

K.E.E.S. - Keyless Electronic Entry System

Chris Condella, Jason Wagner, Samuel Demole
and Joshua Baxter

Dept. of Electrical Engineering and Computer
Science, University of Central Florida, Orlando,
Florida, 32816-2450

Abstract — Over the recent years, home automation and access control systems have become increasingly popular in everyday households and businesses. The recent advance in technology has increased the popularity 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. This paper describes the design, integration and capability of modern access control systems.

Index Terms — Access Control, Radiofrequency Identification, Piezoelectric Materials, Web Services, Face Recognition

I. INTRODUCTION

All of the members in Group 17 Fall/Spring 2013-2014 wanted to take this opportunity to build a useful Senior Design Project that can improve the ease of access for any individual to any home. The KEES provides an innovative way to gain access through the entrance via an everyday door, of a home or office as well as maximizing control and ease of use. The idea of KEES is to allow access to a home or business with ease and comfort via face detection/recognition and RFID or secret knock. An electric strike will be controlled with a micro controller in collaboration with a Raspberry Pi in the system. Access through the door can be granted at the door itself or remotely via the KEES webapp. The hardware of the KEES will allow entry via RFID and knock pattern recognition aka “secret knock” once the individual is recognized within the system. The camera will be used to detect and recognize, as well as provide a live feed of the viewing area around the KEES, and integrated with the software to provide face recognition via the OpenCV software libraries. A web-based application will provide the administrator various ways of managing and controlling the system. The fact that Group 17 is self-funded, the team wanted to integrate and implement the KEES design for the lowest cost but highest quality.

II. SPECIFICATIONS

The design specifications for the KEES project are as follows:

- 1) Door dimensions 4'6"H x 2'6"W x 1'8"D
- 2) Camera snapshot resolution of 320x240
- 3) Face Recognition within 5s
- 4) Video Stream at approximately 2-3fps
- 5) Face Database of up to 5 people
- 6) Unlocked door returns to locked state in 3s
- 7) RFID enclosure dimensions (3" x 4" x 2)
- 8) RFID minimum detection range 3"
- 9) Piezoelectric element resonant freq. 6.3 ± 0.6 kHz
- 10) MCU operational Voltage 5V +/- 0.5V
- 11) MCU lock digital output current 50mA

III. SYSTEM COMPONENTS AND DESIGN

A. System Overview

KEES consists of two main processing units: an ATmega microcontroller and a Raspberry Pi SoC. The RFID and knock sensor are connected to the ATmega, while the camera is connected to the Raspberry Pi. The Raspberry Pi also hosts the KEES webapp used to control the system. The KEES includes its own power system which powers all of the hardware. An overview of the system is shown below in Figure 1.

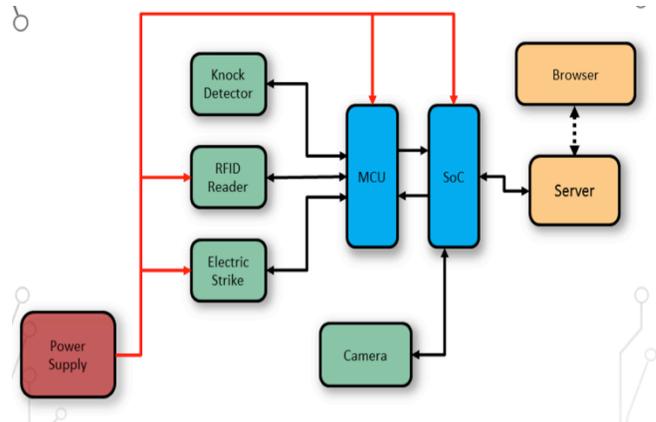


Figure 1. KEES System Block Diagram

B. Door Layout

A small scaled down door with a frame was built to demonstrate the KEES. The Electric Strike is installed in the door so that the door cannot be opened unless the strike is unlocked. The RFID reader is located on the outside of the door on the frame, and the knock sensor is actually located on the door near the handle in order to pick up vibrations from knocks. The camera is located at the top of the door. Behind the door on the frame next to the handle is where the PCB is housed. The Raspberry Pi

will be positioned on top of the door. The door layout is shown below in Figure 2.



