The Cooper Union for the Advancement of Science and Art

End-of-Semester Report

ItsMe: Portable Smart Lock

Advisor: Sam Keene

Authors:

Ivan Chowdhury

Hyungseok Yoon

Min JO

Joshua Go

April 10, 2020

Table of Contents

Table of Contents **1**

1. Abstract **2**

2. Introduction **3**

3. Background **4**

4. Related Work **9**

5. Project Description **14**

5.1. HUB 15
5.2. Sensor 16
5.3. Actuator 16
5.4. Mobile Application 17
5.5. Physical Device 18

6. Status **20**

7. Conclusion **22**

Works Cited **24**

1. Abstract

As more people integrate smart devices into every facet of daily life, they grow to rely on the convenience that they offer. Through the internet of things (IoT), devices can be controlled and information can be collected and shared remotely across a network. One application of such a system is a door lock, which is used extremely frequently in securing homes, offices, and vehicles. However, current solutions available to the average person are difficult and costly to install. The goal of this project is to create a more versatile IoT smart lock that can attach and detach itself from a wide variety of pre-existing door lock knobs to allow for easier installation. Along with this comes control via an android application on a personal smartphone device, while maintaining the existing door lock and key capabilities.

2. Introduction

The internet of things is changing the world we live in and is influencing lifestyles in a plethora of ways. Sophisticated sensors are embedded in physical devices that surround us which transmit data. IoT is the common language that ties all these devices together into a network of interconnected devices, where each can gather transmit data to be integrated and analyzed, then share data to applications to address industry specific needs. This data can ultimately be used to people's benefit by improving efficiency and the user experience [1].

One possible application of such a device is the door lock. With a conventional door lock, keys can be easily misplaced and lost, compromising the important aspect of safety for the user's daily life. In addition, access to guests, family, friends and trusted individuals is cumbersome and requires creating copies of the keys, which too can get misplaced. The convenience that the internet of things brings to daily life is immeasurable and is a foundation of the devices all around. Integrating that into the everyday door lock will allow users to control a lock with a touch of a button through their smartphones, and easily provide their friends and family access to their homes through their own homes. A device such as this saves time and effort, which are people's greatest commodity. Currently implemented solutions for the IoT-controlled lock, or smart lock, are inherently limited, due to the difficult and costly installation process that requires home renovation, and the lack of an easy way to uninstall the device and reinstall it to doors in other homes [2]. The design of our device - the attachable smart lock - circumvents these limitations and adds versatility and convenience by allowing the lock to be freely attached to and detached from standard deadbolt locks.

3. Background

This section of the report will cover the technical background of our project. We will introduce the technologies we plan on using, and explain the conceptual details behind the operation of our project components. The Internet of Things is the notion of creating a "smart" network of devices that operate in conjunction with one another in real-time. This is accomplished by placing sensors and actuators on the devices to send signals between them and a microprocessor, such that the devices can communicate with each other on the same wireless network [1]. Many IoT projects also involve the creation of an Android application that communicates with the network, and allows the user an easy way to control the operation of the network.

An IoT system is generally composed of the following components: a hub, sensors, actuators, and communication module. The hub serves as the focal point of the smart network, working as a controller that interacts with all devices on the network, and allows the user to interact with those devices. Our IoT system will be composed of two main parts: a Raspberry Pi 3B and an Android application. The Raspberry Pi will serve as our backend server, and will be responsible for processing and sending signals between devices, in particular for sending parametrized signals to our rotary actuator in order to rotate the attached lock between locked and unlocked states. The Android Application will serve as a UI for users, allowing them to easily monitor the current state of the system, as well as control the operation of the smart lock remotely [3].

Sensors are physical devices used to monitor the conditions of the environment, to detect and record information in real time, and to send signals to the hub under certain circumstances. Our system does not incorporate any of these sensors, but improvements can be made in the future to expand the scope of the system and utilize this extra functionality to add new features to our smartphone application. Actuators are physical devices that receive signals from the hub and perform designated physical operations, such as rotation or switching. For our project, we will utilize Adafruit's Feedback 360 degree rotation servo as our actuator. The device is capable of rotating to precise degrees according to the signals it receives from the hub, is lightweight and provides adequate torque for turning a deadbolt lock. Lastly, in order to receive the signals being emitted from each device, and provide the appropriate response within other devices on the network in real-time, a hub is needed. The hub is essentially a server hosted on a microprocessor, whose role is to mediate communication between all of the devices on the network; in our case, this includes the Servo motor, lock device, and Android application [3].

The Raspberry Pi is a single board computer that can be connected to typical peripherals like, a mouse, keyboard, monitor, and external circuit board. For our IoT system, we chose to work with the Raspberry Pi 3B+ as our hub. The Raspberry Pi model 3B+ has a 1.4 GHz 4-bit quad-core CPU, 1 GB RAM, Dual-band 802.11ac wireless LAN and Bluetooth 4.2 support, 4 USB ports, Gigabit Ethernet, and support for Linux and Unix operating systems. Compared to previous versions, it offers support for Bluetooth 4.2 Low Energy, improved processor speed, and improved wireless and Ethernet connection speed, among other improvements. The reason this model was chosen over the recently released model 4B is due to the difference in power usage. The Raspberry Pi 4 has been reported to have higher power usage, and thus is more susceptible to overheating during usage. We decided to choose the extra reliability of the model 3B over the improved specs of the model 4B. The Raspberry Pi 3B is set up through installing a customized copy of Debian (Linux) named Raspbian OS onto the computer through a Micro SD card. From there, the circuit board can be connected to the required sensors and actuators through jumper cables or wirelessly [5].

