UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

Knock-Turn Lock

Electrical & Computer Engineering

4/28/2023

Team Introduction

Jack Kelly

Electrical Engineering
Focuses: PCB Design, Actuator and Power Circuitry

Vishal Rajesh

Computer Engineering
Foci: Sensor Verification, Compliance

Adam Frerichs

Computer Engineering
Focuses: Microcontroller, Arduino Programming

Knock, knock.

Who's there?

Doris.

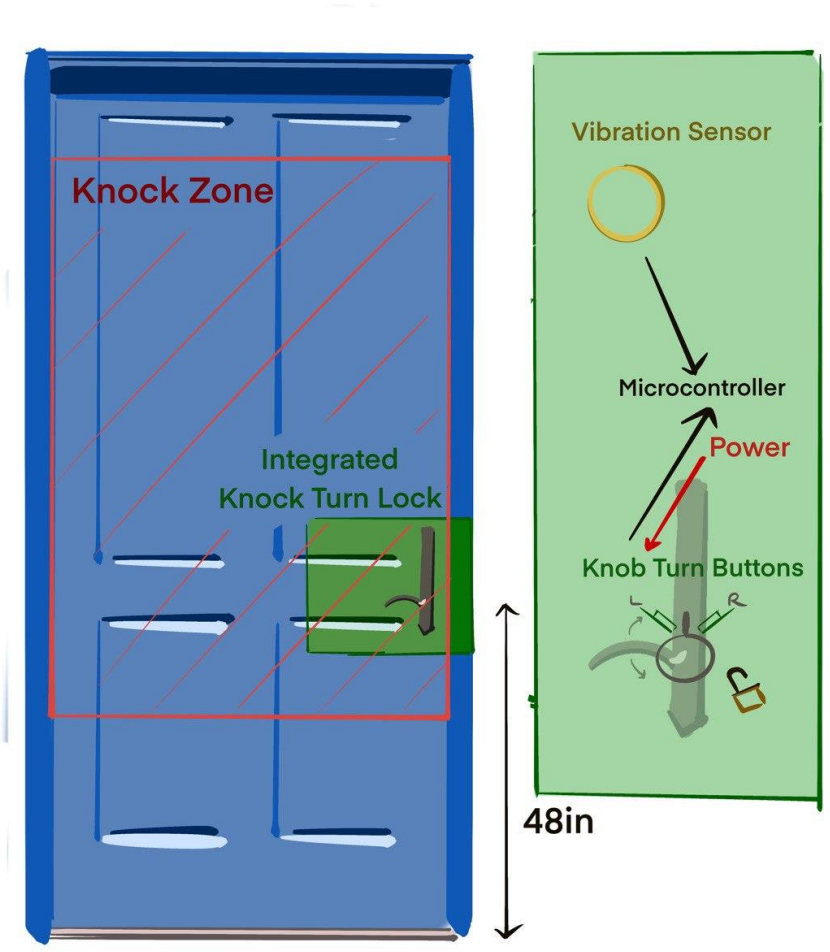
Doris who?

Doris locked, that's why
I'm knocking!

Project Objective

How do we maintain access to a space without a physical token? Consider AirBnB rental use case. There should be a secondary information token.

Motivation



Problem:

It is very common to lose access to a space due to lack of physical token -- losing or not bringing a key. Current solutions are either clearly insecure, such as hiding a key, or convey an unsightly and obvious second path of entry, such as keypad lock.

Keypads

- Often large and unsightly
- Convey an obvious secondary path of entry

RFID Reader

- Convey an obvious secondary path of entry
- Physical key may be lost or stolen, similar to traditional lock/key setups

Hidden Key

- The key under the doormat, in the flower pot, etc. can be easily stolen or lost.
- Not possible in a rental situation

Introducing the Knock-Turn Lock

An auxiliary combination lock based on door knocks and doorknob turns that offers a secure and convenient secondary access alternative.

Hiding the circuit within the door simultaneously maintains the aesthetic and conceals the existence of the secondary combination lock.

