Voice Coded Lock

ECE 445 Team #62 Project Proposal

Aman Thombre
Logan Greuel

TA: Zicheng Ma
02/23/2023

Contents

1 Introduction **2**
1.1 Problem 2
1.2 Solution 2
1.3 Visual Aid 3
1.4 High Level Requirements 3
2 Design **4**
2.1 Block Diagram 4
2.2 Physical Design 4
2.3 Block Descriptions 5
2.3.1 User Interface 5
2.3.2 Keyphrase Recognition 6
2.3.3 Door Lock 7
2.3.4 Microcontroller on a PCB 7
2.3.5 Power Supply 8
2.4 Risk Analysis 9
3 Cost and Schedule **9**
4 Ethics and Safety **11**
5 References **12**

1 Introduction

1.1 Problem

Currently, accessing secure areas usually requires some type of access card or keys. These can easily be misplaced or left at home, leading to being locked out of your area of work Additionally, solutions such as access cards have been known to have security vulnerabilities, exposing government buildings, schools, factories, and companies [1]. These security flaws are a big risk, as according to the FBI, there were over 340,000 burglaries of nonresidence properties in 2019, with an average value of $8781 being stolen [2].

Using access cards or key locks can also be hard to operate with your hands full; trying to get an iCard out while accessing a lab in the ECEB while holding a laptop and FPGA can be quite difficult. Other keyless options requiring physical contact, such as a keypad, may also pose a health concern for some users, especially during cold/flu seasons or a pandemic. These issues show a need for a new form of keyless, contactless entry to secure areas.

1.2 Solution

Our proposed solution is to implement an audio-based locking system for a door. We plan to create an attachment on or near a door which will listen for a user's voice, and upon hearing a user saying some keyphrase, the device will automatically unlock the door. This system will use keyphrase recognition to identify an audio password, allowing for hands free operation of a lock. Keyphrase recognition will be used to verify that the user has access, as only verified users should know the password for the locking system.

This solution eliminates the need for any physical component used to gain access, as only the spoken password is required. The key will only be used as a backup in the event of power failure (so entry would still be possible). This will allow for hands-free operation of the lock, and help reduce the risk of stolen, misplaced, or copied access cards and keys. This solution is intended more for access points with many authorized users rather than for private homes; our solution allows for multiple users to gain access to an area using a single, shared audio password.