An alternative to using the Raspberry Pi as our microprocessor was the Arduino microcontroller. The Arduino is also commonly used for IoT projects, but has different specializations and use cases. The Arduino microcontroller is best for use with simple automated tasks, such as opening blinds, turning on devices, recording and displaying temperature in real time, and other similar activities. For coordinating a network of devices, communicating between multiple devices simultaneously, and doing complex time-based tasks, the Raspberry Pi is more suitable. The Raspberry Pi is also capable of performing multiple tasks simultaneously, as any computer can do, whereas the Arduino can not. For this reason, we decided on using the Raspberry Pi as our hub [6].

The physical operation of rotating a deadbolt lock between locked and unlocked states will be handled by the rotational actuator, or Servo motor, that will be attached to the lock. The 360 degree feedback Servo used for this is capable of rotating a full 360 degrees, as well as rotating to specific angular positions, according to the parametrized signals sent to it from the Raspberry Pi microprocessor. This allows for the rotation of the deadbolt lock to be remotely controlled from the Raspberry Pi, through code written in Python. The Android application provides the user with a simple and straightforward way of remotely controlling the Servo motor's rotation, through sending signals to it via the Raspberry Pi [7].

It is also essential to choose what type of wireless networking protocol and wireless signals will be used for inter-device communication within the IoT network. There are a number of options available for this, some specialized for usage in similar systems. Among our primary considerations for this were Wi-Fi, Bluetooth, Zigbee, and Z-waves. Wi-Fi is the most ubiquitously used wireless networking protocol. Its advantages are that it is compatible with almost all devices, including the vast majority of mobile phones and computers, it supports long-range communication, and provides a strong connection between communicating devices. It is also easy to provide security to a Wi-Fi network through WEP, WPA, WPA2, or WPS encryption. Its disadvantages, however, are that it drains battery life at a faster rate than most other options, and requires a Wi-Fi access point to be in the vicinity. The next option considered was Bluetooth. Bluetooth's main advantage is where Wi-Fi lacks in: battery drain. Bluetooth provides easy-to-setup connections between Bluetooth-compatible devices, with very low battery drain relative to Wi-Fi connections. However, it mainly only works for phones, tablets, a small set of other Bluetooth-compatible devices, and has a weaker range and connection strength compared to Wi-Fi [3].

The last options considered were Z-Waves and Zigbee radio waves. Both are wireless networking protocols designed specifically for smart home networks. They offer strong connections between devices with very low battery drain. Z-waves transmit at 908 MHz, while Zigbee waves transmit at 2.4 GHz. This means that Zigbee is capable of transmitting more data, but over a shorter range than Z-Waves. The disadvantages are that there is a high degree of interference if there are many devices on the same channel. They are also more complicated to set up, and do not offer the same degree of compatibility across devices as more general wireless networking protocols do [3].

For our implementation, we will use the IP address assigned by the home router to communicate with connected mobile devices. Furthermore, we ultimately decided to use Bluetooth Low Energy (BLE), a wireless technology that emphasizes low cost and low power, in order to communicate between the user's smartphone and the server [3]. The reason for this is because the Raspberry Pi, which will host our server and allow communication between the user's android app and the lock, is battery powered. The Raspberry Pi has the ability to last up to an hour on continuous Wi-Fi while powered by a 4000mAh battery. Other sources reported the Raspberry Pi to last up to 2 hours with a 4000 mAh supply [4]. Because the device we're creating is a lock, the server dying after a few hours would be catastrophic, because it would prevent the user from controlling the lock after only a few hours, potentially locking them or their guests out of the home. By using BLE, however, the server may last many days on end. The tradeoff to this is that the stability of the connection between devices, and the strength of the signals, would be significantly lowered. However, since using the smart lock only requires the user's smartphone to be in the immediate vicinity of the lock, this should not be a significant issue, compared to the battery life issue that would arise from using Wi-Fi.

4. Related Work

The term IoT (Internet of Things) was first invented in 1999 and made its way into the mass market in early 2014 [1]. Because IoT has been exposed to the market for several years, there are a plethora of IoT devices available, and hundreds of smart locks, as a simple Google search shows. To compare smart locks, we introduced two main considerations to determine the quality of the product: Price and Features.

The term price refers to these parameters:

1. Device Price: Price is measured by summing up the cost of every device and component that is used. In a smart lock, this includes: the price of the device, the price of maintaining the server that mediates connections between devices on the network, and the cost of installation.
2. Cost of Installation: The term cost is measured by how much damage it inflicts on the door and door lock it is attached to. As mentioned above, our device is intended to provide support for consumers who are renting or sharing their living quarters. A cost is therefore applied if the pre-existing home materials must be repaired after the smart lock has been detached.

