supports at least version 2.3.3-4.3. This should not be difficult, but it does mean that newer interface designs, such as the Action Bar interface will not be able to use. However, there is a third party library, called ActionBarSherlock that is supported on all versions of Android and can be used to provide a sleek, organized interface if desired.

As the app will need to run on as many Android devices as possible, the different screen sizes across Android devices will also have to be taken into account. The table below shows the relative number of devices that have a particular screen size and density.

Table 15 Android Screen Size and Densities Distribution

	ldpi	Mdpi	Tvdpi	Hdpi	xhdpi	xxhdpi	Total
Small	9.2%						9.2%
Normal	0.1%	15.1%		33.4%	22.2%	8.8%	79.6%
Large	0.6%	3.6%	1.2%	0.5%	0.5%		6.4%
Xlarge		4.4%		0.3%	0.1%		4.8%
Total	9.9%	23.1%	1.2%	34.2%	22.8%	8.8%	

According to the android developer website, Xlarge screens are at least 960dp x 720dp, large screens are at least 640dp x 480dp, normal screens are at least 470dp x 320dp, and small screens are at least 460dp x 320dp. The screen densities (dpi) are also defined in ranges where a low density screen (ldpi) has a pixel density of 100-130dpi and a high density screen (hdpi) has a pixel density of 180-280dpi. To account for the varying screen sizes and screen pixel densities, certain steps can be taking such as providing alternative resources for some of the different screen sizes and densities, as well as alternative user interface layout. Also, the android developer website provides tips and guidelines for making user interfaces that automatically resize to fit the screen.

These guidelines will be used to design an Android app for KEES that has a clean and robust user interfaces on as many devices as possible.

The app will communicate with the web server that is hosted on the KEES. As a result, some decisions must be made concerning the functionality and purpose of the app. One option was considered is that an Android app is not even necessary as the front end UI of the server's web page can be designed to provide all of the query functions, such as locking and unlocking a door, checking the lock status of the door, and viewing pictures taken by the camera. The fact that framework such as JQueryMobile can be used to enable the web page to be compatible with desktop, mobile, and tablet browsers makes accessibility virtually ubiquitous. However, if all the functions of the KEES are available on a web page, potentially anyone could access the main functions of the KEES once the web page is discovered. This greatly undermines privacy and security. The app can be used to simply open the web page that will contain all the KEES UI functionality. A client-server model can be used to only allow the web page to be viewed when a request is sent from the app. This would ensure that only owners of the app can make queries to the KEES. Another option is that the app can provide users the option of either using the UI provided on the web server by loading the web page, or using the app's native UI to make queries to the server. The benefit to this approach is that a user may only want to perform a simple task, such as locking the door, without having to load the entire web page. As a result, this option will be pursued in the app's development.

In addition to providing a user interface to the KEES, the app's other main purpose is to notify the user of status, such as telling the user that a person came to the front door while the house was vacant. Although Python can be used as a servlet to dynamically update the web server's page, unless the user is currently viewing the web page, he or she will not be aware of any status messages that are presented on the page. The page would also have to be manually refreshed in order to view the changes. The Android app will be the perfect system to notify the user. The Android app can periodically query the status of the system by using the HTTPClient or URLConnection class to access the server. Java easily supports multithreading and thread synchronization through its Thread class, so this functionality can easily be implemented without blocking the app's execution. The app can also be used to send the name and picture of a person to add to the database.

The app can be designed to have the ability to effectively notify the user when the app is actively used, and when the app is running in the background and not directly viewed by the user. The AlertDialog class is an effective method of displaying a pop-up notification inside the app that disappears upon the user's action. There is also the Toast Class, which can be used to temporarily display text for a specified time period inside the app. Also, the Android SDK provides methods to create custom notifications that display in the phone's notification area at the top of the phone by using the NotificationCompat.Builder

Class. Notifications can also be displayed in more detail in the notification drawer, a pull-down view. The NotificationCombat.Builder Class can be used to create notifications that have a custom audio sound and vibrate pattern. It can even cause the notification to flash any LEDs that some phones, such as the Samsung Galaxy S3, will have. These options will be used to create an app that effectively informs the user. Receiving a notification from the KEES will be analogous to receiving a text message: simple and effective.

3.10.3 - Wireless Communication

The KEES will require wireless communication for an always-on internet connection. This could be made possible using the TCP/IP protocol via Ethernet or WiFi. WiFi has an average range of 150 to 300 ft. depending on the amount of interference and is the most common form of wireless internet communication today in homes around the world. WiFi works similar to the way radios and walkie-talkies communicate. A WiFi device is capable of converting digital information into analog radio waves which can be sent to the router on the network via an antenna in the device. Once the data is received by the router is undergoes an analog to digital conversion, which is then transmitted to the internet using a wired Ethernet connection. The same process occurs when receiving data, just in reverse as the digital to analog conversion would be taking place at the router, and analog to digital conversion taking place on the device.

There are many sub standards based off of IEEE 802.11, which vary in applications. 802.11ac is the newest WiFi standard and improves on the performance of the previous 802.11n by nearly 3 times the amount at the same distances. Many modern routers also include dual-band support and are capable of operating in 2.4 GHz and 5 GHz frequencies simultaneously with a baud rate of around 11 Mbps. Benefits of 802.11 include long range, low cost, and a relatively strong signal. Disadvantages include high power consumption, and complex network configuration. If power consumption was a deciding factor, WiFi may not be the best technology to use. Ethernet would have the highest reliability but WiFi should be reliable enough and secure enough to serve as the link between the KEES and the internet. The user will communicate with the web server wirelessly by making HTTP requests. Data can be encapsulated in JSON or SOAP objects. In most programming languages, to transmit data between two devices over the network, a socket must first be created. Once the webserver is running and the socket is opened, it is able to listen for incoming connections. Unique URLs can be used to call specific scripts to be run on the server. For example the KEES door lock/unlock buttons on the GUI would actually be links which would be processed by the server by analyzing the HTTP requests.

The Hypertext Transfer Protocol (HTTP) was designed specifically to allow communications between a client and server. It's a request/response based protocol where a web browser may act as the client, and the web site hosting the application would be the server. As stated previously, a client may submit an

HTTP request to the server which is processed and then a response would be returned to the client. The following two types of requests can be made by the client; GET and POST. The HTTP GET command requests data from a specific resource on the server. The POST command submits data to be processed by a specific resource on the server. GET requests are able to be cached, can remain in the browser history, are able to be bookmarked, and should be used to retrieve data only. For requests dealing with sensitive data, the HTTP POST method should be used. In a HTTP POST request, query strings are embedded inside the message body rather than inside a specific URL. POST requests are never cached, do not remain in the browser history, and cannot be bookmarked. To be somewhat secure, a POST request would be preferred over a GET request if querying for a user in the system. This way, specific information used to identify a user would not be exposed in the URL, but rather embedded inside the body of the request.

3.10.4 Video Image Processing

The KEES will need to have image processing capabilities in order to fulfill the requirement of informing the user with the identity of the person who came to the door. To fulfill this requirement, the KEES will need the ability to capture images, detect faces, and recognize faces. The open source image processing library OpenCV provides multiple algorithms to and functions to achieve these tasks. It is the most open source computer vision library available, and it supports C,C++, Python and java. The option of writing image processing algorithms from scratch was considered, but it was decided that using OpenCV was the best option for efficiency as the algorithms are optimized. OpenCV can also be cross compiled using the cross-platform, open source build software CMake so that it can run on ARM architectures in addition to x86 architectures.

3.10.4.1 Frame Capture

OpenCV provides the ability to capture images from supported webcams. It can take a frame from video and store it in a memory buffer, with options to convert the image to grayscale, as well as adjust the resolution. There are also other libraries that can be used to capture images. The VideoInput library is a Windows library with similar functionality, and there is a Linux library called video4linux2 that also captures frames from supported cameras. A fast capture frame rate will be desirable in order to increase the chance that a clear shot of the person's face will be obtained. OpenCV also provides the ability to load and save images via the imread and imwrite functions. It can read a variety of image types such as JPEG (.jpeg, .jpg), Windows Bitmaps (.bmp, .dib) and Portable Image format (.pgm).

3.10.4.2 - Face Detection

To recognize faces in an image, a team of classifiers has to be build. A classifier consists of a pattern, or a collection of numbers, and a threshold. An expert is

convolved against faces and non-faces to develop a histogram of values that correspond to faces and non-faces. Then, a threshold is chosen that determines what convolution values indicate a face, or non-face. OpenCV uses the cascade classifier approach which consists of stages in which one uniform classifier is built from many other classifiers. In each stage, simpler classifiers are applied to faces and non-faces that are in the training set. Some classifiers are rejected, and the ones that make it through all the stages are kept. A weighted voting technique is used to determine which classifiers are rejected. OpenCV provides two ways to use a cascade classifier: Haar and Local Binary Patterns (LCB). Local Binary Patterns is a faster implementation since it uses integer features as input for the classifiers unlike the Haar implementation. The quality of the classifier largely depends on the quality of the training images used, so care will be taken in choosing a good database of training images. Both LCB and Haar will be experimented with to determine which implementation provides the best results for the KEES.

The object `opencv_traincascade` can use both Haar and LCB training. To train the classifiers, negative images, or images that don't contain any faces must be provided to the classifier by putting the directory location of each image in a filename. For positive images, or images that contain faces, `opencv_createsamples` is used to create samples. For face detection is imperative to use a large set of positive samples for images. The OpenCV documentation specifies that this amount should be hundreds or even thousands of positive samples. To increase the accuracy of the classifier, all races, age groups, facial expressions, and facial hair should be represented. For this process, a good face database will be used. Fortunately, all of this training only has to be done once: `CascadeClassifier::load` can be used to save the parameters of a classifier to a file, and the trained classifier is automatically saved once finished to the name of the file specified in one of the arguments passed in. To optimize the use of the classifier, the window that the classifier uses to find faces can be changed. Since the classifier finds faces by sliding a two dimensional window over an image, the size of the window chosen greatly affects the computation time and how effective it is at finding faces.

The cascade classifier has the ability to detect more than one face in a frame. This functionality is very valuable because the KEES can include the functionality to notify the user of multiple people that approached the door at a given time. It can also be used for gender classification. Furthermore, the cascade classifier can be used to detect other objects besides faces, such as cars: just simply provide training images of the desired object. This ability adds to the scalability of the image processing abilities of the KEES. If desired, the image processing abilities can be augmented to do more than just recognize faces, and this ability will be considered if there is sufficient time remaining after all of the core features are implemented.

3.10.4.3 - Face Recognition

Once a face in an image is detected, the next step is to determine if the face captured belongs to a person of interest, or a person who is in the database. There are three algorithms provided by OpenCV that can be used to recognize faces: Eigenfaces, Fisherfaces, and Local Binary Patterns. Each algorithm is fairly complex, but will be explained concisely. The OpenCV documentation describes each of these algorithms. Eigenfaces deals with representing a face using a smaller amount of data. Not all pixels of a face are needed since some parts of a face are redundant. Eigenfaces looks for the components of a face that account for most of the information. Principal Component Analysis (PCA) is used on the training images to turn a set of correlated variables into a smaller set of uncorrelated variables. This is accomplished by finding the principal components or directions with the greatest variation which involves matrix and Eigen analysis. This is essentially using only the dimensions that account for most of the meaningful information about a face, greatly reducing the amount of data needed to represent a face without losing much information. A face is recognized by creating principal components to represent all the training face images, which creates a subspace. Then, the target face image is also projected into this subspace. The training image in the subspace that most closely matches the target face image is found. OpenCV provides options to customize the algorithm, such as setting the amount of components to use to represent a face, as well as confidence intervals for prediction. These options will be explored and optimized.

The second algorithm that can be used for face recognition is Fisherfaces. It attempts to account for a shortcoming of PCA: the PCA approach doesn't consider any classes. If variance in the data is caused by an external source, such as lighting, the components identified may not contain any data that is discriminative. Fisherfaces uses Linear Discriminant Analysis (LDA) to undergo class-specific analysis to reduce the dimensions of a face to the most important. It finds and uses the facial features to discriminate between each face, which eliminates the chance of an external source, such as illumination, affecting the results. However, similar to Eigenfaces approximately eight images of each person that is desired to be recognized has to be in the training set.

The third algorithm is Local Binary Pattern Histograms (LBPH). It focuses on extracting local features from images, and summarizing the local features by comparing each pixel to the pixels adjacent to it. If the center pixel's intensity is greater than or equal to its neighbors, it is marked with a one. Otherwise, it is marked with a zero. The end result is a binary number for each pixel. Next, the image is extracted into a certain amount of regions. Then a histogram which captures the amount of pixels that are greater and less than the center pixel is captured from each region. A feature vector for face recognition is then created by concatenating all of the of the local histograms, creating an LBPH. The benefit of LBPH over Eigenfaces and Fisherfaces is that the input image that is

processed for face recognition does not have to be resized unlike the other two algorithms.

All three of these algorithms will be implemented and calibrated to determine which one gives the best results. The FaceRecognizer object is used to implement the three different algorithms. Since the success of these algorithms is heavily dependent on the training images used, great care will be taken to choose a good set of training images. Fortunately, a FaceRecognizer object and its state can be saved to an XML file, and can also be reloaded. This functionality will be useful, as it means the object only has to be filled with the parameters that define how it recognizes faces once. Other factors that need to be considered are preprocessing images. The face captured from the camera will also have to undergo some preprocessing so that its properties, such as illumination, are similar to that of the training images. This is necessary in order to increase the chance of detection as the more similar the target face is to the training images, the greater the chance of accurately performing face detection.

3.10.5 Video Streaming

An objective of KEES is to provide users the ability to view the video that the camera is taking. In order to accomplish this, video streaming software will be needed. There is video streaming software called MJPEG-Streamer that uses Motion JPEG (M-JPEG) as a video format for streaming. M-JPEG is natively supported in Safari, Google Chrome and Firefox. For browsers such as Internet Explorer that don't support M-JPEG natively, external plugins can be installed for support. According to the official website, MJPG-Streamer is an open source command line application that can be used to stream jpeg images over a network so that it can be viewed in a browser such as FireFox. Since there are many browser apps for smart phones, a browser app could be used to view the feed. MJPG Streamer can utilize hardware compression of specific webcams to reduce the amount of CPU time spent compressing the video frame, making it a good choice for embedded applications. The software consists of an input plugin and output plugins. Many different input and output plugins are available, and a developer can write his own plugin to work with MJPG-Streamer. The input plugin copies jpeg images to a global location in memory and informs the output plugin. There is an input plugin called input_uvc.so that comes with the source code. It grabs images from a Linux-UVC V4L2 compatible device. Since the Logitech C300 is a compatible device according to the Linux UVC Drivers and tools website (<http://www.ideasonboard.org/uvcl/>), it can be used with this plugin. The plugin can be used to stream images from the webcam with a resolution up to 960x720 pixels at a high frame rate such as 15 fps without a large strain on the CPU. It can also obtain and compress 1600x1200 uncompressed images, which can then be streamed. The output plugins of the MJPG-Streamer take the images processed by the input plugin and stream them to a source such as an HTTP web page. One output plugin that comes with the software package is output_file.so. It can be used to store jpeg images captured by the input plugin in

a specified directory. It can even be used to forward the images captured to a FTP (File Transfer Protocol) account. Another output plugin that comes with the package is `output_http.so`. It is a functional HTTP 1.0 webserver that can be used to stream a single jpeg image to a specified HTTP website, or can be used to stream many jpeg images. Furthermore, the plugin has multi-threaded ability as many instances of it can be run. The MJPG-Streamer software package even comes with a working example of a website that embeds images and streams captured from the webcam. MJPG-Streamer can be used to fulfill two purposes. One purpose that it can be used for is to feed specified pictures to the server's webpage so that a user can see pictures of people who came to the door. It can also be used to stream a live video feed to the server's website, providing user's the ability to see everything that is happening at the front door.

3.10.6 Embedded Communication

The Keyless Electronic Entry System will require communication between the microcontroller and the Raspberry Pi or BeagleBone subsystem. Since the Camera will be controlled by a different device there must be a way for the two systems to seamlessly communicate with each other. For the projects architecture there will be two main systems the camera system controlled by the Raspberry Pi or BeagleBone as well as the embedded subsystem. The embedded subsystem will consist of the RFID and the Piezoelectric subsystems, as well as the sensor array. Within the embedded environment there will be a need for two lines of communications, one between the RFID and the microcontroller that is selected by the group, and the communication between the microcontroller and the Raspberry Pi. The following paragraphs will discuss possible embedded communications protocols as well as how they are implemented and how they function.

The most predominantly used communications between different embedded systems is serial communications. Serial communications is widely adopted and is an excellent choice if the engineer needs to interface with a device using a personal computer. Most PC's have a serial bus interface that will allow to connect peripherals as well as program and debug embedded devices. Serial is a basic protocol, but is much easier to implement, and can be a perfect choice for communication between embedded devices as long as those devices don't require a very fast connection between each other.

Serial communications also benefit from low pin counts. A serial communication can be implemented with just a single pin if needed, compared to the eight or more needed for parallel communications this could be a good benefit depending on the needs of the architecture. Also many embedded devices commonly support serial communication including the RFID ready that is implemented in the circuit for the Keyless Electronic Entry System. For the KEES project it seems that serial communication will be a valuable tool to understand and to implement for the engineers in the group.

RS-232

RS-232 or TIA/EIA-232-F can be found on almost every personal computer, and is a very common protocol. The RS-232 standard is a well-established standard in which all aspects are specified including the electrical characteristics as well as mechanical and physical characteristics. The hardware connections pin-outs and signal names are also included in this standard. RS-232 is implemented as a point to point interface and can handle speeds up to 115.2Kbps under the right conditions. This standard is capable of full-duplex communication between devices one being the Data terminal equipment and the other being data communication equipment. The personal computer in this setup is typically the data terminal equipment, while the embedded system will be the data communication equipment. If this were to be implemented in the KEES project the Raspberry pi would act as the DTE in this setup.

The transmitters on each side send data by varying the voltage on each of the lines. A voltage that is 3V or above will be a binary zero, and a voltage that is less than a -3V will be a binary one. If the voltage down the line isn't between these values than the data will not be defined. If the architecture that implements the RS-232 uses voltage levels of 0-5V for the logic, such as the atmega that is being considered for the project, can be converted by a conversion IC so that it complies with the standard. For typical RS-232 communication the frames are in the form start bit, data bits, parity bits, and stop bit. On personal computers the typical form is eight data bits, no parity and one stop bit.

Included in the majority of microcontrollers is a Universal Asynchronous Receiver Transmitter, also known as a UART. UART's can be used to communicate with computers and devices using the RS-232 protocol. The architecture is interrupt-driven and can support speeds of up to 115.2Kbps, but this could vary depending on the implementation architecture. This could be a possible choice for communication between the Raspberry Pi and the microcontroller that the group chooses to implement.

RS-422 and RS-485

the RS-422 and RS-485, also known as TIA/EIA-422-B and TIA-EIA-485-A respectively, are balanced twisted-pair interfaces that are capable of much higher speeds than the RS-232 standard. These interfaces can support up to 10 Mbps from distances of as much as 4000 feet. The differential bus use 1.5V and 6V for the logic levels.

RS-422 implements a multi-drop interface meaning that multiple devices can receive the same signal, for up to 10 unit loads. If the devices need to communicate back with the transmitting device the engineer needs to implement a separate bus for this purpose. For the RS-485 it uses a bidirectional bus

between the devices so that communication does not require a separate bus for added functionality. The Keyless Electronic Entry System won't require such high speeds and multi-drop functionality since the communication will predominantly be between just two devices that the communications and logic is in such a way that its not critical to send large amounts of data at high speeds. Many manufacturers add these standards to their microcontroller design so the team will have to make the decision on which implementation to use depending on which microcontroller or manufacturer is selected for the project.

I2C

The Inter-Integrated Circuit bus was developed by Philips Semiconductor. It is a half-duplex, synchronous, multi-master bus that uses two buses: the SDA bus that sends the data, as well as the SCL bus that provides timing for the communication. Both lines require a pull up resistor to operate which is typically above 1.8k on each bus. The lines are controlled by the hardware implementation using open-drain drivers. This standard lets the assigned master to communicate with slave devices using a 7 to 10 bit address that is assigned by the manufacturer of the device.

While the master is communicating with the slave devices, the master device also takes control of the SCL line to provide timing to the other devices on the bus. The master will initiate start and stop messages to the slave devices. In a normal setup of I2C there are typically multiple slaves and one master. Both the master device and the slave devices can transfer data between each other but it will always be the master device that will initiate the transfers. The data transfer speeds on a I2C bus can be one of three speeds, slow being under 100Kbps, fast at 400 Kbps, and high speed that can reach data transfer rates of up to 3.4Mbps. Normally in an I2C implementation the lines will typically be situated on the board but it is possible to run this standard on lines up to 10 feet.

SPI

Serial Peripheral Interface was developed by Motorola and is used on many of their microcontrollers. The SPI bus is a synchronous serial bus that uses four signals the master out slave in, master in slave out, serial clock and active-low slave select. Like the I2C standard SPI is also a multi master/slave protocol. The Master uses the MOSI and MISO lines. The SPI standard operates in full duplex mode, meaning that the data can be transferred and received at the same time on both ends. This functionality increases the speed of this standard. The standard can support speeds between 10 kHz and 100 MHz which is more than fast enough for the Keyless Electronic Entry system.

The typical setup for the SPI standard is between a central processing unit and peripheral device, for the implementation in the KEES project the communication will be setup between the Raspberry Pi and the microcontroller that is chosen.

Depending on the available functionality of the microcontroller, the SPI interface seems to match the specifications and requirements that are needed between the devices.

Microwire

The Microwire standard uses a three wire implementation that uses a clock line to keep timing, this standard was developed by the National Semiconductor and can be seen in many of their COP8 processors. This standard is comparable to the SPI standard and uses a master/slave bus and uses three wires as well. The three wires used in this protocol are the serial data out of the master, and the serial data into the master, and the signal clock to keep timing of all of the devices on the network. Microwire is a full duplex bus meaning that data can be sent and received at the same time by different devices. This standard can support of around 625 Kbps and faster.

Microwire is typically relegated to on board communications but much like SPI it does have capabilities to be implemented over wires of up to 10 feet depending on the capacitance of the wire the standard is being operated on. For communications on wires this long the data rate would have to be significantly reduced for the devices on the bus to be able to communicate reliably with each other. Much like SPI, the Microwire protocol could possibly be implemented into the KEES project depending on the choice of microcontroller that is chosen in the design stage of the project.

Table 16 Embedded Communications Protocols Part 1

Name	Sync /Async	Type	Duplex	Max devices	Max speed (Kbps)	Max distance (Kbps)	Pin count(1)
RS-232	async	peer	full	2	20(2)	30(3)	2(4)
RS-422	async	multi-drop	half	10(5)	10,000	4,000	1(6)
RS-485	async	multi-point	half	32(5)	10,000	4,000	2

Table 17 Embedded Communications Protocols Part 2

Name	Sync /Async	Type	Duplex	Max devices	Max speed (Kbps)	Max distance (Kbps)	Pin count(1)
I ² C	sync	multi-master	half	-7	3,400	<>	2
SPI	sync	multi-master	full	-7	>1,000	<>	3+1(8)
Microwire	sync	master/slave	full	-7	>625	<>	3+1(8)
1-Wire	async	master/slave	half	-7	16	1,000	1s

3.10.7 Voice Recognition & TTS

A type of possible audio addition to the KEES is voice recognition. A user should be able to speak directly to the KEES via voice commands. The command should be received from a user of the KEES and a relative response should be generated and relayed back to the user in a text to speech fashion. This could be useful for adding users to the database, getting information from a user, and for security purposes. Input to the voice recognition/command system will be received via a microphone accessible to users of the KEES. Research must be conducted regarding the type of microphone to use, which hardware (Raspberry Pi, or Arduino) would be best for the task, how to receive and interpret voice commands, and how to relay a message back to the user.

