ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

Knock-Turn Lock

<u>Team 77</u>

ADAM FRERICHS (adamdf2@illinois.edu) JACK KELLY (jacktk2@illinois.edu) VISHAL RAJESH (vrajesh2@illinois.edu)

 \underline{TA} : VISHAL DAYALAN

May 3, 2023

Abstract

A novel approach to domicile security is presented. Current approaches chiefly rely on physical tokens, which can be misplaced or compromised. Information token security is typically centered around an access pad of some sort, the very existence of which relays a vulnerability. We present an information token security method that is entirely undetectable – the Knock Turn Lock.



Figure 1: Conceptual Aid

Contents

| 1 | Intro | oduction | 1 | | |
|---------------|------------------------------------|------------------------------|----|--|--|
| | 1.1 | Problem | 1 | | |
| | 1.2 | Solution | 1 | | |
| | 1.3 | Operation | 1 | | |
| | 1.4 | Power Subsystem | 3 | | |
| | | 1.4.1 Requirements | 3 | | |
| | 1.5 | Sensors Subsystem | 3 | | |
| | | 1.5.1 Requirements | 4 | | |
| | 1.6 | Actuator Subsystem | 4 | | |
| | | 1.6.1 Requirements | 4 | | |
| | 1.7 | Microcontroller Subsystem | 4 | | |
| | | 1.7.1 Requirements | 4 | | |
| | | 1.7.2 Changes | 4 | | |
| | 1.8 | Debug | 4 | | |
| | | 1.8.1 Requirements | 5 | | |
| | | 1.8.2 Changes | 5 | | |
| | | | | | |
| 2 | Desi | gn | 6 | | |
| | 2.1 | Power | 6 | | |
| | 2.2 | Sensors | 6 | | |
| | 2.3 | Actuator | 7 | | |
| | 2.4 | Microcontroller | 8 | | |
| | 2.5 | Debug | 8 | | |
| • | . | <i>re , e</i> | • | | |
| 3 | Veri | fication | 9 | | |
| | 3.1 | Power | 9 | | |
| | 3.2 | Sensors | 11 | | |
| | 3.3 | Actuator | 12 | | |
| | 3.4 | Micro-controller | 12 | | |
| 1 | Cost | | 11 | | |
| 4 | | b Danta | 14 | | |
| | 4.1 | | 14 | | |
| | 4.2 | | 14 | | |
| 5 | Sche | dule | 15 | | |
| 6 Conclusions | | | | | |
| U | 61 | Project Successes | 16 | | |
| | 6.2 | Uncertanties and Future Work | 16 | | |
| | 63 | Ethics and Safety | 16 | | |
| | 0.0 | | 10 | | |
| References 18 | | | | | |
| Aŗ | Appendix A Parts Cost Breakdown 19 | | | | |

1 Introduction

1.1 Problem

Getting locked out of one's apartment or house happens frequently, especially when living alone. Many large apartment buildings have security at the entrance, meaning people often do not feel the need to lock their apartment door, and therefore do not carry a key. As the number of high-rises continues to grow [1], the number of buildings with entrance security will grow, and with it the risk of locking yourself out.

In order to combat this problem, many people hide a spare key or install a keypad lock. Hiding a spare key is a security risk, as it can easily be found by somebody trying to gain entry. Yet in a variety of settings, this unideal solution is used nonetheless, such as for Airbnb rentals. While keypad locks are a somewhat superior alternative, they are often large and unsightly while obviously conveying a secondary path of entry.

1.2 Solution

The Knock-Turn Lock offers a unique solution to the problem of getting locked out of a home or apartment. As previously mentioned, a spare key may be hidden somewhere or a keypad lock may be installed, but both solutions involve several security risks. More generally, the problem of authenticating access without a physical key remains inadequately solved.

Our auxiliary combination lock based on door knocks and door knob turns offers a secure and convenient alternative. The combination input space is simple to use, yet complex enough to avoid unwanted access. Additionally, the lock can be easily reprogrammed in case the combination's secrecy is compromised.