There are three key features to consider:

1. Easy Mounting: To lower the cost of the door, this device should adjust itself to the various heights and sizes of an existing doorknob. The installation and removal should be easy.

2. Lock/Unlock: Since this device should work remotely, users should be able to lock or unlock the door lock from a remote distance. Also in the case when users' phones are out of battery, the door lock should be able to be locked or unlocked manually.
3. Sharing key: When a registered user tries to invite a guest, he/she needs to be able to share keys in order for the guest to unlock the door lock. Keys must be sharable so that another user can access the house.

Depending on the above parameters, the three most similar products are compared in the table below [2].

	August Home ASL-03 Pro	XPRIT Oit'sme Smart Deadbolt Door lock	Candy House Sesame Smart Lock
Need External HUB	Yes (Alexa)	No (internal App)	No (internal App)
Need installation cost	Yes	Yes/No	No
Share Key	No	Yes	Yes
Connection Type	Wifi	Bluetooth	Wifi
Adjustability	No	No	No
Price ($)	279.99	99.96	149.99

Figure 1: Comparison of existing smart locks in the market over the decision parameters.

Comparing the existing products in the market, it is clear that most of them have similar features to what we are trying to build. However, it is also clear that none of the above solutions is highly adjustable for different sizes and shapes of door locks.

The design of a universal grip, a structure that would allow our device to be freely attached onto deadbolt locks of varying dimensions and shapes, was derived from the field of robotics. In industrial robotics, it is necessary to develop a better way to hold an object in order to manufacture a product. The two most commonly implemented designs we found for a universal grip are known as the mGrip and the balloon grip.

A balloon grip is a grip which can hold an object with a balloon which is filled with small particles. It was developed by the Cornell University robotics team and the former version of the balloon grip was used to hold grounded coffee beans. The mechanism of the balloon grip is pressing down the balloon to the object and vacuuming the air inside the balloon so that the balloon can deform and fix the shape to the according object [2].

Similar to the balloon grip, mGrip is a gripping system developed by Softrobotics. The only difference between the balloon grip and mGrip is what they are using to grip an object. While a balloon grip uses a balloon to grip the object, mGrip is using silicon based arms that can flexibly adjust its shape like a human-hand. The mechanism of this process is to put the grip around the object and pull out the air/liquid that is filling the inside of the grip, thus making the shape of the grip fixed to the shape of the object [8].

Both methods of gripping are fairly new and well-developed. However, they still pose certain limitations. One such limitation is that they both require a vacuum state through suction, which can be difficult to implement [9]. The details and cost of such an implementation are not published in detail. Furthermore, they are both difficult to implement within a compact size. In building a smart lock, size is one of the most important aspects of our design; the device must be small enough for convenient installation and handling.

The online group, hackster.io, supervised a project to build “DIY smartphone-connected door locks”. The goal of this project was to build a wired version of the smartphone-connected door lock that can manually close the door lock when the user sends a signal to the device via an application called Blynk [10]. The main function is the same for this project: to build a smart lock that can work remotely using a mobile connection. This product has a compact size, is lightweight, and has a fully functional lock/unlock mechanism.

While the Blynk-hosted smart lock implementation provides many core features that we planned for our own device, our design will also introduce new features that may expand upon their implementation. The key differences between our two products can be classified into the following three categories: software, hardware, and adaptability.

The software of the hackster.io smart lock implementation relied on a third-party IoT platform, known as Blynk, to host a server that would control the communication and behavior of devices on their network [10]. Within our application, however, we will build our own server and run it remotely from the Raspberry Pi. Our application will also provide additional features not present in the aforementioned implementation. Within our application, it would be possible to share a key with friends, granting them temporary access to the lock. This allows for a user’s friends or family to access their home without the user needing to be present.

Furthermore, the hardware of our final device will be structured differently from the hackster.io implementation. We are going to put the overall circuit in the main container and this will be powered by batteries. The advantage of having a power cord is that this machine will not be turned off, so that consistent service is guaranteed. However, if a door is too far from the outlet, then this will occupy much space.

The final difference between our two implementations will be adaptability. In the hackster.io project, they 3D printed the shape of the knob so that it only works on the specific dimension and sizes of a single door that was used for testing [10]. However, the structure used for our design will be more adaptable to variance within the shape and dimensions of the deadbolt locks it will be attached to.

5. Project Description

Our final product is a freely attachable smart lock IOT device, which can be easily attached onto any conventional deadbolt door lock. Once attached, by using the mobile application, the user will be able to automatically lock/unlock the door and get the lock status updated. Overall, just like the conventional IOT system, our IOT system will include four main components: a HUB to receive and transmit signals between devices on the wireless network, a sensor to interact with the HUB, actuator to implement the function and a user friendly mobile application. The physical device of the smart lock will contain the sensor and the actuator of the system and implement the lock/unlock process by the attached grip.

