SENIOR DESIGN

SECURITY OF ELECTRONIC SAFETY LOCKS

PROJECT SPECIFICATION DOCUMENT - GROUP 3

Neal Gardella
Ryan Ostroff
Connor Owen
Russell Owen

May 15, 2023

Contents

1 Overview

1.1 Executive Summary

The purpose of this project is to implement existing results from the previous analysis of a LA GARD ComboGard Pro 39E electronic combination safe lock in order to create a safebuster device that is able to interface with and open the specified lock without knowledge of the pin. The current state of the project is in the analysis phase where as a team we are analyzing the lock in order to find additional vulnerabilities that can be used to create the safebuster device or support the current analysis that has already been given. The tools used for the analysis are a Saleae logic analyzer which scans the digital and analog channels on the internal PCB of the lock and a soldering station for attaching test points. Goals completed during this time were the successful attachment and test analysis of the logic analyzer as well as investigating potential micro-controllers for use in cracking the lock.

1.2 Team contact information and protocols

Name	Email
Neal Gardella	gardelln@oregonstate.edu
Ryan Ostroff	ostroffr@oregonstate.edu
Connor Owen	owenco@oregonstate.edu
Russell Owen	owenrus@oregonstate.edu

Table 1.1: Contact Information

1.2.1 Expected Contributions

Name	Role	Expected Contributions
Neal Gardella	Research and design of electronics.	Research timing attacks. Breadboard development. Test analysis methods.
Ryan Ostroff	Research and design of PCB.	Research circuit architecture vulnerabilities, Research possible components. Test analysis methods. PCB design and fab.
Connor Owen	Research and design of microcontrollers.	Research power/side channel analysis. Microcontroller platform. Code the microcontroller. Test analysis methods.
Russell Owen	Research and design of microcontrollers.	Research motor control vulnerabilities Research possible components. Code the microcontroller. Test analysis methods.

Table 1.2: Expected Contributions

1.2.2 Team Protocols

Listed below are the agreed upon team protocols for work quality and communication.

Topic	Protocol	Standard
On time Deliverables and Tasks	Drafts of all individuals' contributions to teamwork submissions should be fully complete by the deadline as outlined in the tasklist such that as a team we may review them during our scheduled meetings.	Work submitted as complete will include all necessary content and formatting requirements listed in the ECE44X canvas and error free.
Task management and Organization	The team will use the shared Google Drive for all of their organization of files and will use the Task List within the drive to track their record of completion	During team meetings, the team will review the tasks to be completed and assign out parts to individuals. When a task is complete, it should be marked green so that the team can review during the next meeting.
Team Communication	The team will treat every other team member with respect and will try to resolve matters in a civil manner.	During team meetings or during individual lab working time, team members will carry themselves cordially and civilly.
Reaching Out for Help	The team member will reach out for help on a given task before the given deadline if they cannot complete the task.	During any lab time or meeting a team member should reach out for help with their given task if they cannot complete it before the deadline by formally asking another individual team member over text or in person.

Table 1.3: Team Protocols

1.2.3 Project Sponsor Communication and Collaboration Preferences:

1. Project Partners' Interest: Vincent Immler is very interested in the project and would like to contribute in any way possible, through providing equipment to training team members on how to use certain solder stations. Their main role appears to be more hands on and supportive than other project partners in the past.

2. Project Partners' profession: Vincent Immler is a professor for the Oregon State University school of Electrical Engineering and Computer Science.

3. Main Information: Immler is looking to get to know vulnerabilities within a certain set of locks, some of which are already publicly available and some which are private. This is so he can build better analysis and safe cracking methods. Immler also would like to know the solution to opening some of these locks so we as a team can create a safe cracking method that is efficient and universal.

4. Technical Knowledge: Vincent Immler knows a great amount about security vulnerabilities as his area of research and previous work is in the field. He knows the protocols that are used within the locks as well as the types of attacks that can be used to open them.

5. Preferred format and Frequency of Communication: We as a team have been instructed to check in with Immler through email every two weeks and when we have any questions about the project. We may also meet in person or in the Analog Mix Signal Laboratory if we need help beyond that of an email such as soldering or using equipment.

6. Other Individuals: There aren't any others who our Project Sponsor has shown various types of communications to during our initial meeting.

1.3 Gap Analysis

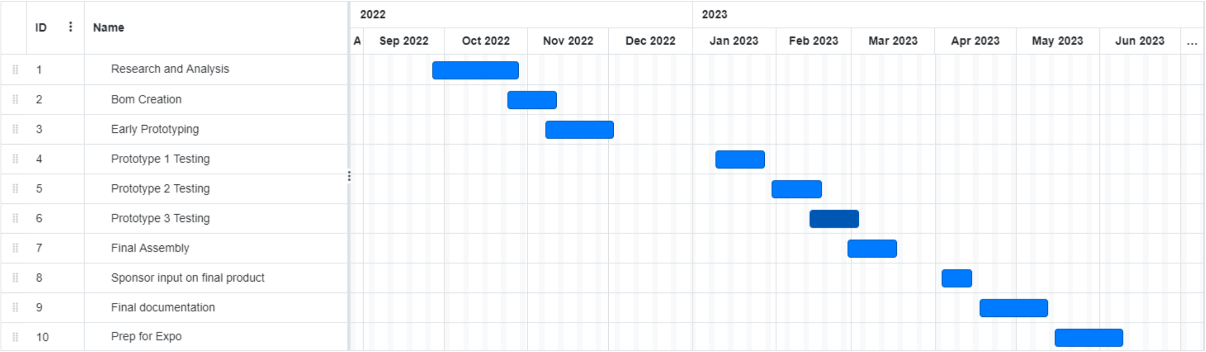
Electronic safe locks are becoming more convenient when compared to mechanical locks but many designs suffer from vulnerabilities that make them prone to timing side-channel analysis, voltage glitching, and lockout-defeat strategies. Since electronic locks are widely used to store valuables, their security is the most important feature. This project looks to take one lock and use the methods stated previously to expose vulnerabilities in the lock to allow manufacturers to make design improvements to improve the security of their locks.

This project will fulfill the need to better characterize lock vulnerabilities based on a manufacturer's particular architecture. It also aims to automate the electronic lock-picking process to better show the repeatability of the vulnerabilities.

When meeting with the project partner, and conducting research through professional communities such as the DEFCON Conference, the needs that are fulfilled by the project are to improve the security of electronic locks through analysis, reverse engineering of lock architecture, and electronic attacks to expose the vulnerabilities of the locks which is relayed back to the manufacturer so that changes can be made.

The end user of electronic locks is the everyday person. Many people use electronic locks to store valuables like jewelry, money, and documents such as passports, and birth certificates. These end users rely on their safe to protect their valuables in the event of burglary or theft so the more secure the safe, the better an experience for that end user. Outside of the end users, the lock manufacturers are significant stakeholders as any improvements on existing electronic lock models will result in a higher quality product.

1.4 Proposed Timeline



ID	Name
1	Research and Analysis
2	Bom Creation
3	Early Prototyping
4	Prototype 1 Testing
5	Prototype 2 Testing
6	Prototype 3 Testing
7	Final Assembly
8	Sponsor input on final product
9	Final documentation
10	Prep for Expo

Figure 1.1: Proposed Timeline

1.5 References and File Links

1.5.1 References

1. DEFCONConference, “DEF CON 24 - plore - side channel attacks on High Security Electronic Safe Locks,” YouTube, 10-Nov-2016. [Online]. Available: https://www.youtube.com/watch?v=lXFpCV646E0. [Accessed: 16-Nov-2022].

2. “EECS Project Portal,” EECS Project Submission Form | OSU. [Online]. Available: https://eecs.oregonstate.edu/capstone/submission/pages/. [Accessed: 16-Nov-2022].

1.5.2 File Links

1. https://drive.google.com/drive/folders/1tpi62i4hFNSUOOfPLSZll8ekPSLUHxg-?usp=sharing

1.6 Revision Table

Date	Name and Update
10/10/2022	Connor Owen: Initial content for team contact information and protocols, and initial content for the executive summary.
10/10/2022	Russell Owen: Added gap analysis section
10/10/22	Ryan Ostroff: Created online gantt chart
10/31/22	Connor Owen: Updated executive summary with additional information
10/31/22	Neal Gardella: Updated file link to Google drive
11/16/22	Connor Owen: Updated references to be in IEEE format.

Table 1.4: Revision Table 1

2 Impacts and Risks

2.1 Design Impact Statement

The modern world, technology has a great impact on everyone's lives and the security of each person is of utmost importance. Electronic devices give this security but have their own vulnerabilities and impacts on not only the environment but our social landscape as a whole. According to [3] the widespread implementation of electronic keys and locks has improved the safety and quality of life for students at an OK state college.