An Arduino would be capable of processing analog speech input. The uSpeech library developed by Arjo Chakravarty provides a software based interface for voice recognition for the Arduino. It features letter based recognition, 80% word based accuracy, and an average response time of 3.2ms. It can be set up by first “Crafting the Vocabulary” and then creating a recognizer. The Arduino speech recognition toolkit is open source and is compatible with most microphones. The developers guide recommends, “a condenser microphone with a pre-amp made using a single transistor”. Although any PC microphone would be sufficient as well or webcam microphone in the Logitech C260/C270 models. Instead of processing the voice using software, the EasyVR Arduino Shield speech recognition hardware module could be used via the UART. It provides commands for basic controls, supports up to 32 pre-defined commands, Arduino libraries, voice passwords (biometric), GUI for voice programming, and sound-playback (PWM audio output to 8 ohm speakers).

The Adafruit “Wave Shield for Arduino” could be used as the speaker for sending a response back to the user. It is capable of playing .wav files of any size (22 KHz, 16-bit) mono output, but is not compatible with the Arduino Mega. The VoiceBox shield is one possible text to speech solution but relies on a SpeakJet

IC voice and sound synthesizer. It doesn't have great feedback on the web but there are tons of resources, guides, and open source examples available. See following figure below for a schematic of the SpeakJet synthesizer chip and next table for the audio amplifier specifications.

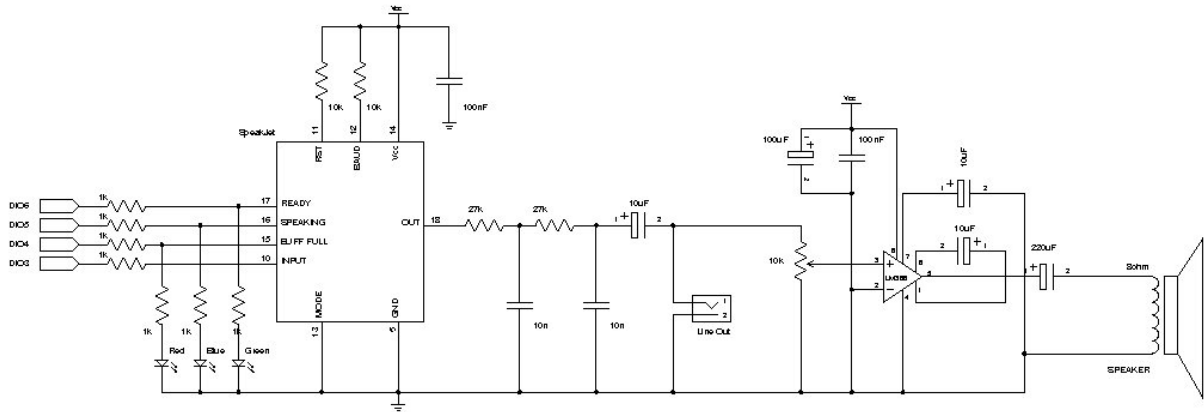


Figure 15 SpeakJet IC Synthesizer

Table 18 SpeakJet IC Amplifier Specifications

1	LM386 audio amplifier IC, DIP-8 package
1	8-pin DIP IC socket
2	10uF electrolytic capacitors (6.3V or greater)
1	100uF electrolytic capacitor (6.3V or greater)
1	220uF electrolytic capacitor (6.3V or greater)
1	1nF (1000pF) ceramic capacitor (marked as "102")
1	100nF monolithic ceramic capacitor (marked as "104")
1	10K trimpot
1	2-pin PCB-mount screw terminal
1	Audio speaker (usually 8 Ohms)

An alternative to the SpeakJet IC is the Emic 2 TTS (Text-To-Speech) hardware module which can be found on SparkFun for \$59.95. The Emic 2 features a Serial TTL interface (9600 bps), on-board 8 ohm speaker, and easy to use ASCII or hex command sequences. It contains all of the logic necessary to parse text and generate natural sounding speech that includes 9 predefined voice styles featuring Perfect Paul, Rough Rita, Uppity Ursula, and Whispering Wendy. According to a SpeakJet vs. Emic 2 review conducted June 25th 2013, the SpeakJet chip sounds very robotic while the Emic 2 provides a more natural sound. See following figure below for a view of the complete circuit.

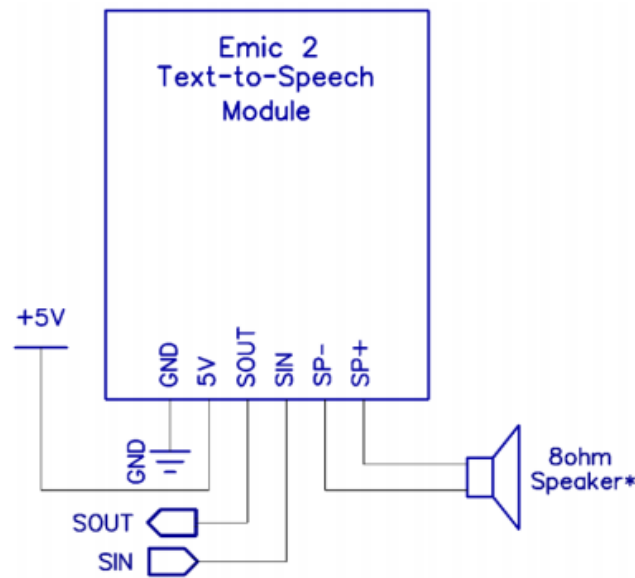


Figure 16 Emic 2 TTS Synthesizer

However based on feedback around the web, this would not be the most ideal way to handle audio synthesis as the reliability is not good and it is noted for being somewhat CPU intensive, which could slow down the processing times for other tasks being handled. For that reason something with a faster CPU would be better for handling audio input. Also considering many webcams available include a built in microphone, it would make sense to handle voice commands from a local KEES user with the same piece of hardware used for controlling the camera.

If a UNIX powered system was used such as a Beaglebone or Raspberry Pi instead, then more powerful software libraries could be utilized instead of writing low level code on the Arduino and requiring additional hardware components. Even a combination of the two devices (Arduino processing, RPi powered) could be used and data could be transmitted via the serial communications port. The accuracy of speech recognition software is heavily dependent on the quality of the microphone being used. Some of the most widely used speech recognition based software includes Dragon Nuance, Julius, and CMU Sphinx.

CMU Sphinx is the most popular open source speech recognition software. Other open source software exists such as OpenVox (related to Julius) but no other has the number of references and tutorials as CMU Sphinx. The CMU Sphinx wiki even includes a guide on increasing performance when running on embedded devices. The two available versions of CMU Sphinx are sphinx4 and pocketsphinx. Sphinx4 is written in java and seems to be more tailored for complex speech recognition systems. Pocketsphinx is written in C and is more light-weight and better suited for simple applications. Pocketsphinx can be used

in the KEES system to process real-time local commands using GStreamer libraries and Python, and integrated into the KEES web page using Web Audio API and a version of Pocketsphinx ported to JavaScript by Sylvain Chevalier. It can also be used for mobile applications in iOS and Android. For a one-hundred word vocabulary, the expected accuracy should be around ninety-nine percent. Even for nearly ten-thousand words, the expected accuracy should be around ninety-five percent.

CMU Sphinx works by first creating a configuration object that is capable of telling pocketsphinx the location of required files. Audio files can then be sent to pocketsphinx from storage or memory where it can be processed. A string representation of the audio will then be returned. Pocketsphinx relies on statistical models when decoding audio. Models are trained on a large amount of data and can be found in many different languages, various acoustic conditions, and different accents. Sphinxtrain is a package that includes training tools to create different models if necessary. Models can be grammar-rule based or probabilistic such as N-gram model.

An N-gram model, models sequences of natural languages using statistical properties. When applied to speech recognition, phonemes (a linguistic unit) and sequences of phonemes are modeled using N-gram distribution. For parsing, words are modeled such that each N-gram is composed of n words. For language identification, sequences of characters are modeled for different languages. For example, the 3-grams that can be generated from “good morning” are “goo”, “ood”, “od”, “d m”, “mo”, “mor”, etc. For sequences of words, a similar 3-gram model can be used but word-based instead of character-based. Primarily N-gram models have been shown to be extremely effective in language modeling.

Similar to CMU Sphinx is Julius, open source speech recognition software that is widely supported in many languages. It is a two-pass large vocabulary continuous speech-recognition (LVCSR) decoder software for developers which is also based on the statistical properties of N-grams. On modern PCs it is able to perform nearly real-time decoding in a 60k word dictation task. Other techniques are also incorporated such as tree lexicon, cross-word context dependency handling, enveloped beam search, Gaussian pruning, Gaussian selection, etc. It is extremely modular and supports shared-state triphones and tied-mixture models. A word dictionary must be provided which is simply a file that contains all the words needing to be identified by the software. A language model and acoustic model must also be provided. The language model supplies the syntax rules of the language and constraints between differences in words, and the acoustic model detects various wave patterns of the audio input. Though Julius is only packaged with Japanese models, VoxForge is a Julius port which includes English acoustic models.

Nuance provides a second option for mobile voice commands. The Nuance Dragon Mobile SDK provides developers with easy to integrate pre-packaged wrappers and widgets to customize any application. The NDEV Mobile developer program, developers are given access to a self-service website and online support forum as well as full code samples including documentation. Once integrated into an application, a 90 day trial period begins for to allow access to the Nuance speech servers. Nuance claims to have the simplest API, small footprint SDK, highly accurate speech recognition results, and minimal integration effort that will enhance the functionality of any mobile application.

Commercial paid voice recognition services claim to use better databases and advanced algorithms that cannot be found in open source solutions such as in the CMUSphinx project. The amount of training data Nuance provides cannot be found in open-source solutions either. The features implemented and data availability in CMUSphinx are much smaller than Nuance. Since Nuance has been around for so long it has had time to collect more data so it's hard for newer projects to acquire the same amount of data. The Nuance Mobile SDK is designed to work under any mobile operating system such as iOS, Android, and Windows. Considering the KEES mobile application will only be used for a prototype and not published to the market, it could be used for free for testing purposes only.

Using the Dragon Nuance SDK seems to be simpler than CMUSphinx. Mainly there are two parts to the SDK which the developer must use. First the recognizer which is used to interpret the audio. A recording is made and transmitted to the Nuance server to be processed and identified. The server can be accessed using an encrypted connection with a unique passcode. The unique passcode is provided upon registering for the NDEV Mobile development program. When the audio has been processed it is returned back to the application. Nuance states that the round-trip average processing time is only one second. Once the authentication has taken place, the audio can be sent to the server for processing.

Programming the recognizer is as simple as creating an object and initializing it with a language type and language model. Upon initializing the recognizer, audio can then be recorded. To stop the recording extra parameters can be given to the initializer so it can be event driven or time based. Developers may want to view the results of the processed audio. Results from the server can be returned to a delegate which is automatically updated once the audio has been synthesized. Error handling can be performed by checking the delegate for any errors found during processing.

Nuance also provides an extensive Text-To-Speech solution in over 30 languages with more than 70 different voices. It includes highly dynamic data capable of transforming any type of content into speech. It brings the interactive experience to another level by having support for animate phrases like

“Welcome!” or “Great!”, and sounds such as crying and laughter. It features user lexicons, prosody controls for modifying the speaking rate or pitch, SSML Support, Mixed Language Support, and a set of dynamic tools to optimize the speech synthesis output for maximum quality.

The Nuance Dragon Mobile SDK may prove itself to be the best option in regards to quality and performance. Nuance provides their cloud services to process the data rather than dedicating that task to the local CPU which could be somewhat intensive at times since it would also be processing data from the camera. Some speech recognition algorithms can be extremely taxing on the processor, and considering the file size of the audio to be processed would be relatively small, it may be faster to transmit the audio over the network to be processed by the Dragon servers.

For mobile app-based voice commands, the Android SpeechRecognizer API should be considered as a viable option. It includes the RecognitionListener interface found inside android.speech API and specific classes such as RecognitionService, RecognizerIntent, RecognizerResultsIntent, and SpeechRecognizer. Implementation of the SpeechRecognizer is meant to be used for streaming audio to the server to perform a certain action, not to be continuously listening for voice input. Once the setRecognitionListener() and startListening() methods are called, a command is given by the user and recorded within the app, which can then be sent to the server to be processed.

An Intent must be created of type RecognizerIntent() with extra parameters defined if needed. Based on the extra parameters, the recognizer prompt is started. The voice is recognized and sent to the server for results to be retrieved. Once a response from the server is received back to the app, it could be processed through the android.speech.tts API and read back to the user using TTS (Text To Speech), also referred to as “Speech Synthesis”. The android TTS API includes native support for multiple languages. Optional parameters can be used to be notified when the audio has finished being played back by using the OnUtteranceCompletedListener() interface. This needs to be done because the speak() method is asynchronous to the text being synthesized and played back.

This way a personalized voice assistant could be created to serve as an interaction link between the KEES system and the user. Responses from the system can then be defined and played in a more conversational manner. The response from the server would then be read back to the user rather than being displayed in a text-based format.

4.0 Design Details

4.1 Risk Assessment

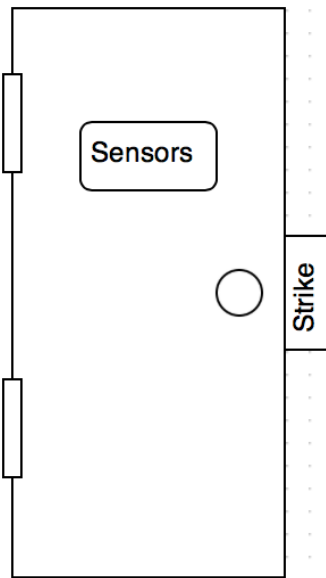
In any project and/or design there will always be risk and security to assess whether it is in the actual design itself, between components of the design or with elements and the environment surrounding the design.

One risk assessment to take into account is where to mount the design relative to the door and frame. Since the design is not wireless, the PCB and the connected sensors will be bound meaning there can be a conflict with the wiring and components depending on the desired and needed placement of the design relative to its environment. It seems there are two different options when choosing where to mount the design, which cause different conflicts; 1) mount the PCB design onto the door itself (first figure) or 2) mount the design on the wall/frame or the static structure adjacent to the door (second figure).

The first way of mounting the PCB means that the entire system will be attached to the door. The RFID reader, camera, photo and PIR sensor will be protruding and/or on the outside of the front side of the door, the piezo sensor will either be placed inside the door or flush against the backside along with the rest of the controlling circuit (PCB and Raspberry Pi etc.). This way has a few potential problems, but the main problem is how to connect the PCB to the strike and how supply the electric strike with 12 volts. The most convenient solution would be to just run wire from the PCB up and around the doorframe to the side of the door to the strike, but using longer connection wire could cause delays and maybe even malfunctions.

The second way of mounting the system would require the RFID reader, camera, photo and PIR sensor be protruding and/or on the outside of the of the wall next to the door. The system would be on the other side of the wall. This seems to be more ideal because the system would be very close to the strike and would fix the problem of the first way but would create another problem. The piezo element would now need to be wired from the PCB to the door. Assuming the element will still need to be on or inside the door, the sensor would have to be wired up and around the doorframe to the side of the door with the hinges because this side opens the least, this could cause delays and malfunctions as well.

Front/Outside View



Back/Inside View

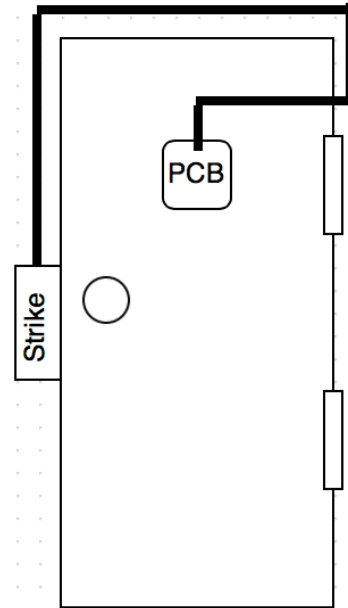
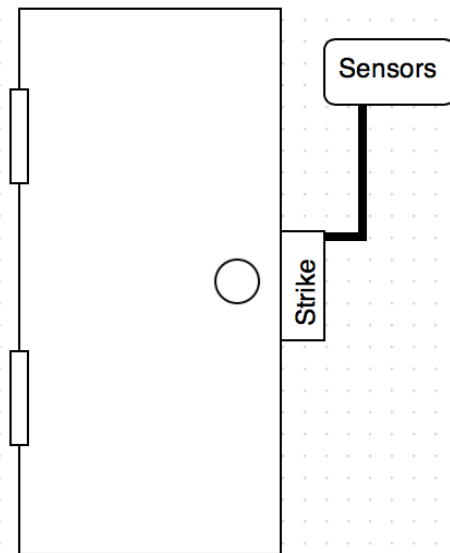


Figure 17 Front and back view of the system mounted on the door

Front/Outside View



Back/Inside View

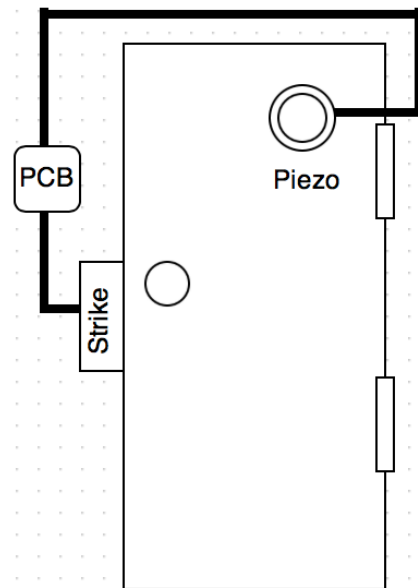


Figure 18 Front and back view of system mounted on the wall

4.2 Hardware Architecture

4.2.1 Power Supply

The power supplied to the design will use a standard AC/DC switched mode power supply. This supply will take an input of 100- 240Vac 50/60Hz and output 12VDC at 500mA. Approx. 120 V ac @ 60Hz. The AC/DC supply will plug directly into an LM7805 voltage regulator to achieve 5 volts to power the microcontroller RFID, photo and motion sensors. The electric strike will need 12 volts to be functional. The TIP31A transistor will trigger the strike enabling 12 volts to be supplied which can be connected directly from the wall or can be connected before the LM7805 regulator. Filtering capacitors will be used to help smooth out any extra ripple coming in from the source. They will be rated at least twice the voltage of what the incoming voltage is, 24 volts or greater. The larger capacitors can also help by holding up the voltage incase of power fluctuations.

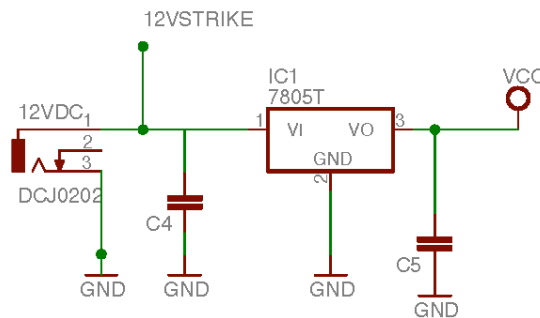


Figure 19 LM 7508 constructed in EgelCad

The LM 7805 voltage regulator circuit was designed and simulated using Multisim (figure 12). The voltage V_{in} was varied from zero to fifteen volts and the output voltage V_{out} was recorded and can be seen in figure 13. The relationship V_{in} vs V_{out} was plotted and can be seen in figure 14. The regulated voltage occurs when the input voltage reaches 7 volts. This means the regulator has a dropout voltage of 2 volts the input voltage must be at least 7 volts in order for the regulator to achieve a regulated output voltage of 5 volts. $7 \text{ volt input} \geq 5 \text{ volt regulated output} + 2 \text{ volt dropout}$. This should not pose a problem because the input voltage being drawn will be approximately 12 volts. The current in I_{in} and the current out I_{out} were also recorded and plotted and can be seen in figures 13 and 15. The power in and power out were calculated and plotted shown in figures 16 and 17.