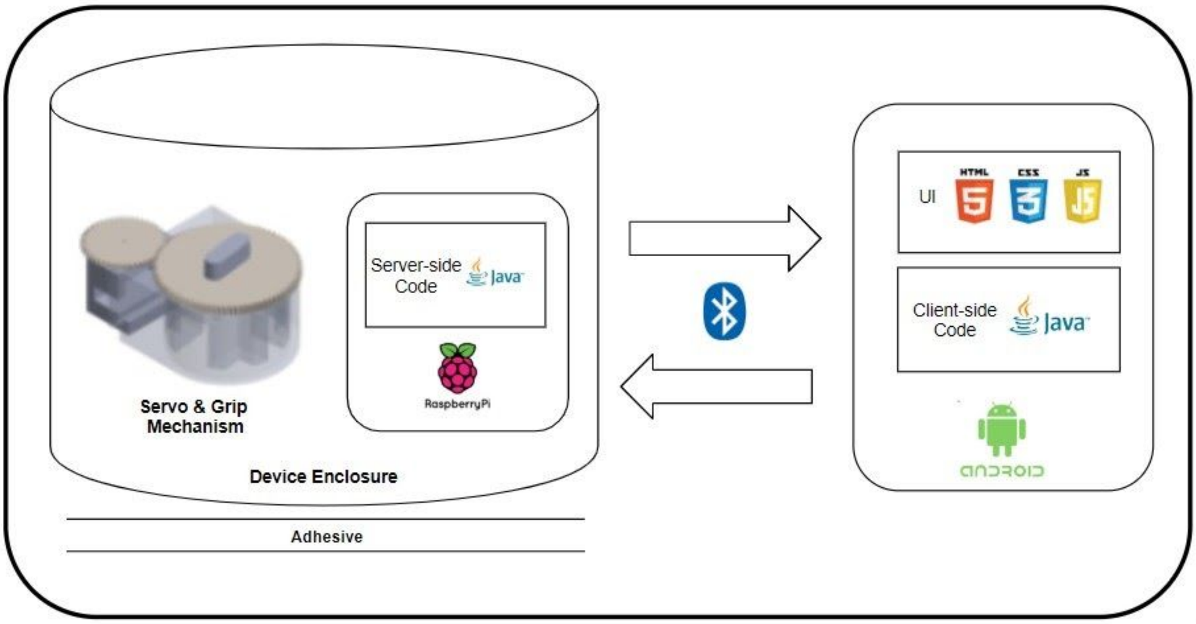

Figure 2: A diagram outlining the architecture of our system

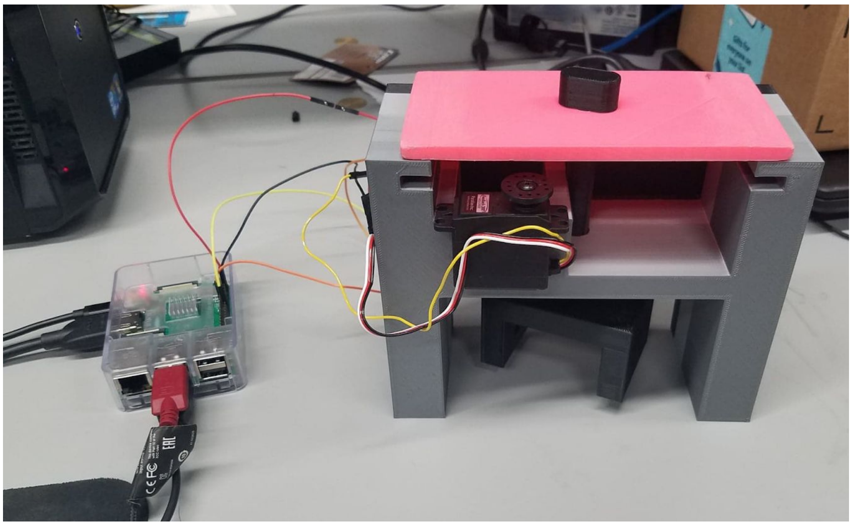

Figure 3: Prototype 1 Setup

5.1. HUB

The HUB of the IOT system makes the connection between all the components of the system. We have decided to use the Raspberry Pi3, which is a single-board computer that can run the Linux operating system and be programmed with Python, to host a server that will handle inter-device communication and signal handling. This will be the core component of our hub. All communication will occur between this server and mobile devices connected to the network. When the server receives a request from a user to manually control the lock, it will operate the Servo motor, which is directly connected to the Raspberry Pi, to rotate the direction and amount required to shift the lock's state.

5.2. Sensor

The biggest advantage of constructing a stable IOT system is that we can easily add any functions by simply connecting an appropriate sensor to the HUB. Currently, we have decided to use a position sensor to fully complete the capabilities of the lock. Using the built in position sensor of the servo motor, we can easily track the change of the motor position through an angular value. When the user first sets up the smart lock and goes for the lock/unlock process for initialization, the HUB will automatically store the status of lock/unlock in a form of angular value. The stored value for a certain type of the lock will then be used to interact with the actuator to make it serve the commands from the user. As for other types of sensors that we deemed appropriate for future expansion, functionality such as fire safety and air quality/carbon monoxide detection through the use of a DS18B20 temperature sensor and an MQ7 sensor. The sensor will be continuously gathering data which will ultimately expand the functionality of the IoT system.