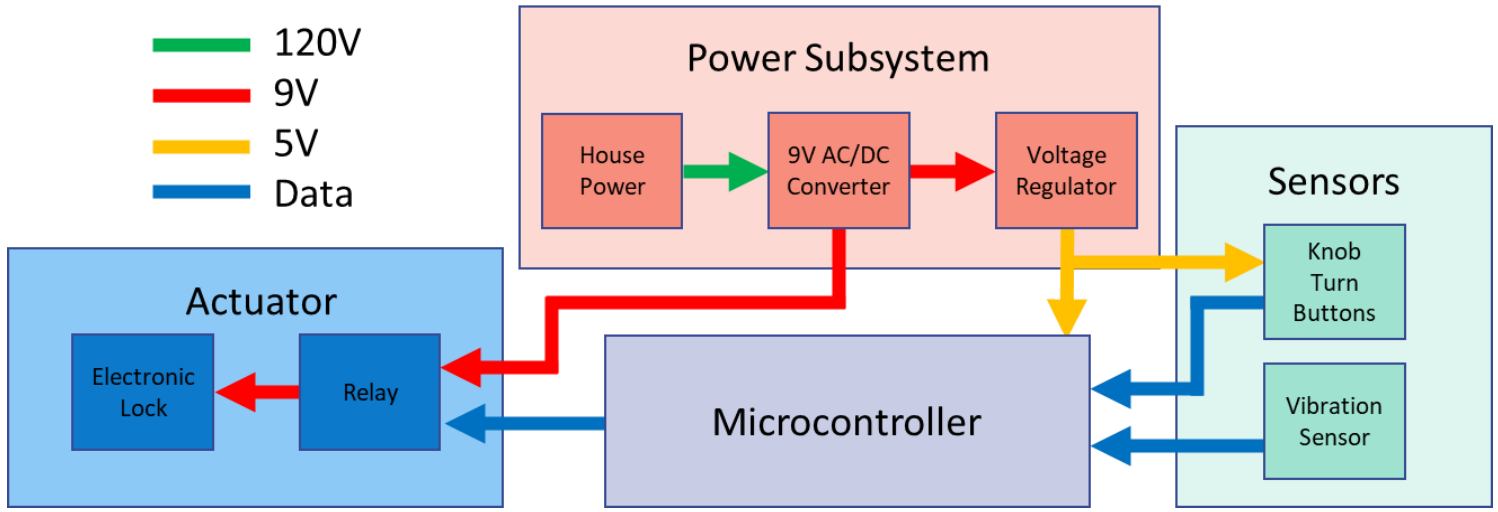
Video Demonstration

Hardware Design

Getting a handle of things

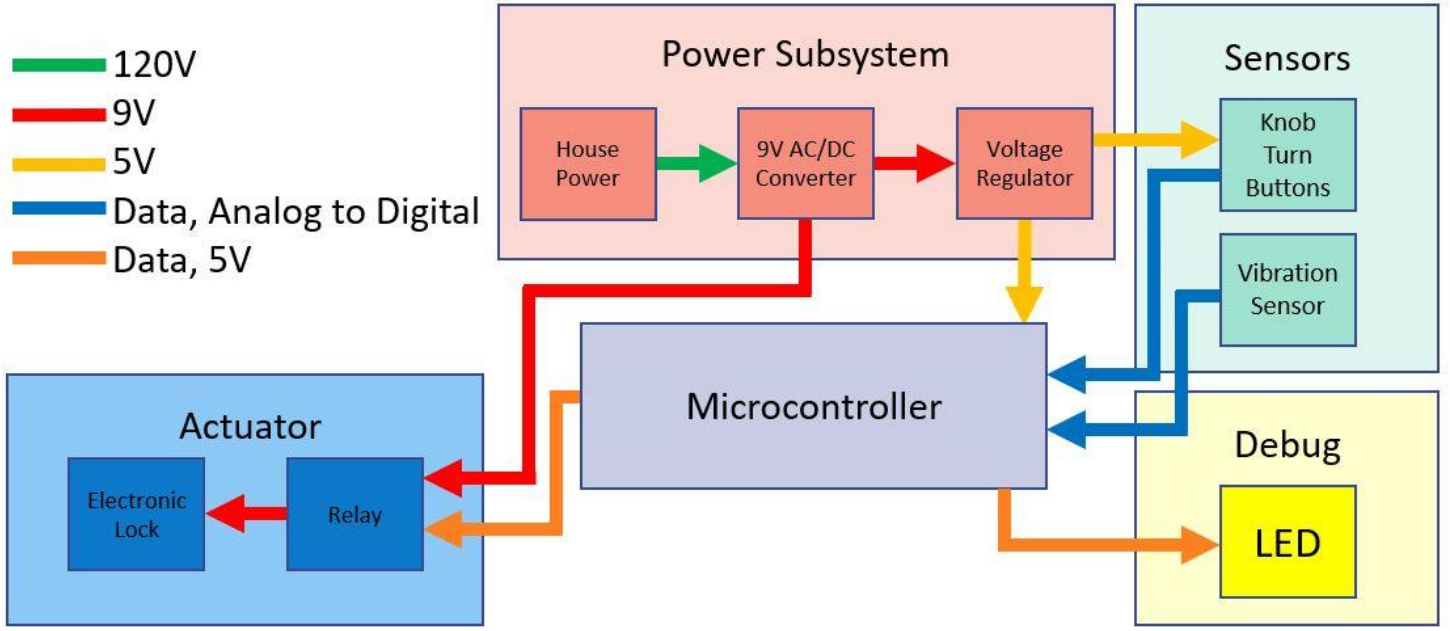
Original Design

- Microcontroller is the central hub
- Sensors are input
- Actuator is the only output



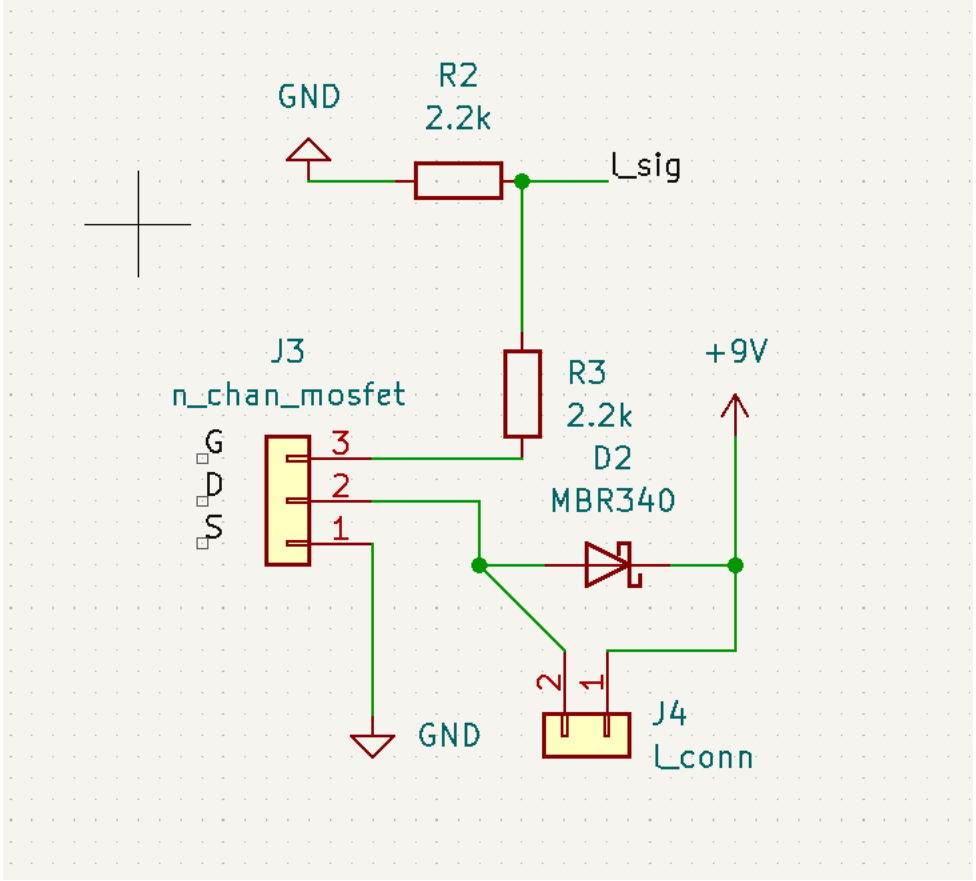
Revised Design:

- LED added for debugging
- Programming header removed
- IC socket added for microcontroller
- Software debouncing

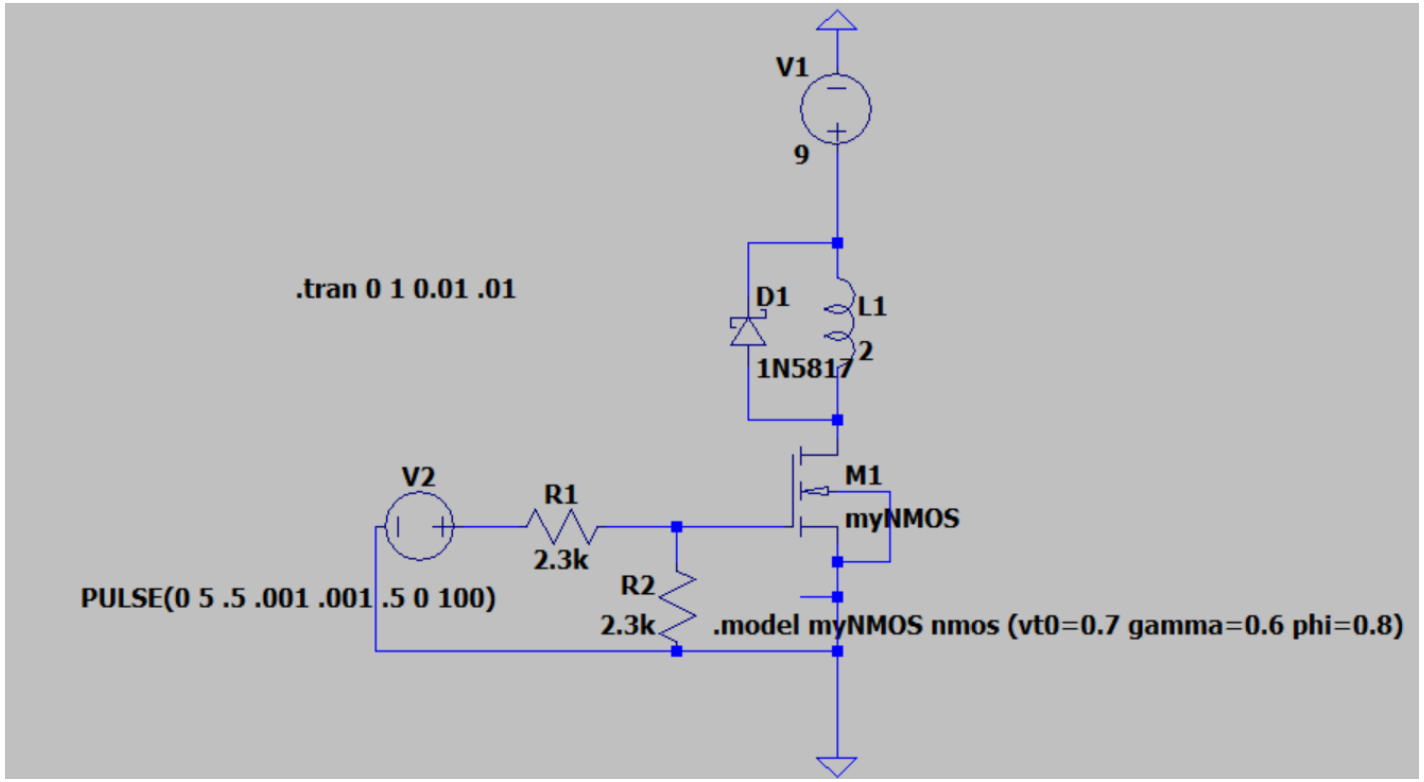


PCB Final Lock Relay Circuit

- MOSFET relay
- Flyback diode



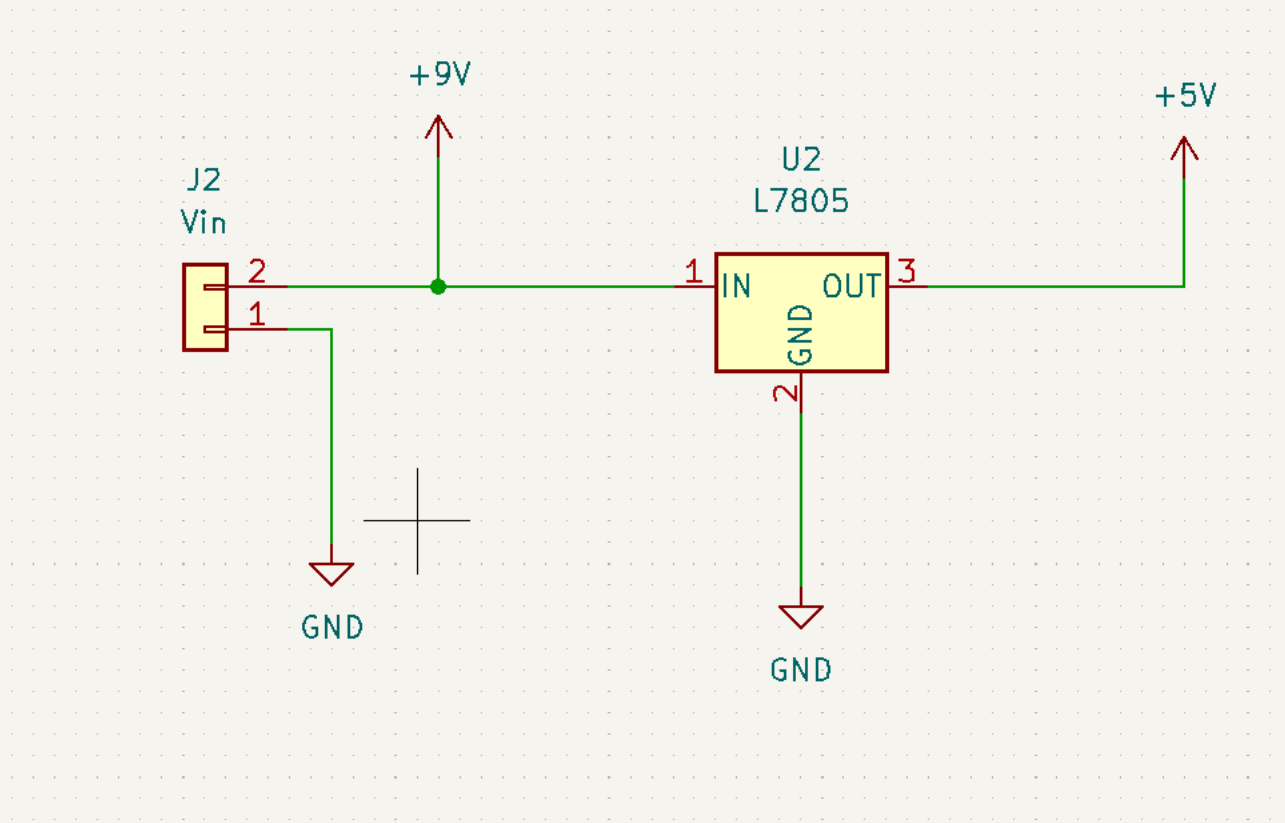
Actuator Simulation



LTSpice Simulation

- Lock modeled as inductor
- Generic NMOS

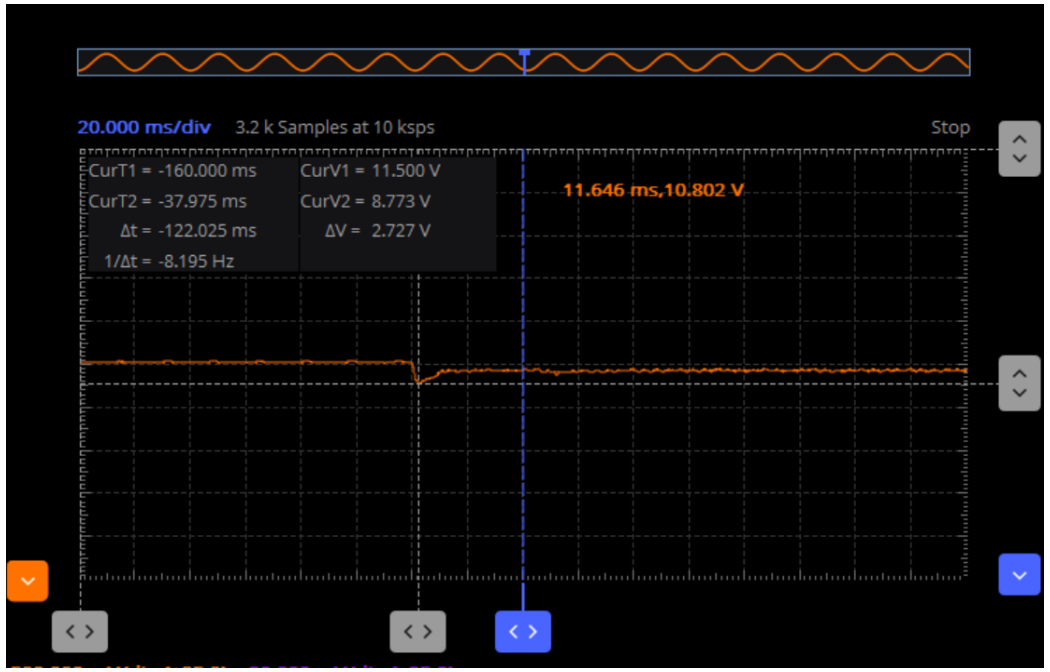
Actuator Simulation