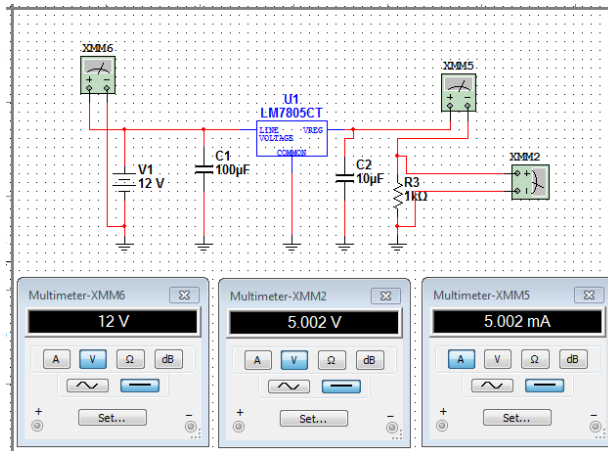


Figure 20 LM 7508Voltage Regulator tested in Multi-sim

Table 19 Voltage Regulator simulated results

Vin (V)	Vout (V)	Iin (mA)	Iout (mA)
0	0	0	0
1	0.0001	2.00E-05	0
2	0.592	5.49E-04	0
3	1.39	3.77E-03	4.00E-06
4	2.54	4.25E-03	1.68E-03
5	3.52	6.70E-03	3.56E-03
6	4.49	9.43E-03	5.45E-03
7	5	1.09E-02	6.47E-03
8	5.001	1.09E-02	6.47E-03
9	5.001	1.10E-02	6.47E-03
10	5.002	1.10E-02	6.47E-03
11	5.002	1.10E-02	6.47E-03
12	5.002	1.10E-02	6.47E-03
13	5.002	1.10E-02	6.47E-03
14	5.002	1.11E-02	6.47E-03
15	5.003	1.11E-02	6.47E-03

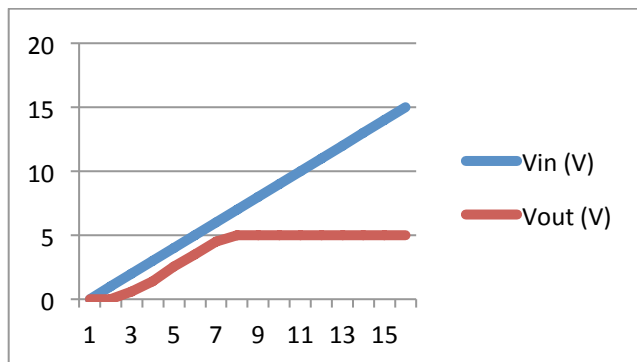


Figure 21 7805 plotted results Vout vs Vin

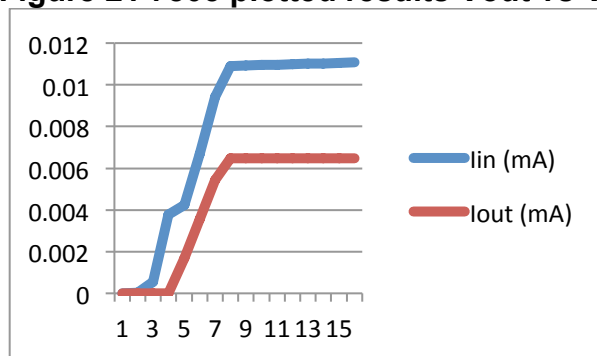


Figure 22 7805 plotted results Iout vs Iin

Table 20 Power In/Power Out values

Vin (V)	Pin (W) = Vin*Iin	Pout (W) = Vout*Iout
0	0	0
1	0.00002	0.00E+00
2	0.0011	0.00E+00
3	0.0113	6.00E-06
4	0.017	4.27E-03
5	0.0335	1.25E-02
6	0.057	2.45E-02
7	0.0763	3.24E-02
8	0.0872	3.24E-02
9	0.099	3.24E-02
10	0.11	3.24E-02
11	0.121	3.24E-02
12	0.132	3.24E-02
13	0.143	3.24E-02
14	0.1554	3.24E-02
15	0.1665	3.24E-02

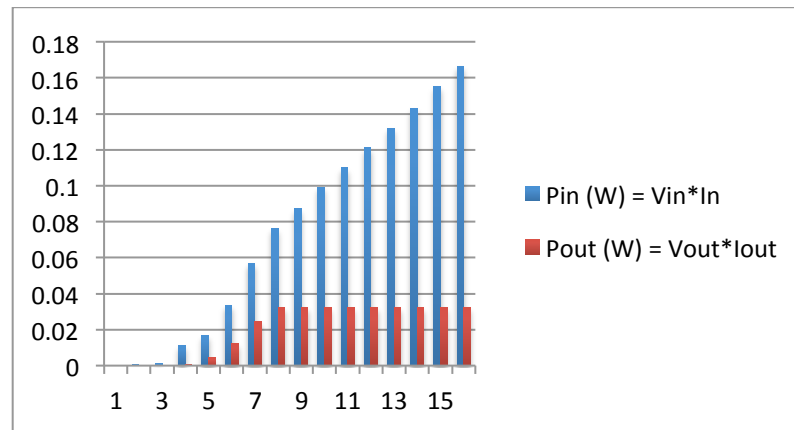


Figure 23 Pin vs. Pout

Vin	Efficiency Pout/Pin
0	0
1	0
2	0
3	0.000531
4	0.251176
5	0.37314
6	0.429825
7	0.42464
8	0.37156
9	0.327273
10	0.294545
11	0.267769
12	0.245455
13	0.226573
14	0.208494
15	0.194595

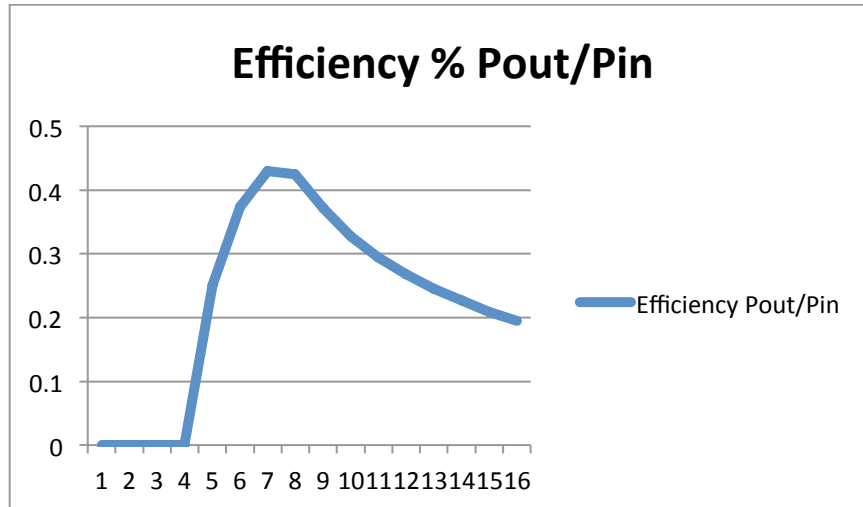


Figure 24 7805 Efficiency (x100%)

The LM22679 regulator circuit was designed using TI's Webench Designer tool. This circuit is intended to take in DC voltages from 4.5 to 42 volts and will deliver a regulated output voltage of 5 volts at 1.2 amperes.

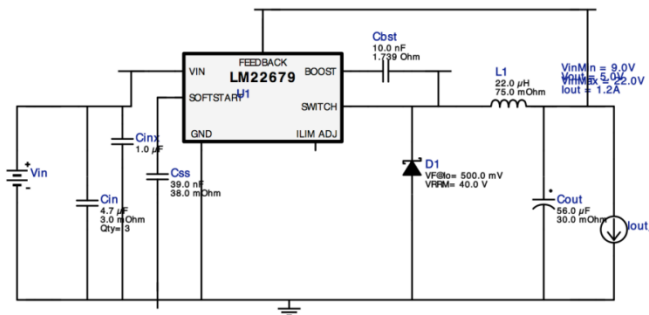


Figure 25 LM22679 Regulator Circuit from Webench

Table 21 Design inputs

Name	Value	Description
lout (max)	1.2 A	Max Output Current
lout1	1.2 A	Output Current 1
Vin (max)	22.0 V	Max Input Voltage
Vin (min)	9.0 V	Min input Voltage
Vout	5.0 V	Output Voltage
Vout1	5.0 V	Output Voltage 1
Base_pn	LM22679	Base Product Number
source	DC	Input Source type
Ta	30.0 degC	Ambient Temp.

Table 22 Operating Values

Name	Value	Category	Description
Cin IRMS	383.749 mA	Current	Input Capacitor ripple current
Cout IRMS	109.361 mA	Current	Output capacitor RMS ripple current
IC Ipk	1.389 A	Current	Peak switch current in IC
Iin Avg	315.84 mA	Current	Avg input current
L Ipp	378.838 mA	Current	P-P inductor ripple current
M1 Irms	594.128 mA	Current	Q lavg
BOM Count	10	General	Total design BOM count
FootPrint	498.0 mm2	General	Total footprint area BOM components
Frequency	500 kHz	General	Switching frequency
IC Tolerance	75.0 mV	General	IC Feedback Tolerance
M Vds Act	62.962 mV	General	Voltage drop across MosFET
Pout	6.0 W	General	Total Output power
Total BOM	\$3.44	General	Total BOM Cost
D1 Tj	41.323 degC	Op_Point	D1 junction temp
Vout OP	5.0 V	Op_Point	Operational Output voltage
Cross Freq.	38.366 kHz	Op_Point	Bode plot crossover frequency
Duty Cycle	24.51%	Op_Point	Duty cycle
Efficiency	86.35%	Op_Point	Steady state efficiency
IC Tj	38.227 degC	Op_Point	IC junction temperature
ICThetaJA	22.0 degC/W	Op_Point	IC junction-ambient thermal resistance
Iout_OP	1.2 A	Op_Point	Iout operating point
Phase Marg.	68.122 deg	Op_Point	Bode plot phase margin
Vin_OP	22.0 V	Op_Point	Vin operating point
Vout p-p	11.49 mV	Op_Point	P-P output ripple voltage
Cin Pd	147.263 μ W	Power	Input capacitor power dissipation
Cout Pd	358.795 μ W	Power	Output capacitor power dissipation
Diode Pd	452.922 mW	Power	Diode power dissipation
IC Pd	376.208 mW	Power	IC power dissipation
L Pd	118.8 mW	Power	Inductor power dissipation
Total Pd	948.468 mW	Power	Total power dissipation

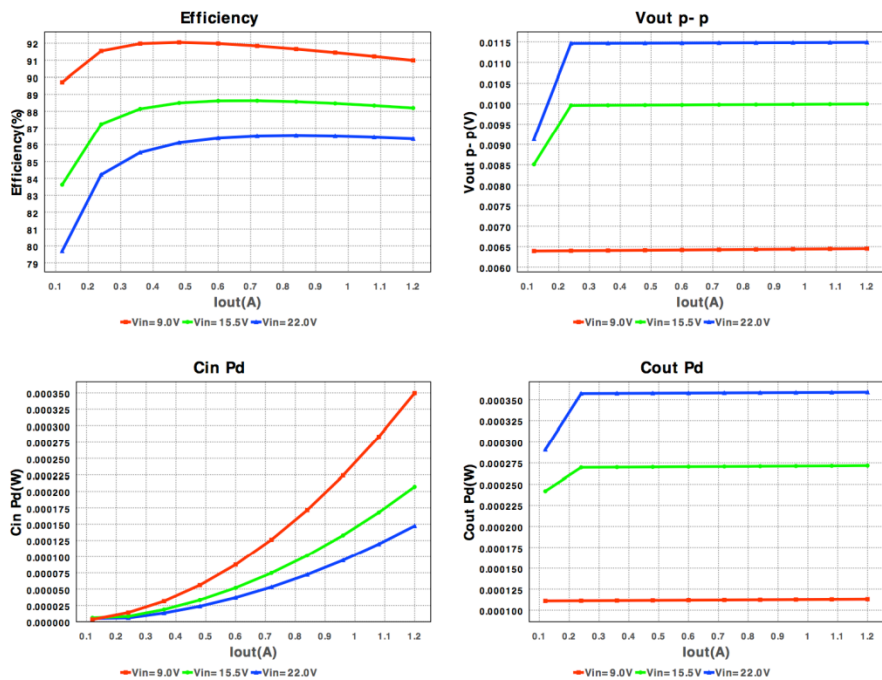


Figure 26 LM22679 First Graphical analysis

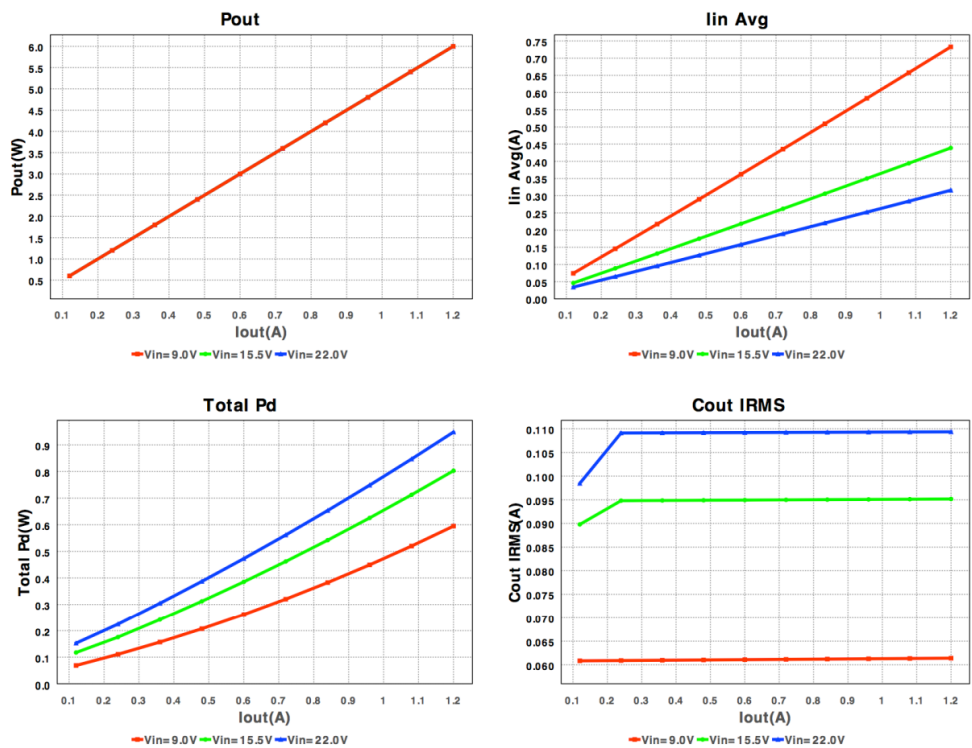


Figure 27 LM22679 Second Graphical analysis

4.2.2 RFID Hardware Architecture

The RFID Hardware Architecture scheme was developed to allow the RFID to communicate with the micro controller to unlock the door if a valid identification was scanned. During the research phase the team had to decide on what RFID reader was going to be implemented into the project. The choices that were up for debate were the Parallax RFID reader module, as well as the Low Voltage Series Reader Modules designed by ID innovations. For the RFID reader the team saw the clear choice was the ID12-LA. While the Parallax RFID reader met all of the specifications required for the implementation of the project, the shorter range of 1.75 inches of the Parallax reader was not sufficient compared to the ID12-LA which can support up to 5 inches in range. The small form factor also aided in the choice of the ID12-LA as well.

As for the microcontroller, the team decided to use the Atmega 168 for the embedded PCB implementation of the KEES project. This microcontroller has more than enough pins to support all of the subsystems of the project. In general the RFID once hooked up to a 5V power supply will automatically be receptive of any RFID card that falls within its range. After stimulus has been applied from an in range card read the RFID reader then will send serial data down the D0 pin in the form of the ASCII representation of the cards ID. The serial port operates at 9600 baud rate with no parity and 1 stop bit, under this setup the Atmega will take in the parsed identification and check whether the scanned ID is allowed access to the door. If the identification is allowed access a signal will be sent to the door lock to unlock it for a specified amount of time that the team chooses.

The Atmega 168 doesn't have an internal clock so it is necessary for the design of the RFID subsystem to include a crystal to provide timing for the microcontroller. For the timing of the microcontroller the team selected a 16 MHz crystal, this crystal will allow the Atmega to interface with other devices as well as provide timing to other devices if needed. The 16 MHz crystal will be tied to pin 7 and 8 on the Atmega 168, these pins are designed to provide timing to the microcontroller. Also in between the crystal and ground the team chose to use two 22pF capacitors. These capacitors will remove the DC ripple effect and allow for a clean signal on the pins.

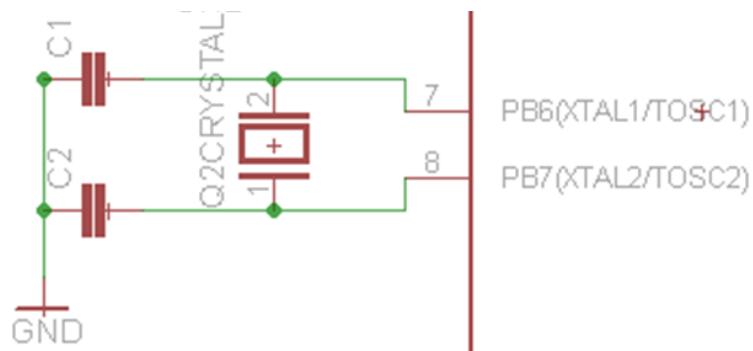


Figure 28 Crystal Schematic

Once an RFID card is scanned the data will be sent down the D0 pin or pin 9 in the schematic. The data sent is the ASCII representation of the ID number that is on the card. This information needs to be processed by the microcontroller to determine whether or not the card is valid, so the door strike state can be changed or stay the same. For this implementation pin 9 from the RFID is tied to pin 30 of the Atmega, which is capable of receiving serial input. Once this input is received the microcontroller will complete the logic for this transaction, and determine the next state of the system. This function of the RFID sub system will be covered in greater length and depth in the software section.

Another one of the useful pins in the RFID is the LED pin. This pin is designed to flash an LED every time that a card is scanned by the reader. Regardless of whether the card is valid for this system the LED will give a flash notification showing that there was a card within range and an identification number was picked up. For the design of the Keyless Electronic Entry System the team decided that this is a useful function but that it's not enough information to tell the user if the transaction will change the state of the lock or not.

As shown in the following figure the team exploited the LED logic by using it to control a transistor used as a switch. Every time a card is scanned the LED signal is activated the Buzzer and the RGB LED will be activated. The color of the LED in this setup will be controlled by the micro controller. The Atmega 168 will control the color by regulating the duty cycle of the PWM pins. The three states will include access denied state, access granted state, and the programming mode state. This functionality as well as an in depth explanation of the RFID subsystem states will be explained with more detail in the software section of this document.

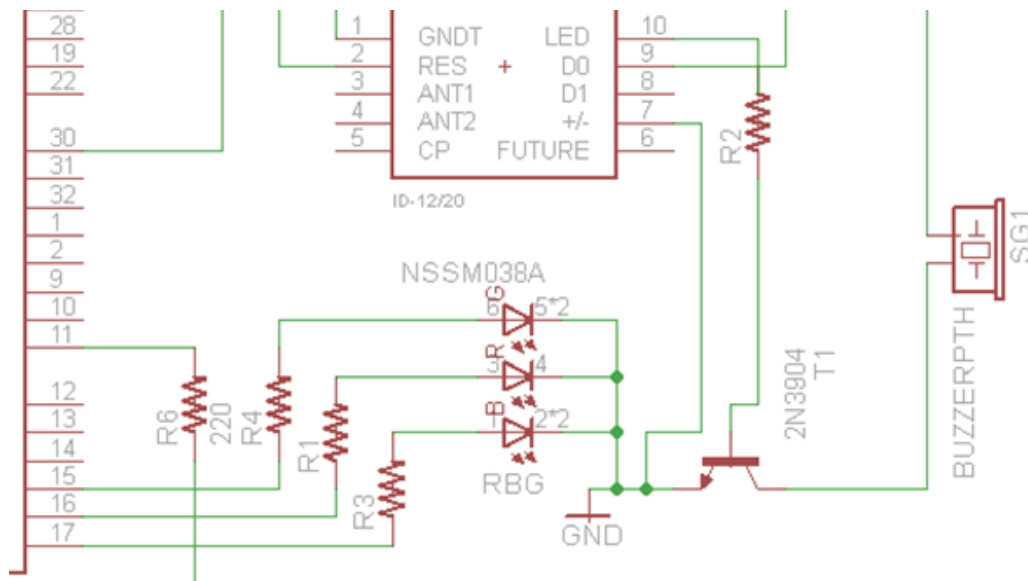


Figure 29 RFID Subsystem Schematic

The following figure depicts the full RFID subsystem; the elements that don't pertain to the RFID side of the project have been taken out for enhanced clarity. This system support three general states that restrict or allow the lock to be accessed as well as allowing new cards to be added. The first state is the access allowed state, this allows the microcontroller to activate the door strike and allow entry through the door and the notification LED turns to green. The second state is the access denied state, when the RFID, in this state the microcontroller doesn't change the state of the door strike, and changes the color of the notification LED to red. The last state is the programming state, once the RFID reader recognizes that the master card has been scanned the microcontroller will allow the next scanned card access to the system. This allows the user that has control of the master keycard to allow access to other key cards for convenience. Once the microcontroller places the system in this state the LED will flash between blue and red colors notifying the user that the system is in the programming state and that the next card to be scanned will be allowed access to the system in the future.

4.2.3 Sensor Array System

The sensor array consists of the piezo sensor, PIR motion sensor, RFID reader and a dark activated relay/LDR.

PIR (Parallax 555-28027)

- Source current up to 12 mA @ 3V, 23 mA @ 5V
- Voltage requirements: 3 to 6 VDC
- Current requirements: 130 uA idle, 3 mA active (no load)
- Communication: Single bit high/low output
- Dimensions: 1.41 x 1.0 x 0.8 in (35.8 x 25.4 x 20.3 cm)
- Operating temp range: 32 to 122 °F (0 to 50 °C)

RFID (ID-12LA Low Voltage Series)

- 125 kHz nominal
- Card format: EM 4001 or compatible
- Power Requirements: +2.8 VDC thru +5 VDC @ 35mA
- Max VCC: 5.5 volt
- Max Current from Antenna +/- 75mA
- RF I/O output current: +/- 200mA PKPK
- Card and Data current drawn: +/- 5mA

Piezo Sensor

- Resonant Frequency: 6.3 +/- 0.6 kHz

Logitech C300 Webcam

- Connectivity and Power: USB

Parallax Single Relay-27115

- Power Requirements: 5 VDC @ ~85 mA (Relay Power), 3.3-5 VDC (input signal)
- Communication Interface: Logic High/Low 3.3-5V DC
- Operating temperature: -13 to +158 °F (-25 to +70 °C)
- Dimensions: 1.57 x 1.06 x 0.71 in (4.0 x 2.7 x 1.8 cm)
- Max Switching Current 10 A @ 250 VAC / 30 VDC

LDR VT-800 Series

- Temperature Range: -40/+75 °C
- Continuous Power Dissipation: 175mW , 3.5 mW/°C

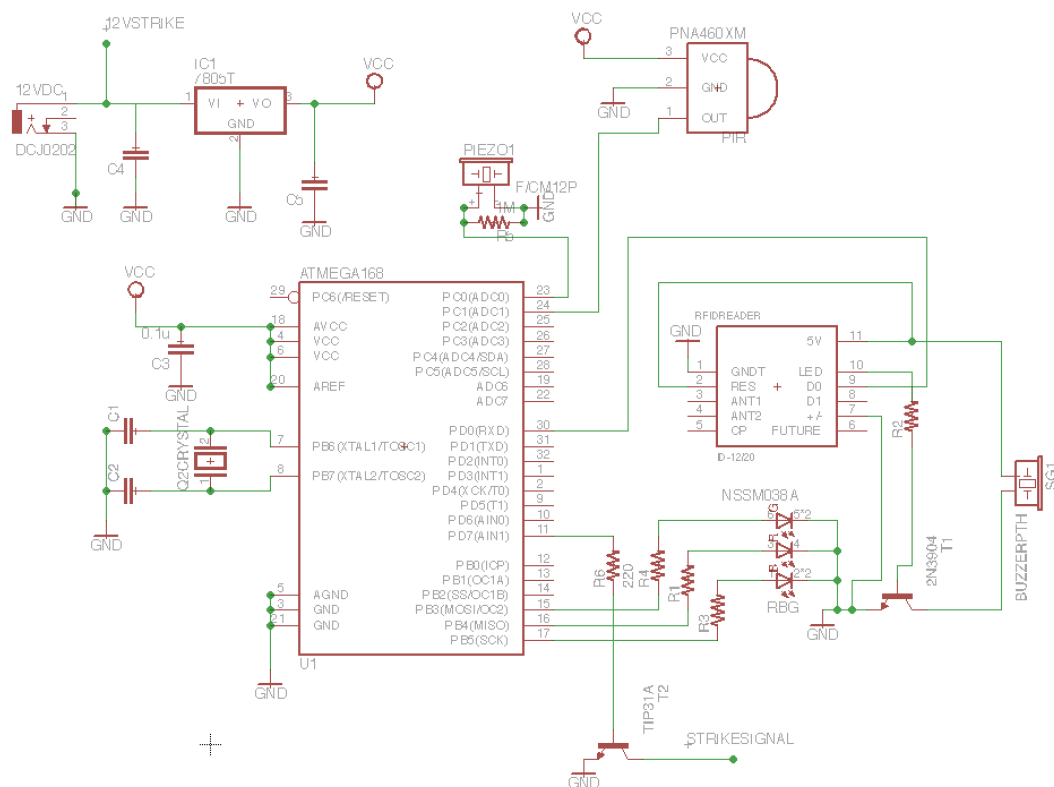


Figure 30 Power Supply Regulator w/ Microcontroller and Sensors schematic

4.2.4 Lock

The schematics below are a simulation and a representation of how the electronic strike will operate. These numerical values are not the exact values that will be present in the actual prototype but will model and show us how the strike will behave. The switch in the circuit represents the microcontroller turning on or off an output which signals the electric strike to open or close. The npn

transistor in the circuit will act as an actual analog switch in the circuit itself. From the current relationships and operating conditions of the transistor when there is no voltage and no current in the base-emitter of the transistor, $V_b < 0.7$ (turn-on voltage) the transistor acts as an open switch. When voltage is applied ($V_b > 0.7$) and current is allowed to flow the transistor acts as a closed switch allowing the collector current (I_c) to flow, opening the strike. Since this is a fail secure type strike the strike will be closed until the controller sends some current through the strike allowing it to release and open for a brief period of time. In figure # (second picture) the switch is closed in the simulation meaning that the microcontroller has sent a signal to the output pin allowing a small current to flow through the strike briefly opening the door. In next figure (first picture) the switch is open meaning the strike is closed. The diode in the schematic is a protection diode, which protects the strike and the transistor from transient voltages. Current flowing through the strike/coil in theory can create a magnetic field, which collapses suddenly when the current is switched off. The sudden collapse of the magnetic field induces a brief high voltage across the strike/coil and tries to keep the current flowing in the same direction as it was when it was switched on, which is very likely to damage transistors and other integrated components. The protection diode allows the induced voltage to drive a brief current through the coil/strike and diode so the magnetic field diminishes away quickly but not instantly. The diode provides a path for current back through the diode. This prevents the induced voltage becoming high enough to cause damage to transistors and other integrated components. The tiny current (μA) still flowing even when the switch is open will not be enough to drive the strike assuming the strike will take around approximately 100~200 mA.