The lock is designed to be hidden from the outside, making it appear as if someone were let in by a roommate. This provides even more security, as it is unlikely that someone would attempt to break in if they believe someone is already inside. Additionally, the discrete design of the lock makes it an ideal solution for those seeking a more aesthetically pleasing option for securing their home or apartment.

In terms of versatility, the lock can be used in a variety of settings. Whether it be for an Airbnb rental, a shared living space, or any other situation where a physical key might not be the best solution, the combination lock provides a secure and convenient alternative. The complete Knock Turn Lock system, if properly installed, satisfies ADA requirements, maximizing the implementation space of the final product. Knocks will be registered as low as 48 inches from the ground, for wheelchair users. [2] In short, The Knock Turn Lock is easy to use, re-programmable, and provides a unique approach to home security.

1.3 Operation

As alluded to above, the lock takes three inputs from the user: door handle turns upwards and downwards, as well as knocks on the door. Once the correct series of inputs



Figure 2: Operation Flow

has been input, the door unlocks. The inputs are stored by the microcontroller, and continually reset after 5 seconds of inactivity. When powered on for the first time, the default combination is 5 knocks in succession.

In order to be accessible to everybody, it is important that re-programming does not require any hardware or software manipulation. Once the user unlocks the lock, they have the opportunity to change the combination. Holding the door handle down for 1 second within 2 seconds of a successful unlock puts the Knock-Turn Lock system into programming mode. Once released, a new series of knob turns and knocks can be input to set the new combination.

In its current prototype form, visual feedback is given by way of LED. A blink indicates a door handle input has been read, a double blink indicates a knock has been read. A triple blink indicates that the inputs have been cleared after 5 seconds of inactivity. A sustained high indicates the lock has been unlocked. After unlocking, once the user holds the door handle down for a second, a blink every second indicates the user has held the door handle down long enough to enter programming mode. Once the user lets go, 5 blinks indicate that programming mode has been entered. Similarly, 5 blinks also indicate that programming mode has been exited after the user has set the new combination.

In production this LED feedback will be eliminated, as it runs contra to our goals of discretion and an unmarred aesthetic. It will be replaced by haptic feedback.



Figure 3: Block Diagram

1.4 Power Subsystem

The power subsection consists of three parts, as can be seen in Figure 3. The first part is AC power coming from the house, as the project is designed to be installed directly to the house power. This gets converted to 9V DC power, which is then routed to the lock via a MOSFET relay. This is the minimum operating voltage for the lock in the design, meaning the microcontroller–with a 5.5V maximum operating voltage–would have to receive a lower voltage. This is done through a voltage regulator, which steps the voltage down to 5V. This connects to both the microcontroller and positive terminal of the door handle contact pads.

1.4.1 Requirements

The only hazards with the power subsystem is failing to meet operating voltage requirements when the electronic lock, which is the largest power draw in the system, switches states. As such, the requirements were to output $9 \pm 0.5V$ with the lock drawing power, and the voltage regulator outputting between 1.8V and 5.5V with the lock drawing power, as this is the operating voltage range for the microcontroller.

1.5 Sensors Subsystem

The sensors used in this lock will be able to detect knocking on the door as well as the turns of the door knob. The knock detection will be done by the vibration sensor, which is a piezo sensor able to detect forces. The turning of the door knob will be detected by buttons within the door knob in production. By placing a button on either side of the turning apparatus, each button will correspond to a separate direction and can be used individually in the combination. In the design prototype we present, this functionality is achieved by contact pads that the door handle makes electrical contact with at each extreme of its range of motion

1.5.1 Requirements

The piezo sensor needs to be able to detect knocks at a standard door knock level. The piezo sensor should also not be able to mistake any other action with a knock. The buttons need to be able to detect the door handle turns once per each turn.