Figure 2. KEES Door Layout

C. ATmega

The microcontroller chosen for the KEES project is the ATmega328, we chose this MCU because all team members are familiar with the C language and the IDE. The ATmega328 has an 8-bit architecture, 20 MHz operating frequency, and comes equipped with 32 kB of onboard Flash memory. The minimal architecture, low power consumption, and proliferated software support on the Arduino development platform where some of the reasons the ATmega 328 was selected. Also, the ATmega328 comes available in a standard 28-DIP package, which a wide amount of variation in the hardware scheme.

D. Electric Strike

The 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.

The two types of configurations “fail secure” and “fail safe”. A fail-secure also called normally opened (N.O. 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 secure (normally opened) type electric strike will most likely be used for this type of

project because it is more ideal for security purposes.

E. RFID

The RFID reader implemented in the project is an ID-12LA, it is a very simple to use RFID reader module designed by ID Innovations. The ID-12LA can read 125 kHz EM4001 64bit RFID tags. Once a tag is read the information is encoded as an ASCII string, and transferred to the microcontroller via a RS232 output. The module also has a built-in antenna, and power module that operates at 2.8V - 5V. Also the ID-12LA is ideal for the KEES project because of the small form factor 25x26mm and the ideal range of approximately 5".

F. Knock Sensor

The Knock Sensor that was selected is a piezoelectric buzzer that can be used to pick up vibrations from knocks. 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. The piezo device is read through an analog input on the microcontroller telling the microcontroller to send the unlock signal to the electric strike. For the use of the application the “7BB-20-6L0” sensor 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).

G. Raspberry Pi

The Raspberry Pi is a \$35 SoC with a 700MHz ARM1176JZFS processor that can overclock up to 1GHz. It has 512 MB SDRAM, 8 pins, and a slot for an SD card. Its peripherals include 2 USB hosts and an HDMI port. It also has a 10/100 Mbps Ethernet port as well as a camera serial interface. It is running a Debian based OS called Raspbian which is the recommended OS for the Raspberry Pi and has support for hardware based floating point calculations as well as the latest drivers for the Raspberry Pi Camera.

H. Raspberry Pi Camera

The camera that is connected to the Raspberry Pi is the Raspberry Pi camera, a camera made specifically for the Raspberry Pi. It connects to the Raspberry Pi via the camera serial interface and it has a much lower latency compared to USB webcams, enabling a higher frame rate. It also supports a wide range of video resolutions up to 1080p, but a resolution of 320 x 240 will be used to

facilitate faster processing. It is also very small and light with dimensions of 20mm x 25mm. To capture frames from the camera for processing, a Python library called Picamera 1.3 is used.

I. Face Recognition and Video Stream

The face recognition and video stream software system developed on the Raspberry Pi is able to detect faces using a method called cascade classification. It is also able to save people to a face database which is accomplished by training the face recognition software in real time. The SDK that is used for image processing capabilities is OpenCV 2.4.6, an open source computer vision library that includes many optimized functions, including face recognition algorithms. Picamera 1.3, the API used to get frames from the camera, can capture frames in a format that is compatible with OpenCV. For the video stream capability a light-weight open source software called MJPEG Streamer is used to upload frames captured from the camera to the web server.

J. Embedded Communication

The Raspberry Pi needs to communicate to the microcontroller to update the state of the system as well as retrieve status of the locking system. The I2C protocol was implemented for communication between these devices because both architectures support this protocol, and it matched the needs for the design of the KEES project. The Atmega 328 is setup as the slave in this scheme and the Raspberry Pi is the master. The team designed a script to allow the Pi to lock, unlock, and notify the embedded system that a valid face has been recognized at the door. The scripts also allow the Pi to track when a RFID, Knock, or Unlock has been detected.

K. Webapp

The Webapp frontend is built on the Bootstrap 3.0 framework which contains numerous HTML/CSS based design templates for various interface components. It allows for a responsive design across multiple browsers and different platforms. For dynamic content the jQuery javascript library is used in addition to the standard HTML code and CSS stylesheets. Combined it allows for a fully featured web-based application capable of managing and controlling the KEES remotely from any device with an internet connection. The webapp includes a secure login authentication system for the owner of the KEES. From the Home page the owner has the ability to control the electric strike, view the live stream from the Raspberry Pi camera, add a guest to the database, and view a history of the most recent events. The Admin page allows the owner to enable master mode for the RFID and knock sensor subsystems, and manage the users in the face recognition database.

See Figure 3 below for screenshots of the KEES webapp UI.