Smart locks are rising in popularity as smart devices become more commonly used and accessible to the average person, causing smart locks to become more ingrained in modern culture. This culture has been termed "ubiquitous culture" [4]. Since this ubiquitous culture is becoming more prevalent due to the use of smart locks, the vulnerabilities of those devices become vulnerabilities in the network, creating an impact on modern culture and by extension, society as a whole.

Technology is moving at a rapid pace and the security measures in smart locks are lagging behind. This allows attackers to gain access to a consumer's home and all smart home devices through inherent vulnerabilities in the hardware. As demonstrated in [5], there are security vulnerabilities in the cutting-edge smart locks by Master Lock, the worst of which allows an attacker to gain management control of the lock. This may be addressed by starting a dialogue with the company to patch the more severe vulnerabilities so that the end consumer will feel safer in a culture driven by technology.

The economic trends show that more and more individuals, be they private users or large companies, are going towards electronic security in order to keep themselves and their own equity safe. Additionally, it has been projected through the 2022-2029 years that there will be a higher demand for more secure electronic security devices [6] This leaves those with funds able to keep themselves secure and those without prime targets for thieves and robbers. This can be addressed by providing extra security measures at a reasonable cost or in the prevention of thieving through extra policing methods. Another way this could be worked around is a make sure that the devices could be loaned to people if/when they need them at very discounted prices

The impact that our project will have on the environment comes down to the e-waste we are creating from our design. Not all e-waste can be recycled leading to disposal into landfills where toxins from the heavy metals can be absorbed into the soil, harming the surrounding environment and its ecosystem [7]. The use of leaded solder in our project as well will have an impact on the environment, so we will have a standard in place to ensure no area is left contaminated. Also, creating a design that only uses electronic parts that we can acquire on OSU's campus without the need to have a shipment of electronic parts delivered so that we never created a notable amount of carbon emissions

The potential to research and develop an attack that works reliably on every lock of the same model would have an extremely detrimental impact on the company that sells the lock as well as overall eroding consumer faith in electronic locks. One potential solution to this could be to implement some element of randomness or obfuscation within the lock's electronic timings. Introducing randomness would prevent a one size fits all hack to all locks of the same model.

2.2 Risks

ID	Description	Category	Probability	Risk Impact	Performance Indicator	Action Plan
1	Logic Analyzer Overvolts	Technical	Low	High	Logic Analyzer stops working	Alert project partner, prevent future overvolting, and seek replacement hardware
2	Working with lead solder	Safety	High	High	Solder is marked as leaded	Wash hands after soldering and do not eat or drink near station
3	Locking ourselves out of lock	Technical	Medium	Medium	Lock doesn't respond to set pin	Try reset code to open the lock, and guess code based on previous inputs
4	Note upkeep not met	Organizational	Low	Medium	Notes are not uploaded to Drive	Set a requirement to upload notes every meeting and reach out to member for their notes.
5	Release of vulnerabilities	Safety/Public Health	Low	High	Project partner is informed of vulnerabilities	To discuss the legal liability of vulnerability with project partner and proceed with student wellbeing in mind
6	AMS Lab Closed	Organizational	Low	High	Keycard access denied	Move to Dearborn lab and use oscilloscope for analysis
7	Desired components not available	Technical	Medium	Medium	Unable to progress with timing analysis	Select alternative component that can achieve similar outcome
8	ESD Damage	Technical	Low	High	Electronic parts (microcontroller) stop working	When using breadboard and prototyping we will use ESD hand straps to help protect against damage

Table 1.5: Risk Identification Table

2.3 References and File Links

2.3.1 References

1. DEFCONConference, "DEF CON 24 - plore - side channel attacks on High Security Electronic Safe Locks," YouTube, 10-Nov-2016. [Online]. Available: https://www.youtube.com/watch?v=lXFpCV646E0. [Accessed: 16-Nov-2022].

2. "EECS Project Portal," EECS Project Submission Form | OSU. [Online]. Available: https://eecs.oregonstate.edu/capstone/submission/pages/. [Accessed: 16-Nov-2022].

3. E. Knight, S. Lord and B. Arief, "Lock Picking in the Era of Internet of Things," in 2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), 2019, pp. 835-842, doi: 10.1109/TrustCom/BigDataSE.2019.00121.

4. R. Kango, P. R. Moore and J. Pu, "Networked smart home appliances - enabling real ubiquitous culture," Proceedings 3rd IEEE International Workshop on System-on-Chip for Real-Time Applications, 2002, pp. 76-80, doi: 10.1109/IWNA.2002.1241340.

5. Author(s) Dale Mathias. (n.d.). Key ingredient: Electronic keys and locks have improved safety and quality of life at one Oklahoma college. Key Ingredient: Electronic Keys and

Locks Have Improved Safety and Quality of Life at One Oklahoma College | Office of Justice Programs. Retrieved November 2, 2022, from https://www.ojp.gov/ncjrs/virtual-library/abstracts/key-ingredient-electronic-keys-and-locks-have-improved-safety-and

6. “Electronic security market: Global industry analysis and forecast (2022-2029),” MAXIMIZE MARKET RESEARCH, 11-Aug-2022. [Online]. Available: https://www.maximizemarketresearch.com/mark report/global-electronic-security-market/31191/. [Accessed: 31-Oct-2022].

7. “Waste its negative effects on the environment,” E. [Online]. Available: https://elytus.com/blog/e-waste-and-its-negative-effects-on-the-environment.html. [Accessed: 02-Nov-2022].

2.3.2 File Links

1. https://drive.google.com/drive/folders/1tpi62i4hFNSUOOfPLSZIl8ekPSLUHxg-?usp=sharing

2.4 Revision Table

Date	Name and Update
10/31/2022	Russell Owen: Added initial content for risk table.
10/31/2022	Connor Owen: Added risks 5 and 6.
11/16/22	Connor Owen: Updated references to be in IEEE format.
11/16/22	Neal Gardella: Added risk #7 to Table. 1.5
11/16/22	Ryan Ostroff: Added risk #8 to Table. 1.5.
4/27/23	Ryan Ostroff: added 2.1 section
5/14/23	Ryan Ostroff: Edits to grammar in section 2.1
5/14/23	Ryan Ostroff: Fixed formatting in the References and File links sections