1.6 Actuator Subsystem

The actuator subsystem has the electronic lock and the MOSFET relay. The relay uses the control signal from the microcontroller as a switch. When the MOSFET receives 5V from the microcontroller, it allows current to flow through the relay from the 9V source into the lock, which unlocks it. As mentioned in the power subsystem, this is to allow the microcontroller to control the state of the lock when it is unable to meet the voltage requirement natively.

1.6.1 Requirements

The actuator needs to be able to receive and cease power quickly in order for the entrant to quickly unlock and lock the door. The actuator also should not require more power than we are able to provide.

1.7 Microcontroller Subsystem

The microcontroller will collect data from the sensors and make decisions about whether to unlock the door based on the current combination programmed into the lock, amongst other things. It will handle the programming flow described in Figure 2.

1.7.1 Requirements

The microcontroller needs 4 lines of I/O for communication with the sensors and the actuator. The microcontroller also needs enough storage to store a simple unlocking program as well as enough processing power to interface with a human.

1.7.2 Changes

The microcontroller had plenty of storage and a reprogramming component was added to the software and still easily fit within the storage requirements. The microcontroller also had an extra I/O line so the debug subsystem was added.

1.8 Debug

The debug subsystem consists of a single LED to provide feedback to us and the user about different states of the program. This subsystem is not crucial to the operation of the project. It greatly reduced debugging time for us and was a great addition.

1.8.1 Requirements

The LED needs to be able to be turned on and off and visible to the user.

1.8.2 Changes

This subsystem was not in the original design, but was included when we realized there was an extra I/O line on the microcontroller.

2 Design

2.1 Power

The first choice for powering the system was where to get power from. Having the lock wired directly to AC power was chosen because a battery would have to be replaced, and renewable sources like solar power would have to be visible outside of the door which would defeat the purpose of hiding the design within the door.

Next was the power for the lock. The solenoid lock was designed to work between 9 and 12V, with higher voltages consuming more power but having quicker response times [3]. The 9V AC/DC converter was used because it means the lock would have a lower power use, and the response time of the lock was negligible for our use case.

Finally, a voltage regulator had to be chosen to get the voltage down to the level of the microcontroller. This could have been anywhere from 2.7-5.5V given the microcontroller datasheet [4]. The voltage regulator chosen was 5V for two reasons. First, it was 0.5V lower than the maximum operating voltage, which meant the microcontroller could tolerate a fair bit of noise. More importantly, this allowed the largest range of values when reading the piezosensor through an analog pin. The microcontroller can use three reference values when doing an ADC conversion: 1.1V, 2.7V, and VCC, with the reference voltage corresponding to the top possible value 1023 [4]. The voltage regulator was ordered at the same time as the piezosensors, so a high VCC value gave the largest range of reference voltages to use in case the piezosensor did not work. The threshold value used to trigger a knock input we used ended up being 150, which corresponds to 5V * 150/1024 = 0.732V. This means future designs could optimize this regulator for cost efficiency for any value between 2.7 and 5.5V, as a high reference voltage is not needed.

The vibration sensor behaves as a capacitor when it senses a force, so this will not need to be connected to power to generate a signal [5].

2.2 Sensors

There are two types of sensors needed in the sensors subsystem: vibration detection and contact detection. For the vibration detection, we decided to use a piezo sensor. Other types of sensors we looked into were flex sensors and pressure sensors. The flex sensors needed to be manipulated too much to get a reading from a simple knock. The pressure sensors would have worked, but was not chosen due to being more expensive and complicated than a piezo sensor [5].

For the contact detection, we decided to simply use wires. We had originally planned on using buttons in the design, but changed to using wires to better fit our physical product. If we were able to have a custom door knob designed, this would have been integrated contact pads within the door handle, but this was outside of our scope

2.3 Actuator

The lock used in our design is a solenoid lock, which uses an inductor to pull the lock in. This means that, on transisition from high to low, the current flows backwards through the lock which can potentially damage a MOSFET. This is why it is recommended to use a flyback diode that prevents the current from flowing back into the relay, which should be a Schottky type diode for a quick response time [3].