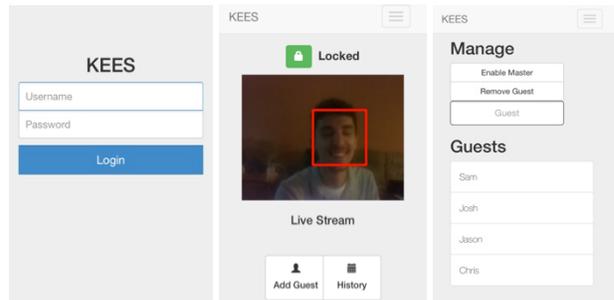


Figure 3 KEES Login, Home, Admin pages

L. I2C Communication

The communication protocol used between the Raspberry Pi and ATmega328 microcontroller is I2C. I2C is a multimaster serial single-ended computer bus invented by the Philips semiconductor division. The group chose to the I2C serial communication protocol because of the need for a simple two-way serial connection. The Raspberry Pi will be able send commands in a byte-stream to the microcontroller to control the state of the embedded system. The microcontroller can also send data to the Raspberry Pi to notify the main system of events that occur. The use cases for communication between the two systems include informing the microcontroller when a face is recognized, and sending commands to unlock the electric strike and enable master mode. The microcontroller must also inform the Raspberry Pi of events that occur such as a valid/invalid knock or RFID, and when the door has been unlocked. A table of valid states for the embedded system can be found below in Table 1.

Code	Function
0	Scan
1	Validate
2	Allow
3	Deny
4	Master
5	Do Nothing

Table 1. Embedded States

M. Power

The KEES relies on two different power sources. A standard 15VDC 8A 120 Watt wall adapter and six rechargeable 1.2V 2300mAh NiMH AA batteries that equal up to 7.2V. The voltages are regulated down to 5V as required for the MCU and Raspberry Pi.

The system current draw from highest to lowest is 1) Electric Strike @ ~1.8A, 2) Raspberry Pi @ ~ 1 A, 3) Embedded Circuit @ ~200mA Max. The max current draw was rounded up to 5A to be safe. The LM1084 voltage regulator will be used to regulate the voltages down to a safe operating voltage of 5VDC for the microcontroller and raspberry pi. This regulator allows a max current of 5 amps to be delivered to the system. A heatsink will be used to help dissipate energy.

Since the Raspberry Pi is constantly pulling current when turned on, it is the part of the system that requires the most energy and has first claim in the power system. The electric strike technically pulls more current but is only periodically operational, opening for only seconds at a time.

The system will always ideally be plugged in and powered by the wall adapter however, in case of a power failure or disconnect the batteries will serve as a backup for the system. While the wall power is on, a pnp power transistor (TIP 42) will allow the raspberry pi to run and twin Schottky diodes (DE5PC3) will allow entire system to pull power from the wall.

The battery back up will only power the MCU and the electric strike if the wall adapter is not present. This means the functionality of the system (RFID + Knock) will still be present except for the web service (Rasp. Pi). The logic of the power system can be followed by the flow charts below.

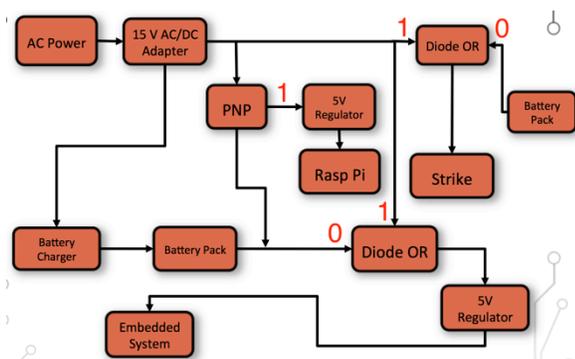


Figure 4. AC Power Connected... 1 = ON, 0 = OFF

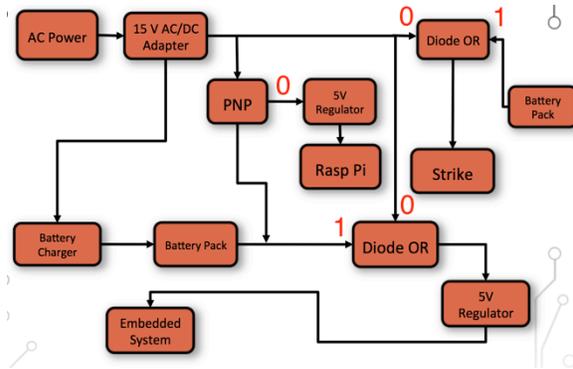


Figure 5. AC Power Disconnected

While the AC power is connected there is a trickle charging circuit that will slowly charge the NiMH batteries. The constant charging current can be varied via trimmer potentiometer from a few mA to a couple hundred mA. With a lower charging current this yields a longer charging period. The batteries will not get overly hot because the current eventually reduces as the batteries become fully charged. It is always good practice to monitor batteries for safety reasons. In the event of overheating the batteries can be easily unplugged and/or switched off.