Table 1.6: Revision Table 2

3 Top Level Architecture

3.1 Block Diagram

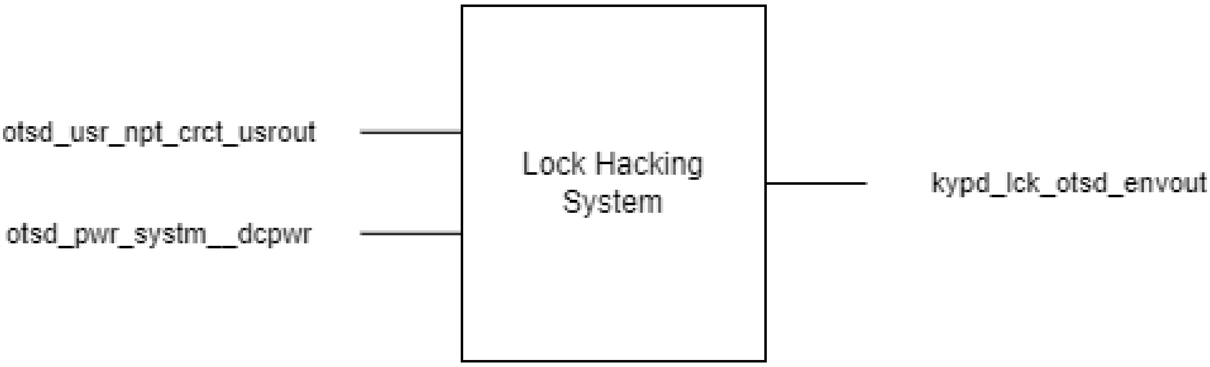


Figure 1.2: Black Box Diagram

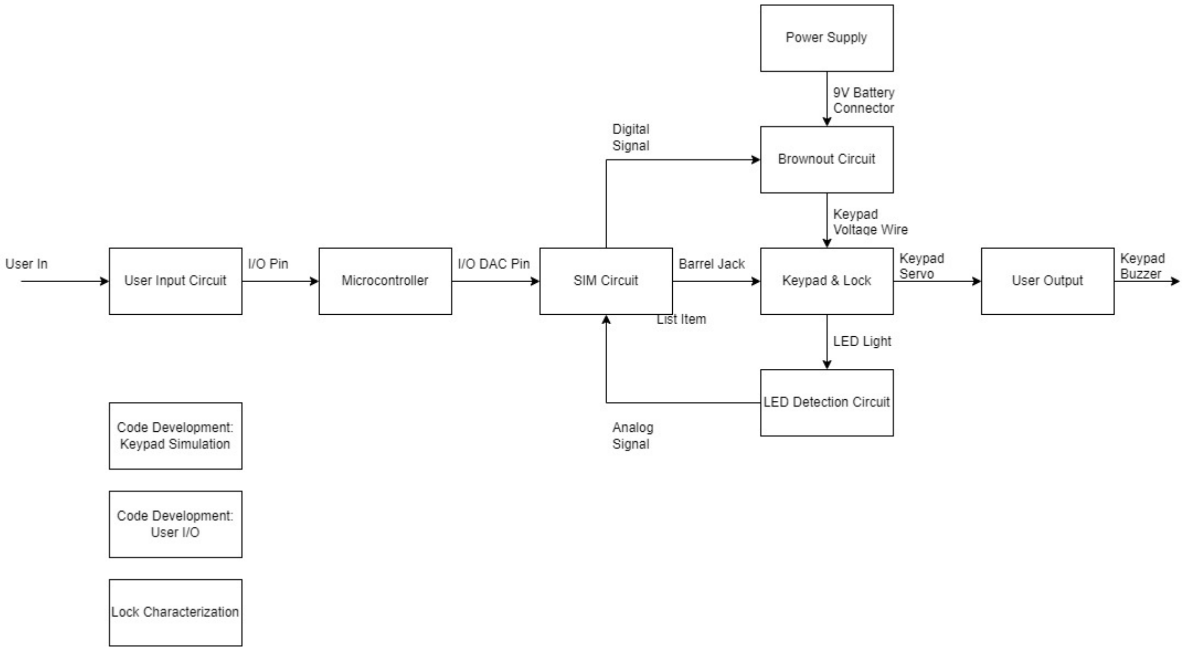


Figure 1.3: Top Level Diagram

3.2 Block Descriptions

3.2.1 LED Detection Circuit

A red light photo-sensitive integrated circuit will be affixed to the keypad enclosure. The LED turns on during user input to the keypad and its response varies depending on if the submitted pin is correct or incorrect. The circuit will measure the pulse width of the LED response.

3.2.2 Lock Characterization

This block is a research-based timing analysis method that will explicitly define the average timing difference (pulse width) between correct and incorrect user inputs on the LAGARD39E electronic safe lock using a Saleae USB logic analyzer and corresponding software. The analysis is conducted using a lock that we know the correct passcode to open. Having access to a lock with a known passcode is essential for observing, quantifying, and exploiting the behavior of the lock. This block is not hardware or software and therefore will not directly interact with other blocks within the system. The data obtained from this analysis will be utilized in the code that analyzes the output from the LED detection circuit.

3.2.3 Brown-Out Circuit

The brown-out circuit is responsible for supplying power to the external lock. The brown-out circuit is an intermediary block that is connected between the microcontroller and the lock. Its purpose is to allow the microcontroller to directly control the power supplied to the lock. The microcontroller must be able to turn on and off the power to the lock at any point during the lock opening attempt.

The circuit itself is composed of resistors and transistors. Two digital signals from the microcontroller are inputs to the brown-out circuit. One input control the power supplied to the lock itself. Another input discharges the lock's power supply quickly to remove any residual charge and ensure a fast turn-off.

3.2.4 Power System

A USB connection from an outside power outlet/Plug/Computer will power the microcontroller and supply 9V to the brownout circuit. A DC-DC boost converter is used to step up 5V input to 9V. The power system is responsible for supplying all the electronic parts within the system with enough power so that opening the external lock does not turn off our device.

3.2.5 Code Development I/O

This code will take feedback from the I/O circuit such as a start or stop and respond by starting the hacking process or resetting it after. The code should be able to receive the correct combination and give this to the output interface such that the user can see the combination.

The code will be written using the Arduino IDE which uses C++, therefore the code will be written using C++.

3.2.6 Microcontroller

This extra block is needed to connect our system and is used to run code for the system while delivering IO. The microcontroller we are using is Teensy 4.1.

3.2.7 User Input Circuit

The user input circuit will take user input through various means such as a touch screen that will then start the hacking process on the lock. The user interface will display the current progress of the hack. After the hack is complete, it will notify the user and display the correct keypad combination such that the lock can be opened using the keypad and not the hacking device. The input should then be able to reset after it is complete and go back to the start menu. The reason for choosing this block is to meet two separate requirements, one is for the user to be able to give inputs to the device in order to start the hack and two is to be able to see the pin code after the lock has been hacked. A touch screen is perfect for this because it can take user input and give updates from the system in the same block.

3.2.8 Code Development: Keypad Simulation

This code will simulate the keypad presses by reading feedback from the LED detection circuit and delivering input signals to the keypad SIM circuit through the microcontroller. It will also interface with the touch screen to begin and repeat the hacking process. This will be a looping algorithm written in Arduino.

3.2.9 Keypad SIM Circuit

This block is a circuit that receives input from the Teensy4.1 microcontroller corresponding to a desired keypress. This will cause the overall circuit to short a particular resistance to ground, simulating what a manual keypress would look like to the LaGard 39E lock. This will be connected to the lock through its barrel jack port to deliver instructions similar to a locksmith debug device. The purpose of this block is to simulate an analog keypress in a digital and automated form to speed up the process of hacking the lock. In this way, the user can run code to automatically hack the lock faster than a human could deliver inputs. This block satisfies the overall system requirement of automating the hacking process requested by the project sponsor.

3.2.10 Keypad Lock

This is the existing LaGarde39E Lock system that we are picking. This block is not designed by any team member but is used for analysis and interfacing with.

3.3 Interface Definitions

Name	Properties
otsd_usr_npt_crct_usrin	**Other: User can restart system using touch.** **Other: User can touch using finger.** **Other: User can use a stylus to touch.**
otsd_pwr_systm__dcpwr	**Other: Imax >= 1A** **Vmax: 5V** **Vmin: 0V**
otsd_lck_chrctrztn_usrin	**Other: Incorrect pin entry** **Other: Correct pin entry** **Timing: Fastest humanly possible input**
otsd_ld_dtctn_crct_envin	**Light: the response time will be less than 10ms** **Light: the circuit will respond to light** **Other: the photodiode will sit flush against the light source**
cd_dvlpmnt__usr_npt_crct_data	**Messages: Sends code information such as graphics and text to circuit.** **Other: Establishes SPI connection to touch screen.** **Protocol: SPI**
usr_npt_crct_cd_dvlpmnt__data	**Messages: Receives touch input from touch screen and responds to it.** **Other: Acknowledges SPI connection from code.** **Protocol: SPI**
usr_npt_crct_mcrcntrllr_data	**Messages: The touchscreen will be the user interface and will display messages such as: "Touch to begin", "Hacking in progress", "Hack complete", "Current pin: ". and "Restart?"** **Other: 3.3v logic level** **Protocol: SPI**
kypd_sm_crct_kypd_lck_dsig	**Other: Digital Potentiometer must be able to change resistance to within 5% of the measured resistances within the keypad** **Other: System needs to deliver input into the keypad through debug port (Barrel Jack Plug))** **Rise Time: Needs to be faster than 1ms to simulate the keypad mechanical rise time**
cd_dvlpmnt_kypd_smltn_mcrcntrllr_data	**Datarate: Code will send 7 digits in sequential order to correspond to a manual keypad input** **Other: Code will display the correct keypad combination during algorithm runtime** **Other: Code will send digital output to the keypad simulation circuit to send a keypad input**
pwr_systm__brwn-ot_crct__dcpwr	**Inominal: 10uA** **Ipeak: 460mA** **Vmax: 9V** **Vmin: 0V**
pwr_systm__mcrcntrllr_dcpwr	**Inominal: 80mA** **Ipeak: 126mA** **Vmax: 5V** **Vmin: 0V**
lck_chrctrztn_otsd_other	**Other: The LEDs minimum response time to the fastest input possible is 223.xms** **Other: The lock's response to a previous incorrect input is 82.xms** **Other: The response time after a correct pin entry is 162.xms**
ld_dtctn_crct_mcrcntrllr_comm	**Other: Inominal of 5mA** **Other: Imax of 10mA** **Vmax: max supply voltage of 10V** **Vmin: min supply voltage of 5V**
mcrcntrllr_kypd_sm_crct_usrin	**Other: BJT must operate like a switch when microcontroller digital signal goes high** **Other: Needs to be able to deliver input within 30ms.** **Other: Needs to use one pin for the possible keypad input**
mcrcntrllr_cd_dvlpmnt_kypd_smltn_data	**Other: Code algorithm will discern between a correct keypad input and an incorrect keypad input based on timing differences** **Other: Must receive digital input in the form of 1's and 0's to indicate a high or low signal from the LED detection circuit** **Other: Code algorithm will wait for input after sending a keypad output**
mcrcntrllr_brwn-ot_crct__dsig	**Logic-Level: Active low (second input)** **Logic-Level: Active High (first input)** **Vmax: 3.3**

Table 1.7: Interface Definitions

3.4 References and File Links

3.4.1 References

1. DEFCONConference, “DEF CON 24 - plore - side channel attacks on High Security Electronic Safe Locks,” YouTube, 10-Nov-2016. [Online]. Available: https://www.youtube.com/watch?v=lXFpCV646E0. [Accessed: 16-Nov-2022].