Another possible use case for implementing sensors would be to allow the lock to detect when the user exceeds a certain distance from the lock, after it's attached to the door. The device may be programmed to automatically lock itself once this limit is exceeded. We named this the "automated lock" function, and have considered implementing it as a stretch feature for our device.

5.3. Actuator

Figure 4: QR code for the link to Actuator demonstration

The main function of the smart lock is making the lock to rotate, while the grip of the physical device holds the lock the actuator and will make the actual rotation by getting the command code from the HUB. At first we were trying to implement a normal servo motor, but this actually caused a problem as normal servo motors only serve a 180 degree rotation. We realized that for many types of locks the rotation angle requires value over 180 degrees. To solve the issue we decided to use the Parallax Feedback 360 Degree High Speed Servo, which can provide a continuous rotation for both directions. The rotation of the motor is managed by a program coded with python. When testing the prototype we focused on the stability rotation, so we also checked the delay between the change of rotational direction (approximately 0.5second) to decide an appropriate sleep time between the continuous signal from the mobile application.

5.4. Mobile Application

For easy use of the smart lock, a mobile application will be provided along with the physical device. The app was developed using Android Studio and is connected with the Raspberry Pi using bluetooth. When using the application, to address the security issue, the user will be allowed to create an account, whose information will be potentially encrypted and stored in a database. When the user finishes the installation of a physical device and login with a created account, the guideline for initialization will show up. The initialization will require the user to first rotate the lock manually to store the data of lock/unlock status. Once the initialization is finished the mobile app will display the lock's current status (locked or unlocked). The user can change the lock status by simply clicking the icon that indicates the lock status.

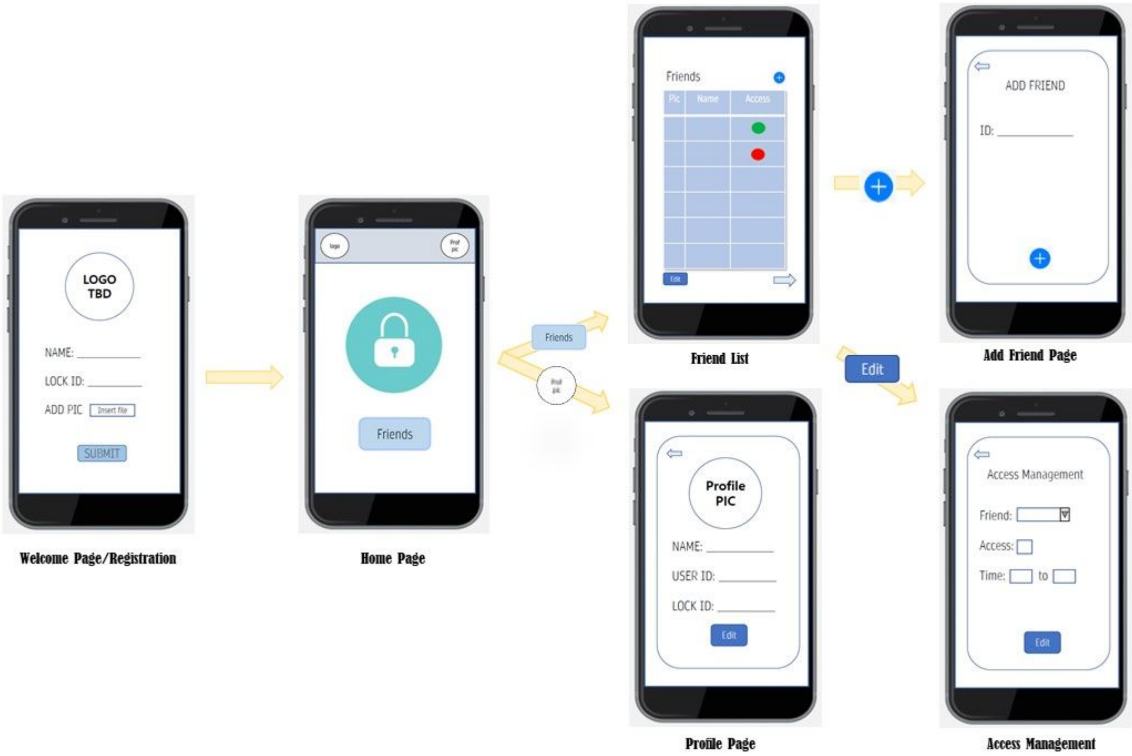

Figure 5: Map for the Mobile App UI

The app also enables the users to manage friends lists. Once the user adds friends, users can send a temporary access code which is used to manage the smart lock for a preset time period. The status of the access approval for each of the friends will be shown in the list and can also be edited in the same page. The temporary access code will be generated each time the users decide to share their authorization and will be deleted once the preset time passes. Users may also edit the profile info whenever needed. To enable the users to keep in track of the battery status of the Raspberry Pi, the app will be showing the status in the main page of the app, this will prevent users from problems occuring due to the low battery.

