Smart Door Lock

(Honors Lab Final Report)

Peter Chiu - jcchiu2 / Johnny Chang - yuchenc2

1) Introduction

a) Statement of Purpose

Moving into college, one of the new experiences is living in a school dorm and sharing a room with your roommate. A very important part of your dorm is security. After accidentally locking ourselves from our dorm, and getting a housing reprimand for borrowing the backup key too many times, we felt the heavy burden of needing to carry a key around. To address this issue, we decided to develop a Smart Door Lock that can be unlocked in multiple ways - without the key and the hassle.

b) Features and Benefit

- **i)** Set personal door knocking pattern as the passcode to unlocking the door.
- **ii)** Integrates a mini button to press in the pattern instead of knocking in case of a need for low noise level or more security reasons.
- **iii)** Allows you to use your phone to unlock the door wireless via Wi-Fi.
- **iv)** Removes the need to have to use a key to unlock the door
- **v)** Reduces the possibility of locking yourself outside the dorm with multiple access.

2) Design:

a) System Overview

Our design uses 4 main components - an Arduino Uno microcontroller, a Piezo sensor, a servo motor, and a Wi-Fi module. The machine is designed with three ways of unlocking the door - the door knocking sequence, button pressing sequence, and Wi-Fi signal.

The door knocking sequence detection uses a piezo sensor attached to the door and a button that works in case of piezo sensor failure. First, there’s a small button on the Arduino that when pressed, allows us to set the knocking pattern and store it. When in use, the sensor detects the knocking pattern and loudness of the knock and outputs the data to the Arduino. The Arduino determines whether the door knocking sequence and timing is correct or not. If the knocking pattern matches the correct pattern, the servo motor unlocks the door.

This unlocking method is supported by a small backup button. The button serves as a way of unlocking the door in case the knocking sensor isn’t working, or you don’t want people to overhear your knocking pattern. By pressing this button in the same sequence as the correct knocking pattern, you can unlock the door.

We also designed our project to allow phones to unlock the door wirelessly. We implemented a Wi-Fi module, which is connected to the Arduino. Using a common router, in this case, our laptop, the module can receive and pass on data from the phone to the Arduino. Looking at the IP address from the module and using it on the phone, we can unlock the door with the phone.

b) **Design Details**

i) **Arduino Uno:**

We use an Arduino Uno microcontroller for our project. It accepts all the inputs and directs outputs, as well as LED indicators. It stores the code needed to store and process door lock patterns and accepts inputs from the piezo sensor, button, and Wi-Fi module. With the code (credit to reference 2), we are able to tweak the sensitivity of the sensor, create a threshold for the minimum knock decibel, record and determine knocking patterns, process Wi-Fi module signals, and provides output for the servo motor.

The Arduino provides the output for the servo motor and determines how far the motor has to turn to lock or unlock the door.

ii) **Piezo Sensor:**

The piezo Sensor is a speaker sensor that we use to detect the knocks. It records the audio of the knock and sends the analogous signal into the Arduino. It is used to both set and detects the knocking pattern. When the knocking pattern is correct, a LED light flashes It is connected to a 2 input OR gate, the other input being a button, and the output goes to the Arduino. We did not use speakers as speakers do not output a range of voltage as piezo sensors do, thus are not applicable to determine the loudness of the knock.

iii) **Wi-Fi Module (ESP8266):**

The Wi-Fi module serves as a server that is connected to our router (a laptop). We connect our phone to this router. By entering the IP address of the Wi-Fi module on our phone's browser, we can connect the phone to the module, which allows wireless control of the Arduino. (Credit for the code controlling the Wi-Fi module goes to reference 1)

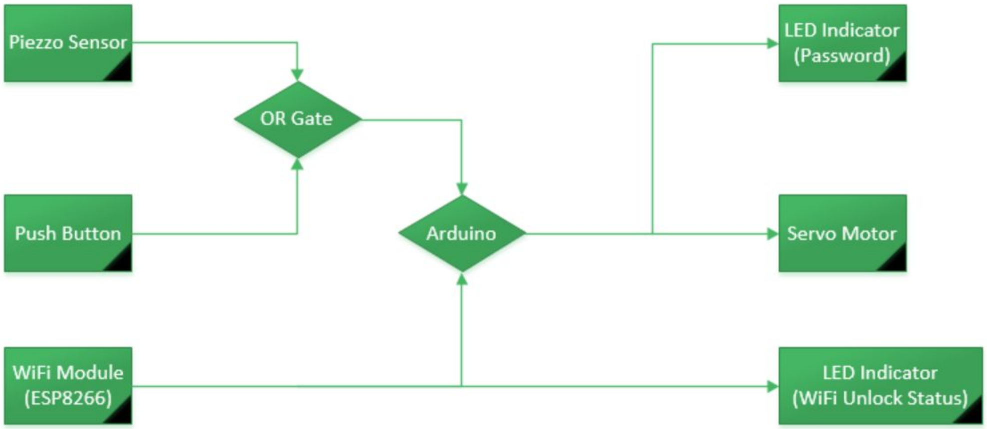
iv) **Servo Motor (MG928):**

We used a high torque gear servo motor to turn the lock. We used Solidworks to design and 3D printed our mount. The servo motor is mounted onto the lock to set it in motion. The servo motor is directly controlled to the Arduino, which has code inside of it to determine how the motor should respond to the signals.