2. “EECS Project Portal,” EECS Project Submission Form | OSU. [Online]. Available: https://eecs.oregonstate.edu/capstone/submission/pages/. [Accessed: 16-Nov-2022].

3.4.2 File Links

1. https://drive.google.com/drive/folders/1tpi62i4hFNSUOOfPLSZll8ekPSLUHxg-?usp=sharing

3.5 Revision Table

Date	Name and Update
3/10/2022	Connor Owen: Created Section and Initial Content
4/25/2023	Connor Owen: Updated section content from Student Portal
5/10/2023	Russell Owen: Updated formatting so seciton headers correctly connect to existing tables
5/14/2023	Ryan Ostroff: Added wording to block descriptions to help show that some blocks are there to help connect all blocks

Table 1.8: Revision Table 3

4 Block Validations

4.1 Lock Characterization

Block champion: Neal Gardella

4.1.1 Description

This block is a research based timing analysis method that will explicitly define the average timing difference (pulse width) between correct and incorrect user inputs on the LAGARD39E electronic safe lock using a Saleae USB logic analyzer and corresponding software. The analysis is conducted using a lock that we know the correct passcode to open. Having access to a lock with a known passcode is essential for observing, quantifying, and exploiting the behavior of the lock. This block is not hardware or software and therefore will not directly interact with other blocks within the system. The data obtained from this analysis will be utilized in the code that analyzes the output from the LED detection circuit.

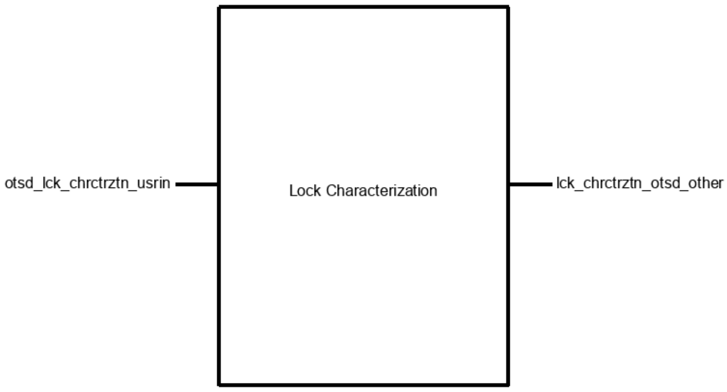


Figure 1.4: Lock characterization block image

4.1.2 Design

This block does not have any interfaces that fit into the mold of this block diagram. The analysis will be conducted using a USB Saleae digital logic analyzer and its accompanying Logic software. The logic analyzer interfaces with the LED and the barrel jack on the keypad. The logic analyzer is connected to the LED on the lock's keypad by opening the enclosure and soldering a lead to the trace connected to the LED. The logic analyzer is connected to the barrel jack by plugging in a barrel jack cord that has been cut and stripped to expose the leads. One lead will connect to V+, the LED and V+ leads will share the common ground on the barrel jack. Each digit on the keypad is associated with a unique resistor value within the keypad's circuit. When the user presses the keypad there will be a voltage drop corresponding to the resistance of the digit that was pressed.The LED blinks once after each user input to the keypad and exhibits a series of blinks after an amount of inputs that match the length of the passcode. For example, a lock with a 7 digit pin will blink after 7 digits have been entered. The delay corresponding to the falling edge of the voltage drop and the rising edge of the LED associated with correct vs incorrect inputs are distinctly different and not perceptible to the human eye. The logic analyzer is used as a visualization and timing analysis tool to support the brute force reverse engineering in order to determine the average timing difference between a correct and incorrect input. The information obtained from this analysis will be implemented within the code that processes the output of the LED detection circuit. The LED detection circuit will be affixed directly to the LED on the face of the keypad and insulated from ambient light. Upon completion of the system the keypad press that normally depends on user input during analysis will be simulated using the keypad SIM circuit. The LED detection circuit will detect the LED during each simulated keypress. The microcontroller will analyze the timing difference between the falling edge of the simulated keypad entry and the rising edge of the LED's response to iterate through every possible keypad combination and determine if each key entry is correct or incorrect. There are 10n possible combinations for an n-length pin.

4.1.3 General Validation

This design specifically meets the needs of this system because the timing analysis is passive and non-invasive[2]. The goal of the system when completed is to be able to successfully open a LAGARD 39E electronic safe lock by only exploiting externally available information. When a lock is affixed to a safe the only user accessible elements are the keypad, LED, and barrel jack. This analysis will not interfere with any of the locks computing or attempt to alter its behavior, instead this will quantify the lock's behavior for future exploitation. Even though the ultimate goal of the system is to be non-invasive the research is easier to conduct when the inside of the keypad is accessible and the LED can be directly probed. The LED behavior can be clearly analyzed using the Saleae digital logic analyzer and its accompanying software purpose-built for accurately analyzing digital and analog signals. This analysis method isolates the information from external variables and expedites the research process by being conducted in parallel with other blocks of the project. The particular lock we are analyzing has a 5 minute lock-out period where no user inputs are possible after 3 consecutive incorrect attempts, and then a 10 minute lock-out period if the next attempt immediately following the 5 minute lock-out is incorrect. The lock will exit lock-out mode and clear the wrong attempt counter after one correct pin entry. Brute forcing the combination is simply not possible.

4.1.4 Interface Validation

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
otsd_lck_chrctrztn_usrin : Input		
Other: Incorrect pin entry	An incorrect input will have an explicit delay between the falling edge of user input and the rising edge of the LEDs response.	The lock and keypad's behavior have been analyzed to determine their response to correct and incorrect inputs
Other: Correct pin entry	A correct input will have an explicit delay between the falling edge of user input and the rising edge of the LEDs response.	The lock and keypad's behavior have been analyzed to determine their response to correct and incorrect inputs
Timing: Fastest humanly possible input	The keypad can be used in quick succession faster than the LED is able to respond.	The LED's minimum response time between user inputs is 223.xms. Regardless of how quickly the keypad was used the LED has an explicit minimum reponse.
lck_chrctrztn_otsd_other : Output		
Other: The LEDs minimum response time to the fastest input possible is 223.xms	The keypad can be used in quick succession faster than the LED is able to respond.	The LED's minimum response time between user inputs is 223.xms. Regardless of how quickly the keypad was used the LED has an explicit minimum reponse.
Other: The lock's response to a previous incorrect input is 82.xms	An incorrect input will have an explicit delay between the falling edge of user input and the rising edge of the LEDs response.	The lock and keypad's behavior have been analyzed to determine their response to correct and incorrect inputs
Other: The response time after a correct pin entry is 162.xms	A correct input will have an explicit delay between the falling edge of user input and the rising edge of the LEDs response.	The lock and keypad's behavior have been analyzed to determine their response to correct and incorrect inputs

Table 1.9: Interface Properties

4.1.5 Verification Process

1. Open the external housing of the keypad to expose the PCB and the LED.
2. Solder a wire to one of the LED terminals for probing with the logic analyzer.
3. Obtain a barrel jack connector cable. Plug the barrel jack into the keypad, cut the other end of the wire and strip the insulation to expose the individual V+ and GND strands contained inside.
4. Connect the logic analyzer probe to the wire soldered to the LED.
5. Connect the logic analyzer probe to the V+ terminal of the barrel jack cable.

6. Connect the logic analyzer GND associated with the LED and V+ channels to the common GND of the barrel jack cable. Ensure a stable GND connection for a clean signal.
7. Connect the battery to a multimeter to verify the battery is fresh (9V) the lock will not function properly if the battery is below 8.5V.
8. Connect the Saleae logic analyzer to your computer, install and launch the accompanying Logic software.
9. Within the software, enable the channel connected to the LED as digital and the channel connected to the barrel jack V+ as analog. Select the highest sampling rate for both channels and make sure all other channels are disabled. There should only be two active channels.
10. Start recording within the Logic software and enter a single combination to the keypad, starting with 1-1-1-1-1-1. Stop recording after the lock has finished its incorrect entry response.
11. Observe the voltage drop on the analog channel when the keypad is pressed and the LED's response on the digital channel.
12. Using the Measure tab, place a pair between the falling edge of the analog channel and the rising edge of the digital channel to quantify the delay and record the result. The pair marker automatically snaps to edges on digital channels but you will need to zoom in a lot for precise placement on the analog channel.
13. Begin iterating through different combinations, 2-1-1-1-1-1, 3-1-1-1-1-1. . . .etc. and make the same measurements using the marker pairs.
14. ALWAYS enter the correct passcode after two consecutive incorrect attempts or the lock will enter lock-out mode after the third incorrect attempt and you will have to wait 5 minutes until the keypad is usable again. There is currently no workaround.
15. Collect several measurements of the same passcode entry to obtain an average value, don't rely on the first measurement as absolute.
16. Collect several measurements of the correct passcode entry to obtain an average value when every keypad entry is correct.