5.5. Physical Device

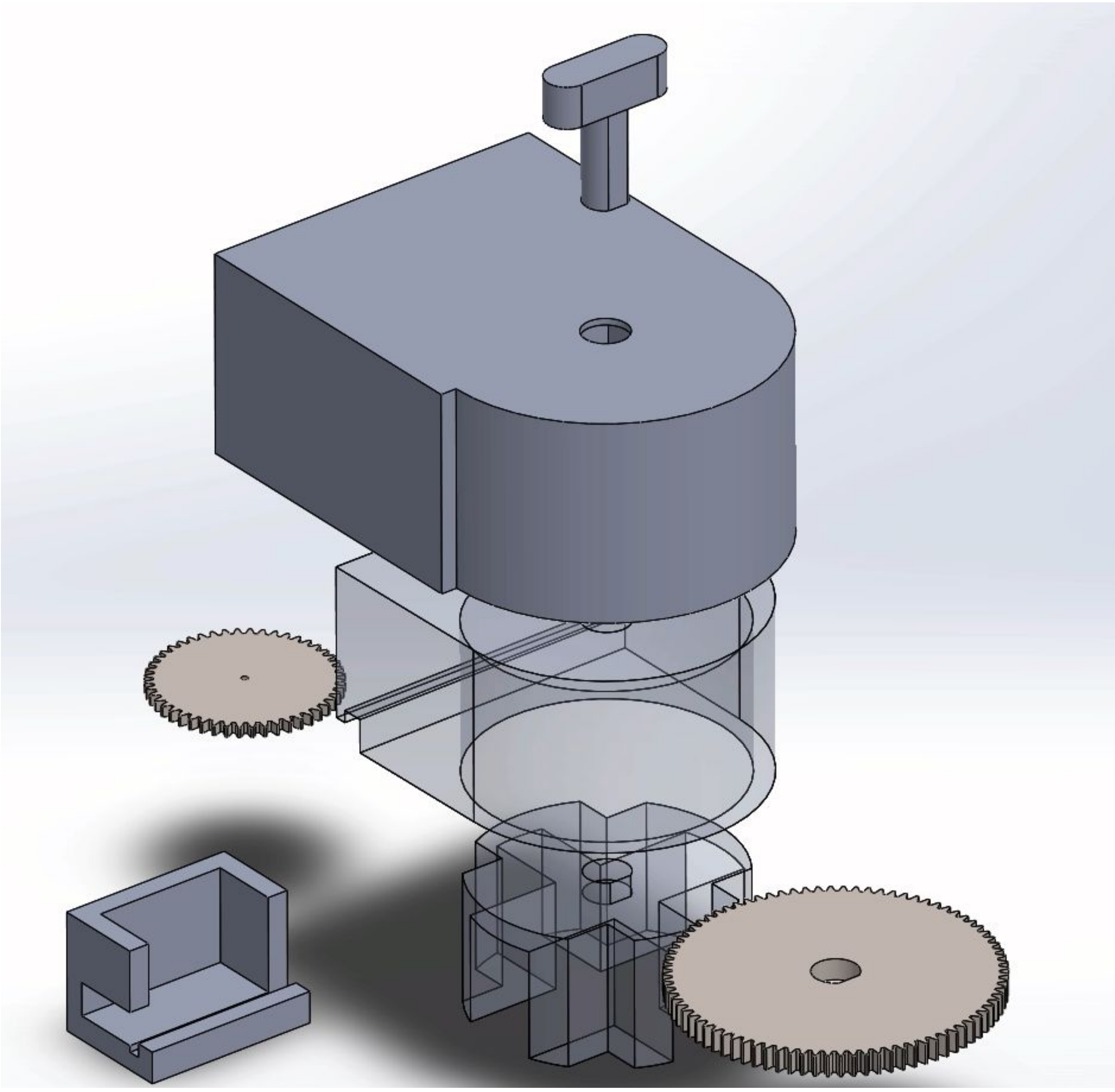

Figure 6. Final Prototype Design (Exploded View)

5.5.1. Design Description

Our physical model consists of the adaptive grip and rotational structure. All the components of the physical device are 3D printed except Raspberry pi and grip. The main goal of this device is to be attachable to any type of the doors and to be adjustable with various dimensions of conventional deadbolt locks. As shown in the figure above, our final prototype looks like 'Jimmy Proof Deadbolt lock'. Figure 6 (a) is the shape of what 'Jimmy

Proof Deadbolt lock' looks like, and Figure 6 (b) shows how our prototype design looks like in Solidworks. The reason for this prototype model is to effectively cover the existing deadbolt lock design. According to the structure, most of the conventional door locks are just a circular design. Additional rectangular space for the raspberry pi and the servo motor is accommodated on the enclosure. When all parts are connected, the prototype will be attached to the door using 3M's double sided tape, which can hold upto 90 pound per inch.

5.5.2 Rotation Mechanism

Conventional deadbolt door locks have a rotational mechanism in which a user just rotates the door knob to open/close the lock. By research, most of the deadbolt locks are in an 'open' state when the knob is turned to the right, and 'close' state when the knob is turned to the left. To control the rotation, Parallax's 'continuous rotation servo motor' will be used. When the servo is installed, two rotating gears will connect the servo with the grip. Since two linked gears will have opposite rotational direction, installation of our model should be guided. Also, in order to have a steady rotation, larger gear will be used to rotate the grip to make the lock rotate slower. The ratio between small and large gear is 4:7, allowing a higher torque to be yielded in turning the lock knob.