Power Design

- High tolerance for DC stepdown
- Low power use in standby

Power Analytics

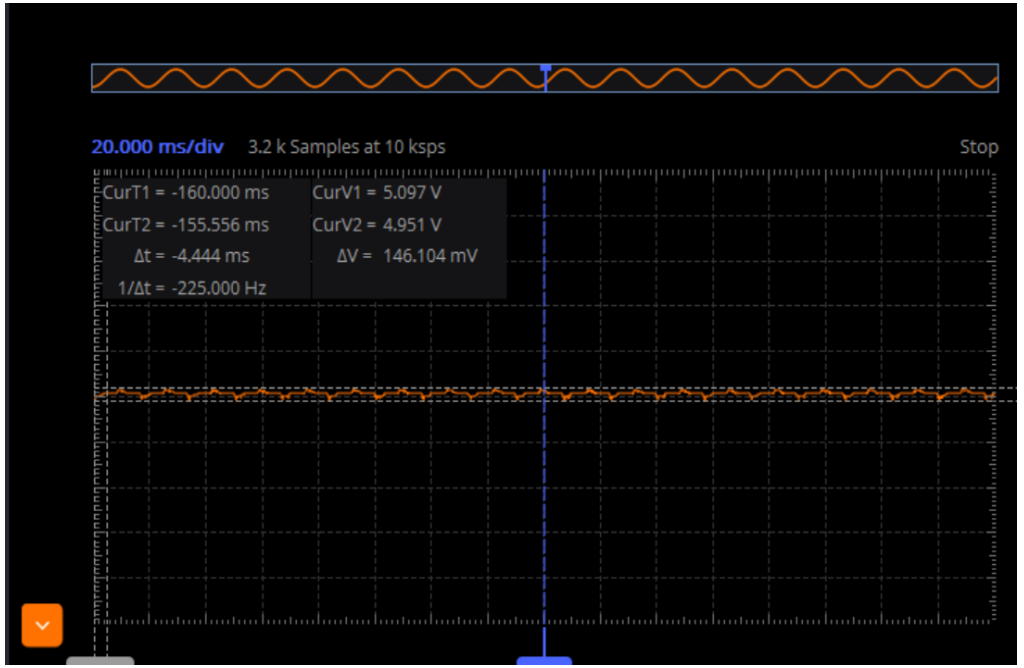


Low Standby Power

- Voltage drop on lock unlocking
- Low standby due to lock being active on low

High Tolerance

- 1.8-5.5V for microcontroller
- Power of 60mW for microcontroller, 2.525mW for sensors

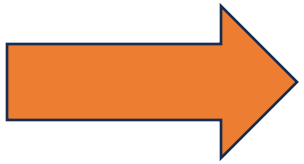


Software Flow