4.1.6 References and File Links

1. DEFCONConference, “DEF CON 24 - plore - side channel attacks on High Security Electronic Safe Locks,” YouTube, 10-Nov-2016. [Online]. Available: [Accessed: 08-Feb-2023].

2. F.-X. Standaert, “Introduction to side-channel attacks,” Integrated Circuits and Systems, pp. 27–42, 2009.

4.1.7 Revision Table

1/20/23	Neal Gardella: initial document creation
2/8/23	Neal Gardella: Clarified Description section and added more detail to Design section per comments Updated figures More explanation in General Validation per comments Updated interface definitions table and added caption per comments Updated Verification Plan and added figure with caption Fixed references for IEEE formatting per comments

Table 1.10: Revision Table

4.2 LED Detection Circuit

4.2.1 Description

A red light photo sensitive integrated circuit will be affixed to the keypad enclosure. The LED turns on during user input to the keypad and its response varies depending if the submitted pin is correct or incorrect. The circuit will measure the pulse width of the LED response.

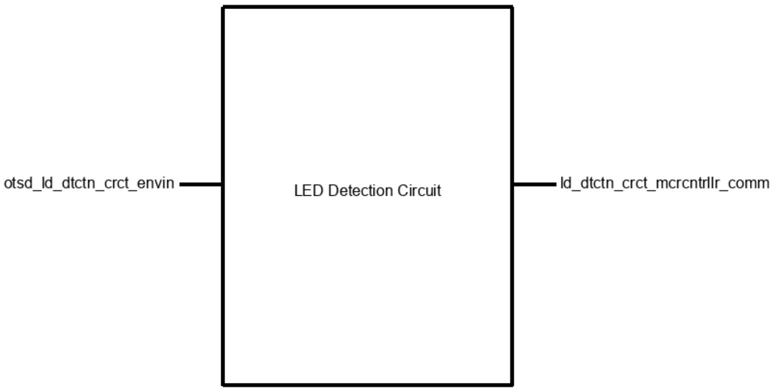


Figure 1.5: LED block diagram

4.2.2 Design

The LED detection circuit is a transimpedance amplifier that consists of a photodiode connected to the inverting and non-inverting inputs of an op-amp and a resistor connected the non-inverting input and Vout of the op-amp.

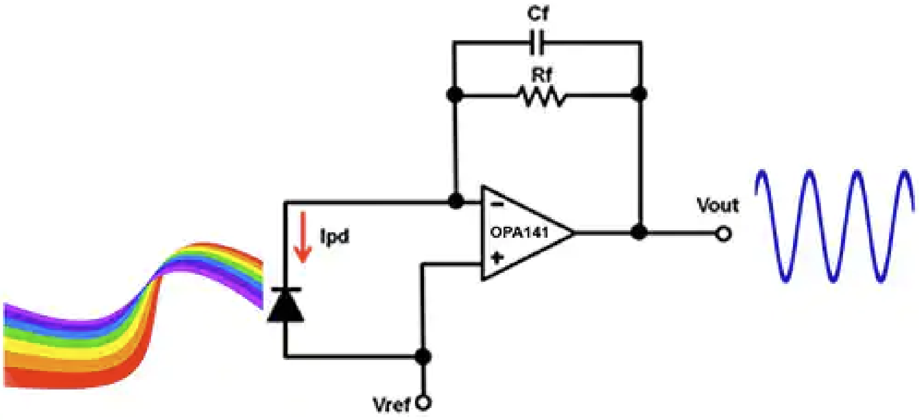


Figure 1.6: LED block diagram

4.2.3 General Validation

Transimpedance amplifiers (TIAs) are commonly employed to convert the current output of sensors, such as photodiodes, into an output voltage. The circuit itself is quite simple and widely used, consisting of only a few components. Voltage is ubiquitous and easy to read for most devices.

4.2.4 Interface Validation

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
otsd_ld_dtctn_crct_envin : Input		
Light: the response time will be less than 10ms	The circuit must respond quickly enough to the LED for the microcontroller the process the falling edge	The microcontroller can read and respond to signals within this amount of time
Light: the circuit will respond to light		
Other: the photodiode will sit flush against the light source	The photodiode is very sensitive and insulation from ambient light is essential for accurate and consistent performance	The diode will be sealed off from outside light and thus the output of the circuit will not be influenced by light other than the LED
ld_dtctn_crct_mcrcntrllr_comm : Output		
Other: Inominal of 5mA	This is the operating current of the op-amp as listed in the data sheet	The operating parameters of the components are specified within the data sheet. Unless faulty components will always work within their operating range
Other: Imax of 10mA	This is the max current of the op-amp as listed in the data sheet	The operating parameters of the components are specified within the data sheet. Unless faulty components will always work within their operating range
Vmax: max supply voltage of 10V	This is the max supply voltage of the op-amp as listed in the data sheet	The operating parameters of the components are specified within the data sheet. Unless faulty components will always work within their operating range
Vmin: min supply voltage of 5V	This is the minimum supply voltage of the op-amp as listed in the data sheet	The operating parameters of the components are specified within the data sheet. Unless faulty components will always work within their operating range

Table 1.11: Interface Properties

4.2.5 Verification Process

1. Connect the LED detection circuit to 5-10V supply.
2. Connect a multimeter to the output of the op-amp and set it to DC voltage mode.

3. Observe the voltage output while the photodiode is completely insulated from ambient light, when the photodiode is exposed to ambient light, and the photodiode is deliberately exposed to light other than ambient light (flashlight, LED).

4.2.6 References and File Links

1. PDF, D. (n.d.). Stabilize your transimpedance amplifier. Stabilize Your Transimpedance Amplifier | Analog Devices. Retrieved March 12, 2023,

4.2.7 Revision Table

3/10/23	Neal Gardella: Created section and initial content

Table 1.12: Revision Table

4.3 Keypad SIM Circuit

4.3.1 Description

This block is a circuit that receives input from the Teensy4.1 microcontroller corresponding to a desired key press. This will cause the overall circuit to short a particular resistance to ground, simulating what a manual key press would look like to the LaGard 39E lock. This will be connected to the lock through its barrel jack port to deliver instructions similar to a locksmith debug device. The purpose of this block is to simulate an analog key press in a digital and automated form to speed up the process of hacking the lock. In this way, the user can run code to automatically hack the lock faster than a human could deliver inputs. This block satisfies the overall system requirement of automating the hacking process requested by the project sponsor.

4.3.2 Design

The design section includes the overall black box diagram as well as the working schematic for the keypad SIM circuit. From a high level, the keypad SIM circuit will simulate the behavior of the LaGard 39E keypad to automate keypad input. This circuit will be receiving power from the Teensy4.1 microcontroller and will be plugged into the LaGard 39E lock through its barrel jack debug port. This barrel jack will be soldered to the circuit which can be seen in the KiCad schematic.

Currently, the working design is using a boost converter to boost 3.3v to 5v for switching given a 3.3v input. A 330 ohm resistor is used to bias the transistor as a switch which will short the digiPOT to ground. This will simulate the behavior of the lock when a keypad is pressed.

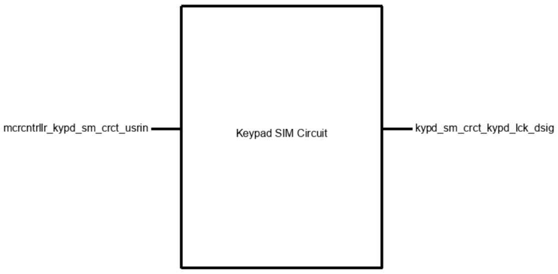


Figure 1.7: Sim Circuit Block Diagram

Instead of using a resistor ladder like the LaGard 39E, a digital potentiometer is used so that the microcontroller can write I2C instructions to change the resistance based on the corresponding keypad input to be simulated.

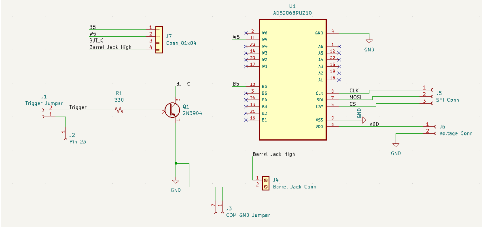


Figure 1.8: Sim Circuit Schematic

To give more context to the design, below in Figure 1.9 is a picture of the LaGard 39E. In the right side of the picture, a small hole can be seen. This is a barrel jack plug which is used by locksmiths to open up a lock whose passcode has been locked, or run debug diagnostics. This is the port that the Keypad SIM Circuit will plug into to hack into the keypad in a very similar way to how locksmiths would open this lock.