5.5.3. Grip Mechanism

There have been lots of attempts to develop a method to grab universal shapes with one device. The original attempt for this project was to import one of those methods to grab various shapes of conventional deadbolt locks. However, it was difficult to implement those technologies due to weight limitation. In order to be attached on the door perpendicular to the axis, it was necessary to make it as light as possible to ensure safety and convenience. Technologies that we have under consideration were elaborated in the previous sections.

This Figure on the right is called 'door knob grip', completely made with silicon and helps to open the door knob easily by increasing the surface area and resistance. Silicon or other soft and elastic materials were considered to be used since it was necessary to adapt to various sizes.

Figure. Silicon door knob grip

Our final model for the grip looks like the figure on the right. There are two different sections for the grip; narrow and wide. This area depends on the size of the door knobs. Then the grip is filled with silicon gel which is used as a supportive insole of the shoes. When the door knob is inserted, the silicon gel is filling the rest of the area so that it effectively holds the knob. The size of the knob is varying so the minimum width and maximum width is based on our neighborhoods' door knob sizes.

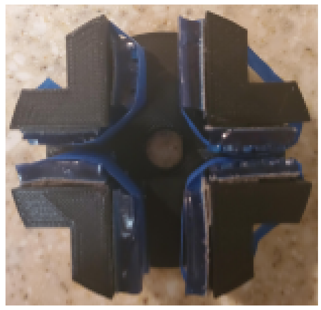
Figure. Prototype grip

6. Status

Smart Lock

ACTIVITY	PLAN START	ACTUAL START	PERCENT COMPLETE
Group Building	9.03	9.03	100%
Topic Assign	9.03	9.24	100%
Activity Log	9.03	10.1	100%
mid-Presentation	10.08	10.22	100%
Design Proposal	10.08	10.22	100%
Report outline	11.05	11.09	100%
Prototype	11.05	12.03	80%
final-Presentation	12.03	12.10	75%
Report Final	11.05	12.10	100%
Actual Product	next semester		

Figure 7: Fall 2019 Activity Log

The above figure displays a Gantt Chart modeling our work timeline for the Fall 2019 semester. During the group building and topic assignment phases, we formulated our group and brainstormed on a project to work on. This phase took longer than it for most groups, because our group wasn't finalized until 3-4 weeks into the fall semester. The mid-presentation and design proposal stages highlight the time spent finalizing our design and presenting it to our peers and faculty. After this phase, we outlined and drafted our end-of-semester report, and began the building phase for a first prototype. By the end of the fall term, the first model for our lock-gripping mechanism was designed and 3D-printed, our Raspberry Pi was set up and configured with the Raspbian operating system, and a basic script was configured to operate the Servo motor.

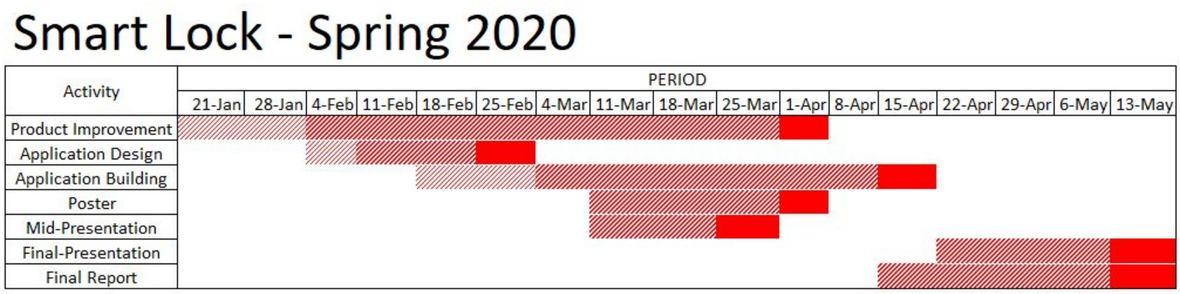

Figure 8: Spring 2020 Activity Log

The figure above was the plan for the most recent semester. We have finished our product improvement, application design process and so on. Due to the school being inaccessible, the device is not completely polished, but it has been finished in an appropriate amount and serves its foundational purpose. The code and 3D design will remain as an open source so that anyone who wants to access our design can download and print out the part and all the components used in this project will be recorded. Video of the demo and installation guide will be documented in order to know how our product would work in the market if any further improvements are made. The figure below shows the current device.