The relay circuit requires a transistor to act as a relay, so there was a choice between a BJT and a MOSFET. We chose a MOSFET due to the faster switch times and low gate current so as to not damage our microcontroller [6].

Finally, the circuit design was done in simulation in order to ensure correct response times. As seen in Figure 4, the response time in simulation was $32\mu s$, which is a practically inconceivable delay to the user.



Figure 4: The simulated circuit (top) and current response time (bottom)

2.4 Microcontroller

The design phase for the microcontroller consisted of searching for a microcontroller that fit all the requirements. The first microcontroller we researched was the ATtiny85. The reasoning for researching this one first is because we were already familiar with it as it was used in the CAD and soldering assignment. The ATtiny85 had enough I/O pins and also an analog to digital converter for reading the piezo sensor. The only other requirements we needed were a high enough clock speed to read the piezosensor and contact pads quickly, and enough memory to fit the entire program. In both of these cases, we decided to give large amounts of leeway to avoid problems while designing. The debounce circuits for the contact pads have a time constant of $\tau = R * C = 10000 * 1 * 10^{-6} = 10ms$, and the microcontroller reads high for voltages higher than 2.7V [4]. Using the formula $V(t) = V_0 * e^{\frac{-t}{RC}}$ and solving for time, we get $t = -RC * ln(\frac{V(t)}{V_0})$, which yields a time of 6 milliseconds to read a high signal. Therefore a clock signal of 164 Hz would be acceptable to read these inputs, and the ATTiny85 clock speed of 8MHz would be more than enough [4]. It also has 8kB of memory, which fit the preliminary program size of 1.5kB with lots of room for expansion. Therefore, the ATTiny85 was chosen as it exceeded our design requirements and we were familiar with its functionality.

2.5 Debug

The design of the debug subsystem was simple. All the debug subsystem needed to do was be able to give feedback to us and the user. This could have come in the form of audio or visual and we decided just an LED would be sufficient feedback.

3 Verification

3.1 Power

The requirements for power were simply not failing under the stress of the lock, as this required the most power. When running the lock at 9V, it draws 500mA for a power consumption of 4.5W [3]. In contrast, the button and piezosensor circuits only power consumption is due to the pull-up resistors which are 10k each. This leads to a power consumption of $P = \frac{V_{cc}^2}{R} = \frac{25}{10000}$, or 2.5mW. Finally, at our clock speed of 8mHz and Vcc of 5V, the active supply current of the microcontroller is 5mA, which is a power draw of 25mW. This means the only non-negligible power draw was the lock, and in order to maintain functionality we needed the power supply to remain within $9 \pm 0.5V$ and the voltage regulator to remain within the microcontroller rated voltages of 5.5-2.7V. As can be seen in Figure 5, this was achieved.

| Requirements | Verification | |
|---|---|--|
| The AC/DC converter should be able to output enough voltage to run the highest voltage requirement and have enough current to ensure there will not be a failure due to lack of power. The AC/DC converter should output 9 ± 0.5 V even with varying loads, no matter the state of the actuators. | This can be tested by cycling the lock and ensuring there is no failure due to lack of power nor voltage drops outside of acceptable range. Using an oscillo- scope, we read the voltage output of the AC/DC converter during a lock cycle. [3]. | |
| The voltage regulator should be able to output a voltage acceptable for the micro-controller specifications, in order to not damage the part. The micro- controller VCC and I/O pins are rated for 2.7-5.5V, so the regulator will be rated for 5V and should always remain in the range of 2.7-5.5V even while the lock is cycling twice a second [4]. As such, the voltage regulator should re- main within the 2-5.5V even when cy- cling the lock. | This can be tested by cycling the lock, which consumes the most power, and ensuring there is no failure due to lack of power nor voltage drops outside of acceptable range. Using an oscilloscope, we read the voltage output of the volt- age regulator during a lock cycle. | |

Table 1: Power Requirements and Verification



Figure 5: 9V power source (top) and voltage regulator (bottom) responses to lock being driven

3.2 Sensors

The requirements for the sensors used in this lock were for them to be able to detect knocking on the door as well as the turns of the door knob. The knock detection done by the piezo sensor should be robust to noise. The turn detection done by the contact pads must make proper electrical contact at each extreme of its range of motion. The time frames listed below are mostly a function of the micro-controller program, but also indicate sensor functionality.