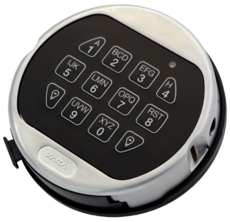


Figure 1.9: LaGard 39E Keypad With Debug Port

4.3.3 General Validation

The design on this block was chosen for the simple functionality of shorting nodes to ground in a power efficient way. This was necessary because the Teensy 4.1 only operates with 3.3v logic which introduces issues when most designs for similar functionality use 5v logic. Originally, level shifters were going to be used but the use of a boost converter means that a 5v source does not need to be provided. The 330 ohm resistors with a 5v level on the analog pins will allow for the transistor to enter saturation and act like a switch. The digital potentiometer was used to reduce the amount of pins needed, down from 10 dedicated pins to only 3. These resistance values are the same ones used within the actual keypad that have been measured. This complete design will allow the user to run the routine and simulate keypad pressing in an accurate manner. The main research that was done for this block comes from information gathered from experimenting with the LaGard 39E and taking measurements. The LaGard 39E is not designed in a way where strict electrical requirements must be met due to the use cases of the lock ranging very drastically.

4.3.4 Interface Validation

The interface validation table below explains the interface values and why the design details for the block will meet the property. The verification plan ties directly to this table but does reword the exact interface property to be more readable without removing the context of the interface property.

Interface Property	**Why is this interface this value?**	**Why do you know that your design details for this block above meet or exceed each property?**
kypd_sm_crct_kypd_lck_dsig : Output		
Digital Potentiometer must be able to change resistance to within 5% of the measured resistances within the keypad	This interface is this value because the lock has certain resistor values that have a wide variance. As long as the resistance value is close enough, the lock will recognize the input.	Per measurements conducted when opening the keypad of the lock, resistance values of up to 5% will be read as a correct input.
Other: System needs to deliver input into keypad through debug port (Barrel Jack Plug)	To avoid having to open the lock which can cause damage, there is a debug port which uses a barrel jack plug to interface. This is more seamless and does not damage the lock.	This design utilizes a barrel jack which soldiers directly to the SIM circuit.
Rise Time: Needs to be faster than 1ms to simulate the keypad mechanical rise time	The physical keypad has a rise time of its input signals of roughly 1ms. This behavior must be simulated when performing the automated input from the microcontroller.	This design will meet this property because the resistance will be set before the keypad input is set, and the rise time of a 2N3904 is rated at 35ns
mcrcntrllr_kypd_sm_crct_usrin : Input		
Other: Needs to be able to deliver input within 30ms.	When manually using inputs to cycle the lock, the average time to click a keypad button was around 30 ms since this was the time it took a human to press the keypad.	This circuit uses very fast switching BJTs at 35ns with a delay time of 35ns. This is much faster than 30ms. The microcontroller is also rated for 600MHz when performing its operations. This combination should be sufficiently fast.
Other: BJT must operate like a switch when the microcontroller signal goes high	This interface is needed because the BJT is functioning as switches which ground the resistor ladder based on whichever keypad input was requested.	The Teenzy 4.1 microcontroller outputs 3.3v when a pin is set to high. Using the 330 Ohm resistors will result in 15mA in the base of the 2N3904. The minimum rated current for base current is 1mA in saturation.
Other: Needs to use one pin per possible keypad input	This interface is this way because each keypad input has a specific resistance associated with it to communicate to the lock. The zero digit is unique and does not have a resistor which is not reflected.	The Teensy 4.1 microcontroller has enough analog pins to have enough outputs. These pins are all connected to the unique resistors required to cause input.

Table 1.13: Interface Validation

4.3.5 Verification Process

The verification plan for this block is done mostly by observation. This is because the strict electrical requirements of this circuit are not important to its functionality due to the LaGard 39E lock having very wide margins for its functionality. An example of this is the resistance of the digipot having a generous boundary of operation for functional success.

Micro-controller Input:

One pin per possible keypad input:

1. This can be verified by making sure the schematic has sufficient inputs labeled for each possible keypad input.

BJT must operate like a switch:

1. Measure initial voltage on both sides of the resistor (Should be 5v due to lock battery)
2. Turn on microcontroller
3. Run simulation code
4. Measure voltage on both sides of the resistor (Should see a voltage drop drop of 1.8v for a keypad "5" input)

Needs to deliver input within 30ms:

1. Attach logic analyzer probes to each side of the circuit
2. Set logic analyzer to trigger on rising edge
3. Run simulation code
4. Measure time between two signals on logic analyzer (Expecting to see <5ms)

Keypad SIM Circuit Output:

Digital Potentiometer must be able to change resistance to within 10% of the measured resistances within the keypad:

1. Attach multimeter probes to the RW pin and RL pin.
2. Run Simulation code for setting wiper resistance.
3. Change code to set each corresponding keypad input resistance value.
4. Compare output resistance values versus the actual resistance values expected.

System needs to deliver input into keypad through Barrel Jack:

1. Lock needs to be opened.
2. Attach logic analyzer to internal green wire trace (Known internal lock signal pathway).
3. Run simulation code.
4. Use the logic analyzer to see that the intended signal was delivered to the green wire trace.

Needs to have at least 1ms rise time:

1. Attach logic analyzer or oscilloscope probes to digital output pins.
2. Run simulation code.
3. Measure signal rise time to required voltage (1.8v for a keypad “5” input).

4.3.6 References and File Links

[1] “2N3906 general purpose transistors - onsemi.”
https://www.onsemi.com/pdf/datasheet/2n3906-d.pdf. [Accessed: 20-Jan-2023].
[2] “La Gard Electronic,” Kaba Electronic Combination Safe Locks - Safe Lock ComboGard Pro.
https://www.dormakaba.com/us-en/solutions/products/safe-locks/la-gard-electronic/la-gard-combogard-pro-293216. [Accessed: 10-Feb-2023].
[3] ”Teensy® 4.1 Development Board,” PJRC.https://www.pjrc.com/store/teensy41.html.
[Accessed: 19-Jan-2023].

4.3.7 Revision Table

Date	Section	Action Taken
1/18/23	Document Created	Wrote in initial information
1/19/23	Interface Validation	Updated interface validation information to be more precise and in-line with datasheets
2/10/23	Design, General Validation, Interface Validation, Verification Plan, Revision Statement	Updated design of circuit to be more optimized. This has affected the following sections to make sure information is up to date.
2/10/23	Description, Interface Validation, and Verification Plan	Updated description based on Rachaels feedback for tying the description more to the purpose and overall system requirements.
2/10/23	References and file links	Added an IEEE citation to the LaGard 39E manual based on feedback from Joseph Borisch
2/10/23	Description	Added more specific outcome of the block based on feedback from Chris Viray
2/10/23	Design	Added a picture of the LaGard 39E to give more context to the reader about the design of the lock. Following feedback from Chris Viray
2/10/23	References and file links	Updated the file links to better fit with IEEE citation standards

Table 1.14: Revision Table

4.4 Keypad SIM Code

4.4.1 Description

The Keypad SIM Code is the routine that receives input from the micro-controller and will run the hacking process on the lock. This routine will tell the micro-controller which smaller blocks require outputs by setting certain pins to low or high. It is also responsible for running the SPI interface connection to the digital potentiometer as well as the touch screen.

4.4.2 Design

The Keypad SIM code has many different sub routines build inside of it but is designed to be multiple for loops that will iterate through the different keypad combinations. It will start with a base case, "1000000", and the code will poll the LED circuit and look for a specific timing discrepancy. The code will continue to increment the most significant digit until the timing discrepancy is found. Once it is, the code will then move to the next digit and repeat the process. Every five writes to the keypad, the brown-out circuit will be activated to remove the memory of previous keypad inputs from the lock. This is done so the circuit can have an unlimited number of inputs without the lockout procedure being triggered from too many incorrect inputs.

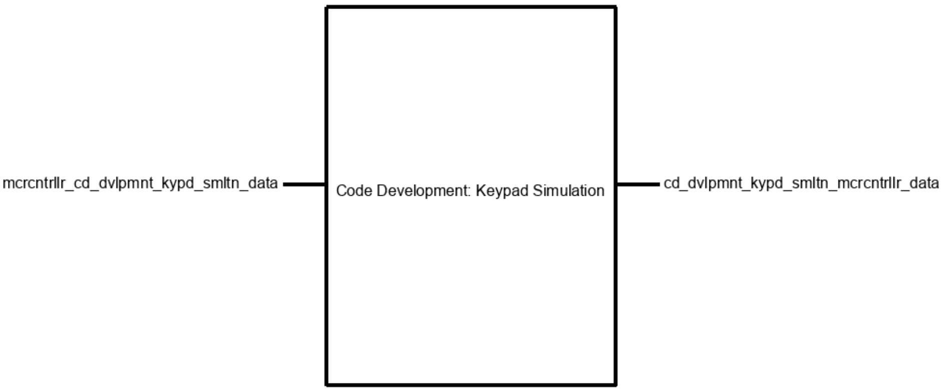

mcrcntrllr_cd_dvlpmnt_kypd_smltn_data
Code Development: Keypad Simulation
cd_dvlpmnt_kypd_smltn_mcrcntrllr_data

Figure 1.10: Keypad SIM Code Block Diagram

4.5 User Input Circuit

4.5.1 Description

Block Champion: Connor Owen
The user input circuit will take user input through various means such as a touch screen that will then start the hacking process on the lock. The user interface will display the current progress of the hack. After the hack is complete, it will notify the user and display the correct keypad combination such that the lock can be opened using the keypad and not the hacking device. The input should then be able to reset after it is complete and go back to the start menu. The reason for choosing this block is to meet two separate requirements, one is for the user to be able to give inputs to the device in order to start the hack and two is to be able to see the pin code after the lock has been hacked. A touch screen is perfect for this because it can take user input and give updates from the system in the same block.