1.3 Visual Aid

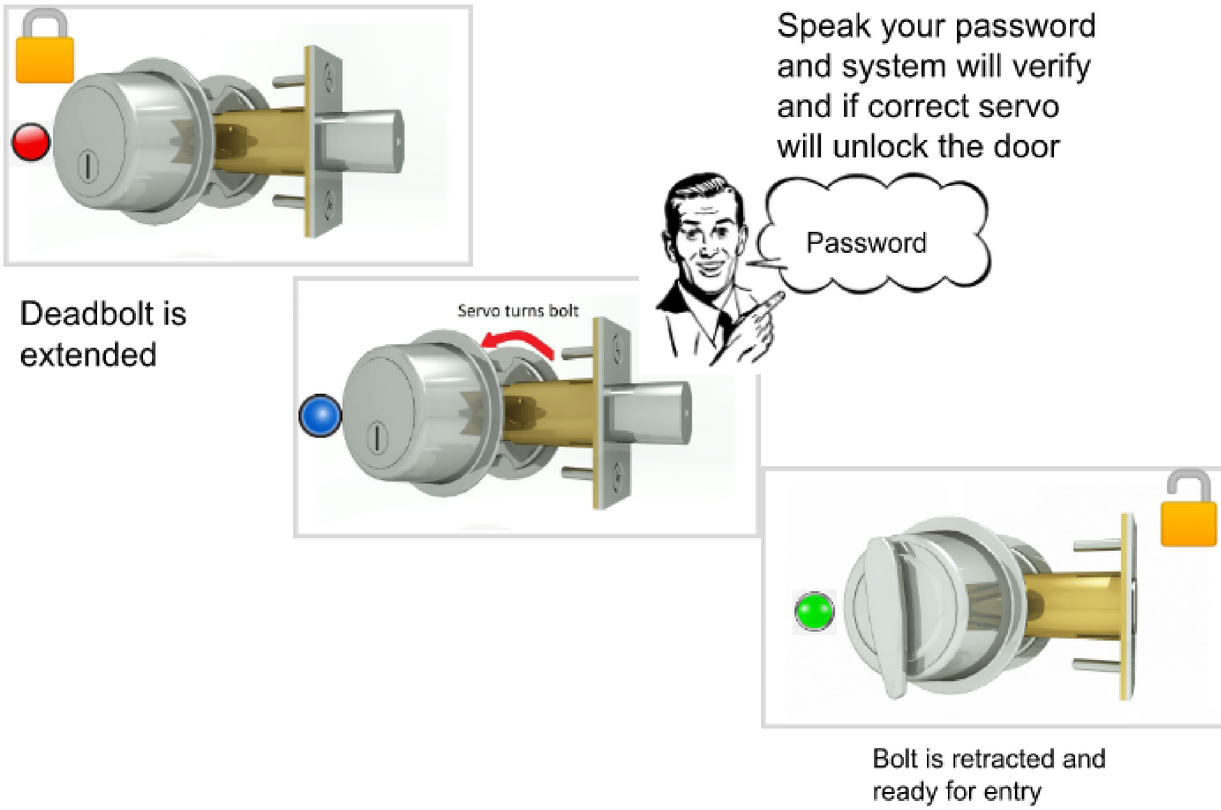


Figure 1: Visual Aid

1.4 High Level Requirements

1. Keyphrase recognition should be able to consistently correctly classify audio password as correct/incorrect.
2. Operation of the locking system, from time of saying password to unlocking should be performed in reasonable time (<8 sec).
3. System should be able to operate a door lock automatically.

2 Design

2.1 Block Diagram

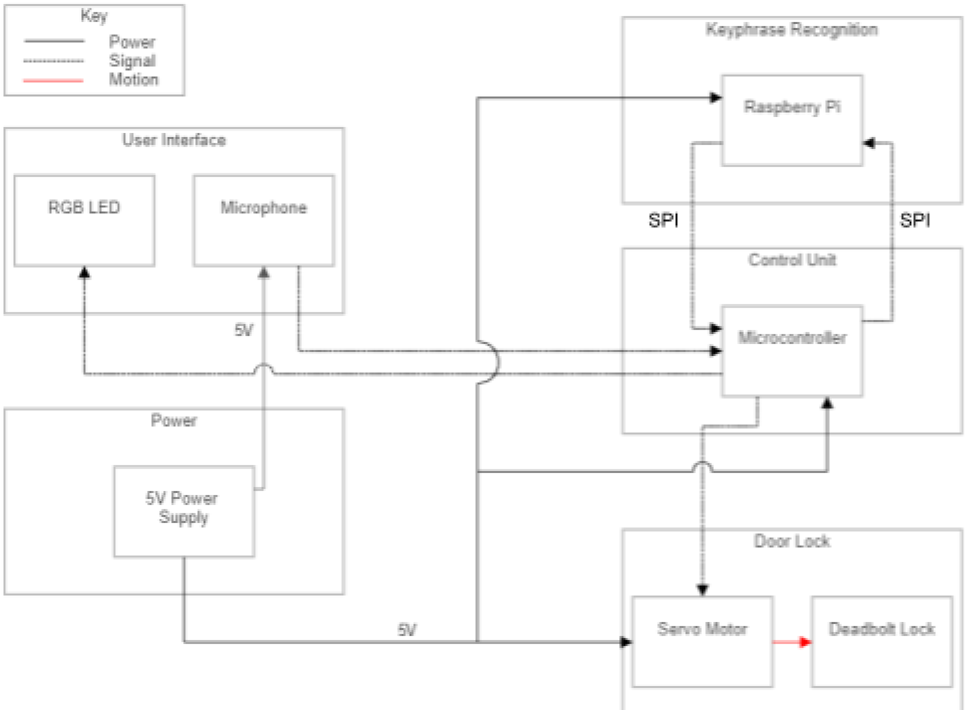
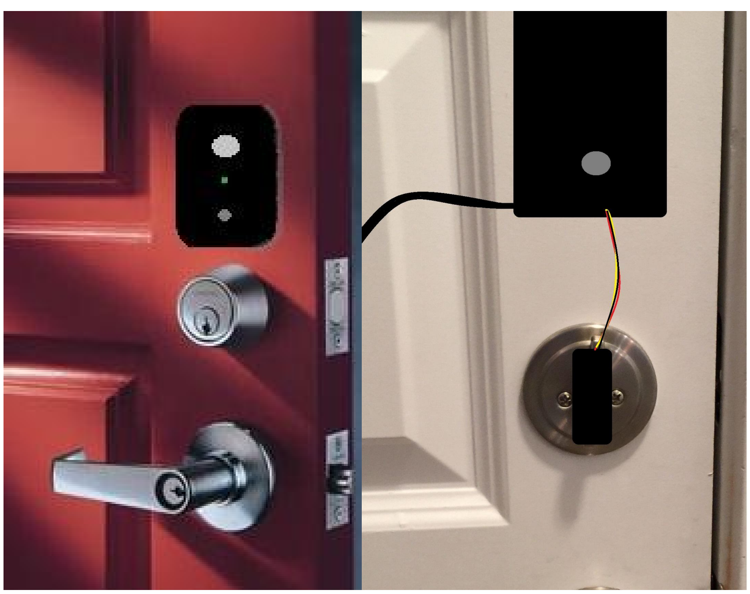