Twists and Turns

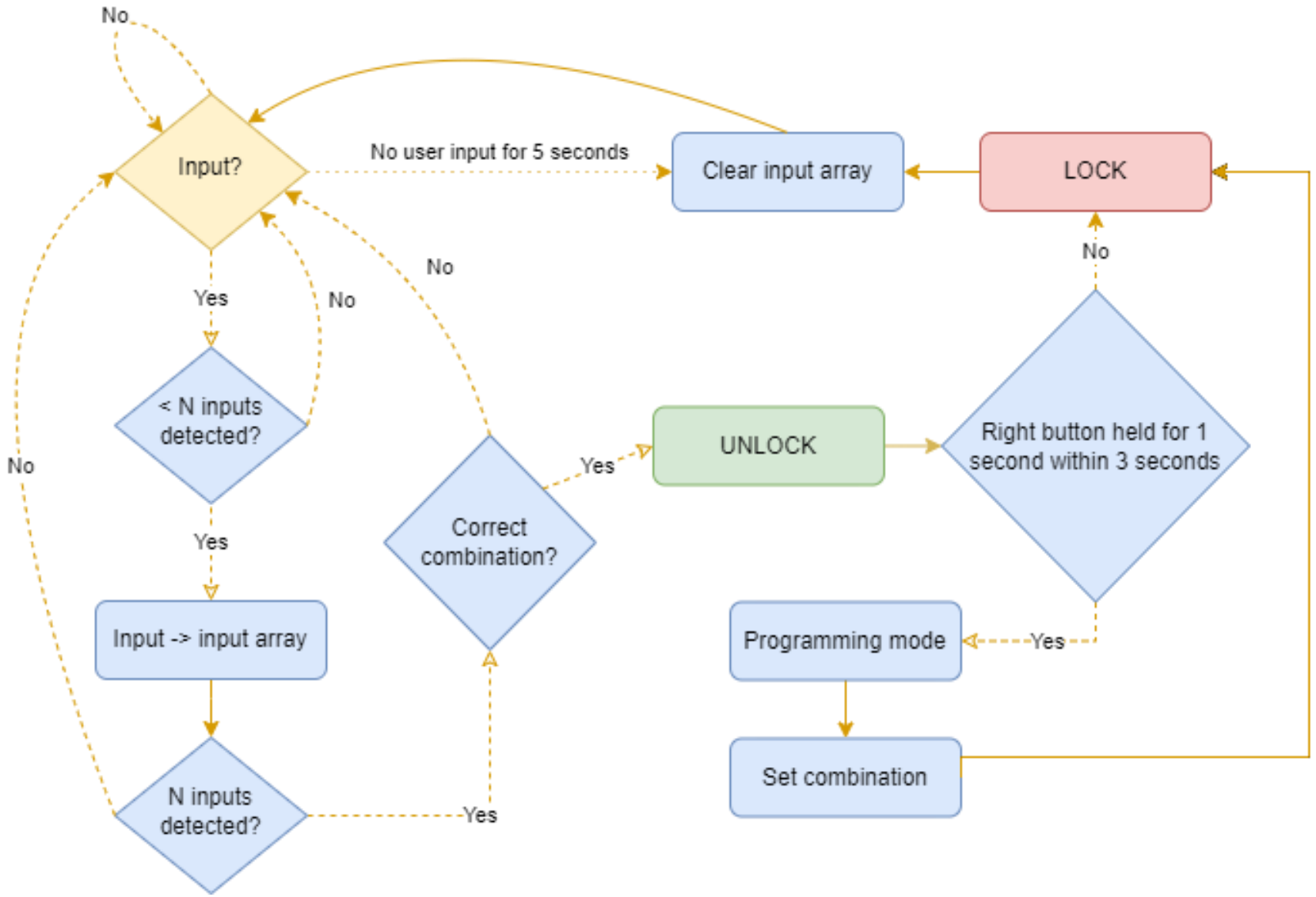
Overview

Setup:
1. Configure pins
2. Set initial combination



Main Loop

Main Loop


No
Input?
No user input for 5 seconds
Clear input array
LOCK
No
Yes
No
No
< N inputs detected?
No
Yes
Correct combination?
Yes
UNLOCK
Right button held for 1 second within 3 seconds
Input -> input array
Programming mode
Yes
N inputs detected?
Yes
Set combination