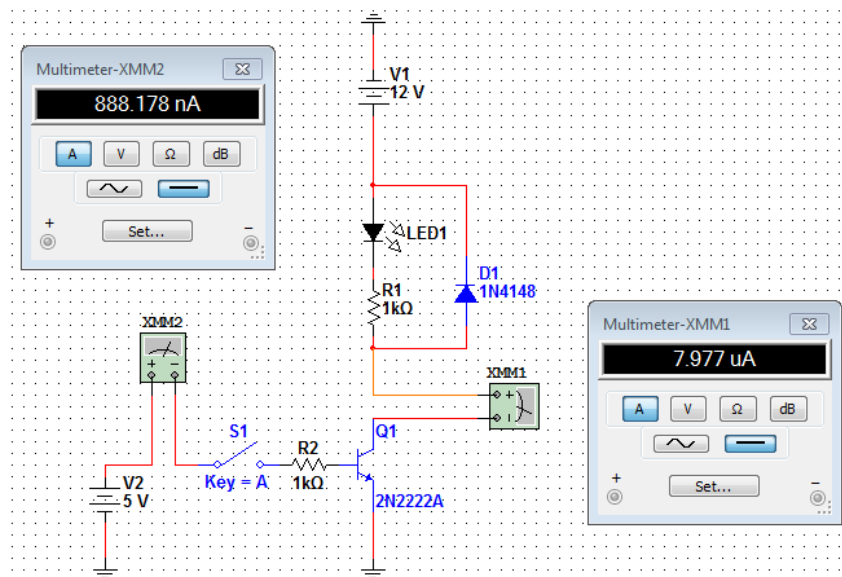


Figure 31 Strike Open Switch State

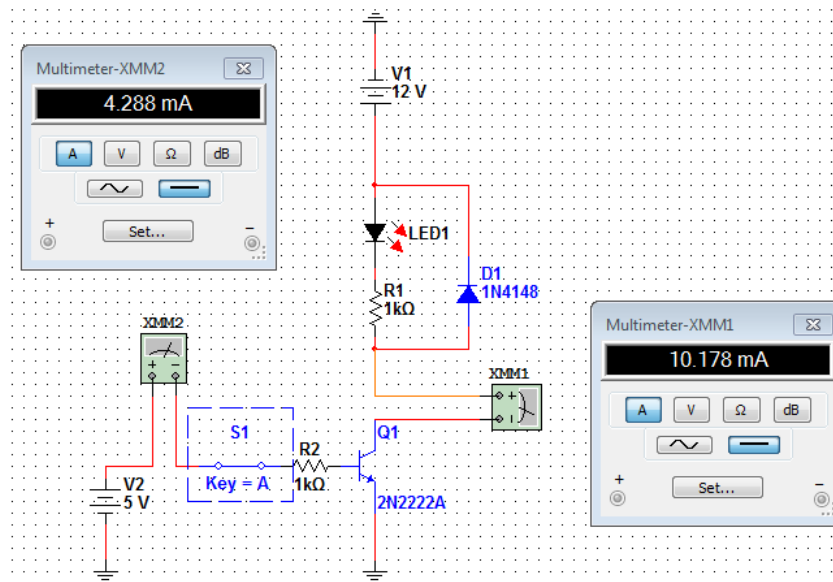


Figure 32 Strike Closed Switch State

4.3 Software Architecture

4.3.1 App Software

High Level Description and Design: The Android app is a very integral component of KEES. It will enable the user to communicate directly with the system for supplying commands, querying the status of the system, and even providing input such as pictures of people to add to the database. As a result, the app must have certain security features in case to prevent it from being used by someone else other than the user. To facilitate this, upon installation the app will request the user to create a username and password that will be saved for future use. This can be easily accomplished as there is a function that can be called when the app is opened for the first time. To further facilitate security, the login credentials will be encrypted when stored in the file. In the event that the owner of the phone loses his phone, another person will be unable to discover the login credentials by simply plugging the phone into a computer to view the login text file. The user will also be forced to use a password of at least 10 characters, with at least one number, uppercase letter, and one symbol. The user will also have to change the password every month. Furthermore, the app will lock after a specified amount of idle time has passed to avoid another person from gaining access to the system if the person leaves their phone unattended.

There will also be two variants of the app: an administrative version and a user version. The administrative version will have the ability to change the settings of the system, such as the faces of people to be added to the database, adding another tag to be recognized to the FRID system on the Atmega (by instructing the Raspberry Pi to set the Atmega to programming mode), as well as assigning songs to play when certain people approach the door. The user version will only have the ability to receive notifications about the system, and to unlock the

door. Having two versions of the app is very applicable: in a family the parents would want to have administrative rights to the system so that they could change settings, while the children would only require user access.

The user interface will be slight different for the administrative and user version of the app due to the features previously stated. However, both user interfaces will share certain functions. A diagram of how the GUI interface for the administrative version of the app will look like is shown below in the following figure.

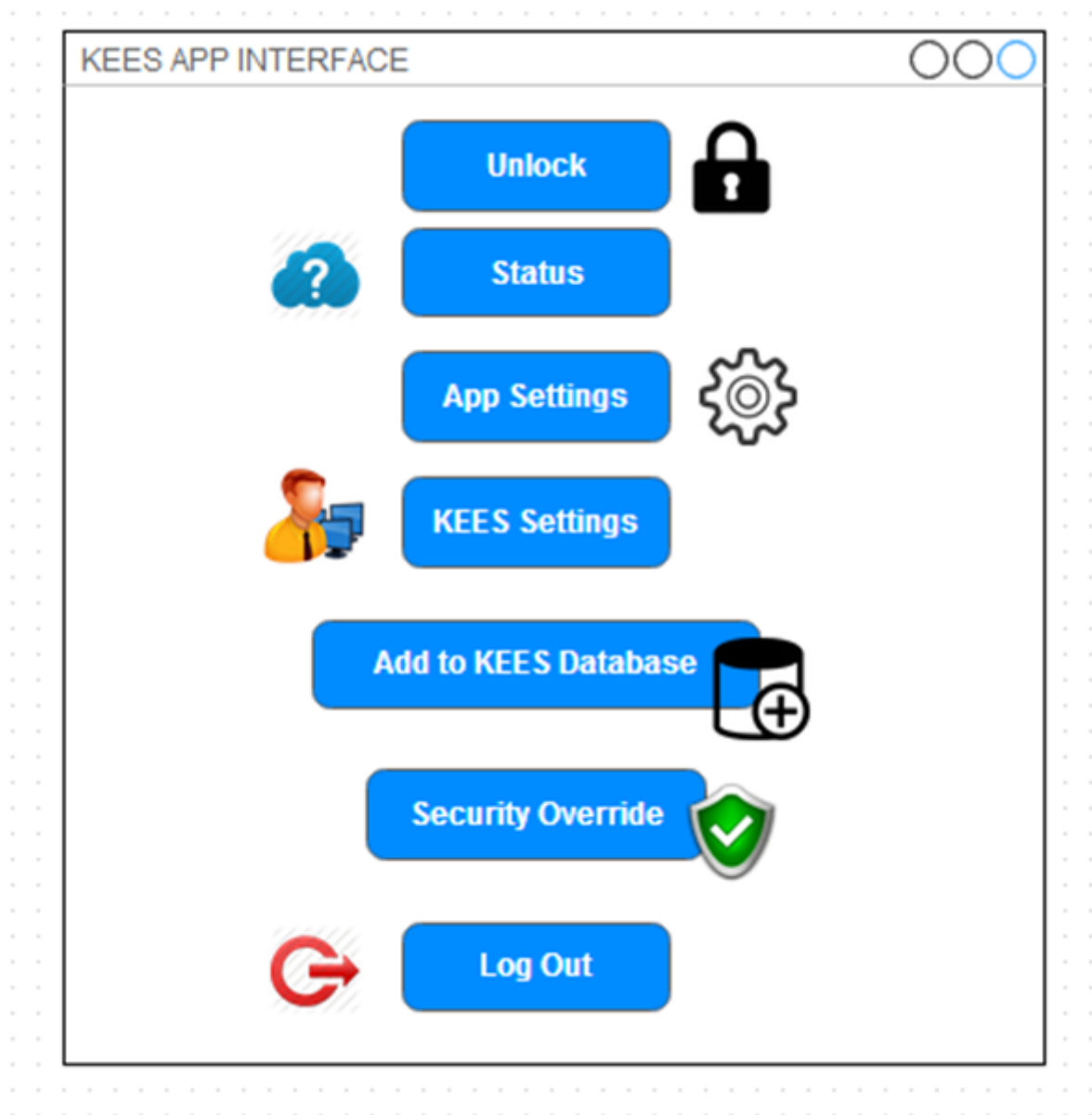


Figure 33 KEES APP GUI

Both the administrative and user version of the app will be able to open the server's status page via the "Status" button that displays a record of who has visited (if the person is in the database), the time, and a picture of the person's face. Both users will also have access to the "Unlock", "App Settings", and "Log Out" Buttons. The "App Settings" button will enable the user to change how often the app communicates with the server to obtain status alerts that manifest in the form of notifications. It will also allow the user to specify the amount of idle time that should pass before the app locks itself. The user will also be able to specify how the app's notifications will manifest, including whether or not to display an icon in the taskbar, as well as the sound that the notification makes. "KEES Settings", "Add to KEES Database" and "Security Override" buttons will only be available on the administrative version of the app. The "KEES Settings" button will allow the user to change the aforementioned settings on the KEES system. The "Add to KEES Database" will enable the user to add a person's face, name, and whether or not to unlock the door upon recognizing their face to the KEES database. The "Security Override" button is a very advanced feature that can be used to ensure maximum security if necessary. This button will enable the user to render the apps of non-administrative users unusable in the sense that they will be unable to access KEES at all via the "Lock" and "Status Page". Upon pressing the button, the user will be forced to enter his or her login credentials. The low level details of this will be discussed in the Class Diagram section. The motive behind this is that in the event that someone's app was compromised, the administrative user could execute this command and select which of the user's apps to disable. It could also be used in the event that a person's phone was lost to eliminate any chance of an outsider using the app to unlock the door, and gain unauthorized access to the system.

State Diagrams: Since the app has a GUI interface, it has many states that stem from pressing its buttons. A comprehensive state diagram of the app is shown below in following figures. The first diagram displays the login state, the second displays the state diagrams for "Unlock", "App Settings" and "Status". The third diagram shows the state for the remaining buttons on the App's main page.

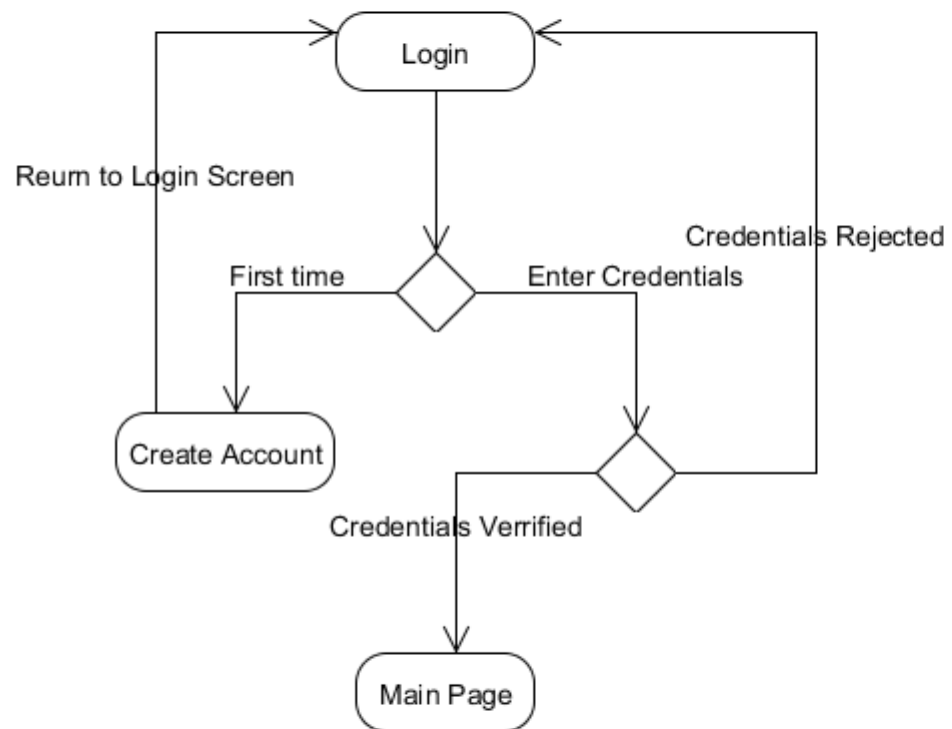


Figure 34 KEES App State Diagram: Login

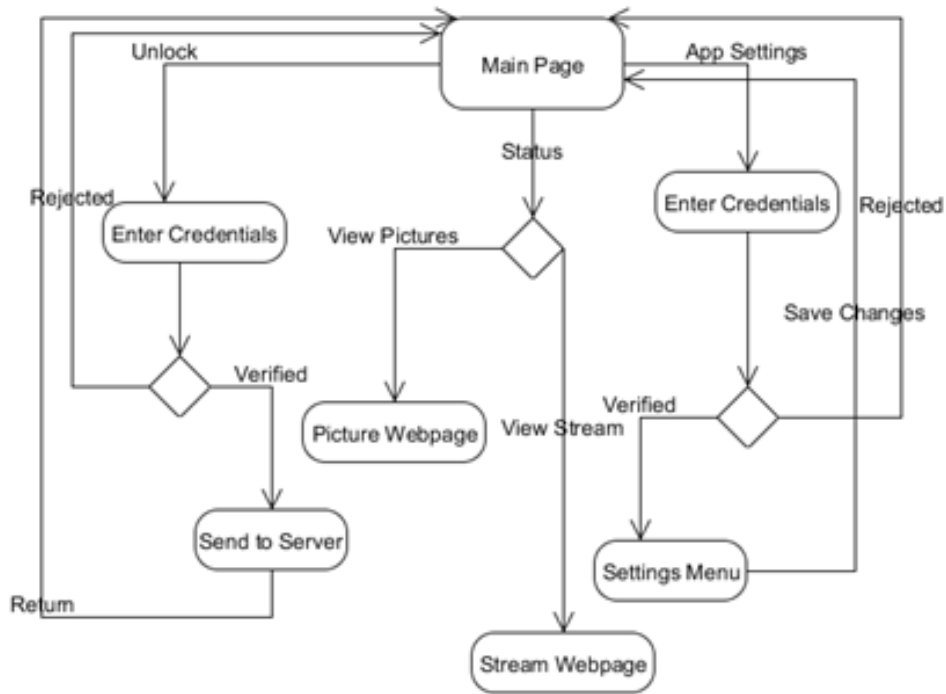


Figure 35 KEES App State Diagram 1



Figure 36 KEES App State Diagram 2

As depicted in the state diagrams, login credentials were required to modify KEES' settings, unlock, security override, and to modify the app's settings. By default, executing these commands will require the user to enter credentials, but this can be changed via the app's settings if desired.

Class Diagram: The app's design will consist of six classes. The class Main will be responsible for displaying the main page of the app. In Android development, each main process or user interface page is a class. A class diagram is shown below in following figure.

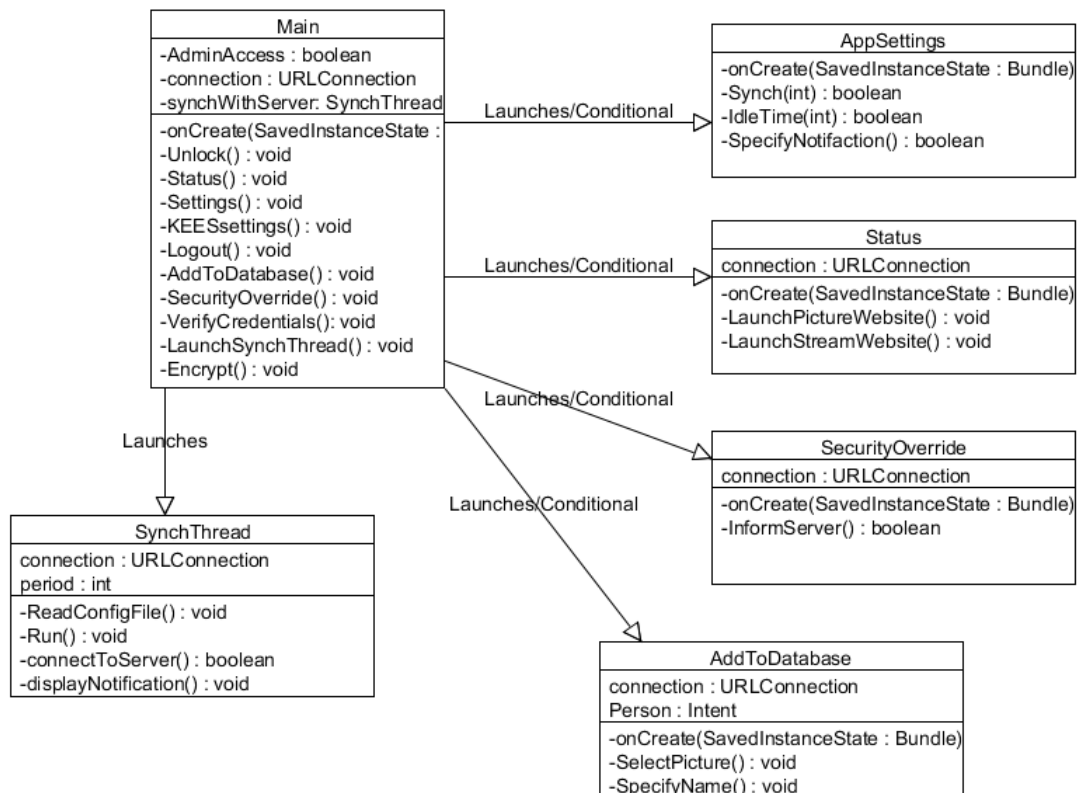


Figure 37 KEES App Class Diagram

The onCreate function is used by the classes to display UI elements that are specified in XML files. It also serves as “main” in that the function is called whenever the user interface page is loaded. In Android objects called Intent are used to launch a new UI page. The main class launches the UI page for the “Add to KEES Database”, “Security Override”, “Status”, and “App Settings”, respectively. The Launches/Conditional arrow specifies that the respective UI page is only launched when the button is pressed on the main page. The only exception is SynchThread class. Main automatically creates a SynchThread object which is an object that extends Java’s Thread class. This object is a thread that will periodically connect to the KEES server to receive notifications. It

will only run after the specified period obtained from the “appSettingsConfig.xml” file has expired, and will suspend otherwise. There isn’t a separate UI for KEES Settings as the server’s website is simply launched after verifying that the user is an administrator.

The login credentials will be saved in a text file named “login.txt” in the phone’s internal storage in a region that is only accessible to the KEES app, and will be parsed using Java’s String.Split() function. To further facilitate security, the login credentials will be encrypted when stored in the file using a hashing function. In the event that the owner of the phone loses his phone, another person will be unable to discover the login credentials by simply plugging the phone into a computer to view the login text file. The SecurityOverride class will connect to the server to return a list of all the users that have an app. This list will be displayed to the user in list format, and the user will be able to select which of the apps to disable. The app to disable will be sent to the server via a HTTP request, and the server will reject any requests sent by that app.

4.3.2 Web Server & Database

The web server will follow an MVC (Model-View-Controller) architecture pattern. The MVC architectural methodology separates the representation of information from the user’s interaction with the information. Web applications are composed of Models (files that contain a description of the data representation), Views (files that contain a description of the data representation), and Controllers (files that contain a description of the workflow). Controllers send commands to the model which update the model’s current state. Commands can also be sent to change the view representation. The Model notifies its related views and controllers when a state change has been made. The notifications sent by the Model enable the views to product an updated output, and controllers to change the available commands. The View simply requests information from the Model required for creating an output presented to the user. See the next figure below for a visual representation of how the components of the MVC architecture are related.

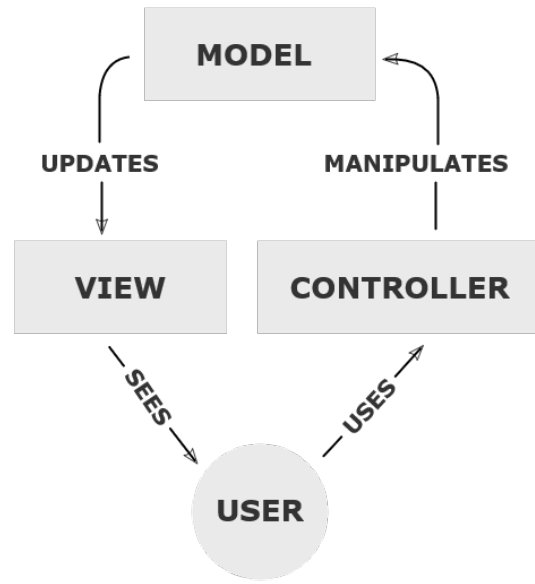


Figure 38 MVC Architecture Component Relationships

Using web2py as a framework will enable to use of an MVC based architecture. Web2py is based on Python making it faster and more scalable by providing a comprehensive web-based administrative interface and included libraries. It is compatible with lighthttpd and nginx web servers, databases such as SQLite, and MySQL, and many protocols such as HTML/XML, JSON, and REST. The MVC architecture is satisfied through the Controllers which consist of functions associated to a URL which is called when the URL is visited, Models which are executed independently before the function is called, and Views which are called when the function returns data and renders the data in the proper format. Web2py is less verbose and its syntax is much cleaner than PHP based frameworks thus making web development simple, easier to read, and easier to maintain.