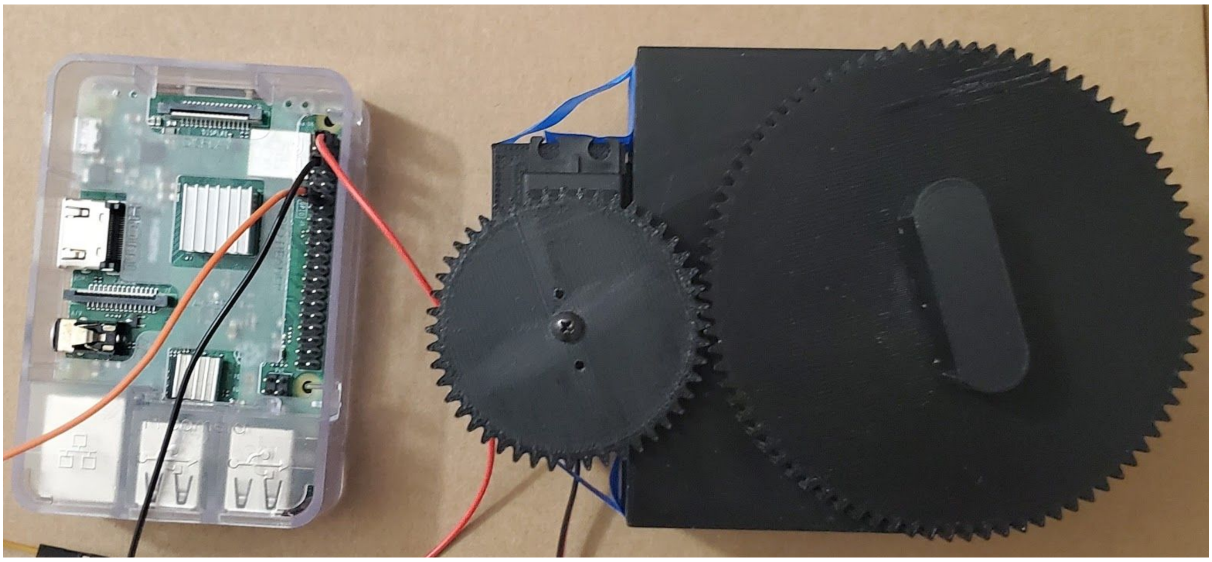

Figure 9. Final Prototype

VII. Conclusion

The purpose of this project is to create a smart lock device that can be freely attached onto any conventional deadbolt lock. The user can then remotely control the lock through the accompanying Android application. The motivation behind this project was that many conventional smart lock devices currently in the industry pose challenges in cost and adaptability, particularly in the flexibility of installing and reinstalling the lock, and the costs required to purchase, install, and maintain the lock. This design attempts to provide a cost-efficient, reliable, and easy-to-install smart lock that is practical to use for all users, and does not require home renovation or expensive server maintenance costs to use.

The project is an Internet-of-Things system containing a hub, actuator, and communication devices. A Raspberry Pi model 3B+ was used in conjunction with a smartphone application for the hub, and will be responsible for handling bluetooth signals sent between the various devices on the network. A 360 degree feedback servo motor will be used for the precise angular rotation of the deadbolt lock that the device will be attached to, into locked and unlocked states.

At this juncture, the mechanical design and the software designs have been fully integrated into a working final prototype. While many more improvements can be made, the inability to access important resources at our institution hinders us from fully achieving all the intended goals and vision.

Works Cited

1. Knud Lasse Lueth. "Why the Internet of Things is called Internet of Things: Definition, history, disambiguation". Iot-analytics.com. https://iot-analytics.com/internet-of-things-definition/ (12/16/2019)
2. Various Smart Lock implementations in the industry, https://www.amazon.com/August-Smart-Lock-Connect-technology/dp/B0752V8D8D https://www.amazon.com/XPRIT-Oitsme-Smart-Deadbolt-Door/dp/B07WPFZBV5/ref=sr_1_1?keywords=XPRIT+Oit%27sme+Smart&qid=1582236184&sr=8-1 https://www.amazon.com/CANDY-HOUSE-Sesame-Enabled-Champagne/dp/B0763MY9NG?th=1
3. Bell, Tyler, et al. *Smart Sense*. 2016, *Smart Sense*.
4. Moll, Luke, et al. "How Long Does the RPi Last on a Battery?" *Raspberry Pi Stack Exchange*, 3 July 2014, raspberrypi.stackexchange.com/questions/19812/how-long-does-the-rpi-last-on-a-battery
5. Hattersley, Lucy. *Raspberry Pi 4 vs Raspberry Pi 3B+*. The MagPi Magazine, 1 Oct. 2019, magpi.raspberrypi.org/articles/raspberry-pi-4-vs-raspberry-pi-3b-plus.
6. Alisa. "Raspberry Pi or Arduino – Which Is Best for IoT Home Solutions?" *Seeedstudio*, 1 Jan. 2019, www.seeedstudio.com/blog/2019/01/29/raspberry-pi-or-arduino-which-is-best-for-iot-home-solutions/.
7. Block, Michael, and Derek Toub. *AFLAT: Automatic Fast Locking Actuated Tuner*. 2013, *AFLAT: Automatic Fast Locking Actuated Tuner*.
8. Anne Ju. "Balloon filled with ground coffee makes ideal robotics gripper". news.cornell.edu. https://news.cornell.edu/stories/2010/10/researchers-develop-universal-robotic-gripper (11/05/2019)

9. Eugene Demaitre. “mGrip modular gripper is Soft Robotics’ response to user needs”. Therobotreport.com. https://www.therobotreport.com/mgrip-modular-gripper-soft-robotics/ (12/04/2019)
10. Hacker Shack (Hackster.io), “How to Make a Smartphone Connected Door Lock”, (Jan 17, 2017). Accessed Dec 10, 2019. [Online Video]. https://www.youtube.com/watch?v=bAcK80fm1_0