Program Size

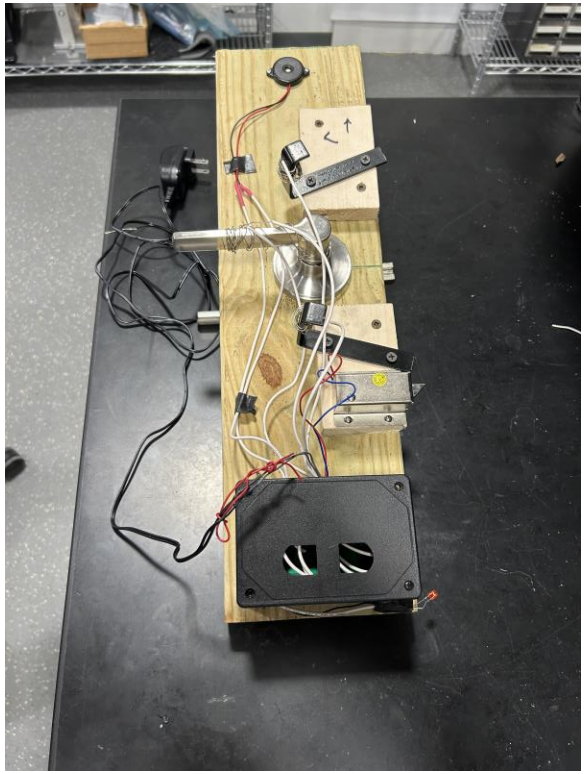
Program is efficiently written and fits well within the 8kB requirement

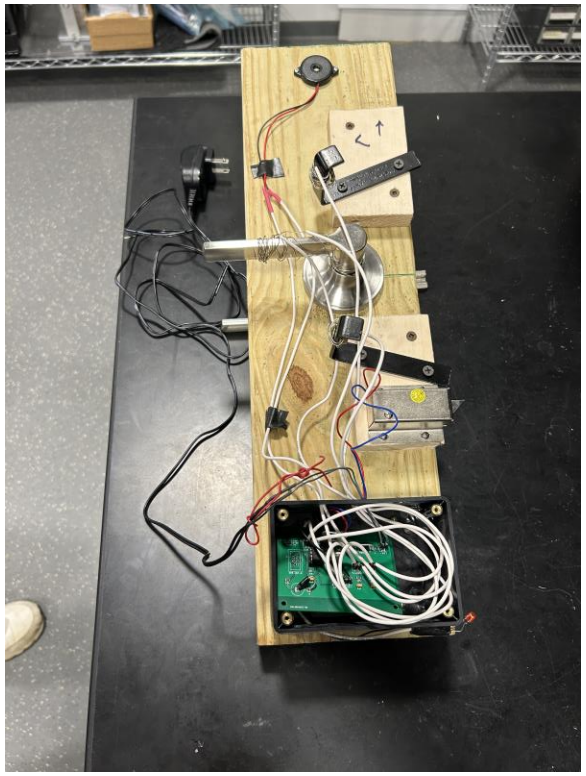
```
Done compiling.
"C:\\Program Files\\WindowsApps\\ArduinoLLC.ArduinoIDE_1.8.57.0_x86__mdqgnx93n4wtt\\tools-builder\\ctags\\5.8-arduino11/ctags" -u --la
Compiling sketch...
"C:\\Users\\freri\\Documents\\ArduinoData\\packages\\arduino\\tools\\avr-gcc\\7.3.0-atmel3.6.1-arduino7/bin/avr-g++" -c -g -Os -w -std
Compiling libraries...
Compiling core...
Using precompiled core: C:\Users\freri\AppData\Local\Temp\arduino_cache_91630\core\core_attiny_avr_attiny_cpu_attiny85,clock_internal8
Linking everything together...
"C:\\Users\\freri\\Documents\\ArduinoData\\packages\\arduino\\tools\\avr-gcc\\7.3.0-atmel3.6.1-arduino7/bin/avr-gcc" -w -Os -g -flto -
"C:\\Users\\freri\\Documents\\ArduinoData\\packages\\arduino\\tools\\avr-gcc\\7.3.0-atmel3.6.1-arduino7/bin/avr-objcopy" -O ihex -j .e
"C:\\Users\\freri\\Documents\\ArduinoData\\packages\\arduino\\tools\\avr-gcc\\7.3.0-atmel3.6.1-arduino7/bin/avr-objcopy" -O ihex -R .e
"C:\\Users\\freri\\Documents\\ArduinoData\\packages\\arduino\\tools\\avr-gcc\\7.3.0-atmel3.6.1-arduino7/bin/avr-size" -A "C:\\Users\\f
Sketch uses 2028 bytes (24%) of program storage space. Maximum is 8192 bytes.
Global variables use 35 bytes of dynamic memory.
```