The verification for sensor response times were done by monitoring the LED on different sensor inputs, and ensuring it changes state within the designated time frames below.



Figure 6 below shows voltage spike when a knock occurs.

Figure 6: Piezo Response

| Requirements | Verification |
|--|--|
| Upon a turn to the left or the right, the input should be read within 0.2 seconds. | The LED will flash within 0.2 seconds of a door turn being input. |
| The piezo sensor input will be read within 0.2 seconds. | The LED will double flash within 0.2 seconds of a door turn being input. |
| The piezo sensor should not detect han- dle turns as knocks. | Handle turns should not trigger the LED to double flash, which indicates a knock input being read. |
| The door handle turns are properly de- bounced so as to only trigger once. | The LED only flashes once when the door handle is turned. |

Table 2: Sensors Requirements and Verification

3.3 Actuator

The only constraints with the actuator subsystem were the lock having a fast response time given a control signal from the microcontroller, and the relay circuit being properly designed so as to not cause the microcontroller to fail when driving the lock. The lock is quick enough so that any delay is inconceivable, which mimics the simulation results in Figure 4. Furthermore, upon many successful lock and unlock cycles, the circuit has yet to fail.

Table 3: Actuators Requirements and Verification

| Requirements | Verification |
|---|--|
| The relay circuit has a small enough power draw so as to not turn the micro-controller off. | Activating the lock through the relay, due to correct combination input, does not restart the microcontroller. |
| The lock opens right after a correct com- bination input. | The lock unlocks within one second of the combination being input. |

3.4 Micro-controller

The requirements for the micro-controller are mainly centered around the requisite program to be loaded. The physical specifications required to fit the whole program and route in all three inputs (piezo sensor, door knob down, door knob up) and route out the decision output.

Testing combinations shows that the inputs are read, due to the changing of LED states, and correct combination inputs unlock the lock for two seconds. Other actions, such

as combination reset on incorrect combination entry are also visible from the LED response.

The program compiled to a size of around 2kB which is well under the 8kB requirement.



| TI - | D | • 1 | 1 • 1 | •1• |
|-------------|--------------|----------|--------------|-----------|
| H1011100 // | Program | CIZO CHO | 117n 117h110 | compiling |
| | THOULAND | SIZE SHO | | |
| I IGUIC / · | I I USI MIII | | | compring |
| 0 | 0 | | | 1 0 |

| Requirements | Verification | |
|---|---|--|
| Program reads all three inputs and is able to control the output. | Single LED flash for both door handle turns, double LED flash for knock inputs, and the lock is able to unlock. | |
| Program is efficiently written to fit within board memory requirements. | Program is less than 8kB (ATTiny85 max memory sensor). [4] | |
| Program allows for new lock combina- tions to be input without reprogram- ming the microcontroller. | Perform reprogramming inputs to en- ter programming mode, which allows a new combination to be entered. This new combination now unlocks the lock. | |
| Solenoid lock stays open for more than 2 seconds to allow entry. | Perform correct combination and view the lock staying open. | |
| Combination resets after incorrect at- tempt. | Perform incorrect combination and wait for LED flashes to indicate reset. | |
| Combination is not easily able to be brute forced. | Combination is at least 8 entries long and timeout is at least 5 seconds requir- ing greater than 9 hours to guess every combination. | |

Table 4: Microcontroller Requirements and Verification

4 Costs

4.1 Parts

The parts cost breakdown is in Appendix A. The breakdown includes extra parts ordered in case of failure.

The raw total cost of all of the parts is \$92.78. Shipping costs are around 5% and sales tax is around 10%. This brings the total cost of parts delivered to be \$106.70.