The web2py web framework includes all the necessary features required for the backend database and web services. Services can be created and written in Python, and executed directly via a specific URL, or called within a webpage upon interaction by the user. Database connections and access methods can be written without hassle thanks to the DB-API provided with the web2py framework. The frontend UI which will be presented to the user via a web browser can be dynamically generated and formatted based on the type of device the user is on by using powerful web languages such as HTML5, CSS3, and jQuery. Services will be created to display and interpret information from the backend such as the live video stream, notifications, and administrative functions.

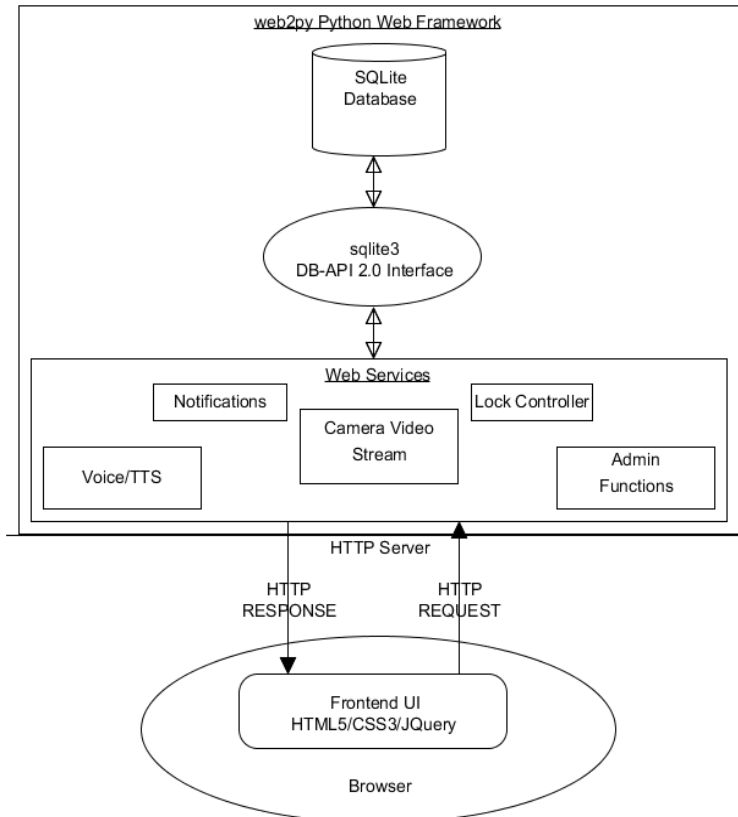


Figure 39 KEES Web Server Architecture

4.3.3 RFID Software

The logic for the RFID subsystem will be largely controlled by the Atmega 168. When power is applied to the RFID reader, the reader automatically is scanning for input from an RFID card. Once an RFID card is in range of the reader, the reader's transceiver will pick up the signal from the cards antenna and receive the data information from the card. This data which can be seen in the following figure, is sent down the serial out pin of the RFID reader, which is tied to the Atmega's receive pin. This data once received will be interpreted by the microcontroller against saved RFID's to verify whether or not the card will be allowed entry through the door.

Output Data Structure - ASCII - 9600 Baud, No Parity, 1 stop bit.

Output = CMOS (Push Pull) 0-Vdd

STX (02h)	DATA (10 ASCII)	CHECK SUM (2 ASCII)	CR	LF	ETX (03h)
-----------	-----------------	---------------------	----	----	-----------

Figure 40 RFID Data Encoding

The microcontroller will have the functionality to put the RFID system in three general states. The first state is the access allowed state, this allows the microcontroller to activate the door strike and allow entry through the door, after an allotted amount of time for the user to gain entry and closes the door, the system goes back into the access denied state. The second state is the access denied state, which is the state in which the RFID subsystem is scanning for new RFID cards as well as putting the microcontroller in a wait state. When the RFID is in this state the microcontroller doesn't change the state of the door strike. The last state is the programming state, once the RFID reader recognizes that the master card has been scanned the microcontroller will allow the next scanned card access to the system. This allows the user that has control of the master keycard to allow access to other keycards for convenience. Also functionality will be added to delete a card's access from the system if the card that is scanned after the system is in programming mode is in the system, then the card will be deleted from the microcontroller's memory and be an invalid RFID card. The following state diagram displays the state changes as the RFID system moves from one state to the next.

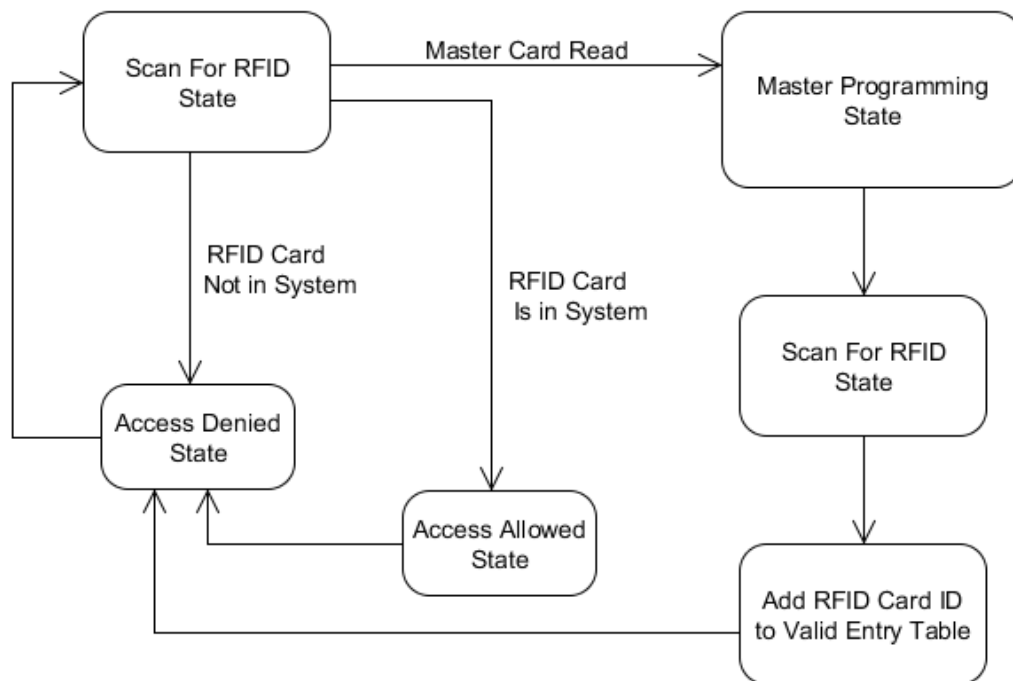


Figure 41 RFID State Diagram

The RFID system will be capable of notifying the user of what state it is in. The RGB LED tied to the pins of the microcontroller will be able to notify the user if the state has changed by changing between three distinct colors. To signify the user that access is allowed and the door is unlocked from a valid RFID key card entry, the system will display a green light from the LED. This light will only be active for the amount of time that the door is in the unlocked position. Once the

door transitions to lock the LED turns off. The next state that the LED displays is the access denied state. When the RFID system verifies that the RFID card that was scanned is an invalid card entry, the RGB LED will turn red to notify the user that the card is invalid and will be needed to be programmed into the system to gain access. The LED will show this state for approximately three seconds then return to the previous state. Once the RFID system verifies that the scanned RFID is the master card, which is programmed into the microcontrollers EEPROM, the system will flash in between red and blue. The red and blue LED flashing light will notify the user that they are in programming mode and that the next card that will be swiped by the system will have their RFID access toggled. The following figure displays the different states that the system will transition through.

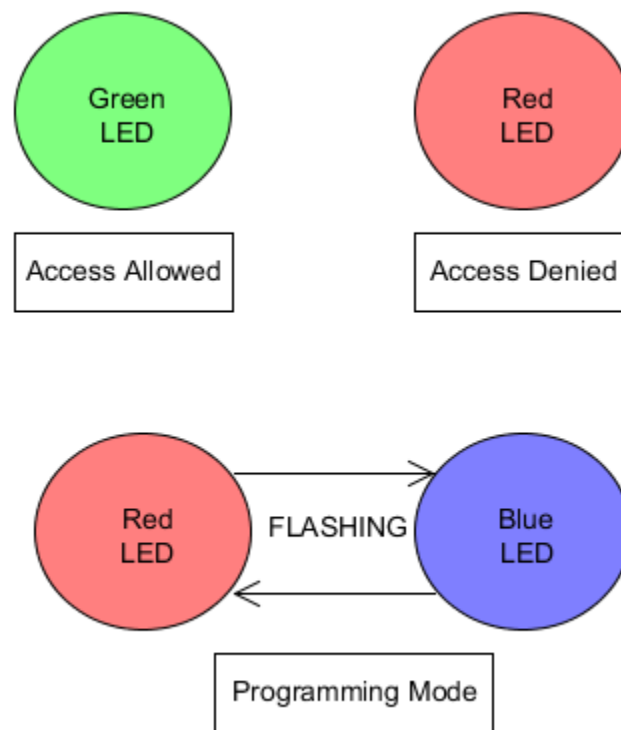


Figure 42 RFID Notification State Diagram

4.3.4 Piezoelectric Sensor Software

The piezoelectric knock sensor will be utilized to unlock the door if a user's knock pattern is the same as the knock pattern that is saved to the microcontroller. This subsystem will be capable of unlocking the door as well as have the capability to change the saved knock pattern. The piezo element can accomplish this task by picking up the vibrations of the knocks on the door's surface. Once the knocks are sensed, the microcontroller will take care of the logic necessary to verify whether the entered knock is valid, or if the knock is invalid. The user will be able to tell if the knock is invalid or valid from the notification rendered by the LED.

This section will further explain the programming architecture and design implemented by the piezoelectric sensor subsystem.

There will be a few states that the microcontroller will transition through in the piezoelectric sensor subsystem. The initial state that the microcontroller will be in is the Waiting for Knock State. While the microcontroller is in the Waiting for Knock State, the pin that the piezoelectric sensor is will constantly being scanned. The input pin will scan the pin and match the voltage input with a specified threshold. If this threshold is exceeded, this signals to the microcontroller that a knock has been received. Once the knock is received from the input pin the microcontroller will transition into the next state.

The next state that the microcontroller will enter is the Storing Knock Parameters state. While in this state the user will be initiating the knock pattern in an attempt to match the secret knock pattern to gain entrance. During this period every knocks timing will be stored into an array, not to exceed a maximum of twenty knocks. For example if after the knock is completed the knock pattern array stored 5 elements as follows: 25, 60, 25 90, and 0. The meaning of these numbers is as follows initial knock entered 25ms until second knock, 60ms time elapsed until third knock, 25ms time elapsed until the fourth knock ...etc. The timing of the knocks is what will be of interest when the secret knock pattern is matched against the users knock pattern. After the users knock pattern is determined and stored in the Storing Knock Parameters State, the piezoelectric subsystem will transition into the following state.

Once the users knock pattern is determined the system will transition to the Validating Knock Pattern State. While in this state the users knock will be matched against the current secret knock, and it will be determined if the users knock is valid or invalid. The way in which the validation process works is each knock elapsed time will be compared to the time of the secret knock. If all of the knocks are within a given threshold the knock will be validated, if the knocks are off of the given threshold then the knock will be determined as invalid. For an invalid knock the user will be denied access to unlock the door and the system will transition back to the Waiting for Knock state. If it is determined that the users knock matches the secret knock then the door will be unlocked for a specified amount time the user needs to enter the door, and then the system will transition to the Waiting for Knock State

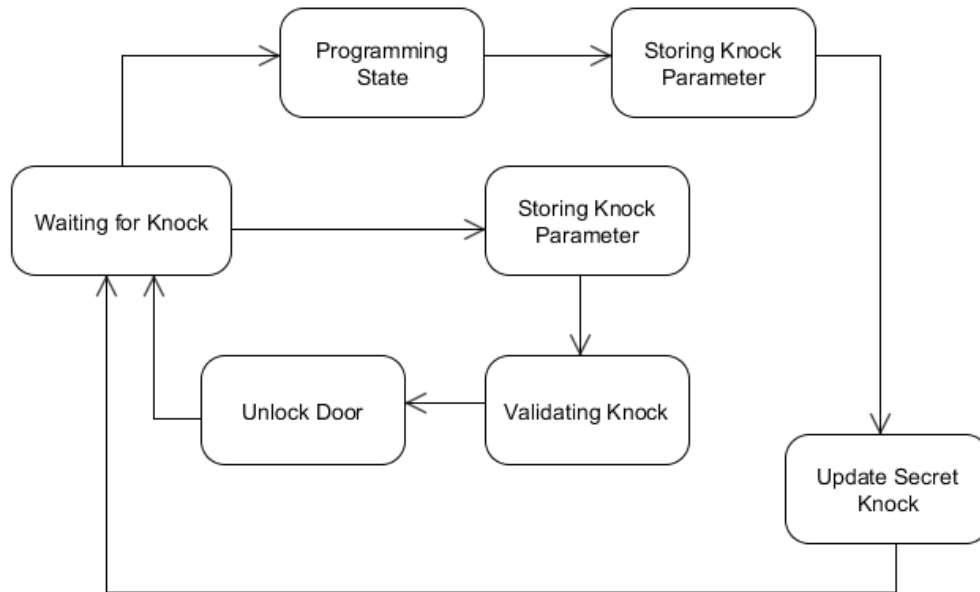


Figure 43 Piezoelectric State Diagram

The Piezoelectric sensor software also has the ability to notify the user of the different states that the system transitions to. While the sensor system is in the Waiting for Knock state the user will be notified by the LED. The LED will be blue while in this state signaling that the system is listening for a knock. During the Storing Knock Parameter State the LED will continue to stay blue while the user is attempting to enter a valid knock. If the knock is valid the user will gain access through the door and the LED will turn green for the specified amount of time that the door is opened. Conversely if the users knock doesn't match the stored knock, the user will not gain access to the door and will be notified by the LED transition to blinking red. As with the previous RFID system software, when the system is in the Programming state, and waiting for a knock, the LED will blink between red and blue. The following figure displays the notification patterns output to the user for the different states mentioned.

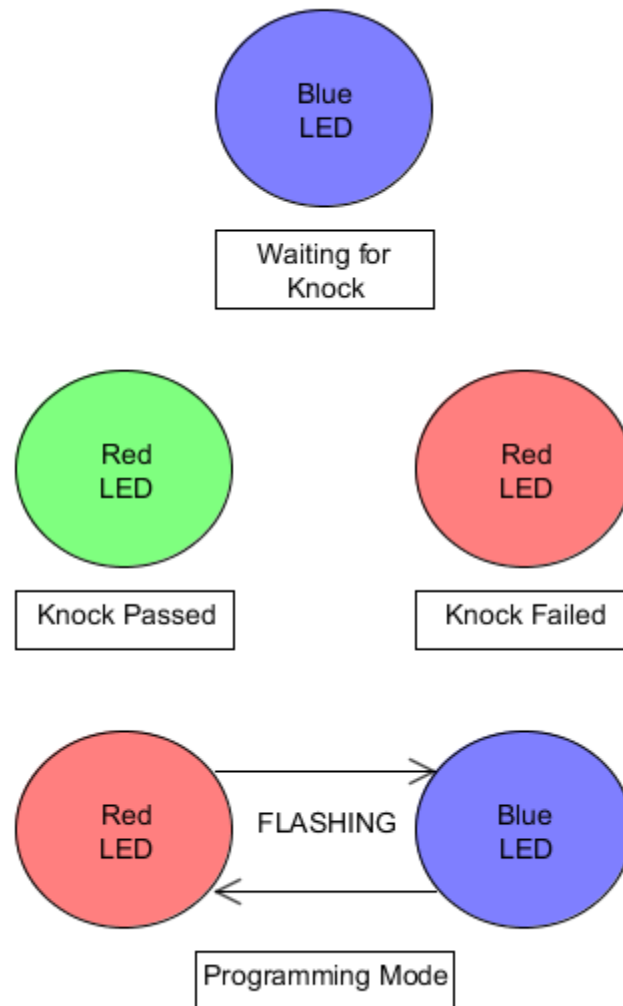


Figure 44 Piezoelectric Notification State Diagram

4.3.5 Embedded Software

The embedded communication protocol that the team decided to use for the Raspberry Pi to microcontroller control is I2C. The group chose to go with this serial communication protocol because the need for a simple serial connection was needed. Both the Raspberry Pi and the microcontroller that is implemented in the design, the Atmega168, support this protocol. The Raspberry Pi will be able to control the functionality of the microcontroller by sending commands through the I2C connection. The information sent down the serial connection is in the form of ASCII characters and will be decoded by the Atmega to control useful functions of the embedded system. Some of the capabilities of the code will include locking, unlocking, and changing the states of the RFID subsystem as well as the Piezoelectric subsystem.

The first commands, the image processing and voice recognition needs to implement is the lock and unlock commands. The unlock sequence the Raspberry Pi will send a ASCII character 'U' down the I2C communication line. Once the Atmega168 receives the 'U' code from the image processing subsystem the microcontroller will execute the unlock command to change the state of the door lock. After this command is executed the microcontroller will send back notification in the form of ASCII letter 'C' to the Raspberry Pi to verify that the command was completed. The next command that the image processing and voice recognition subsystem will need the micro controller to execute is the lock command. To lock the door the Raspberry Pi will send the ASCII character 'L' to the Atmega. After this command is executed the microcontroller will send back notification in the form of ASCII letter 'C' to the Raspberry Pi to verify that the command was completed. Also if the door is already locked and there is no change to the system, the microcontroller will still send back 'C', to verify that the door is in the locked position.

Another potentially useful command that the image processing subsystem will be capable of is putting the system into Programming Mode. This functionality is useful so the user can take full control of the system without the need of the Master RFID card. If needed, the user can use the application software on their handheld device to simulate the RFID Master card. This will give the user the capability of allowing the person at the door to program a new secret knock, or to enter a new RFID card into the system. To put the embedded system into Programming Mode the Raspberry Pi will send the ASCII character 'P' to the Atmega. After this command is executed the microcontroller will send back notification in the form of ASCII letter 'C' to the Raspberry Pi to verify that the given command was completed.

The motion detection portion of the project allows the camera to be turned off while there isn't anyone presently at the door. Since the motion detector is integrated into the embedded system, whenever motion is detected the Atmega will have to communicate with the Pi to turn the camera to the on position. To complete this task the Atmega will send an ASCII character down the serial connection to the Raspberry Pi. The communications will be to tell the image processing subsystem if there is currently someone in front of the door, or inversely if there isn't anyone present. To signify to the camera subsystem that there is someone present, the microcontroller will send an ASCII character 'M' to signify that there was motion detected. The Atmega will send a ASCII character 'N' if the camera subsystem needs to verify whether or not anyone is present.

The image processing subsystem also needs to know the states of certain elements of the embedded system. The Raspberry Pi needs to be able to poll the Atmega to get the information on all of the subsystems. To poll the embedded system in the Raspberry Pi will send the ASCII character 'P' to the Atmega. After the microcontroller receives this signal it will respond by reporting state information to the image processing subsystem. The Atmega will send the state

codes for the motion detector back to the Raspberry Pi, signifying that there was motion detected or that there is no one presently in front of the door. Also once polled the Atmega will send back the state of the lock to the Raspberry Pi. The microcontroller will indicate whether the lock is in the locked or unlocked state. The Raspberry Pi will need some of this vital information to be able to function effectively with the embedded system.

Figure 45 Embedded Communication Script Commands

ASCII Code	Meaning	Function
'L'	Lock	Locks the door
'U'	Unlock	Unlocks the door
'N'	No Motion	Sent if no motion is detected by the motion detector
'M'	Motion	Sent if motion is detected by the motion detector
'P'	Poll	Poll the microcontroller for state information of devices
'C'	Complete	Microcontroller signals that the command in question was executed

4.3.6 Image Processing/Video Streaming

Low Level Description and Design: The Image Processing and Video Streaming Software use the camera as input, as it provides the frames. The Image Processing modules use OpenCV for grabbing the frames from the camera. OpenCV is able to recognize a large range of cameras, including the Logitech C300, and the video4linux driver provides compatibility for the camera in a Linux OS. OpenCV automatically turns on a camera when it wants to capture a frame, but the camera will always be on so that the video stream software can always have access to the camera's frames. For the image processing software, a queue is used to store all the frames captured. The queue only captures frames when it receives a "Motion Command" from the Atmega. This command will be an ASCII M and is sent whenever the PIR sensor picks up any motion in the vicinity of the system. A python GPIO Reader module is used to obtain such a command from the Atmega. As a result, processing power is only used when necessary. The queue will capture a predetermined amount of frames, such as 6 or 7. Then it will pass those frames to the Face Detector module, which will perform any necessary preprocessing of the images, such as resizing, changing resolution, etc.

The preprocessing of the images will not occur in between capturing frames as this method will lead to a lower frame rate, which reduces the possibility of a decent picture of the person in the vicinity being captured. The Face Detector module will use the Cascade Classifier method to detect any faces in the frame. The Cascade Classifier is loaded from an xml file “facedetector.xml”. Once it finds a face in the picture, it will not check any of the remaining frames in order to save processing time. It will find faces by using a window size to scan the image. This window size will be experimented with to achieve optimal results. A diagram of the image processing/video streaming software system is shown below in following figure.

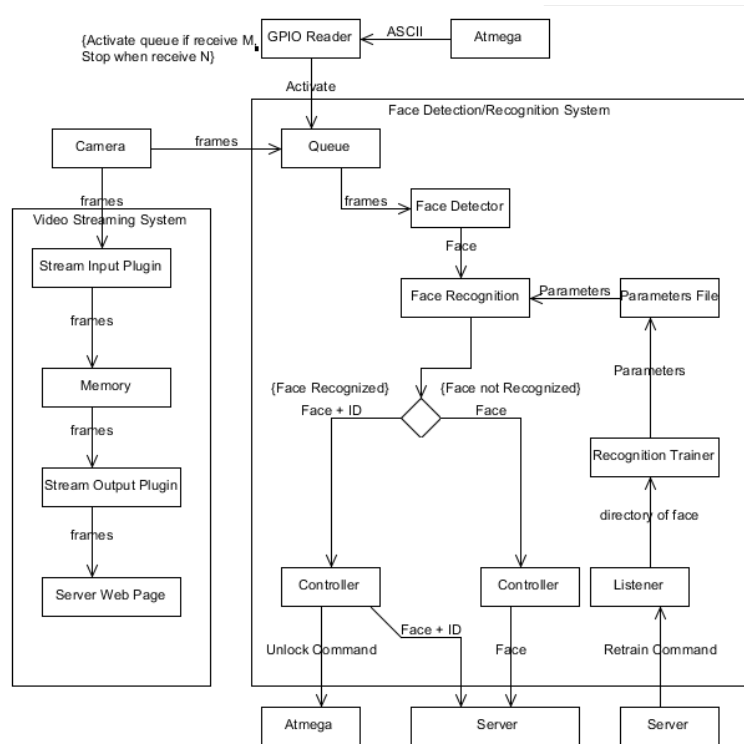


Figure 46 Image Processing/Video Streaming System Diagram

The face that the face detector finds will be extracted and sent to the face recognition module. The face recognition module will use the most optimal algorithm (eigenfaces, fisherfaces, or local binary patterns) to see if the face belongs to any of the faces that the system is set to recognize. The face recognition module will read its parameters from a file called “facestorecognize.xml” which will contain all the information that will be used to predict which face that the captured face most resembles. The module’s prediction will only be accepted if it falls within a 95% confidence interval in order to ensure that only accurate predictions are used. If the face recognition module’s prediction meets this criterion, the face and its ID (a string that indicates the name of the person that the detector determined that the face captured belonged to) will be passed to the controller module. If the setting to unlock the

door if a person who is in the KEES database comes to the door is activated, the controller module will use a GPIO library to send an unlock command to the Atmega that will cause the electric door strike to unlock. It will also interact with the server and send the face and the identity of the face to the server so that it can be added to the web page. If the face is not found to belong to anyone in the database, only the face will be sent to the server.