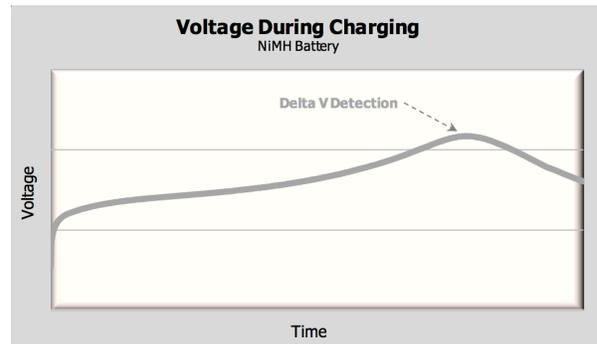


Figure 6. NiMH voltage charge curve

Running on solely batteries the system stays functional for approximately 24 hours. However the lifespan will depend on how much use the system gets (i.e. how often the door gets unlocked). However the good thing about this system is that if power is completely lost (AC and Battery) the door will function as an ordinary door!

IV. SOFTWARE DETAIL

The main software components of the KEES are embedded software running on the ATmega, and the higher level software running on the Raspberry Pi, which includes the webserver, listener services, face recognition, and video stream. For development of the embedded software the Arduino IDE was used. The face recognition

software was programmed in Python since it is the official language of the Raspberry Pi. Rasbian, one of the available OS's for the Raspberry Pi, comes pre-installed with Python support and an official Python camera interface, Picamera 1.3, which is available in the OS repository. Furthermore, OpenCV has robust Python support. The listener services and scripts to send commands to the microcontroller are also written in Python. The KEES webapp frontend is built using the Bootstrap framework and jQuery, and the webserver itself is written using Node.js due to its light-weight architecture, performance, and scalability.

A. Face Recognition and Video Stream

Once frames with a resolution of 320 x 240 are captured from the camera, they are saved on a location in the file system that is mounted to RAM in jpeg format so that MJPEG Streamer can capture the frame, which can then be uploaded to the web server. The frames are saved in RAM due to reduce the effects of read and write latency. MJPEG Streamer serves as the middle man between the camera driver and the webserver since it is able to listen to a directory for a jpeg file in a manner that uses less than 1% of CPU processing power. It can handle a frame rate of approximately 4 fps. The video stream process is shown below in Figure 7.

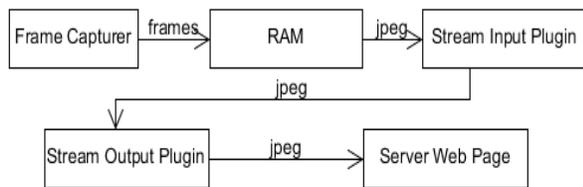


Figure 7. Video Stream Process

The face recognition software reads a state file to determine if the face database has anyone in it, or if it is empty. Then, the frame saver is launched, and it is provided with the state information. Once the frames are captured from the camera via the Python camera API Picamera 1.3, the frames are converted to grayscale and passed into the face detector. Converting to grayscale greatly speeds up the process of detecting a face since a grayscale frame is 1/3 the size of a color frame.

The face detector module will use a method called cascade classification to detect any faces in the frame, and will be able to detect any faces that are within approximately 2.5 feet distance from the camera. The face that is extracted then undergoes pre-processing to prepare it for face recognition. This includes resizing the face to a standardized size, such as 100 x 100, reducing

noise in the image by applying a kernel, binary histograms to even out brightness and contrast in the face, as well as cropping as much as the background out as possible using a method known as Canny edge detection. Another method that can be used to crop the face is by detecting the eyes, and using the distance between the two eyes to crop the face. This removes more of the background than with using Canny edge detection, and the frame can be reduced to only contain the main components of the face: eyes, mouth, and nose. The face recognition module will use the algorithm known as Local Binary Patterns to recognize a person's face since it allows for fast retraining due to its ability to simply add a person's face data to the existing face data without retraining the entire face recognition module, unlike the other two algorithms Fisherfaces and Eigenfaces.