Figure 2: Block Diagram

2.2 Physical Design



Our design will place a servo motor onto an existing deadbolt lock and then have an additional unit connected where the other components are housed. The exterior side will have the microphone, LED, and a lock button while the internal will have a lock button. The unit will be powered by an adapter and circuitry will be inside. (Servo mounting will be done by Machine Shop)

Figure 3 (left) and Figure 4 (right) showing exterior and interior view

2.3 Block Descriptions

2.3.1 User Interface

The user will interact with our locking system through a microphone and LEDs. Keyphrases will be listened for using a microphone, which will send the audio signal it records in real time to our microcontroller. An RGB LED will be used to signal to the user the state of the locking system - we plan to use different colors to indicate locked, unlocked, and listening.

Microphone

- The microphone will record user input, and send data to the microcontroller.
- *Requirements:* The microphone must send audio recording to the microcontroller in real time, and voltage to microphone will be set to match input required by Raspberry Pi (5V) [3].

RGB LED

- The RGB LED will display a different color for each status of the lock (locked, unlocked, listening).
- *Requirements:* The LED must be able to display at least three colors, depending on inputs given and have forward voltage of 2-4.5V [4] .

Table 1: RV for User Interface Subsystem

Requirements	**Verification**
• The microphone must be able to accurately record users speaking at a reasonable volume at a distance of 3 feet.	• Measure microphone output at baseline, with no audio input. The microphone output should only be noise. • Measure the microphone output when playing a fixed, 440 Hz tone. The microphone should output a 440 ± 5% Hz sine wave. • Measure the microphone output while a user is speaking into the microphone at a reasonable volume (the same volume used for day to day speech), from a distance of 3 feet away and ensure that a reasonable output is given.
• The LED will display three different colors based on status: ○ Red, meaning door locked ○ Green, meaning door open	• Initialize the lock system with no one talking and no ambient sound, ensure that the LED is red. • Then, speak an incorrect password to

○ Blue, meaning processing	the system. The LED should turn blue for processing time, and then turn back to red. ● Then, speak the correct password to the system. The LED should turn blue for processing, and then green as the door unlocks.

2.3.2 Keyphrase Recognition

Audio signals will be fed from the microcontroller to a Raspberry Pi Zero, which will perform keyphrase recognition. We plan to code this software in Python, and use some machine learning model (determined by which models give us best results in testing) to determine whether a given keyphrase is correct or incorrect. Current plans for models include hidden Markov Models and convolutional neural nets. Signals will be fed from the Raspberry Pi to the microcontroller, signaling whether the keyphrase heard was correct or incorrect.