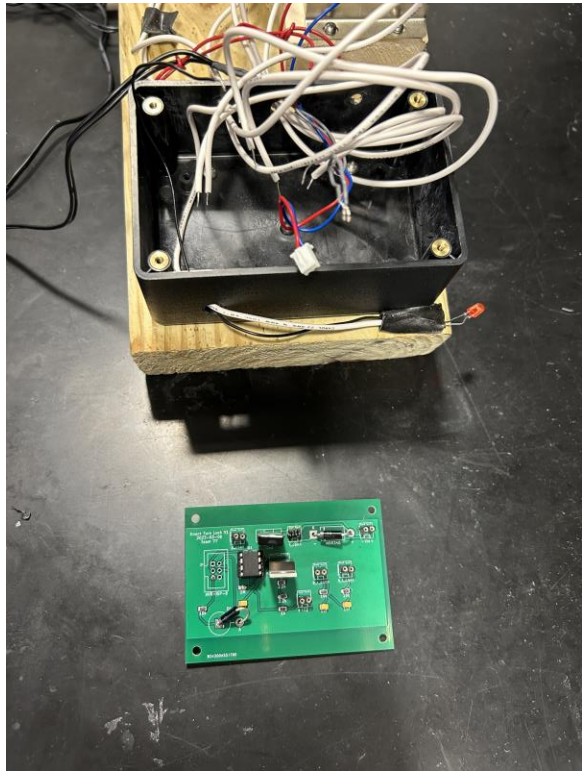
Final Build

Knock on Wood

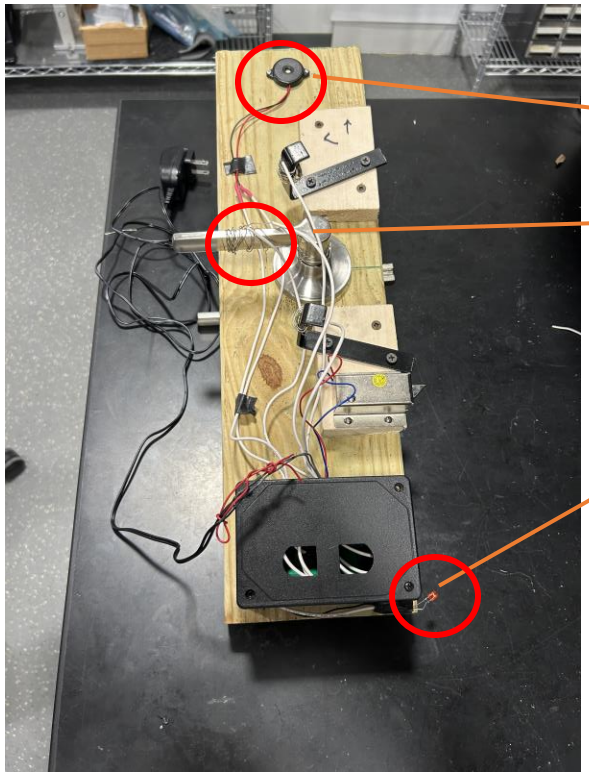
Behind the Door





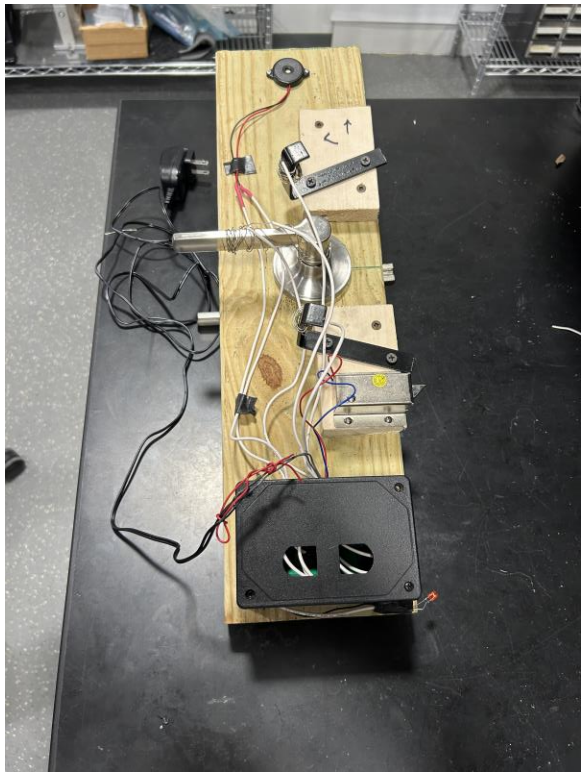


Behind the Door



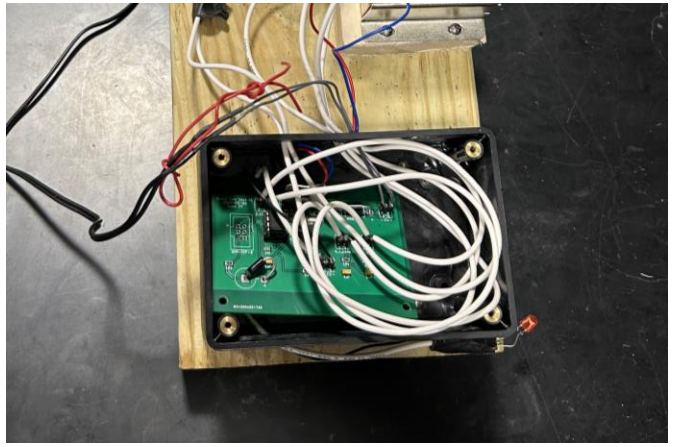
Annotated Electronics

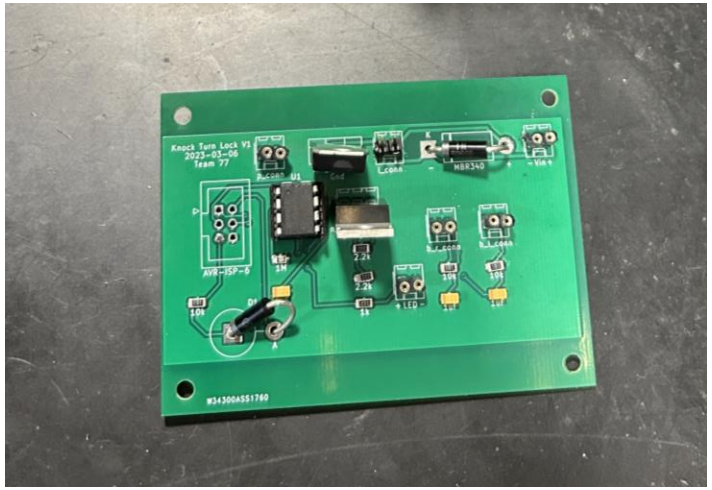
- Single Piezo Sensor
- Contact wire on door handle
- Debug LED