There are three different states of the face recognition system: uninitialized mode, training mode, and face prediction mode. When the face recognition software is launched without any people in the database it is in uninitialized mode meaning that whenever a face enters the system, it will be categorized as "Unknown." Training mode is activated by the server whenever it sends a command to add someone to the database. When this occurs, snapshots of the person's face will be taken, and the face recognizer will add the person's data so that in subsequent times the system will attempt to recognize the person and will enter face prediction mode. When the server sends a command to remove someone from the database, the system will retain all of their data but will simply mark the person's name as "Unknown" in order to avoid the time it would take to retrain the face recognizer. The person will no longer be able to access the RFID module.

In face prediction mode, there is at least one person in the database and the software will attempt to determine if the person in front of the door is in the database or not. The face recognition software will make a prediction to match the person's face with a face in the database that is most similar. In order to determine if that match is valid, threshold values are used. If the confidence of the prediction is less than the threshold value, the person will be categorized as 'Unknown' and they will not be able to access the RFID system. The confidence value can be static, meaning that one value is set for all guests, or it can be dynamic meaning that this value is individually set for each guest during training mode.

If the face of the person is recognized, an I2C command is sent to the ATmega, and the RFID system is activated, allowing the guest to swipe his RFID tag. In both cases (face recognized, face not recognized), the server is informed of the result via writing to a file that the server periodically checks for updates. The face recognition software informs the server of the date and time of the

result in addition to the identity of the person. An overview of the face recognition system is shown below in Figure 8.

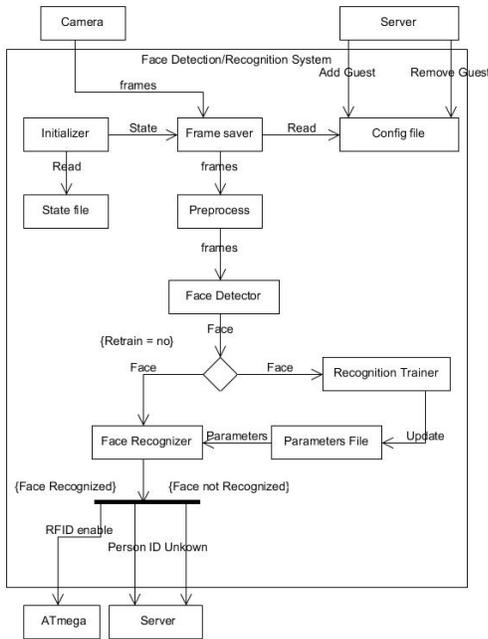


Figure 8. Face Recognition System Overview

The Face Recognition software system is comprised of three main classes: one for face detection, face recognition, retraining the face recognizer, and a main controller class which contains instantiations of all the other classes. The class diagram is shown below in Figure 9.

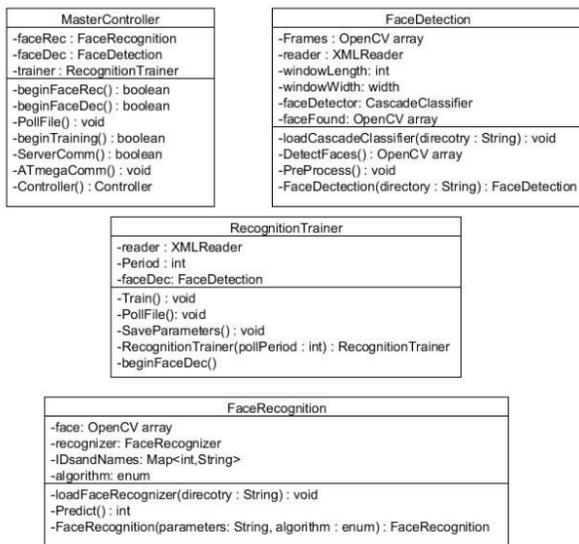


Figure 9. Face Recognition Class Diagram

B. Webserver

The webserver runs inside Node.js, a software platform for scalable server-side applications. It is based on a single threaded, event-driven, non-blocking I/O architecture and utilizes Google's V8 Javascript engine. It relies on the express package, a node.js web application framework providing libraries to easily model an MVC architecture. Simple RESTful services can be created to send data in JSON format to the client-side such as the list of current guests in the database or event history from the log file. With express, requests can be handled and routed accordingly with ease to redirect the user to another page, handle data received by the client, and/or return data to the client. The server architecture can be seen in Figure 10 below.