The other component in the image processing software system is the recognition trainer module. This module is activated if the server sends a command to the listener module to retrain the face recognizer due to the fact that a new person was added to the KEES database. This command will be sent by setting a node in the xml file "retrain.xml" to the value "retrain." The listener module will periodically read this file to determine if retraining needs to be done. This will largely occur when the system is idle which occurs when no frames are being captured by the image processing system. The method of using a socket that always listened on a specified port on the loopback address 127.0.0.1 was considered, but this method would require a separate thread, as listening on a socket blocks the thread until it receives data. A thread that is idly waiting is a waste of processing power, so this method was ruled out. This listener module will then activate the recognition trainer module. There will be a file called "facestorecognize.csv" that will contain all the directories of the faces that the recognition trainer module needs to use for training. The server script that sent the "retrain" command adds the directory of the new face that was added to the database to this file. For each face directory, the corresponding name of the person and a unique ID number will also be in the file. The recognition trainer will then use "facestorecognize.csv" to read in all the images of the people that KEES needs to recognize, and will generate parameters that distinguish the faces from each other. These parameters will be saved to the file "facestorecognize.xml" which the face detector module uses to make predictions. It then saves the frames in an accessible location in memory that an output plugin can access. The frames

The video streaming primarily consists of plugins. The input plugin input_uvc.so uses the video4linux driver to automatically capture frames from the camera's video. This plugin will be optimized for efficiency. The output plugin output_http.so will read the frames that are stored in memory. It will communicate with the server and stream it the frames that were captured. This process will always happen. If this process is observed to noticeably slow down the system, the stream will only happen when a user accesses the web page that displays the video stream.

Class Diagram: Since the video stream system consists only of plugins and scripts, a class diagram is not applicable. The class diagram is shown below in the following figure.

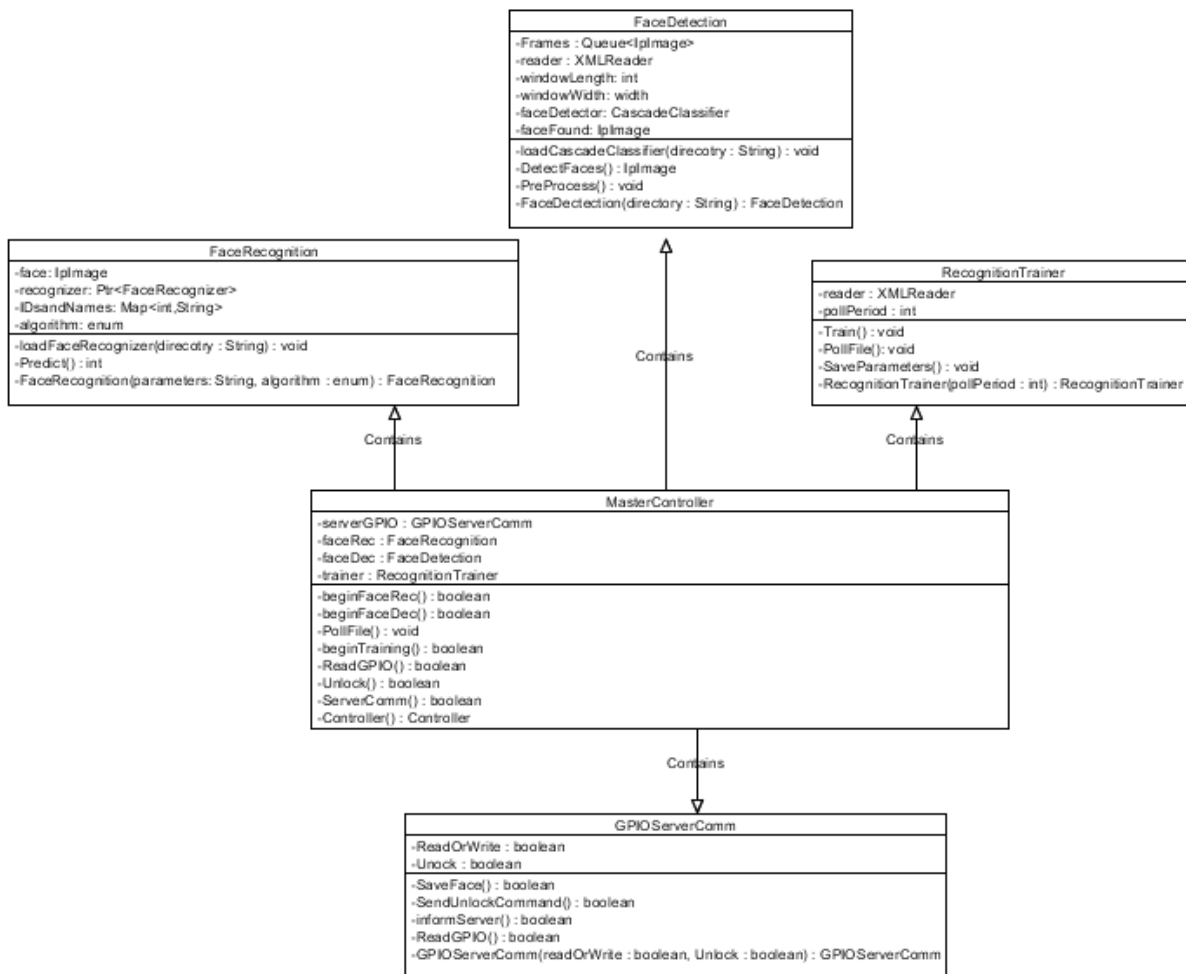


Figure 47 Image Processing Class Diagram

The FaceRecognition, FaceDetection, and RecognitionTrainer belong to the image processing functionality. The GPIOServerComm class is used to communicate with the Atmega through the GPIO pins, and is also used to communicate with the server as well. Whether it communicates with the server, Atmega, or both is specified by the parameters passed into its constructor. The MasterController class acts as the main controller for the image processing software. It contains a FaceRecognition, FaceDetection, RecognitionTrainer, and GPIOServerComm object. Its purpose is to organize all the functions of the software system into one main class that can be instantiated.

5.0 Design Summary of Hardware and Software

5.1 Hardware Summary

The design consists of several separate components all integrated into a single function system. Capacitors can help suppress higher frequency noise and short power dips. Therefore some capacitors may be located near the microcontroller

to help with short bursts. An atmega microcontroller will serve the brains of most of the system. The microcontroller will be processing data from the RFID Reader receiving serial data in on its Rx line. A piezo buzzer and status RGB will reflect an accepted or denied RFID tag, green and buzz for accepted, red and buzz for denied and blue for system waiting. The controller will be loaded with code so the piezo sensor may be activated with a secret knock. The fail-secure electric strike, light detector will be under the control of the atmega as well as integrated to a raspberry pi with a webcam connected through usb for image processing and sending an “unlock” signal to the atmega. The PIR motion detector connected through the atmega will send a signal to the webcam when motion is detected to allow the cam to gather frames for facial recognition. This saves power because the system won't be running constantly

The Arduino and raspberry pi are connected via their I2C (inter-integrated circuit) pinouts. The raspberry pi is operating at a 3.3 volt level while the Arduino is operating at a 5 volt level. This poses a problem however the reason this works is because the Arduino does not have any pull-ups resistors, but the headers on the Raspberry Pi have 1.8kohm resistors to the 3.3 volt power rail. The data is transmitted by pulling the lines to 0 for a high logic signal. For the low logic signal, it's pulled up to the rail voltage level. But because there are no pull-up resistors on the Arduino and the fact that 3.3 volts is within the low logic level range for the Arduino, everything should work normally. Although it is presumed safe to directly connect the two devices together a logic level converter may be used to make sure that the voltage is stepped up or down appropriately.

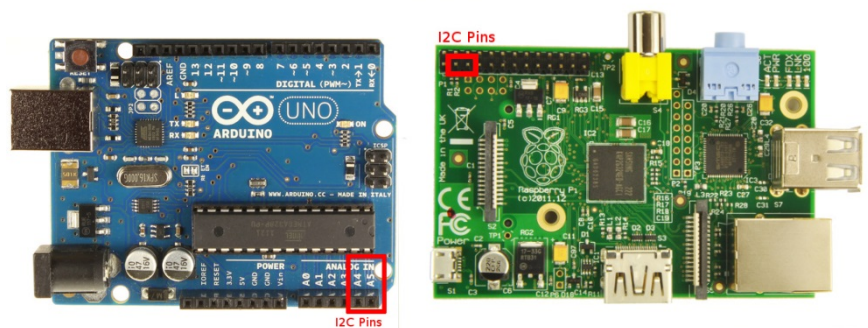


Figure 48 I2C Embedded Communications Pinout

5.2 Software Summary

The software architecture on the Raspberry Pi contains a webserver, services for the webserver, video streaming, image processing, a database, and a GPIO reader for I2C communication. There is a lot of communication among the various software modules, and a lot of time will be spent integrating all of the various software components. The software architecture on the Raspberry Pi is shown in the following figure.

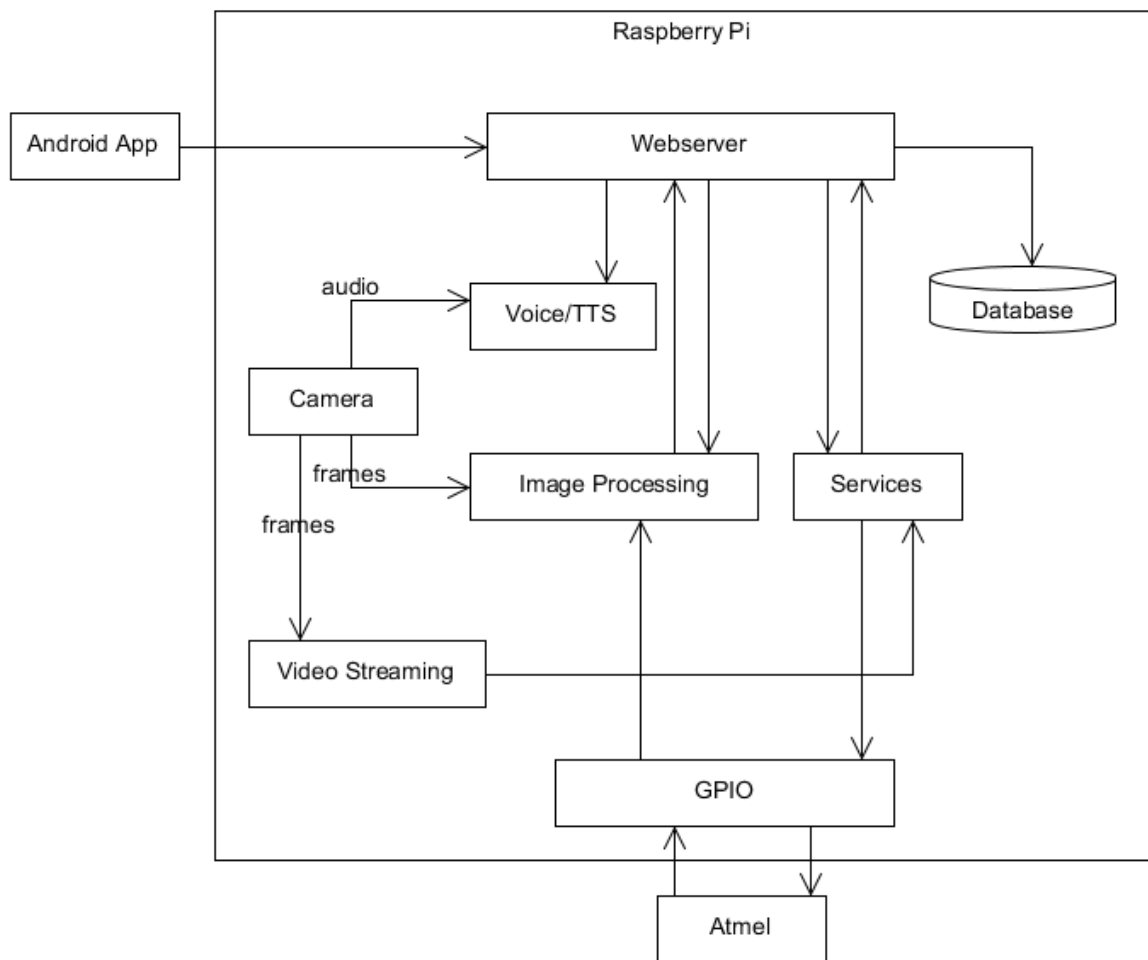


Figure 49 Raspberry Pi Software Architecture and Android App

6.0 Project Prototype Construction

6.1 Parts Acquisition/Financial Budget

The total cost will be split evenly amongst the group members. All the supplies will be bought during the Christmas break. Most of the supplies will be obtained from SparkFun, Amazon, and Element14.

Table 23 Project Budget

Part	Cost Per Unit	Quantity	Total Cost
Apache Web Server Software	\$0.00	1	\$0.00
Android SDK	\$0.00	1	\$0.00
OpenCV SDK	\$0.00	1	\$0.00
Raspberry Pi	\$35.00	1	\$35.00
USB Wifi Module	\$12.00	1	\$12.00
Arduino Uno R3 (ATmega328)	\$30.00	2	\$60.00
Logitech C300 Webcam	\$20.00	1	\$20.00
Electric Door Strike	\$30.00	1	\$30.00
RFID Starter Kit	\$49.95	1	\$49.95
125khz RFID Token Tag	\$1.25	3	\$3.75
RFID Reader Breakout	\$0.95	1	\$0.95
Wall Adapter Power Supply-12VDC 600mA	\$5.95	1	\$5.95
Circuit Components: LEDs, Transistors, Capacitors, Wires, Breadboard etc.	\$10.00	1	\$10.00
Piezo Element	\$1.50	1	\$1.50
PIR Sensor	\$12.95	1	\$12.95
RGB Sensor	\$8.00	1	\$8.00
		Total	\$217.55

6.2 PCB Vendors and Assembly

A major part of the design will be producing a printed circuit board (PCB) for the final design and project. Having a PCB not only will clean up the final design but will allow the project to appear well thought out planned and professional. There are two options when creating a PCB for the design, either use circuit CAD software available online for free or for purchase and use a company that will create and send the PCB. It is also important to note that many PCB manufacturers use and will allow much different PCB software. So the PCB may be created with certain software but may be sent to a cheaper/faster/or better quality manufacturer. The other option is to obtain the necessary parts and components for the design and PCB and create and make the PCB without a professional manufacturer, creating it ourselves.

4PCB offers a Full Spec 2-Layer Prototype Special for \$33 and a 4-Layer for \$66 with no minimum quantity order. 4PCB also offers PCB artist for free which is their own PCB layout software and a free PCB file check.

Eagle Cad is software that can be purchased and/or used for free depending on how extensive a PCB needs to be that includes schematic and layout editor and auto-router. This software is very popular amongst the hobbyists and DIY community. The software creates standard files that are accepted by many PCB fabrication companies. The price of the PCB will vary on the actual PCB and the manufacturer chosen.

OSH-Park is a community PCB order which reduces costs of the PCB. They offer a 2 layer order for \$5 per square inch where three copies of the design will be manufactured. They also offer a 4 layer order for \$10 per square inch where three copies of the design will be manufactured. "You can order as many copies as you want, as long as they are in multiples of three." According to the website. This may serve as some interest in case other tinkering or experimenting with the other copies of the pcb's. They process Eagle Cad files as well.

Another option is to create the PCB ourselves. This option may be slightly cheaper however the time, effort and patience needed to create a pcb without having it professionally manufactured would most likely outweigh the price factor.

6.3 Prototype Construction and Configuration

6.3.1 Software Build Plan

Two of the group members will develop the software that runs on the Raspberry Pi which consists of the webserver, web services, image processing, video streaming, and voice recognition. One group member will develop the webserver its web services, and voice recognition (if time permits) on one Raspberry Pi. The other group member will develop the video streaming and image processing

software on a separate Raspberry Pi. For the image processing software, development will be done by using a virtual machine to emulate a Linux OS. A cross compiler designed to target the Raspberry Pi and its hardware floating point ability will be used to compile code that can be run on the Raspberry Pi. The software will then be copied to the Raspberry Pi via the Linux “scp” command. Once both group members complete their development, all software will be integrated onto one Raspberry Pi. This can easily be accomplished by using an ssh client to copy files from one Raspberry Pi onto the other. Then integration testing will begin to test the integrity of the system. Once all the software for the Raspberry Pi is completed, both of the software developers will create the Android app, which will then be tested with the webserver.

6.3.2 Hardware Build Plan

The remaining two group members will focus on the implementation of the hardware system. The team will design the schematics and all of the pinout diagrams for the embedded system and build the first prototype on a breadboard. The secondary responsibility will be to implement the code for the RFID and Piezoelectric subsystem. The code will be implemented using the design specifications as well as the state diagrams in the design section of the document.

7.0 Project Prototype Test Plan

In this section the team will discuss testing methods and procedures of the testing process for the Keyless Electronic Entry System. In the test environment subsection a description of the hardware tools as well as the tools used to test the software will be explained. In the following section unit testing will be performed on the separate devices and circuits that will be used in the project. Once the unit testing is complete then the process of testing that the various systems included in the project actually function and work together is discussed in detail in the integration test subsection. Finally the teams will device tests and methods to verify that the whole system can't be broken in the regression testing subsection of this document.

7.1 Test Environment

The environment that will be used to test the Keyless Electronic Entry System will vary depending on what type of test as well as what sub element of the system is being tested. To thoroughly test the end product in the final stages of the process the team will be testing in the senior design lab as well as in one of the team member's garages. This will allow for extensive testing to verify that the final product matches the specification that has been set out. For the unit testing portion of the project each member will be testing the separate components alone, before the final project will be implemented. During this testing phase each

team member will choose a place suitable for them to do the tests that will allow for a safe test environment and accurate test results.

7.2 Unit/Functional Testing

7.2.1 RFID Testing

RFID Card Scan Test

Purpose and objective: The RFID Scan Test's objective is to verify that the RFID reader is capable scanning valid RFID cards.

Supplies:

- ID-12LA RFID reader
- RFID card
- Buzzer

Preparation

Connectivity and power for supplies, check specifications

- Make sure there isn't large amounts of RF noise in the area during testing

Procedure:

1. Verify all components are connected correctly and correct amount of power is supplied
2. Slowly move the RFID card within the RFID readers range
3. Wait for RFID Buzzer to sound and LED notification light to turn on

Expected Results:

The outcome of the RFID Card Scan Test is that the RFID reader subsystem will respond with buzzer and LED notification when a card is scanned

RFID Range Test

Purpose and objective: The RFID Range Test's objective is to verify that the RFID reader can detect and identify valid RFID cards at a sufficient range established within the specifications of the given design.

Supplies:

- ID-12LA RFID reader
- RFID card
- LED
- Ruler

Preparation

Connectivity and power for supplies, check specifications

- Make sure there isn't large amounts of RF noise in the area during testing

Procedure:

1. Verify all components are connected correctly and correct amount of power is supplied
2. After device is powered position ruler perpendicular to the RFID reader's antenna input
3. Slowly move the RFID card within the RFID readers range
4. Once notification LED blinks notifying the card read was successful record the distance the card was read from the RFID device
5. Repeat steps 2-4 5 times
6. Analyze results and verify card reads fall within hardware specifications

Expected Results:

The outcome of all the card reads must be at a minimum of 2 inches in order to pass the RFID Range test.

RFID Serial Port ASCII Code Test

Purpose and objective: The RFID Serial Port ASCII Code test's objective is to verify that the RFID reader is in a readable ASCII format.

Supplies:

- ID-12LA RFID reader
- RFID card
- Atmega 168

Preparation

Connectivity and power for supplies, check specifications

- Make sure there isn't large amounts of RF noise in the area during testing

Procedure:

1. Verify all components are connected correctly and correct amount of power is supplied
2. Verify that the format selector pin is tied to ground (format selected is ASCII)
3. Slowly move the RFID card within the RFID readers range
4. Once RFID card read is successful verify correct ASCII ID code was sent over the serial connection

Expected Results:

The outcome of the RFID Serial Port ASCII Code Test is that the RFID code present over the serial connection is the same ASCII representation of the physical RFID card.

RFID Add/Delete Card Test

Purpose and objective: The RFID Add/Delete Card Test's objective is to verify that the RFID reader subsystem can Add/Delete RFID cards from the Atmega 168 memory.

Supplies:

- ID-12LA RFID reader
- MASTER RFID card
- LED
- Atmega 168

Preparation

Connectivity and power for supplies, check specifications

- Make sure there isn't large amounts of RF noise in the area during testing

Procedure:

1. Verify all components are connected correctly and correct amount of power is supplied
2. Slowly move the MASTER RFID card within the RFID readers range
3. Once the MASTER RFID card read is successful the buzzer will sound and the LED will flash blue and red
4. Slowly move a RFID card within the RFID readers range that **is not** registered in the system
5. Once the card is scanned verify that the card **can** unlock the door
6. Slowly move the MASTER RFID card within the RFID readers range
7. Once the MASTER RFID card read is successful the buzzer will sound and the LED will flash blue and red
8. Slowly move a RFID card within the RFID readers range that **is** registered in the system
9. Once the card is scanned verify that the card **cannot** unlock the door

Expected Results:

The outcome of the RFID Add/Delete Card Test is that the RFID reader subsystem is capable of going into Programming mode. Once in programming mode the RFID subsystem is capable of adding or removing cards from the system.

7.2.2 Piezo Testing

Piezo Serial Test

Purpose and objective: To test the functionality of the Piezo element.

Supplies:

- Piezo element
- Arduino Uno Dev. Board
- Arduino software environment
- Breadboard

- Small piece of wood

Preparation

- Connections and power for supply, program simple test code to Arduino.
- Make sure piezo element is correctly wired.

Procedure:

1. Verify all components are connected correctly and correct amount of power is supplied.
2. Apply external force to the piece of wood with piezo element attached.
3. The analog to digital converter will convert the applied external force and transform the voltage into a value in the range 0 to 1024. 0 represents 0 volts, while 1024 represents 5 volts.
4. The voltage values will be sent over the Arduino's serial port to the Arduino GUI. The values can then be observed and recorded.

Expected Results:

The outcome of the Piezo Serial Test is that the piezo element responds to an applied external force. The response can then be seen via the Arduino GUI.

Piezo Differential Test

Purpose and objective: To test the functionality of the Piezo element with programmed series "rhythm" of knocks.

Supplies:

- Piezo element
- Arduino Uno Dev. Board
- Arduino software environment
- Breadboard
- Small piece of wood
- LED to simulate lock

Preparation

- Connections and power for supply, program test code to Arduino.
- Make sure piezo element is correctly wired.
- Test strongly relies on the code uploaded to the Atmega.