Raspberry Pi Zero

- Will perform keyphrase recognition based on inputs given from microcontroller.
- Will send a signal to the microcontroller if the audio password was recognized.
- *Requirements*: Software must be able to consistently distinguish correct/incorrect keyphrases from one another with >80% accuracy, which is around what is achieved using probabilistic models [5]. Must be able to provide 3.3V at GPIO pins [6].

Table 2: RV for Keyphrase Recognition Subsystem

Requirements	**Verification**
● The keyphrase recognition software should be able to handle audio inputs up to 5 seconds long.	● Audio inputs up to 5 seconds long will be provided to the keyphrase recognition software. ● Ensure that the software can accurately detect the correct password if it is uttered in the 5 second audio input, passing accuracy requirements detailed below.
● Keyphrase recognition should not take more than 5 ± 1 seconds.	● After inputs are fed to the keyphrase recognition system, the Python time package will be used to time how long keyphrase recognition takes.
● Should be able to distinguish correct/incorrect keyphrases with at	● Each team member will provide 20 attempts to unlock the door (10 with

least 80% ± 5% accuracy. ● Precision, recall, and specificity should also pass this same benchmark.	the correct password and 10 with the incorrect password for 40 attempts total). ● Results will be noted in a confusion matrix, which will be used to calculate accuracy, precision, recall, and specificity. ○ Accuracy = TP+TN/Total ○ Precision = TP/TP+FP ○ Recall = TP/TP+FN ○ Specificity = TN/TN+FP

2.3.3 Door Lock

We plan to use a deadbolt style lock which retracts when audio is verified. This can be achieved by using a servo motor to engage/disengage the lock. Lock should still be accessible with a key in the event of power failure (to make entry possible still). Servo motor is run with a PWM signal.

Servo Motor

- A servo motor will be used to operate the deadbolt.
- *Requirements:* Servo motor must be able to operate the deadbolt, and requires 4.8-6.0V (will be run at 5V) [7].

Table 3: RV for Door Lock Subsystem

Requirements	Verification
● The servo motor must have enough torque to operate the deadbolt lock.	● Assemble servo onto the lock and test varying PWM signals to check that lock will turn

2.3.4 Microcontroller on a PCB

The microcontroller will be used to send signals to/from our user interface, perform some processing of the audio signal, and send signals to our door locking subsystem. The microcontroller will receive audio signals from the microphone, and when an audio signal passes over a certain threshold, the microcontroller will start sending this audio signal to the Raspberry Pi for keyphrase recognition. When the audio signal is verified, the microcontroller will send a signal to the locking system to disengage the lock, and after a period of time, send a signal to re-engage the lock. The microcontroller will also send signals to our RGB LED, displaying whether the door is locked, unlocked, or whether a phrase is being processed.

Microcontroller

- Will be used to send signals from the microphone to Raspberry Pi.
- Will be used to take signals from Raspberry Pi and operate door lock and LED.
- *Requirements:* Must be able to sink/source 5V +/- 5% at GPIO pins to send/receive inputs, must have an ADC to receive microphone inputs. Must be able to communicate via SPI to Raspberry Pi at speeds greater than 4.5 MBPS.

Table 4: RV for Microcontroller Subsystem

Requirements	**Verification**
● Must be able to convert analog audio input into digital signals for raspberry pi zero	● Receive input from the microphone and run ADC converter to confirm digital output recognizable by Raspberry pi
● Send PWM signals to the Servo to control locking and unlocking controls	● Connect servo and confirm output from microcontroller will rotate into desired positions

2.3.5 Power Supply

Every component in our design requires power, so we will need a consistent supply. Since our design will be mounted on a door, we will use an AC voltage adaptor in a nearby outlet to power our design. All the components are able to be run off of 5V DC and current draw is primarily dominated by the Raspberry Pi, which uses at maximum 750mA. Anything over 1A would provide sufficient current.

AC Adaptor

- Convert the AC wall power to 5V DC for our system to use.
- *Requirements:* Convert AC 120-240V power from wall into DC 5V +/- 5% for our circuit to use.