4.5.2 Design

The design section includes schematics that detail exactly how the user input circuit should function as well as the outputs and inputs that will be sent to the system from this block. From a higher level, the user input circuit will contain a touch screen utilizing the ILI0341 controller chip and a Teensy4.1 board from PJRC. A black box image of this design can be seen in Figure **??**.

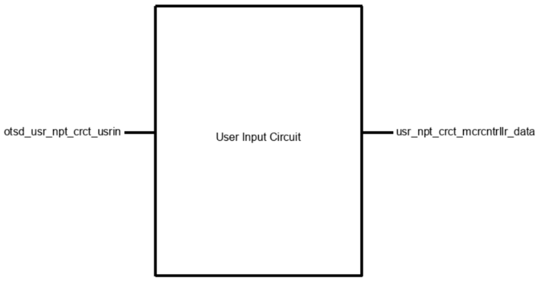


Figure 1.11: User Input Circuit Block Diagram

The usr_npt_crct_mcrcntrllr_data will send and receive inputs that have been given to the user to the microcontroller that controls the entire system. These inputs will be logic level 3.3v on a digital pin that will be outputted by this control circuit and sent to the microcontroller in order to start or stop certain processes. This can be put on any digital pin such as A0-A17 depending on the necessary hardware that will be used besides the touch screen.

The User Input Circuit will rely solely on a simple touch screen circuit that can be seen in Figure 1.13. On the left-hand side is a breakout board that connects all of the SPI communication pins from the microcontroller on the right (U1) to the touch screen through header pins. There are some repeat communication ports that are used which reduces the number of wires from the microcontroller to the actual touchscreen board.

An example of the final assembly can be seen in Figure 1.12 as per PJRC's technical datasheet that includes a Teensy3.1 instead of the Teensy4.1 we are using in this project. The components are rather simple, with only three resistors and a fuse. The fuse helps to regulate the voltage that is currently going into the touchscreen board in order to meet its manufacturing specifications. The resistors regulate the amount of current going into individual pins or onboard functions as specified by the ILI0341 datasheet.

4.5.3 General Validation

The user input circuit is necessary for the system because of the inherent need for user control over the system. The industry standard for electronic safe lock hacking devices is some sort of user input with an automatic readout that lets the user know the pin. This can typically be found in safe buster devices that locksmiths use when opening a safe or lock for someone that has forgotten their pin. Therefore, we need a way to output the correct pin for the user after hacking

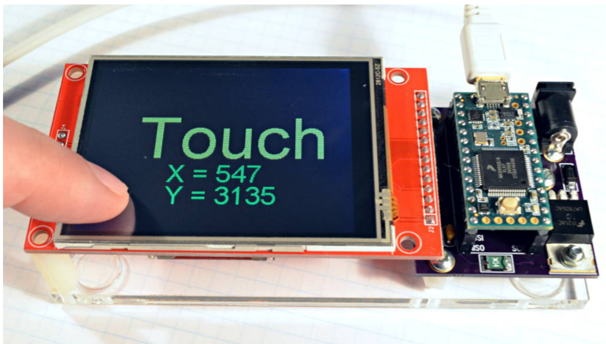


Figure 1.12: PJRC Touch Screen with Teensy 3.1

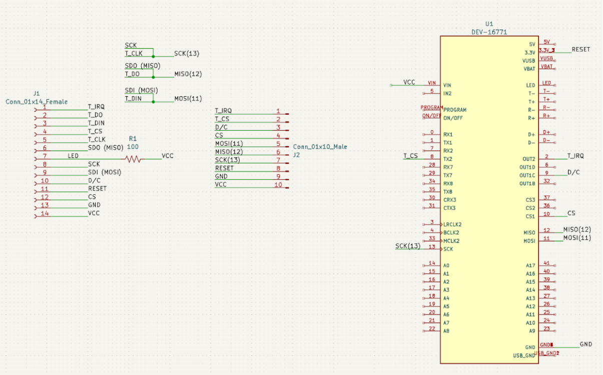


Figure 1.13: User Input Circuit Schematic

the device. Additionally, control over the process such as a start and stop button are needed in order to let the user control how the system operates. It is also necessary to have current status available so the user knows the state that the device is in. These are all necessary items that a display must do and that is why we decided to use the touch screen board. It can display all of these items in a visually pleasing and easy to understand way, especially in a modern world where most interfaces such as cell phones and devices use a touch screen as both a keypad and a display.

The cost of this circuit is also rather small. The only necessary part to get the touchscreen board to interact with the Teensy4.1 is a 100ohm resistor [1]. The parts are going to be at most $1 in total depending on size. The real cost is in the microcontroller and the touchscreen board. The microcontroller is about $30 and the touchscreen board is about $18. For our project we need a microcontroller that is fast enough to emulate analog signals and process digital signals, therefore the Teensy board is a cost that is acceptable for the performance that is necessary for the project. The touch screen board could be replaced with an LCD screen and buttons as those would be cheaper and more cost-effective. The issue would come from a visual and compatibility standpoint. As the touch screen is built for the Teensy line of development boards by PJRC, it is easily set up and configured in order to operate which allows more time for code development and troubleshooting. Time, in this case, is worth more than the cost of the screen as setting up LCD screens and buttons might incur more development costs. This is because we would need to research how the screen interacts with the microcontroller.

The project partner also had requirements for us as far as the microcontroller went. This was set because the electronic lock sends very fast signals and we must have a microcontroller that can read these and process them before the next signal is given to the controller on the lock. The Teensy board is one of the only microcontrollers that can actually do this while having compatibility with the touchscreen, making this user interface block a perfect choice for a soft requirement that was given by the project partner. Additionally, the Teensy4.1 board operates on a 3.3v logic which is perfect for the touchscreen as it also runs on 3.3v logic, allowing for the use of direct connections. This could have been a problem as the lock uses 5v and 9v logic which would require either a voltage regulator or some type of level shifter in order to get the correct voltages to operate the device.

A recommendation from the project partner was that we use a pre-built rail system that could mount both the lock and the device as well for displaying during the engineering expo. This is a rather nice idea as it would allow individuals who visit the option to see everything on display. This rail system is not very big, meaning that if we utilized it the screen would need to be lightweight, small, and also have readable screen. The Teensy4.1 development board is already rather small, so the touch screen that would accompany it would also have to be quite small as well. Other options would have worked just as well, but the compatibility of the already necessary device is too great and is one of the greatest reasons we are using this touchscreen board. The other attributes of the device also help contribute to the choice we have made.

One worry that the group had was the availability of parts. Both the touchscreen and the microcontroller are subject to the chip shortage that is currently going on which could make ordering more parts take weeks if they are out of stock. Current stock checks of PJRC say that there are plenty of touch screens in stock and Teensy4.1 development boards, but there have been times when both are sold out and take months to get back in stock. This could be very bad if it takes a month or two to obtain the device and begin testing. That is why my group has already ordered the components in order to secure them in case they go out of stock. Another worst-case scenario would be using one of the other options such as an LCD screen or Bluetooth device to communicate the user control. This would be unideal but usable as we could dedicate time to researching these components and making use of them within our design. We would lose compatibility with the Teensy board but that is why we chose the touchscreen board. The last worst-case scenario would be if we overvoltage the touchscreen board as if they were out of stock which could result in delays caused by the shipping and manufacturing.

Lastly, as for other parts, there were two separate ideas during brainstorming. One was the LCD screen and button controls that would have been more cost-effective yet had more research time and the other was using Bluetooth or some wave device in order to communicate wirelessly with your phone or computer. This latter device was seen as even more expensive than the touchscreen and prone to research costs that were too large, making it an unavailable option along with the LCD screen and button. Although, it would have been very useful to hack into devices using only your smartphone and not needing an actual user interface on the device.

4.5.4 Interface Validation

The interface validation contains both the input and output interfaces that the device will be using in order to function. The ostd_user_npt_crct_usrin is a user input interface that allows the user to utilize the touchscreen functionality and explains how the interface works. The usr_npt_crct_mcrcntrllr_data interface is an output that communicates with the microcontroller using SPI and uses a 3.3v logic level.