Procedure:

1. Verify all components are connected correctly and correct amount of power is supplied.
2. Apply specific preprogrammed knock to the piece of wood with piezo element attached.
3. The peizo element will transform the knock to digital values and if the knock matches the programmed knock the Arduino will turn the led on. The led is simulating the electric strike unlocking. The code converts the absolute timing of the knocks to the rhythm of the knocks. This allows knocks ot be

fast or slow as long as the rhythm correct the led will turn on aka the strike will unlock.

4. Repeat steps 2 and 3 but try the knock at different speeds to verify that the strike will unlock no matter the speed of the individual knocks.

Expected Results:

The outcome of the Piezo Differential Test is that the strike will unlock when the system recognizes a series of knocks.

7.2.3 PIR Unit Testing

PIR Range Test

Purpose and objective: The PIR Range Test's objective is to verify that the PIR sensor is able to detect people that are moving 15 feet away and people that are moving 30 feet away.

Supplies:

- Parallax 555-28027 sensor
- Atmega 168

Preparation

Connectivity and power for supplies; check specifications. Make sure supply voltage that is in the range of 3V-6V to the Parallax sensor.

Procedure:

1. Verify that the Parallax sensor is connected correctly to the Atmega 168. One pin is for ground, the second is for voltage, and the third pin is the output pin.
2. Adjust the jumper on the Parallax sensor to change the detection range to 15 feet.
3. Measure a distance of 15 feet from the sensor and start moving towards the sensor. Verify that the LED is not on when you are out of the sensor's range.
4. Verify that when you move into the detection range, the LED on the Parallax turns on. Verify that when you stop moving, the LED on the Parallax turns off.
5. Repeat the test with a detection range of 30 feet by adjusting the jumper on the sensor.

Expected Results:

The outcome of the PIR Range Test is that the LED will turn on when a person moves into a range, and will turn off once the person stops moving.

PIR Stability Test

Purpose and objective: The PIR Stability Test's objective is to verify that the PIR sensor is stable in an outside shaded environment.

Supplies:

- Parallax 555-28027 sensor
- Atmega 168

Preparation

Connectivity and power for supplies; check specifications. Make sure supply voltage that is in the range of 3V-6V to the Parallax sensor.

Procedure:

1. Verify that the Parallax sensor is connected correctly to the Atmega 168. One pin is for ground, the second is for voltage, and the third pin is the output pin.
2. Place the sensor outside in a shaded area.
3. Adjust the jumper on the Parallax sensor to change the detection range to 15 feet.
4. Move out of the range of the sensor
5. Verify that the LED does not turn on.
6. Repeat the test with a detection range of 30 feet by adjusting the jumper on the sensor.

Expected Results:

The outcome of the PIR Stability Test is that the LED will not turn on when no one is in the range of the sensor.

7.2.4 Strike Test(s)**Fail-Secure Strike Unlock Test**

Purpose and objective: To test the functionality of the Fail-Secure lock making sure the strike unlocks when current is passed through.

Supplies:

- Fail-Secure Strike
- Arduino Uno Dev. Board
- TIP31 Transistor
- 12 V DC Source
- Breadboard
- Push button
- Pull-Up/Down resistor

Preparation

Connections and power for supply, program simple test code to Arduino.

- Make sure strike and transistor terminals are correctly wired.

Procedure:

1. Verify all components are connected correctly and correct amount of power is supplied.
2. Press the push button allowing current to pass through the strike.
3. Wait for strike to then open (unlock) for a brief period of time.

Expected Results:

The outcome of the Fail-Secure Test is that the strike will open when a small amount of current is passed through it.

Fail-Secure Strike Unlock with Piezo element Test

Purpose and objective: To test the functionality of the Fail-Secure lock making sure the strike unlocks via the piezo element.

Supplies:

- Fail-Secure Strike
- Arduino Uno Dev. Board
- TIP31 Transistor
- 12 V DC Source
- Breadboard
- Push button
- Pull-Up/Down resistor
- Piezo element

Preparation

- Connections and power for supply, program simple test code to Arduino.
- Make sure strike, piezo and transistor terminals are correctly wired.

Procedure:

1. Verify all components are connected correctly and correct amount of power is supplied.
2. Perform simple vibration test to piezo element.
3. Wait for strike to then open (unlock) for a brief period of time.

Expected Results:

The outcome of the Fail-Secure Piezo Test is that the strike will open when the piezo picks up external vibrations. The microcontroller allows a small amount of current to pass through the strike.

Fail-Secure Strike Unlock with RFID Test

Purpose and objective: To test the functionality of the Fail-Secure lock making sure the strike unlocks via the RFID system.

Supplies:

- Fail-Secure Strike
- Arduino Uno Dev. Board
- TIP31 Transistor
- 12 V DC Source
- Breadboard
- Push button

- Pull-Up/Down resistor
- RFID system + RFID card

Preparation

- Connections and power for supply, program simple test code to Arduino.
- Make sure strike and transistor terminals are correctly wired and RFID system is correctly integrated.

Procedure:

1. Verify all components are connected correctly and correct amount of power is supplied.
2. Display RFID card to the RFID card reader.
3. Once card is read wait for strike to then open (unlock) for a brief period of time.

Expected Results:

The outcome of the Fail-Secure RFID Test is that the strike will open when the RFID card reader detects a valid card.

Fail-Secure Strike Unlock with Voice recognition Test

Purpose and objective: To test the functionality of the Fail-Secure lock making sure the strike unlocks via voice recognition.

Supplies:

- Fail-Secure Strike
- Arduino Uno Dev. Board
- TIP31 Transistor
- 12 V DC Source
- Breadboard
- Push button
- Pull-Up/Down resistor
- Voice recognition system

Preparation

- Connections and power for supply, program simple test code to Arduino.
- Make sure strike and transistor terminals are correctly wired and Voice recognition system is correctly integrated.

Procedure:

1. Verify all components are connected correctly and correct amount of power is supplied.
2. Speak to system.
3. Once system verifies the voice, wait for strike to then open (unlock) for a brief period of time.

Expected Results:

The outcome of the Fail-Secure Voice Recognition Test is that the strike will open when the system validates a voice.

Fail-Secure Strike Unlock with Face recognition Test

Purpose and objective: To test the functionality of the Fail-Secure lock making sure the strike unlocks via Face recognition.

Supplies:

- Fail-Secure Strike
- Arduino Uno Dev. Board
- Raspberry Pi
- TIP31 Transistor
- 12 V DC Source
- Breadboard
- Push button
- Pull-Up/Down resistor
- Face recognition system

Preparation

- Connections and power for supply, program simple test code to Arduino.
- Make sure strike and transistor terminals are correctly wired and Face recognition system is correctly integrated.

Procedure:

1. Verify all components are connected correctly and correct amount of power is supplied.
2. Present a face to the camera system.
3. Once system verifies the face, wait for strike to then open (unlock) for a brief period of time.

Expected Results:

The outcome of the Fail-Secure Face Recognition Test is that the strike will open when the system validates a face.

7.2.5 Strike Test

Fail-Secure Strike Unlock Test

Purpose and objective: To test the functionality of the Fail-Secure lock making sure the strike unlocks when current is passed through.

Supplies:

- Fail-Secure Strike
- Arduino Uno Dev. Board
- TIP31 Transistor
- 12 V DC Source
- Breadboard
- Push button
- Pull-Up/Down resistor

Preparation

- Connections and power for supply, program simple test code to Arduino.
- Make sure strike and transistor terminals are correctly wired.

Procedure:

1. Verify all components are connected correctly and correct amount of power is supplied.
2. Press the push button allowing current to pass through the strike.
3. Wait for strike to then open (unlock) for a brief period of time.

Expected Results:

The outcome of the Fail-Secure Test is that the strike will open when a small amount of current is passed through it.

Fail-Secure Strike Unlock with Piezo element Test

Purpose and objective: To test the functionality of the Fail-Secure lock making sure the strike unlocks via the piezo element.

Supplies:

- Fail-Secure Strike
- Arduino Uno Dev. Board
- TIP31 Transistor
- 12 V DC Source
- Breadboard
- Push button
- Pull-Up/Down resistor
- Piezo element

Preparation

- Connections and power for supply, program simple test code to Arduino.
- Make sure strike, piezo and transistor terminals are correctly wired.

Procedure:

1. Verify all components are connected correctly and correct amount of power is supplied.
2. Perform simple vibration test to piezo element.
3. Wait for strike to then open (unlock) for a brief period of time.

Expected Results:

The outcome of the Fail-Secure Piezo Test is that the strike will open when the piezo picks up external vibrations. The microcontroller allows a small amount of current to pass through the strike.

Fail-Secure Strike Unlock with RFID Test

Purpose and objective: To test the functionality of the Fail-Secure lock making sure the strike unlocks via the RFID system.

Supplies:

- Fail-Secure Strike
- Arduino Uno Dev. Board
- TIP31 Transistor
- 12 V DC Source
- Breadboard
- Push button
- Pull-Up/Down resistor
- RFID system + RFID card

Preparation

- Connections and power for supply, program simple test code to Arduino.
- Make sure strike and transistor terminals are correctly wired and RFID system is correctly integrated.

Procedure:

1. Verify all components are connected correctly and correct amount of power is supplied.
2. Display RFID card to the RFID card reader.
3. Once card is read wait for strike to then open (unlock) for a brief period of time.

Expected Results:

The outcome of the Fail-Secure RFID Test is that the strike will open when the RFID card reader detects a valid card.

Fail-Secure Strike Unlock with Voice recognition Test

Purpose and objective: To test the functionality of the Fail-Secure lock making sure the strike unlocks via voice recognition.

Supplies:

- Fail-Secure Strike
- Arduino Uno Dev. Board
- TIP31 Transistor
- 12 V DC Source
- Breadboard
- Push button
- Pull-Up/Down resistor
- Voice recognition system

Preparation

- Connections and power for supply, program simple test code to Arduino.
- Make sure strike and transistor terminals are correctly wired and Voice recognition system is correctly integrated.

Procedure:

1. Verify all components are connected correctly and correct amount of power is supplied.
2. Speak to system.
3. Once system verifies the voice, wait for strike to then open (unlock) for a brief period of time.

Expected Results:

The outcome of the Fail-Secure Voice Recognition Test is that the strike will open when the system validates a voice.

Fail-Secure Strike Unlock with Face recognition Test

Purpose and objective: To test the functionality of the Fail-Secure lock making sure the strike unlocks via Face recognition.

Supplies:

- Fail-Secure Strike
- Arduino Uno Dev. Board
- Raspberry Pi
- TIP31 Transistor
- 12 V DC Source
- Breadboard
- Push button
- Pull-Up/Down resistor
- Face recognition system

Preparation

- Connections and power for supply, program simple test code to Arduino.
- Make sure strike and transistor terminals are correctly wired and Face recognition system is correctly integrated.

Procedure:

1. Verify all components are connected correctly and correct amount of power is supplied.
2. Present a face to the camera system.
3. Once system verifies the face, wait for strike to then open (unlock) for a brief period of time.

Expected Results:

The outcome of the Fail-Secure Face Recognition Test is that the strike will open when the system validates a face.

7.2.6 Camera/OpenCV Unit Testing**Camera-Linux Compatibility Test**

Purpose and objective: The Camera-Linux Compatibility Test's objective is to verify that Logitech C300 works with the Linux OS on the Raspberry Pi.

Supplies:

- Logitech C300 Webcam
- Raspberry Pi
- Micro USB charger
- HDMI monitor/TV
- Keyboard and mouse

Preparation

Power the Raspberry Pi by using the Micro USB charger. Plug in the Logitech C300 into one of the Raspberry Pi's USB ports, and plugin the keyboard.

Procedure:

1. Boot up the Raspberry Pi.
2. Download guvcview webcam viewer by opening up a terminal and typing in the following commands:

```
sudo apt-get update
```

```
sudo apt-get install guvcview
```
3. Verify that the webcam is working. Should be able to change the resolution and view the camera's output.
4. If the webcam is not working, verify that the video4linux driver is installed.

Expected Results:

The outcome of the Camera-Linux Compatibility Test is that the camera is recognized by the OS, and you can change the camera settings.

OpenCV Image Processing Test

Purpose and objective: The OpenCV Image Processing Test's objective is to verify that Logitech C300 works with OpenCV, and that face detection and face recognition are properly performed.

Supplies:

- Logitech C300 Webcam
- Raspberry Pi
- Micro USB charger
- HDMI monitor/TV
- Keyboard and mouse
- OpenCV SDK

Preparation

Power the Raspberry Pi by using the Micro USB charger. Plug in the Logitech C300 into one of the Raspberry Pi's USB ports, and plugin the keyboard.

Procedure:

1. Boot up the Raspberry Pi.
2. Run the sample OpenCV python file “facedetect.py.” Insert the code for grabbing frames from the camera. This code can be found in the OpenCV python file “camera.py.” Load the file “haarcascade_frontalface_alt.xml” which contains the parameters that will be used to detect faces.
3. Place your face in front of the camera.
4. Verify that the frames grabbed by OpenCV are displayed in a window.
5. Turn the grabbed frames to grayscale, and change the resolution of the pictures to 640x480. Also verify that OpenCV places a rectangle around your face that is shown in the video feed.
6. Save the grabbed frames to the SD card, and verify that the saved files can be opened.
7. Prepare a csv file named “facetrain.csv” that contains the directory of all the faces that need to be added, as well as an ID that indicates which person the face belongs to. These faces will belong to the group members.
8. Run the sample OpenCV file “facerec_demo.cpp.” Copy code for obtaining frames from the camera from “camshiftdemo.cpp.” Train the face recognizer on face images belonging to members of the group by reading the csv file, and loading the faces and the corresponding ID in the face recognizer.
9. Save the parameters generated from the training to a file “facerecognitionparameters.xml.”
10. Have each group member take turns going in front of the camera. Verify that a rectangle around the member’s face, along with the member’s name is shown in the video feed.

Expected Results:

The outcome of the OpenCV Image Processing Test is that the camera is compatible with OpenCV. Also, a frame rate can be observed when capturing frames from the camera using OpenCV. The Face Detection and Face Recognition algorithms are performed with great accuracy and under 400ms total.

7.2.7 Web Server, Web Services, & Database:**Test Name: Local Web Server Access**

Objective: The objective of this test is to verify an HTTP connection can be established within the same local network.

Supplies:

Single-board Computer
 HTTP Server software
 Router
 Mobile Device/Laptop/PC

Preparation: Install any compatible HTTP server software of choice onto the single-board computer running in a UNIX environment. Connect the single-board computer to the network router via Ethernet or WiFi. Connect the mobile device/laptop/PC to the same router on the network via Ethernet or WiFi. Note the default port for an HTTP server is port 80. Note the Internal IP of the single-board computer (assigned by the router).

Procedure: Open the browser on the mobile device/laptop/PC. Initiate an HTTP connection over the local network via the URL: `http://<Internal IP>`, where Internal IP is the IP assigned by the router of the single-board computer running the HTTP server.

Expected Results: Depending on the HTTP server used, a default welcome/home page should display to the user on the mobile device/laptop/PC. If errors occur such as HTTP 404 Page Not Found, then there is something not configured properly within the HTTP Server config file or in the router settings.

Test Name: Remote Web Server Access via Internet

Objective: The objective of this test is to verify an HTTP connection can be established across remote networks via the internet.

Supplies:

Single-board Computer
HTTP Server software
Router
Modem for Internet Connection
Mobile Device/Laptop/PC

Preparation: Install any compatible HTTP server software of choice onto the single-board computer running in a UNIX environment. Connect the single-board computer to the network router via Ethernet or WiFi. Ensure the router has a valid internet connection from the modem. Also ensure the mobile device/laptop/PC has an internet connection and is not on the same network as the single-board computer. On the network hosting the HTTP server, verify the router has port forwarding enabled for port 80. Note the default port for an HTTP server is port 80. Note the External IP of the single-board computer (IP of computer viewed from outside of the network).

Procedure: Open the browser on the mobile device/laptop/PC. Initiate an HTTP connection over the internet via the URL: `http://<External IP>`, where External IP is the IP of the single-board computer to the outside world, running the HTTP server.

Expected Results: Depending on the HTTP server used, a default

welcome/home page should display to the user on the mobile device/laptop/PC. If errors occur such as HTTP 404 Page Not Found, then there is something not configured properly within the HTTP Server config file or port 80 on the router has not been opened and could be blocked by the firewall.

Test Name: Establish Database Connection

Objective: The objective of this test is to verify a connection to the database can be established from within various services and scripts.

Supplies:

Single-board Computer
Database software
Web Framework with Database API

Preparation: Install the Database software of choice on the single-board computer. Install a Web Framework of choice compatible with the selected database. Create a basic Database Accessor class with a single method for initializing a connection to the database. Code and compile a simple script which instantiates the Database Accessor object and calls the method for connecting to the database.

Procedure: Compile the script written and execute the script to open and connect to the database.

Expected Results: The script executes successfully and does not produce any syntax or run-time errors.

Test Name: Web Service Frontend Integration

Objective: The objective of this test is to verify a web service can be created for processing inputs (requests) and generating valid outputs (responses).

Supplies:

Single-board Computer
HTTP Server software
Web Framework

Preparation: Install the HTTP Server software of choice on the single-board computer. Install a Web Framework of choice compatible with the selected HTTP Server. Create a basic test web page with a single element such as a button. Create a simple script which is called when the button is pressed to display some information on the page back to the user.

Procedure: Compile the script, open the browser, and open the newly created test web page.

Expected Results: Verify the test webpage displays as expected and that once the button is clicked, the script is executed and some text is displayed back to the user.

Test Name: Web Service Backend Integration

Objective: The objective of this test is to verify a web service can be created for connecting to a database and querying for specific data.

Supplies:

Single-board Computer
Database software
Web Framework with Database API

Preparation: Install the Database software of choice on the single-board computer. Install a Web Framework of choice compatible with the selected database. Create a basic script which performs a query on the database to retrieve specific data.

Procedure: Compile the script written and execute the script to connect to the database and query for specific data.

Expected Results: The script executes successfully, does not produce any syntax or run-time errors, and returns the expected data.

7.2.8 Video Streaming/Camera Test

Purpose and objective: The test's objective is to verify that the video stream input and output plugins function correctly.

Supplies:

- Logitech C300 Webcam
- Raspberry Pi
- Micro USB charger
- HDMI monitor/TV
- Keyboard and mouse

Preparation

Power the Raspberry Pi by using the Micro USB charger. Plug in the Logitech C300 into one of the Raspberry Pi's USB ports, and plugin the keyboard.

Procedure:

1. Boot up the Raspberry Pi.
2. Use guvcview webcam viewer to see the output of the camera.

3. Use the input plugin to automatically capture and save frames from the camera in memory. Modify the output plugin to save the frames captured to a specified directory.
4. Plug out the camera after 30 seconds, and verify that many pictures were saved in the specified directory.

Expected Results:

The outcome of the test is that the input plugin is able to automatically capture frames from the camera, and the output plugin can obtain those frames from memory.

7.2.9 Light Sensor Test(s)

Light Sensor Test

Purpose and objective: To combine a light sensor with a relay (switch) to test the capability of turning on and off a light due to the amount of ambient radiated light.

Supplies:

- House Light
- Arduino Uno Dev. Board
- Single Relay
- Light sensor

Preparation

- Simple relay circuit and power supply, program simple test code.
- Make sure house light is correctly integrated.

Procedure:

1. Verify all components are connected correctly and correct amount of power is supplied.
2. Shield light sensor from light to see if the relay is triggered and house light turns on.
3. Once light turns on expose excess light to the sensor to trigger the relay to turn house light off.

Expected Results:

The outcome if the test is to verify the system switching on the light in simulated night time and turn off the light in simulated day time.

7.2.10 Voice Recognition & TTS:

Test Name: Voice Recognition via Microphone

Objective: The objective of this test is to verify speech input can be received by the single-board computer via an attached microphone and converted to text.

Supplies:

Single-board computer running UNIX
Webcam with built-in microphone

Preparation: Connect the webcam with built in microphone to the single-board computer. Ensure the single-board computer has a valid internet connection. Install relevant voice recorder software and voice recognition API. Ensure the webcam is recognized by the computer.

Procedure: Create and compile a simple script to listen in for speech from the microphone, process it, and synthesize it into text. Run the script and speak into the microphone.

Expected Results: Verify the speech input is received from the microphone, processed, and is converted to text.

Test Name: Text to Speech via Speaker

Objective: The objective of this test is to output audio converted from text on a single-board computer.

Supplies:

Single-board computer running UNIX
Speaker
Audio software

Preparation: Connect a speaker to the analog port of the single-board computer. Ensure the single-board computer has a valid internet connection. Install a Linux compatible audio player, and relevant sound drivers and utilities. Setup the config file for audio output. Create a basic script to process text and generate an audio file.

Procedure: Compile and execute the script with a sample text string. Play the generated audio file.

Expected Results: Verify the audio plays from the speaker connected to the single-board computer, is audible, at a decent volume level, and translated correctly from the test text string.

7.3 Integration Testing**7.3.1 Embedded Integration Testing****Embedded Communication Verify Test**

Purpose and objective: The Embedded Communication Verify Script Test's objective is to verify that the embedded microcontroller is capable of sending and receiving scripted commands from the Raspberry Pi.

Supplies:

- Embedded PCB
- I2C serial connection
- Raspberry Pi
- Electronic Strike

Preparation

- Connectivity and power for supplies, check specifications
- Connect monitor and input devices to the Raspberry Pi needed for testing

Procedure:

1. Verify all components are connected correctly and correct amount of power is supplied
2. Navigate to the Raspberry Pi's terminal emulator
3. Using the terminal emulator and keyboard, enter "TEST"
4. Verify that the terminal emulator received "PASS"

Expected Results:

The outcome of the Embedded Communication Verify Script Test is that the following the entering of the "TEST" command to the microcontroller receives the "PASS" string; this verifies communication is sending and receiving.

Embedded Lock/Unlock Script Test

Purpose and objective: The Embedded Lock/Unlock Script Test's objective is to verify that the embedded microcontroller is capable of executing scripted commands Lock and Unlock from the Raspberry Pi.

Supplies:

- Embedded PCB
- I2C serial connection
- Raspberry Pi
- Electronic Strike

Preparation

- Connectivity and power for supplies, check specifications
- Connect monitor and input devices to the Raspberry Pi needed for testing

Procedure:

1. Verify all components are connected correctly and correct amount of power is supplied
2. Navigate to the Raspberry Pi's terminal emulator

3. Using the terminal emulator and keyboard, enter the ASCII code 'U'
4. Verify that the electronic strike is in the **unlocked** position
5. Using the terminal emulator and keyboard, enter the ASCII code 'L'
6. Verify that the electronic strike is in the **locked** position

Expected Results:

The outcome of the Embedded Lock/Unlock Script Test is that the following the entering of the 'L' and 'U' commands to the microcontroller, the electronic strike will lock and unlock respectively.

Embedded Poll Motion Detect Script Test

Purpose and objective: The Embedded Poll Motion Detect Script Test's objective is to verify that the embedded microcontroller is capable of executing scripted commands Motion commands from the Raspberry Pi.