4.2 Labor

For labor costs of the 3 team members, we will use the rate of \$50 an hour. We estimate we have spent around 64 hours each on the project. We did not make use of the machine shop, so we do not have any costs related to them. This brings the total cost of labor to be $50 \times 64 \times 3 \times 2.5 = $24,000.00$.

Adding together both the parts and labor costs, the total cost for the project is \$24,106.70.

5 Schedule

| Week | Task | Involved |
|-----------------|------------------------------|---------------|
| Feb 20 - Feb 26 | Order first round parts | Adam |
| | Start PCB design | Jack |
| Feb 27 - Mar 5 | Design review | Everyone |
| | Finalize PCB design | Vishal |
| Mar 6 - Mar 12 | Order second round parts | Jack |
| | Begin software program | Adam & Vishal |
| | Order PCB | Adam |
| Mar 13 - Mar 26 | Assemble & test PCB | Adam & Vishal |
| | Revise PCB | Jack |
| Mar 27 - Apr 2 | Order second PCB | Adam |
| Apr 3 - Apr 9 | Assemble second PCB | Jack |
| | Revise software program | Adam & Vishal |
| Apr 10 - Apr 16 | Test PCB | Jack |
| | Team Contract Fulfillment | Everyone |
| | Full test of Knock Turn Lock | Everyone |
| Apr 17 - Apr 23 | Mock demo | Everyone |
| Apr 24 - Apr 30 | Final demo | Everyone |
| | Mock presentation | Everyone |
| May 1 - May 4 | Final Presentation | Everyone |
| | Final paper | Everyone |

6 Conclusions

6.1 **Project Successes**

Overall, the design functioned very well. Every function we set out to implement worked, and the design seemed to have reasonable use cases. As we got more accustomed to using the device, it became trivial to unlock the door and even program a new code.

The device is able to take any five inputs consisting of knocking, turning the door handle up, and turning the door handle down. Upon successful input of the code a different combination can be set by holding the door handle down for one second, and any different five inputs can be set as the new combination.

6.2 Uncertanties and Future Work

The only unsatisfactory result from our project was the knock threshold. While it was not difficult to get a knock input to read, it required you to knock harder than we would have liked.

The main thing we want to implement in the future is complete integration into the door. This means having a door handle with integrated contact pads, as well as having a separate entrance system using a key.

We also want to allow the combination to be more complex. This could entail adding more knock sensors to different parts of the door, so knocking in different places are different inputs. It could also be adding a timing component to the combination, so the lock remembers the rhythm and not just the pattern.

6.3 Ethics and Safety

Given our position as computing professionals, our actions and the resultant products of our intellectual property have the potential to change the world, in both positive and negative ways. As such, a careful analysis of all the ways in which our work could be used is warranted. What follows is a thorough overview of the Knock Turn Lock in its relation to the ethical standards proposed by the IEEE and ACM. While both codes differ in their minutiae, the overarching themes are largely congruent, and all the main themes relevant to the Knock Turn Lock will be covered.

With regards to IEEE, the main relevant code is "to uphold the highest standards of integrity, responsible behavior, and ethical conduct in professional activities," specifically "to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, to protect the privacy of others." ACM emphasizes that we must " design and implement systems that are robustly and usably secure" and "use care when modifying or retiring systems."

Our product being a novel lock that aims to replace existing tried and tested systems, we must abide by this ethos in a very literal sense. Through rigorous testing, we have

ensured that our lock is robustly prepared to handle the actions of bad actors whilst also allowing easy access to authorized users. Beyond this, before our product should come to market we would apply for ANSI/BHMA commercial hardware certification, ensuring without a doubt that any consumer of our product is properly informed of the security grade of the Knock Turn Lock.

In addition to the Knock Turn Lock abiding by the relevant ethical standards as proposed by IEEE / ACM, it is equally important for the product to abide by the safety standards put into place by the relevant regulatory authorities [2]. Given that the Knock Turn Lock is designed for residential, not commercial use, we could sidestep many regulations aimed exclusively at the latter, especially given that residential door lock regulations are largely non-existent. However, since we aim to allow the Knock Turn Lock to be used in as many places as possible, its design attempts to follow commercial regulations as well, whether or not we officially advertise this compliance.