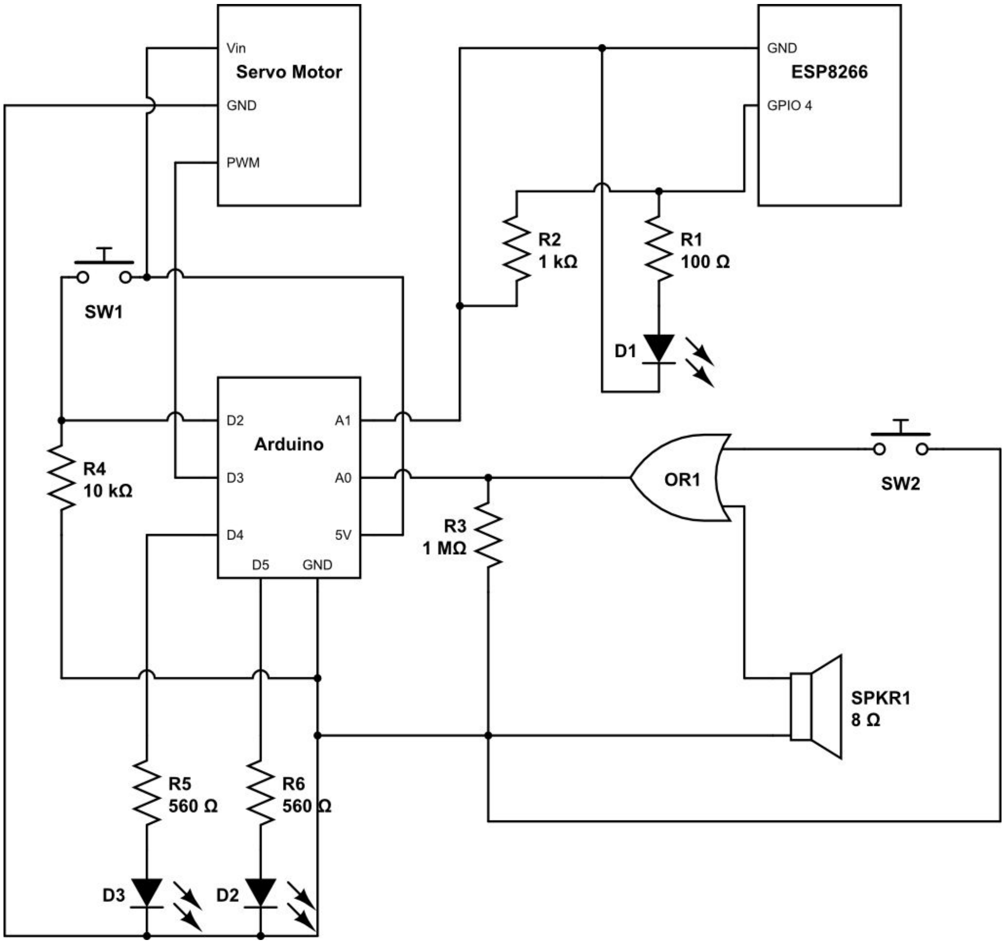
v) **Buttons:**

There are two buttons in our project, one serves as the programming button for inputting new knocking password and the other one serves to be a more robust way of unlocking the door via pressing pattern should the piezo sensor fail to operate correctly.

c) Block Diagram



d) Circuit Schematic



3) Results

a) Characterization of Piezo Sensor

Experimental setup:

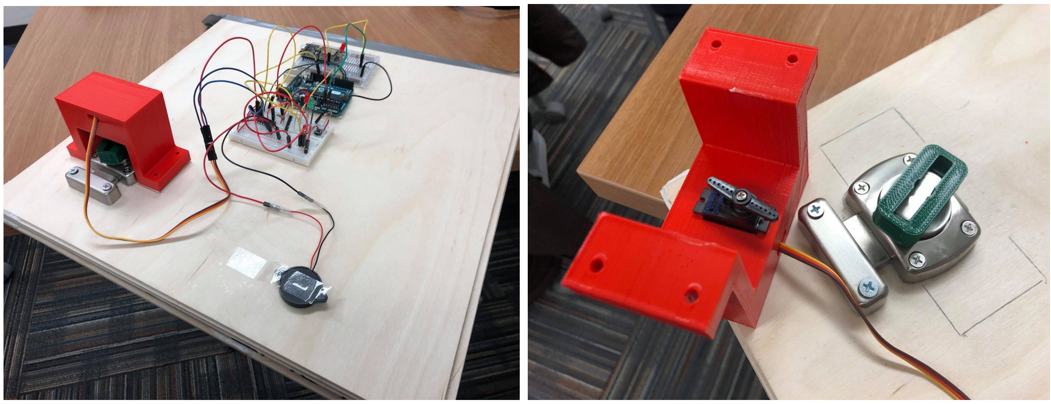
We connect the probes of the oscilloscope parallel to the piezo sensor's positive and negative pins and read the voltage value from the display screen.