Supplies:

- Embedded PCB
- I2C serial connection
- Raspberry Pi
- Camera

Preparation

- Connectivity and power for supplies, check specifications
- Connect monitor and input devices to the Raspberry Pi needed for testing

Procedure:

1. Verify all components are connected correctly and correct amount of power is supplied
2. Navigate to the Raspberry Pi's terminal emulator
3. Have a test subject actively moving in front of camera subsystem
4. Using the terminal emulator and keyboard, enter the ASCII code 'P'
5. Verify that the terminal emulator received 'M' code
6. Make sure there is no activity in front of the camera
7. Using the terminal emulator and keyboard, enter the ASCII code 'P'
8. Verify that the terminal emulator received 'M' code

Expected Results:

The outcome of the Embedded Lock/Unlock Script Test is that the following the entering of the 'P' command to the microcontroller, the embedded subsystem will send the correct codes for motion detection.

7.3.2 OpenCV/Camera/PIR/Server/Video Stream Integration Test

Purpose and objective: The test's objective is to verify that the image processing, video stream, server, and PIR motion sensor can function coherently as a system.

Supplies:

- Raspberry Pi
- Atmega with Electronic Strike and Parallax 555-28027 (PIR sensor) connected
- Logitech C300 webcam
- Keyboard and mouse
- HDMI compatible screen

Preparation

Power the Raspberry Pi and Atmega by using a 12V source. Connect the Raspberry Pi to the Atmega via I2C. Plug in the Logitech C300 into one of the Raspberry Pi's USB ports. Plug in a compatible screen into the HDMI port on the Pi for debugging purposes.

Procedure:

1. Have a person who is in the database approach the door. Also have a person who is not in the database approach the door. Verify that the Parallax sensor's LED turns on, and that the image processing software is activated.
2. Verify that OpenCV begins capturing frames from the camera, and that it stops capturing frames once the Parallax Sensor's LED is turned off.
3. Case of person who is in the database: verify that the face of the person and the corresponding name/ID is sent to the server it can be uploaded to the status webpage.
4. Case of person who is not in the database: verify that the face of the person is sent to the server so that it can be uploaded to the status page. Also, verify that the door is unlocked (if this setting is enabled on the Raspberry Pi).
5. Verify that the input plugin of the video stream is also capturing the frames of the camera. Observe if the output plugin successfully receives these frames and sends it to the server so that it can be uploaded to the webpage. Navigate to the video stream portion of the webpage to verify.

Expected Results:

The outcome of the test is that communication between the Atmega and Raspberry Pi works properly, as well as the communication between the server and the video stream, and between the server and image processing software.

7.3.3 Web Server, Web Services, & Database:

Test Name: Frontend UI to Backend Database Integration

Objective: The objective of this test is to verify data can be accepted from a web

form on the frontend UI, processed, and stored in the backend database.

Supplies:

Single-board Computer
HTTP Server software
Database software
Web Framework with Database API

Preparation: Install the HTTP Server and Database software of choice on the single-board computer. Install a Web Framework of choice compatible with the chosen HTTP Server and Database. Create a basic test web page with some fields related to a record store in the database. Create a basic script to be called by the web page with the parameters and values input into the form. Set the script to connect to the database and update the record with the values input into the form.

Procedure: Open the browser and navigate to the test web page hosted by the HTTP Server. Input data in the fields on the page and submit the form.

Expected Results: Verify the database record is updated with the values input into the form from the test web page.

Test Name: Backend Database to Frontend UI Integration

Objective: The objective of this test is to verify data can be retrieved from the database, processed, and presented to the user via the frontend UI.

Supplies:

Single-board Computer
HTTP Server software
Database software
Web Framework with Database API

Preparation: Install the HTTP Server and Database software of choice on the single-board computer. Install a Web Framework of choice compatible with the chosen HTTP Server and Database. Create a basic test web page with an empty view which can be populated with data returned from the database, and a button to initiate the database query. Create a basic script to be called when the button is pressed to query the database for a specific data set and populate the empty view on the frontend UI with the data returned by the query.

Procedure: Open the browser and navigate to the test web page hosted by the HTTP Server. Click the button to execute the database query.

Expected Results:

The test web page is populated with the data pulled from the database matching

the specified query.

Test Name: Administrator Login Service

Objective: The objective of this test is to verify that only an administrator account is able to successfully login to the server.

Supplies:

Single-board Computer
HTTP Server software
Database software
Web Framework with Database API

Preparation: Install the HTTP Server and Database software of choice on the single-board computer. Install a Web Framework of choice compatible with the chosen HTTP Server and Database. Add an administrator user record to the database with a pre-defined username and password. Create a basic test web page with a username and password field, and a login button. Create a basic script which takes in the username and password, connects to the database, and verifies the login credentials match the credentials stored for the administrator record. Program the script to return a success/denied notification to the UI.

Procedure: Open the browser and navigate to the test web page hosted by the HTTP Server. Input credentials into the username and password fields and submit the credentials to the server by clicking the login button.

Expected Results: There are two cases for expected results. If the credentials entered are invalid and do not match the credentials stored in the database for the administrator account, then the user is unable to login. If the credentials entered are valid, the user is able to login successfully.

Test Name: Video Stream Service

Objective: The objective of this test is to verify the video stream data from the camera can be viewed over an established HTTP connection in real-time.

Supplies:

Single-board Computer
Webcam
HTTP Server software
Stream software
Web Framework

Preparation: Connect any compatible webcam to the single-board computer and install any necessary drivers and configuration utilities. Install the HTTP Server software of choice on the single-board computer. Install a Web Framework of

choice compatible with the chosen HTTP Server. Configure the HTTP Server to allow RTP (Real-Time Transport) protocol tunneling. Install streaming software and configure it for HTTP. Create a basic test web page on the HTTP Server which serves as a hook to the video stream server process.

Procedure: Open the browser and navigate to the test web page for the video stream hosted by the HTTP Server.

Expected Results: Verify the video feed from the camera is visible, quality is sufficient, and is updating in real-time.

Test Name: Lock Controller Service

Objective: The objective of this test is to verify the lock controller service is able to send the unlock signal request to the Arduino for the door to be unlocked.

Supplies:

Single-board Computer
HTTP Server software
Web Framework
Arduino microcontroller
Electric Strike

Preparation: Connect the Electric Strike to the Arduino and configure it to allow commands to be received via the serial interface connection. Setup the single-board computer and connect it to the Arduino. Install the HTTP Server software of choice on the single-board computer. Install a Web Framework of choice compatible with the chosen HTTP Server. Create a basic script on the server to control the electric strike by sending an unlock request to the Arduino.

Procedure: Open the browser and point the URL directly to the test script created for sending an unlock request to the Arduino.

Expected Results: Verify the unlock request is sent by the single-board computer and received by the Arduino microcontroller. Verify the electric strike is unlocked for some period of time, and then locked again until another request is made.

7.3.4 App/Webserver/Camera/OpenCV Integration Test

Purpose and objective: The test's objective is to verify that the communication between the app and webserver is functioning correctly. It also verifies that the app receives notifications whenever OpenCV detects faces that come to the front door. It also verifies that the app's user interface works correctly.

Supplies:

- Two Android Phones with app installed (Samsung Galaxy S3)

- Raspberry Pi
- Atmega with Electronic Strike connected
- Logitech C300 webcam

Preparation

Open the administrative version of the app. Power the Raspberry Pi and Atmega by using a 12V source. Connect the Raspberry Pi to the Atmega via I2C. Plug in the Logitech C300 into one of the Raspberry Pi's USB ports.

Procedure:

1. If this is the first time using the app, create an account and login. If not, login. Verify that you can only login if the correct credentials are entered.
2. Verify that the main page displays the following buttons: "Unlock", "Status", "App Settings", "KEES Settings", "Add to KEES Database", "Security Override", and "Log Out." On the non-administrative version of the app, verify that the buttons "Security Override", "Add to KEES Database", and "KEES Settings" are not available.
3. Press the "Unlock" button. Enter credentials and verify that the door unlocks.
4. Press the "Status" button. Verify that the server's webpage opens up.
5. Press the "App Settings" button. Verify that a menu opens up when credentials are properly entered. The settings menu should include the options "Notifications", "Lock Settings", and "Synch." Press "Notifications" and verify that you are able to change how the app notifications manifest, as well as the sound that accompanies them. Press "Lock Settings" and verify that you are able to specify how much idle time should pass before locking the app, as well as to disable/enable entering credentials after pressing the app's various buttons. Press "Synch" and verify that you can change how often the app communicates with the server.
6. Press the "KEES Settings" button. Verify that the server's administrative webpage with a list of settings is displayed.
7. Press the "Add to KEES Database" button. Enter credentials and verify that you are presented with an interface that enables you to choose a picture to send, as well as the corresponding name. Select a picture, and verify on the server end that the data is received and added to the database. Also verify that the face recognizer is retrained with the new face.
8. Press the "Security Override" button. Enter credentials and verify that you are presented with a list of other users that have the app. Select the app that you wish to disable. Verify on the server end that the user is removed from the database. Also verify that the user's app is unable to make any queries to the server, and that it displays an "app remotely shutdown" message in an alert dialog.
9. Press the "Log Out" button. Verify that the app exists.
10. Have a person who is in the KEES database approach the door. Also have a person who is not in the KEES database approach the door. Verify that the app displays a notification that alerts the user that a known person came to the door for the former case, and a notification indicating that an unknown

person came to the door for the latter case. Also verify that OpenCV captures and saves the images, which are uploaded to the server.

Expected Results:

The outcome of the test is that the app's user interface is responsive and works correctly. All communication between the app and the server is quick in a strong wifi/mobile data environment. The image processing software effectively notifies the server when it captures a face, and also retracts a face when the server commands it to.

7.3.5 Voice Recognition & TTS:

Test Name: Speech-to-Text & Text-to-Speech Integration

Objective: The objective of this test is to verify that speech input can be recognized from a microphone and processed by the system. The system should perform the specific action (if any) and then generate a relevant response and output it via the connected speaker.

Supplies:

Single-board computer running UNIX
Webcam with built-in microphone
Speaker
Audio software

Preparation: Ensure the single-board computer has a valid internet connection. Ensure the webcam is recognized by the computer. Install relevant voice recorder software and voice recognition API. Install a Linux compatible audio player, and relevant sound drivers and utilities. Setup the config file for audio output. Ensure audio can be played back to the speaker, and speech and be accepted from the microphone.

Create a basic script on the server with a specific keyword that can be used to initiate audio recording via the microphone and convert the proceeding speech to text format. Create a second script to process the received audio input in the form of text and to validate it using a list of pre-defined commands. Create a third script to convert the relative response to speech, and output it via the connected speaker. Create a main wrapper script which links all the scripts together. The main script will be used to continuously listen in for the keyword which begins the speech synthesis process.

Procedure: Run the script on the single-board computer and initialize audio recording by saying the valid keyword such as "KEES". Following the keyword say a valid command such as "unlock door".

Expected Results: Verify the command is processed by the system in a timely

manner, and the expected function based on the command given is performed. Verify that a valid response is played back to the user from the speaker saying something such as “door has been unlocked”.

7.4 Regression Testing

Regression testing for the KEES project must be done for the entire software system to ensure the stability and performance of the software. The point of regression testing is to reveal bugs in the system which have existed or been introduced based on a change in the code. Regression testing must verify that one change to a part of the software, did not mess with any of the functionality in another area of the software. When a significant change is made to the underlying code, the software system should be retested based on the functional and integration tests available. Results from previous test cases should match the results found after running the test cases again once a change has been made.

Regression testing is not defined as being done to strictly test the correctness of a program, but is often used to also track the quality of the produced outputs. The quality of the outputs of a program or service is important depending on how critical it is to the system. If the output of a program generates the wrong user name, then it's simple enough to track down the cause of the bug and make the proper fix. If the output of a program produces an exception or a null result, then there is a more critical failure in the code which must be fixed before further testing can proceed.

It is common practice that when developing the software if a bug is found, then a test case should be recorded exposing the bug and it's cause. Thus when code is changed in the future, the same test can be conducted to see if the bug has been re-introduced. Regression testing may also include test cases which attempt to break the system, or produce invalid results. Scenarios such as these may include attempting to login through the administrator page with invalid credentials, or attempting to add a user to the database which already exists in the system.

Usually regression testing is done automatically by first initially recording test cases, and playing them back whenever a change is made. Automation of test cases is time consuming and can be difficult for complex scenarios. At times, it can be more time consuming managing and maintaining automated tests than the development process itself. Automated test cases would be most beneficial once the software system has been designed and completed and moved into the maintenance phase. Considering the KEES project will be short lived and most likely not maintained in the future, manual testing will be sufficient. Manual regression testing will be performed after the foundation to the software has been developed and only whenever a code change is made to any of the server-side scripts or code running on the Arduino microcontroller. For example if a code

change is made to the Arduino program, manual testing should be done to ensure the server-side script for controlling the door lock still functions properly. Testing should also be done to ensure the Arduino microcontroller itself is still functioning properly.

8.0 Summary

The KEES project has provided the group with lots of information regarding home automation and smart home projects. Throughout the research phase the group was able to become familiar with various hardware and different types of components. The group has learned about various power supply configurations and hardware sensors such as the Piezo element, photo, motion, and RGB sensors. The group has also learned about RFID, single-board computers such as the Raspberry Pi and Beagleboard, and microcontrollers such as the Arduino. Research needed to be conducted to determine the type of web camera best suited for the KEES project, and how the locking mechanism would be implemented.

In regards to the software of the KEES project, research was conducted about the various web servers' available, mobile development platforms available, and image processing and voice recognition software both open source and paid. Due to the large amount of references and similar projects around the web, it was not as difficult making software design decisions than hardware design decisions. By breaking down the project into hardware and software, the group was able to separate the amount of work by the different components making up the system. Each member of the group conducted research and planned the design and test phases for specific components. Following the research phase, design challenges were faced by some members in which they were eventually able to overcome.

So far, the KEES project is composed of many parts in which some are integrated with other features while others are completely independent. The hardware of the KEES system will allow entry via RFID and knock pattern recognition. The motion detection sensor will be used and integrated with the software to allow notifications to be sent when a personal presence is detected. The camera will be used to provide a live feed of the viewing area around the KEES system, and integrated with the software to provide face recognition via the OpenCV software libraries. The mobile application will be developed side by side with the web application to provide the administrator various ways of managing and controlling the KEES project.

Through the collaboration of team members and careful planning, the KEES project has thus far been a success. The group is on track for completing a working prototype in the amount of time remaining and has already begun ordering parts and working on the hardware construction. Many things can be learned from the KEES project and each group member has a specific area of

focus in which they can reference in their resumes which will hopefully help with their career. The KEES project experience provides a foundation for group members to build upon in their many years to come in the field of engineering.

9.0 Appendix

Sources:

App:

- http://developer.android.com/guide/practices/screens_support.html#screen-independence
- <http://developer.android.com/guide/topics/resources/providing-resources.html#AlternativeResources>
- http://developer.android.com/guide/practices/screens_support.html
- <http://developer.android.com/about/dashboards/index.html>

Video Streaming MJPEG Streamer:

- http://sourceforge.net/apps/mediawiki/mjpg-streamer/index.php?title=Main_Page

OpenCV:

- http://docs.opencv.org/modules/contrib/doc/facerec/facerec_tutorial.html#fisherfaces
- <http://docs.opencv.org/trunk/modules/contrib/doc/facerec/>
- http://docs.opencv.org/trunk/modules/contrib/doc/facerec/tutorial/facerec_video_recognition.html#introduction
- http://docs.opencv.org/modules/contrib/doc/facerec/tutorial/facerec_save_load.html
- http://docs.opencv.org/doc/tutorials/introduction/display_image/display_image.html

Compatible cameras for Raspberry Pi:

- http://elinux.org/RPi_USB_Webcams

Cameras:

- <http://www.raspberrypi.org/camera>
- http://www.shopping.hp.com/en_US/home-office/-/products/Accessories/Webcams/A5F64AA

- http://logitechenamr.custhelp.com/app/answers/detail/a_id/13852/~/~quickcam-pro-9000-technical-specifications
- <http://www.logitech.com/en-us/product/hd-webcam-c270>
- http://reviews.cnet.com/webcams/logitech-c300-black/4507-6502_7-33916318.html

Motion:

- <http://learn.parallax.com/KickStart/555-28027>
- <http://media.digikey.com/pdf/Data%20Sheets/Panasonic%20Electric%20Works%20PDFs/AMN%20Design%20Manual.pdf>
- Parallax Single-Relay-Board-Guide-v1.0.pdf
- BISS0001 Micro Power PIR Motion Detector IC PDF
- <http://www.digikey.com/us/en/techzone/sensors/resources/articles/optimizing-proximity-sensing.html>

Piezo:

- Gautschi, G (2002). Piezoelectric Sensorics: Force, Strain, Pressure, Acceleration and Acoustic Emission Sensors, Materials and Amplifiers. Springer.
- The Columbia Electronic Encyclopedia, 6th ed. Copyright © 2012, Columbia University Press.
- <http://www.doitpoms.ac.uk/tlplib/piezoelectrics/dipole.php>
- MuRata Piezoelectric Sound Components P37e.pdf

RFID:

- <http://electronics.howstuffworks.com/gadgets/high-tech-gadgets/rfid.htm>
- http://en.wikipedia.org/wiki/Radio-frequency_identification

Embedded Communications:

- <http://www.embedded.com/design/connectivity/4023975/Serial-Protocols-Compared>

- <http://blog.oscarliang.net/raspberry-pi-Arduino-connected-i2c/>

Power:

- EEL 4309 Lab Experiment #3- Linear Voltage Regulators
- μ A7800 SERIES Positive Voltage Regulators PDF
- Texas Instruments LM78xx 3-Terminal Positive Regulators PDF
- Webench Design Report: LM22679
- TI Simple Switcher Step-down Voltage Regulator PDF
- <http://www.smpstech.com/tutorial/t01int.htm>
- Microelectronics: Circuit Analysis and Design: Neamen

Photoresistor:

- <http://www.acroname.com/howto/photoresistor/photoresistor.html>
- http://www.radio-electronics.com/info/data/resistor/lldr/light_dependent_resistor.php
- CDS Photoconductive Cells GL5528 PDF
- VT500 Photoconductive Cells and Analog Optoisolators PDF

Electric Strike:

- <http://www.unikey.com>
- <https://lockitron.com/>
- http://grathio.com/2009/11/secret_knock_detecting_door_lock/
- <http://www.kwikset.com/SmartSecurity/Electronic-Locks.aspx>
- <http://today.ucf.edu/ucf-engineering-alum-gets-in-shark-tank-may-18/>
- <http://idighardware.com/2012/07/fail-safe-vs-fail-secure-when-and-where/>

RGB LED:

- <https://www.sparkfun.com/datasheets/Components/YSL-R596CR3G4B5C-C10.pdf>

- <http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Components/General/YSL-R1047CR4G3BW-F8.pdf>

Permissions:

Piezo Element PDF (P37E.pdf): 7BB-20-6L0 sensor

← REPLY ←← REPLY ALL → FORWARD ...



ccondella
Sat 11/2/2013 2:36 PM

mark as unread

To: intl@murata.co.jp;

Hi, I am an electrical engineering undergraduate senior at the University of Central Florida. I am currently working on a senior design project and it involves the use of piezo element made by your company. I Downloaded the "piezoelectric sound components PDF (P37E.pdf)" and would like to use the schematic of the element "7BB-20-6L0" in my report. The university requires that I obtain permission to use any pictures I may find, I would like to ask for permission to use the image. Thank You- Chris

Permission Status: Pending

Parallax RFID Reader Picture

info@parallax.com

Permissions

Hello,

I am a Computer Engineering student at University of Central Florida working on a senior design project. I would like to ask for permission to use some of the pictures and information in the Technical Document for [RFID Reader Module \(#28140\)](#). This will be used for a student project, and using the information in this document would be a great help. Thanks!

Best Regards,
Jason Wagner

Permission Status: Pending

Parallax RFID Reader Picture

help@ID-Innovations.com

Permissions

Hello,

I am a Computer Engineering student at University of Central Florida working on a senior design project. I would like to ask for permission to use some of the pictures and information in the Technical Document for [ID-2LA](#), [ID-12LA](#), [ID-20LA](#) Low Voltage Series Reader Modules. This will be used for a student project, and using the information in this document would be a great help. Thanks!

Best Regards,
Jason Wagner

Permission Status: Pending

Arduino Model Size Comparison Figure 10

<http://digitaldiner.blogspot.com/2012/10/Arduino-uno-vs-beaglebone-vs-raspberry.html>

Permission Status: Pending

Electric Strike Figure 12

Permission Status: This file is made available under the Creative Commons CC0 1.0 Universal Public Domain Dedication

Server Memory Usage Comparison

http://wiki.dreamhost.com/Web_Server_Performance_Comparison#Memory_Usage

Content available under:

Creative Commons Attribution-ShareAlike 3.0 United States

Permission Status: Accepted

Voice Recognition / TTS

Speakjet IC Synthesizer Schematic

<http://www.practicalArduino.com/schematics/speech-synthesizer-schematic.jpg>

Permission request to use image found in your article

Hugh Blemings <hugh@blemings.org>

Tue, Nov 26, 2013 at 11:16 PM

To: "S.Demole" <sdemole@gmail.com>, Jonathan Oxer <jon@oxer.com.au>

Hi Samuel,

Thankyou for your email, nice to hear from you!

On 26/11/13 10:54, S.Demole wrote:

> Hi there, we have been asked by our instructor to request permission for
> any images found online which we would like to feature in our University
> Senior Design project document.

Cool!

> I'd like to ask for permission to use an image found in your article:
> <http://www.practicalarduino.com/projects/speech-synthesizer>
>
> Here is the specific image I would like to use with your permission:
> <http://www.practicalarduino.com/schematics/speech-synthesizer-schematic.jpg>

I had a chat with Jon (CC'd) and we have no objection to you using the image, if appropriate please note the source in your footnotes or bibliography though.

Good luck with the project, hope it's going well!

Cheers,
Hugh

Permission Status: Accepted

Emic 2 TTS Synthesizer

<http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Components/General/30016-Emic2TextToSpeech-v1.1.pdf>

Permission request to use image found in your data sheet

2 messages

S.Demole <sdemole@gmail.com>

Mon, Nov 25, 2013 at 7:02 PM

To: support@parallax.com

Hi there, we have been asked by our instructor to request permission for any images found online which we would like to feature in our University Senior Design project document.

I'd like to ask for permission to use an image found in your pdf:
<http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Components/General/30016-Emic2TextToSpeech-v1.1.pdf>

The specific image I would like to use, with your permission, is the example circuit of the Emic 2 Text-to-Speed Module found on page 2.

Thank you,
Samuel Demole

Support Account <support@parallax.com>

Mon, Nov 25, 2013 at 7:10 PM

To: "S.Demole" <sdemole@gmail.com>

Hello,

You can use our diagrams and documentation freely for use in an educational role. The only time we object to use of our documentation or images is if they're modified and re-distributed. Good luck on your project.

Respectfully,
Chris Savage
Engineering Tech, Parallax Inc.

[Quoted text hidden]

-

Permission Status: Accepted


MVC Controller

<http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>

Summary [\[edit\]](#)

Description	English: The model, view, and controller (MVC) pattern relative to the user.
Date	May 2010
Source	Own work
Author	RegisFrey
Other versions	MVC-Process.png

Licensing [\[edit\]](#)

	I, the copyright holder of this work, release this work into the public domain . This applies worldwide. In some countries this may not be legally possible; if so: <i>I grant anyone the right to use this work for any purpose, without any conditions, unless such conditions are required by law.</i>
---	---

Permission Status: Accepted