Annotated Electronics

- Single Piezo Sensor
- Contact wire on door handle
- Debug LED



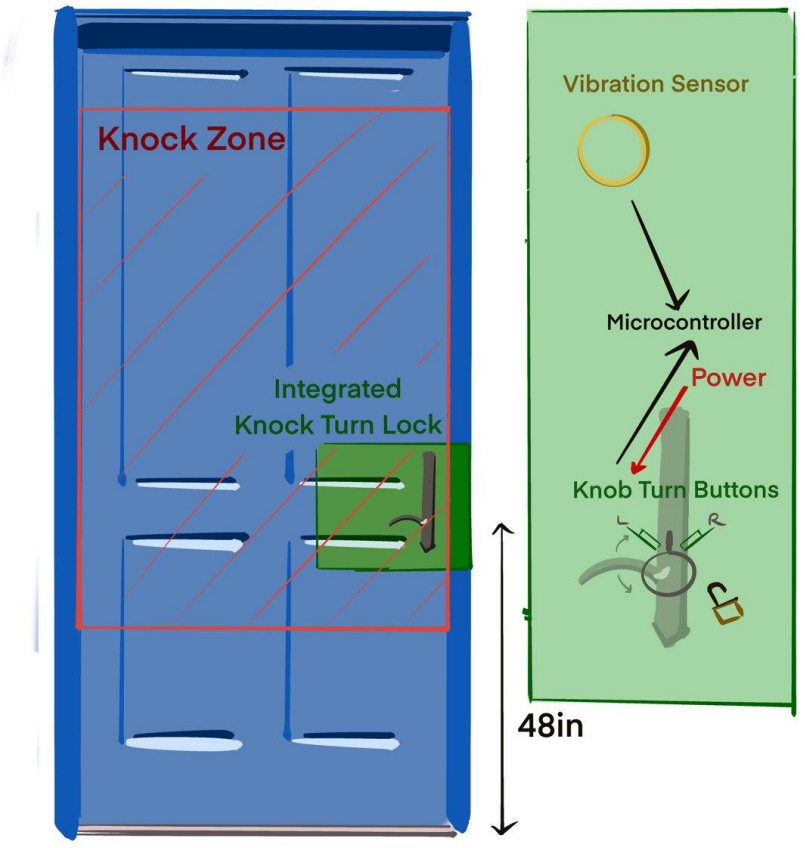


Moving Forward

One door closes ...

Additional Features

- Rhythmic knock sequence
- Variable sequence length
- Advanced door handle design
  - Integrated contact pads
  - Integrated key fail safe
- Piezo sensor array
- Audio user feedback



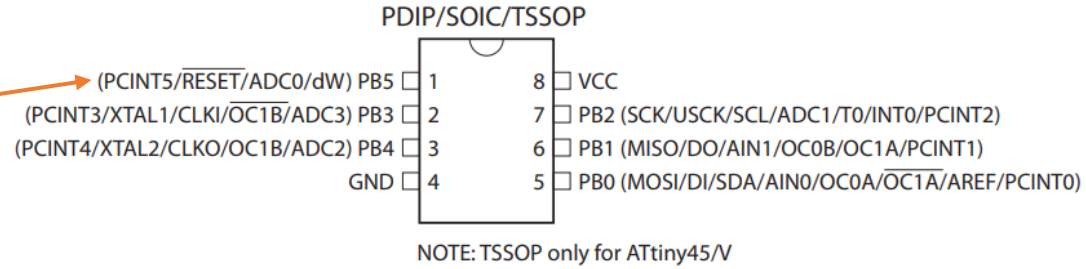
Conclusions

Locking in a final design

Lessons Learned

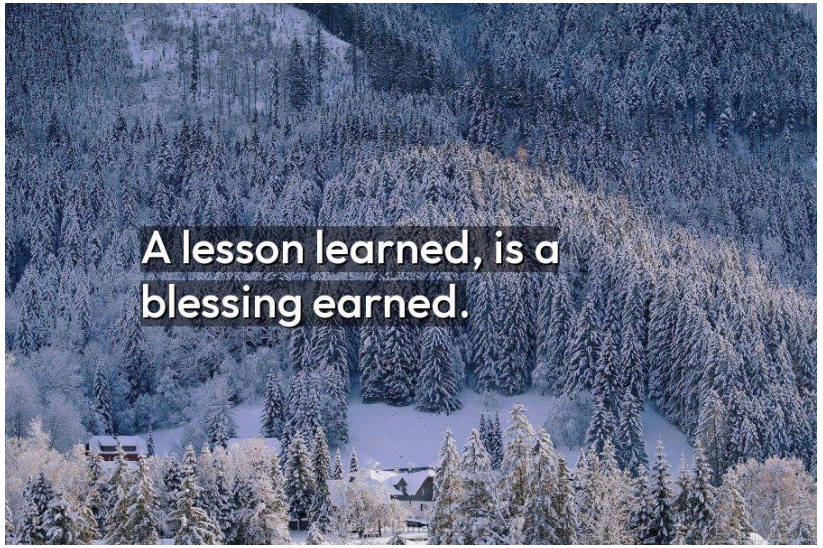
- Simulate Circuit Conditions
- Test, test, test!
- Module testing through program hot-swapping
- Understand datasheets (RESET active low)
- Add test points to PCB

Figure 1-1. Pinout ATtiny25/45/85



Design Improvements

- Hollow Door
- Lower threshold for knocks



Questions?

Adam Frerichs (adamdf2@illinois.edu), Jack Kelly (jacktk2@illinois.edu), Vishal Rajesh(vrajesh2@illinois.edu)