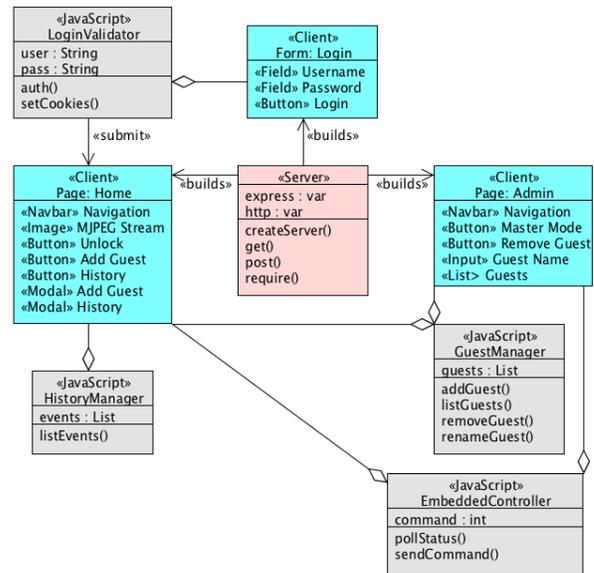


Figure 10. Webserver architecture

A simple use case for controlling the electric strike is as follows. The user presses the unlock button on the webapp Home page where a request is sent to the server. The server handles the request and then executes a Python script to send the unlock command over I2C to the microcontroller. Enabling Master mode for the KEES is done in a similar manner. Node also interacts with the file system when reading log files and guest data. The server sends data back to the client when handling requests for viewing the recent event history or for managing guests in the system. When a guest is added into the system, the name of the guest to be added is sent along with the request to add a guest. Upon receiving the request on the server-side, the guest name is extracted from the request and passed as an argument to the face recognition training program.

C. Embedded

The software for the microcontroller was designed as a Mealy finite-state machine where the output values are determined both by its current state and the current inputs. The state for the embedded system are as follows SCAN, VALIDATE, ALLOW, DENY, MASTER, and DO_NOTHING. The default state that the embedded side is in is the DO_NOTHING state. While in this state the Atmega328 is waiting for indication from the Pi that there has been a valid face recognized by the system. Once a valid person is recognized the system will enter the SCAN state. While in SCAN the microcontroller will be listening to the analog pin the piezo is connected to, and the RS232 connection with the RFID reader. If a knock or RFID card is detected the system will enter the VALIDATE state. The microcontroller will determine if the knock matches the knock saved to the EEPROM, or an RFID is scanned the system will determine if the RFID card is in the system or if the RFID isn't in the system. After the system validates or invalidates the knock, or card the user will be allowed or denied entry via the ALLOW, or DENY states. if Allowed the door will open for 5 seconds, and change the LED indicator green. Inversely if denied the door will stay locked and the LED indicator will turn red. The MASTER mode is the main programming mode for the embedded system. The MASTER mode can be accessed in 2 ways, either the user sets the mode through the web application, or the user scans the MASTER RFID card. While in master mode the user will be able to update the knock that is saved in the EEPROM, or the user can add or delete RFID cards from the system.

V. PCB

The KEES schematic shown below in Figure 11 was designed in Eagle CAD. This shows the entire simplistic embedded system.

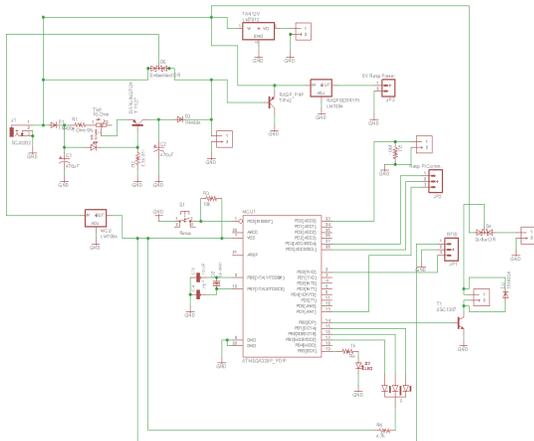


Figure 11. PCB EagleCAD Schematic

During the testing phase of the project prototype, there were system failures and inconsistencies due to unknown, at the time, factors among the system components and the system integration. This led to several design changes and component changes along the way of testing the system as a whole. The final integrated system can be seen in the power section also in our final PCB design below.