What follows is a thorough overview of the Knock Turn Lock in its relation to the safety standards outlined by the ADA and HIPAA. While all commercial locks must abide by the ADA, HIPAA applies to specific industries where patient privacy is in question. Again, since the Knock Turn Lock aims to be used in as many places as possible, all efforts were made in this pilot product to satisfy as many regulations as possible, and official compliance can be established by a later version. The additional stipulation placed by HIPAA is an access control system. Although separate combinations for separate users so as to identify them by their combination of entry are not available on our prototype, such functionality is trivially extendable for later versions.

Regulations exist for both doors and locks. Our product will be assumed to be installed on a compliant door. Thus this discussion will be limited to the scope of lock regulations. The ADA requires:

- 1. All locks and handles attached to your doors must be operable with one hand and not require any extreme grasping or twisting.
- 2. Any lock or handle hardware must be mounted no more than 48 inches above a finished floor; this ensures that those in wheelchairs are able to reach door locks as needed.

The Knock Turn Lock will use a lever handle, and allow knocks from below 48in of height. Easy egress will also be implemented automatically by the lever handle.

References

- [1] N. Routley. "There is a global race to build even taller skyscrapers." Accessed: 2023-02-09. (Jan. 2018), [Online]. Available: https://www.weforum.org/agenda/2018/01/theres-a-global-rush-to-build-ever-higher-skyscrapers.
- [2] S. Chris. "Door lock compliance: What business owners need to know." Accessed: 2023-02-09. (Jan. 2019), [Online]. Available: https://www.gvlock.com/blog/doorlock-compliance/#Specific_ADA_Requirements_for_Commercial_Door_Locks.
- [3] A. Industries. "Lock-style solenoid 12vdc." (), [Online]. Available: https://www.adafruit.com/product/1512.
- [4] Atmel. "Atmel 8-bit avr microcontroller with 2/4/8k bytes in-system programmable flash." (Aug. 2013), [Online]. Available: https://ww1.microchip.com/downloads/ en / DeviceDoc / Atmel - 2586 - AVR - 8 - bit - Microcontroller - ATtiny25 - ATtiny45 -ATtiny85_Datasheet.pdf.
- [5] A. Industries. "Small enclosed piezo w/wires." (), [Online]. Available: https://www. adafruit.com/product/1740.
- [6] E. Tutorials. "Relay switch circuit." (), [Online]. Available: https://www.electronics-tutorials.ws/blog/relay-switch-circuit.html.

Appendix A Parts Cost Breakdown

| Description | Manufacturer | Quantity | Total Price (\$) |
|-------------------|--------------|-----------|------------------|
| Solenoid Lock | Adafruit | 1 | 14.95 |
| Piezo Element | Adafruit | 2 | 1.90 |
| Piezo Element | Adafruit | 2 | 1.90 |
| Piezo Element | CUI Devices | 2 | 1.96 |
| Piezo Element | CUI Devices | 2 | 4.16 |
| Power Supply | CUI Inc. | 1 | 15.79 |
| ATtiny85 | Microchip | 8 | 12.48 |
| Button | SparkFun | 4 | 2.20 |
| Wire | general | 10 ft | 0.10 |
| N-Channel MOSFET | Fairchild | 2 | 3.94 |
| Resistor | general | 20 | 2.00 |
| Capacitor | general | 20 | 2.00 |
| Diode | general | 8 | 2.00 |
| LED | general | 2 | 0.10 |
| IC socket | general | 2 | 0.10 |
| Male connectors | general | 20 | 1.00 |
| Female connectors | general | 20 | 1.00 |
| Voltage Regulator | onsemi | 2 | 3.20 |
| РСВ | PCBWay | 3 | 15.00 |
| Wood | general | 2x6x18 in | 5.00 |
| Wire mounts | general | 2 | 2.00 |

Table 6: Parts Cost Breakdown