Table 5: RV for Power Subsystem

Requirements	**Verification**
● Provide 5V DC +/- 5% from an input of 120-240V AC	● Connect adaptor to multimeter to test for voltage output of 5V +/- 5%
● Provide at least 1 A of current	● Connect adaptor to multimeter to confirm current is at least 1A

2.4 Tolerance Analysis

The riskiest block of our design is the keyphrase recognition; the main feature of the system is that this lock will be able to recognize a keyphrase, and then operate a lock based on this recognition. In order to implement this, we plan on implementing a keyphrase recognition algorithm in Python on a Raspberry Pi board; many possible methods for this exist, and our final implementation will determine what method we end up using.

Algorithms that currently exist for keyword recognition include using probabilistic models, such as hidden Markov models, or using some machine learning models. Currently, we plan to implement a few models, including a probabilistic model based on HMMs and a convolutional neural net model. These have been shown to work for the problem of keyword spotting [8], and we intend to implement multiple models and test for which one gives best performance. These methods have been shown to be able to achieve accuracies of around 80% [8], so we believe that we will be able to achieve similar results. Testing models will be done by providing example cases and noting results in a confusion matrix, as described in table 2. If these methods do not work, we could use existing speech recognition packages to implement our design as a last resort.

Another risk with this module is the availability of the Raspberry Pi; they are currently out of stock due to chip shortages. However, Raspberry Pi has said that they plan to make more devices available early this year [9]. If this does not work out, we can use an Arduino UNO Rev3 for this module as a replacement, as this board also allows us to program in Python.

3 Cost and Schedule

3.1 Cost Analysis

Hourly wages will be $40/hr, found using average expected salary for ECE graduates at UIUC [12] and converted to an approximate hourly rate. There are 2 in our group and we expect to work around 12 hours per week for the 10 weeks.

$$Total\ Labor\ Cost\ = 2 \cdot \frac{\$40}{hr} \cdot 2.5 \cdot 12 \frac{hrs}{week} \cdot 10\ weeks\ = \$24,000$$

Part Description	Provider	Part Number	Quantity	Cost
Microphone, condenser, PCB mount	ECE Supply	102-1721-ND	1	$0.95
ATmega328 Microcontroller Bootloader Uno	ECE Supply	X000048	1	$7.20
SERVO MOTOR HS-311	ECE Supply	HS-311	1	$13.54

Single Cylinder Stainless Steel Deadbolt	Home Depot	NA	1	$12.47
Raspberry Pi Zero - Version 1.3	Adafruit	2885	1	$5.00
AC/DC Wall Mount Adapter 5V 10W	Digi-Key (Adafruit)	276-ND	1	$7.95
LED RGB Clear T-1 3/4 T/H	Digi-Key (Kingbright)	754-2153-ND	1	$2.02
Switch Push SPST	Digi-Key (E-Switch)	EG4694-ND	2	$3.88
Total Parts Cost				$53.01
Labor				$24,000
Total Cost				$24,049.13

3.2 Schedule

Week	Class Deadlines	Logan	Aman
1	• Design Review	• Microcontroller Research • Circuit and PCB Design	• Research viable models, identify papers with promising results
2	• 1st Round PCB Orders	• PCB Finalization • Order PCB	• Begin coding first model (HMM)
3	• Spring Break	• Develop code for Analog to Digital Conversion and I/O	
4		• Develop code for Analog to Digital Conversion and I/O • Install board components	• Finalize first model, perform initial testing to ensure proper function • Begin coding second/third models (CNN +other)
5	• 2nd Round PCB Orders • Individual	• Establish Communication between	• Establish Communication between

	Progress Reports	microcontroller and Raspberry PI ● Test Microcontroller functionality	microcontroller and Raspberry PI ● Finalize other models + test functionality
6		● Assembly	● Implement models on the locking system and test which model works better on our system
7	● Debugging and Finalization ● Team Contract Fulfillment	● Debugging and Finalization	● Implement final model, debug as needed
8	● Mock Demo ● Final Testing	● Prepare for Demo	● Gather performance data (accuracy, etc.) ● Prepare for demo
9	● Final Demo	● Work on final presentation/paper	● Work on final presentation/paper
10	● Final Presentation/Paper	● Continue work on presentation/paper	● Continue work on presentation/paper

4 Ethics and Safety

Our project should not have any major ethical or safety concerns. The main ethical concern that could be relevant would be due to recording audio for the lock. If this were being used to record conversations this would be a violation of people's privacy which would break the IEEE Code of Ethics section I part 1 [10] and also the ACM code part 1.6- Respect Privacy [11]. However, our design will only record short inputs which prevents it from recording entire conversations and will contain any audio recordings internally. Audio recordings will not be shared with any other device and will not be stored, so we do not pose the risk of violating user's privacy or sharing sensitive information.