The PCB was designed using Eagle CAD and manufactured through OSH Park. OSH Park is a community PCB order company. The company brings high quality, lead free boards (ENIG finish), manufactured in the USA. The standard two-layer board costs \$5 per square inch, which includes three copies of the design. The group decided on a two layer-board with the top layer devoted to signal traces and the bottom layer a ground layer. The dimensions of the 2 layer board are 3.94x3.15 inches and cost approx. ~\$60.00 US.

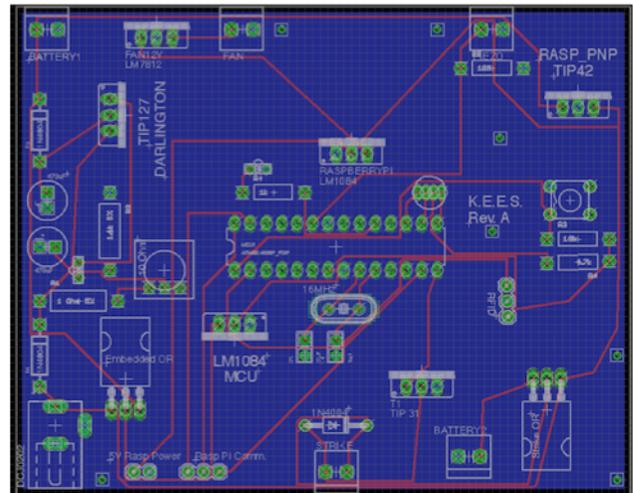


Figure 12. PCB Eagle Fab. Board

VI. CONCLUSION

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.

VII. Acknowledgement

Group 17 of the Fall 2013 - Spring 2014 Senior Design class would like to acknowledge and thank the Department of Electrical Engineering and Computer Science at the University of Central Florida.

VIII. THE TEAM



Samuel Demole is currently a senior at the University of Central Florida. He is expected to graduate with his Bachelor's of Science in Computer Engineering on May 1, 2014. Upon graduation he plans to pursue a career as a Software Engineer. He enjoys programming and gaming.



Joshua Baxter is currently a senior at the University of Central Florida. He is expected to graduate with his Bachelor's of Science in Computer Engineering on May 1, 2014. Upon graduation he plans to pursue a career as a Software Engineer. His hobbies include working out and programming.



Chris Condella is currently a senior at the University of Central Florida. He is expected to graduate with his Bachelor's of Science in Electrical Engineering on May 1, 2014. After graduation he plans to pursue a career as an Electrical Engineer. He enjoys playing guitar and astrophysics.



Jason Wagner is currently a senior at the University of Central Florida. He is expected to graduate with his Bachelor's of Science in Computer Engineering on May 1, 2014. Upon graduation he plans to pursue a career as a Software Engineer. He enjoys playing guitar.

IX. References

- [1] OpenCV Face Recognition Tutorial
<http://docs.opencv.org/modules/contrib/doc/facerec/facerec_tutorial.html#fisherfaces>
- [2] OpenCV Face Recognition in Videos
<http://docs.opencv.org/trunk/modules/contrib/doc/facerec/tutorial/facerec_video_recognition.html#introduction>
- [3] Raspberry Pi Python Camera API
<<http://picamera.readthedocs.org/en/release-1.3/>>
- [4] MJPG Streamer
<http://sourceforge.net/apps/mediawiki/mjpg-streamer/index.php?title=Main_Page>
- [5] LM1084 Texas Instruments, 1999. March 2013.
<<http://www.ti.com/lit/ds/symlink/lm1084.pdf>>
- [6] DE5PC3 Schottky Twin Diode SHINDENGEN, March 2013.
<<http://www.mouser.com/ds/2/366/J532DE5PC3-191003.pdf>>
- [7] Express API
<<http://expressjs.com/3x/api.html>>
- [8] jQuery API
<<http://api.jquery.com>>
- [9] Bootstrap Framework References
<<http://getbootstrap.com/javascript/>>
<<http://getbootstrap.com/components/>>