Tables of measurements:

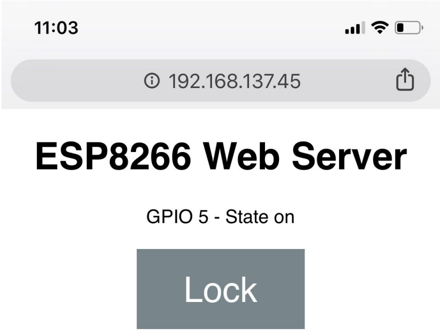
Knock/No Knock	***Voltage (mV)***
No Knock	0
Knock	400 (avg)

b) Verification of Functionality

Our project is able to listen to knocking patterns very precisely and unlocks the door lock only when a correct pattern is inputted. We can also easily enter password changing mode by pressing the programming button in our circuit. The 3D printed mount fits perfectly with the servo motor and the door lock, so we did not have to sand any part of the mount. The wifi module successfully connects to the WiFi router and acts as a web server, and we are able to unlock the door lock through our phones' browsers.



(Left) Picture of the entire design. (Right) Picture of 3D Printed Parts with a servo motor and door lock.



11:03

192.168.137.45

ESP8266 Web Server

GPIO 5 - State on

(Left) Picture of WiFi unlocking method on the phone browser.

4) Problems and Challenges

a) Servo Motor

When we connected the servo motor to the Arduino, we used codes and commands to give the servo motor its next position, which in turn led to it turning into the next state. However, we couldn't get the motor to rotate more than 15 degrees. When the next position more than 15 degrees off from the original angle, the servo motor doesn't respond. This is possibly caused by the limitation of the servo motor since we have tried providing voltage through an external power supply to eliminate the possibility that the Arduino cannot provide enough power to turn the motor.

This problem has been solved with modifying the code. Instead of having the motor turn 180 degrees all the way, we had the motor turn 15 degrees at a time and ran it in a loop with the correct delay. With this method, we were able to get the motor to rotate to the correct position as well as acquiring more accuracy to the motor's position.

b) OR Gate

The smart door lock has multiple advanced ways of unlocking a door without a key, but we also implement a more fail-proof way of unlocking the door should the more complex ways fail to work. We used a small button that functions as another input for the knocking sequence, with each press of a button as a definitive knock. We linked both the button and Piezo sensor to an OR gate, expecting it to work if either knocking or button pressing is ongoing. However, since connecting the piezo sensor to the OR gate meant the input from the OR gate to the Arduino is no longer analogous, every voltage from the piezo that exceeds the minimum input voltage level is detected as a 1, which made our sensor detecting a nonstop knock(too high sensitivity).

We have tried to ground and add resistors to our piezo sensor output in hopes of lowering its output voltage (decreased sensitivity), however, we have not yet figured out a way to balance it so it can reach optimum sensitivity.

c) WiFi Module

When we bought our first ESP8266 Wifi module from Amazon, we did not care about the quality of the module and got the cheapest option. However, after we received the product, we were stuck with connectivity problems and were not able to receive information from the module. We gave up on the module after a few hours of debugging and searching online and bought an ESP8266 (ESP-12E) Wifi module from HiLetgo. The new module immediately established a stable connection, and we were able to run our code for it perfectly.

d) Door lock mount

From multiple sources, we attempted to try out different ways of mounting the device onto the door lock. We tried direct gluing, mounting with wood planks and screws, but none of them were precise or robust enough to handle the weight and torque from turning the lock. Eventually, we used SolidWorks to design our mount and parts, and 3D printed it.

We replicated the door lock and our design onto a wooden board. However, due to the lack of equipment (drill bits, power drill, screws), we couldn't drill holes small enough to mount our lock. Thus, we found screws that were slightly too small and glued them into the holes made with a hand drill to secure the lock and design.

5) Future Plans

Our main future plans for this door lock design is to make the OR gate design work. Currently, we have not fixed the sensitivity of the piezo sensor via circuitry, so we have not been able to implement it with the OR gate to allow both button and piezo sensor to detect door knocking sequence. It would be extremely helpful if the door knocking way of unlocking had another method, as it could lower the chance of the design malfunctioning significantly.

Other improvements would be to find a way to increase the speed of the motor unlocking the door to save precious time. Currently, the delay in turning the motor 15 degrees at a time is noticeable, thus it would make the design much smoother if we can find another way to have the motor move into position in one motion.

Overall, the design is very customizable and effective, and with further improvements can really be a good replacement for standard keys.

6) References

1. **Build an ESP8266 Web Server - Code and Schematics. (n.d.). Retrieved December 9, 2018, from https://randomnerdtutorials.com/esp8266-web-server/**
2. **Grathio, & Instructables. (2017, November 06). Secret Knock Detecting Door Lock. Retrieved December 9, 2018, from https://www.instructables.com/id/Secret-Knock-Detecting-Door-Lock/**