Safety hazards for the project will also be very minimal. The highest voltage used will be due to using the wall adaptor but everything else in the design will be very low voltage. The mechanical portion of the design is the servo which will only be used when the door is actively locking or unlocking, so there will be very few moving parts. Therefore, the design poses no significant safety hazards or ethical concerns.

5 References

[1] Cameron, Dell. "Critical Flaws Leave Some Government Access Cards Vulnerable to Attack." *Gizmodo*, Gizmodo, 15 Jan. 2019, https://gizmodo.com/critical-flaws-leave-some-government-access-cards-vulne-1831785262.

[2] "2019 Crime in the United States." *FBI UCR*, Federal Bureau of Investigation, https://ucr.fbi.gov/crime-in-the-u.s/2019/crime-in-the-u.s.-2019/topic-pages/tables/table-23 . Accessed 9 Feb. 2023.

[3] "Electret Microphone Amplifier - MAX4466 with Adjustable Gain : ID 1063 : $6.95 : Adafruit Industries, Unique & Fun DIY Electronics and Kits." *Adafruit Industries*, Adafruit Industries, https://www.adafruit.com/product/1063#technical-details. Accessed 9 Feb. 2023.

[4] "Diffused RGB (Tri-Color) LED [Common Anode] : ID 159 : $2.00 : Adafruit Industries, Unique & Fun DIY Electronics and Kits." *Adafruit Industries, Unique & Fun DIY Electronics and Kits*, Adafruit Industries, https://www.adafruit.com/product/159?gclid=CjwKCAiA0JKfBhBIEiwAPhZXD3SFC1-uvJZXTvw7qj_EFlj4a_3rZqow17SQUkMDHsOSb_Yb-Od5ZhoC6g0QAvD_BwE. Accessed 9 Feb. 2023.

[5] Rose, R. C., and D. B. Paul. "A Hidden Markov Model Based Keyword Recognition System." *International Conference on Acoustics, Speech, and Signal Processing*, IEEE. *Crossref*, doi:10.1109/icassp.1990.115555. Accessed 9 Feb. 2023.

[6]"Raspberry Pi Documentation - Raspberry Pi Hardware." *Raspberry Pi*, https://www.raspberrypi.com/documentation/computers/raspberry-pi.html. Accessed 9 Feb. 2023.

[7] "Continuous Rotation Servo." *Adafruit Industries*, Adafruit Industries, https://www.adafruit.com/product/154?gclid=CjwKCAiA0JKfBhBIEiwAPhZXDyM_2kg296y0_3YSy4qWPKv9_87anO0Qg7oYSY_LauXVC0iskXSNUhoCwEQQAvD_BwE. Accessed 9 Feb. 2023.

[8] Silaghi, Marius. "Spotting Subsequences Matching a HMM Using the Average Observation Probability Criteria with Application to Keyword Spotting." *American Association for Artificial Intelligence* , American Association for Artificial Intelligence , 2005. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=d94ba6d52cfe095dce25a3563a4dd47f754ee495

[9] Upton, Eben. “Supply Chain Update - It’s Good News! - Raspberry Pi.” *Raspberry Pi*, https://www.raspberrypi.com/news/supply-chain-update-its-good-news/.

[10] “IEEE - IEEE Code of Ethics.” *IEEE*, IEEE, https://www.ieee.org/about/corporate/governance/p7-8.html. Accessed 9 Feb. 2023.

[11] “Code of Ethics.” *Association for Computing Machinery*, Association for Computing Machinery, https://www.acm.org/code-of-ethics. Accessed 9 Feb. 2023.

[12] “Salary Averages.” *The Grainger College of Engineering Electrical and Computer Engineering*, Electrical and Computer Engineering, https://ece.illinois.edu/admissions/why-ece/salary-averages. Accessed 21 Feb. 2023