Locked and Loaded

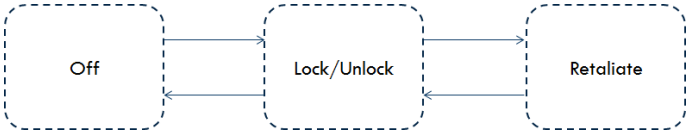
Team members: Quinn Johnson, Harrison Tsai, David Wu

Project Vision and Content

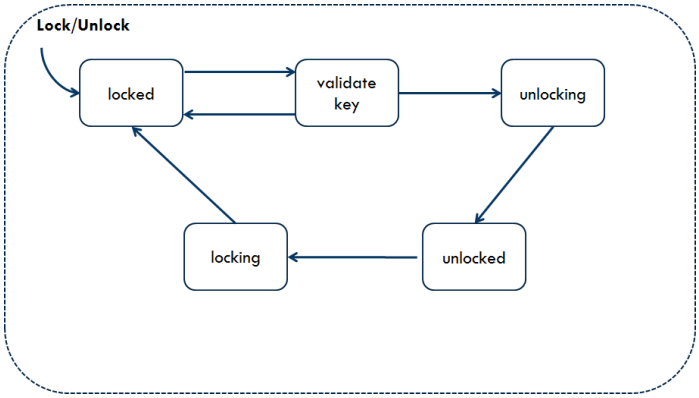
The project vision is to create a retaliatory smart lock. Most of the initial vision remains the same; however, several design changes were made due to the safety critical nature of the project. We diverted from our original plans by fleshing out our retaliate state. Rather than simply firing on all possible intruders we wanted to minimize the number of false positives. We created a separate region to distinguish between an armed and unarmed state, with LTL formulas that specify when retaliation can occur. Additionally, we wanted to enable override systems and add an additional IR sensor to detect if an intruder was actually present.

Milestone Updates

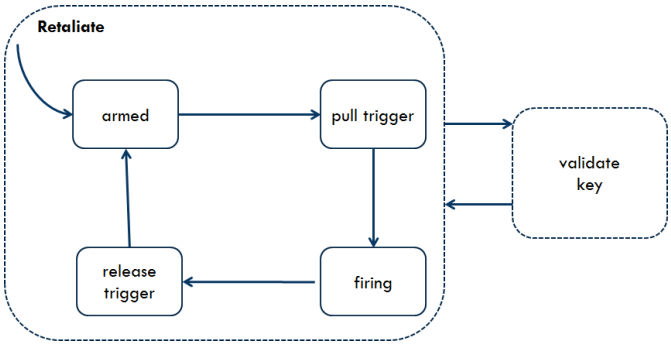
We still plan to unlock via bluetooth and retaliate with a nerf gun, but some of the guards to enter into a retaliation attempt were changed to include a position sensor. One key change was to include an initial off state to enable the user to simply turn the system off. Additionally, override features were added throughout the system to enable the user to override the armed state. On a high level, our finite state machine looks like:



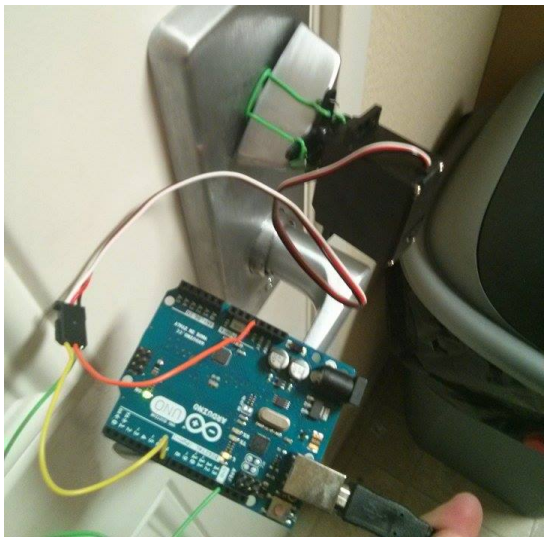
Our Lock/Unlock region is:



Our retaliation region is:



Hardware is starting to come in as we will be picking up the nerf gun tomorrow and will be starting to assemble every component together. We ran into a lot of issues getting the FRDM-KL25Z boards to connect to the computer. We've begun testing the servo independently with an Arduino board against our apartment's lock as shown below.



Mini-update Questions

One comment made by Professor Lee was about the consideration of a stepper motor. After conducting additional research, we have concluded a stepper motor is more appropriate. A stepper motor enables better positional control compared to a servo motor. "Where servos require a feedback mechanism and support circuitry to drive positioning, a stepper motor has positional control via its nature of rotation by fractional increments."[1] Given the wide variety of locks, sacrificing "high speed and high torque"[2] for positional accuracy is appropriate as we do not want to break the lock and want to accurately support multiple locks; however, since we have already committed to a servo, making a change now for a proof of concept is not appropriate.

As mentioned in the milestone updates, we ran into issues with getting the mbed boards to work. We have started to work with an Arduino to at least get the servo motor working. In the meantime, we will continue to work with the GSI's to fix the issues and start writing up the code to program the servo and wireless connectivity.

Upcoming Plans

At this point, we have iterated through our model with feedback from instructors, and have tested some individual hardware components. Our next step is to combine these hardware components to implement portions of the FSM model. We will begin by implementing the Lock/Unlock region, without any bluetooth key validation or transitions out of the region. With that completed, we will separately test the IR sensor detection, and integrate a transition into/out of an armed state. From there, we will repeat the process, integrating the remaining hardware components (key validation / accelerometer / Nerf servo).

[1] https://www.modmypi.com/blog/whats-the-difference-between-dc-servo-stepper-motors
[2] http://components.about.com/od/Components/a/Stepper-Motors-Vs-Servo-Motors-Selecting-A-Motor.htm