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
ostd_user_npt_crct_usrin: Input		
Type: User can use a finger or stylus to touch the screen.	**The touch screen board only responds to touches that are on the front screen.**	The datasheet on the part says that that is the interface for this device. Additionally, I have tested it myself and it responds to touch with a finger or a stylus.
usr_npt_crct_mcrcntrllr_data : Output		
Messages: The touchscreen will be the user interface and will display messages such as: "Touch to begin", "Hacking in progress", "Hack complete", "Current pin: " and "Restart?"	**This property has these values as these will be the current display values before, during, and after hacking the lock, and are necessary for the user to see in order to properly operate the device.**	The touch screen board has a datasheet that shows all of the different data it can display such as set sentences, emojis, and different characters [1]. This means that all of these messages are well within its capabilities. Additionally there are different GitHub repositories that contain custom libraries with even more messages. There is also a system for creating custom characters on this device.
Other: 3.3v logic level	The touch screen cannot be run on anything larger than 5.5v as a maximum or it will destroy the device.	This is known because the datasheet specifies an operating range from 3v-5v and recommends 3.3v for its operating voltage [1, 2]. Additionally, the Teensy4.1 platform runs on 3.3v logic.
Protocol: SPI	The communication ports that this device uses are the SPI communication ports on the Teensy4.1, therefore it must be using SPI.	They meet this property because using the SPI communication ports I can program the controller.

Table 1.15: Interface Validation

4.5.5 Verification Process

The best way to verify the User Input Circuit is to test the device and use a sample code that would show the use case that the device will be under in the capstone. The following plan goes through the steps that a user would normally be doing when operating the device. The usr_npt_crct_mcrcntrllr_data can be verified by measuring the SPI pins and observing that the pins are 3.3v during communication.

1. Power up the touch screen.
   (a) Does it display the beginning message?
   (b) Does the screen communication pins output 3.3v?
2. Press the touch screen button that says “Touch to begin.”
   (a) Does the button respond to the touch?
   (b) Does a touch anywhere else on the screen start the hack?
   (c) Does the screen correctly display the message?
   (d) Does the screen communication pins output 3.3v?
3. The screen displays “Hacking in Progress.”
   (a) Does it correctly display the message?
   (b) Does the screen display some sort of progress meter?
   (c) Does the screen allow touch during this time?
   (d) Does the screen properly leave this display after it is complete?
   (e) Does the screen communication pins output 3.3v?
4. The screen displays “Hack Complete.”
   (a) Does the screen display the message?
   (b) Does the screen move from this to the next screen in a readable time?
   (c) Does the screen communication pins output 3.3v?
5. The screen displays “Current Pin: “
   (a) Does the screen display the message?
   (b) Does the screen allow the user to continue past this screen?
   (c) Does the screen display the correct pin?
   (d) Does the screen communication pins output 3.3v?

6. The screen displays “Restart?”
   (a) Does the screen display the message properly?
   (b) Does the screen allow the user to continue past this screen?
   (c) Does the screen restart to the beginning message?
   (d) Does a touch anywhere else on the screen restart?
   (e) Does the screen communication pins output 3.3v?

4.5.6 References and File Links

[1] “Color 320x240 TFT touchscreen, ILI9341 controller chip,” PJRC. [Online]. Available: https://www.pjrc.com/store/display_ili9341_touch.html. [Accessed: 19-Jan-2023].
[2] “Ili9341 datasheet - adafruit industries.” [Online]. Available: https://cdn-shop.adafruit.com/datasheets/ ILI9341.pdf. [Accessed: 19-Jan-2023].

4.5.7 Revision Table

Date	**Name**	Section	Action Taken
1/18/23	**Connor Owen**	Document, Description, Design	Created a document and filled out description and design sections.
1/19/23	**Connor Owen**	General Validation, Interface Validation, Verification Plan, References and File Links	Filled out the following sections with information and formatted for IEEE formatting.
2/10/23	**Connor Owen**	Design, General Validation, Interface Validation, Verification Plan, References and File Links	Updates figures in Design to fit updates schematics and reflect changes on capstone website, updated general validation information to reflect up to date schematics, updated interface validation to reflect the student portal, updated verification plan with testing 3.3v logic level, and updated references and file links with citation numbers.
2/10/23	**Connor Owen**	Revision Statement	Added section and filled it out.

Table 1.16: Revision Table

4.6 User I/O Code

4.6.1 Description

Block Champion: Connor Owen
This code will take feedback from the I/O circuit such as a start or stop and respond by starting the hacking process or resetting it after. The code should be able to receive the correct combination and give this to the output interface such that the user can see the combination. The code will be written using the Arduino IDE which uses C++, therefore the code will be written using C++.

4.6.2 Design

The design section includes schematics and pseudo-codes that detail exactly how the user input code should function as well as the inputs and outputs that will be sent to the system from this block. From a higher level, the user input code will contain Arduino code ran on the Arduino

IDE using pre-built libraries provided by the manufacturer of the touch screen and microcontroller. A black box image can be seen in Figure 1.14.

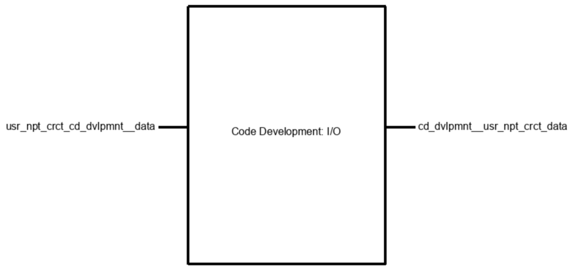


Figure 1.14: User Input Code Block Diagram

The code structure uses three main functions that help to display all the necessary items for the user. The first is a main menu that shows the welcome message and gives instructions as to how to use the device. The code will then wait for a touch input on the hardware which signals that it should continue to the next function. Here it will poll for keypad inputs while the safe-busting circuit tries to find the correct combination. On a successful find of a combination, it will display that on the screen and continue until the entire combination has been found. It will end with a graphic that shows the final code as found by the system and prompts the user to replay the program. There has been graphics built into the code that help give a more flavorful experience to the user and give up-to-date readings on the code the system has currently found.

4.6.3 General Validation

The reason Arduino was chosen for this project is that it is a universal compiler that is used with many different microcontrollers. It also contains a plethora of libraries that must be used in order to sync the hardware with the software. That is why we as a group felt like it was the best method. Another consideration were using a python compiler but the issue was that it did not work well with the Teensy4.1 development board.

4.6.4 Interface Validation

Since this is code there are no interfaces to validate.

4.6.5 Verification Process

If the code compiles and runs on the Arduino, then it has passed the verification process.

4.6.6 References and File Links

References

1. DEFCONConference, “DEF CON 24 - plore - side channel attacks on High Security Electronic Safe Locks,” YouTube, 10-Nov-2016. [Online]. Available: https://www.youtube.com/wa tch?v=lXFpCV646E0. [Accessed: 16-Nov-2022].
2. “EECS Project Portal,” EECS Project Submission Form | OSU. [Online]. Available: https://eecs.oregonstate.edu/capstone/submission/pages/. [Accessed: 16-Nov-2022].

File Links

1. https://drive.google.com/drive/folders/1tpi62i4hFNSUOOfPLSZll8ekPSLUHxg-?usp=sharing

4.6.7 Revision Table

Date	Name and Update
3/10/2023	Connor Owen: Created Section and Initial Content as per document standards.
5/11/2023	Neal Gardella: Minor grammatical fixes

Table 1.17: Revision Table

4.7 Power System

4.7.1 Description

Block Champion: Ryan Ostroff
A USB connection to a power supply will use the input 5V to power the micro-controller and will also create an output 9V to be used by the brownout circuit. The Power system will act as the sink for the USB 2.0 source. There will be 5.0V and 1.5A of power available to the board at all times it is plugged in.

4.7.2 Design

The power system needs to be able to power the Mico-controller and act as a power source for the light detection op-amp and the 9V power line that will power the external lock. With 5V and up to 1.5A as a source from the USB A boost converter that was >90 percent efficient would be able to run at 9V and 300mA along with the controller at 5V and 120mA and op-amp at 5V and

30mA. The boost converter would be a bought block as efficient boost converters with potentiometers that change output voltages are easy to use and very cheap.

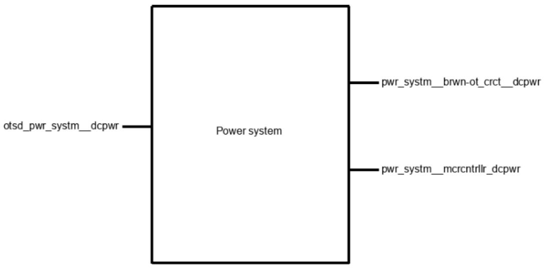


Figure 1.15: Power System Block Diagram

the power system block is very simple as there is only one connection outside of the system and that is the USB connection. This is simplified as only the Vin and Grnd pins on the USB connector need to be used as there is no data stream only power being transferred.

4.7.3 General Validation

The design of this power supply will be very straightforward as it just needs to be able to create a 9V power line from the 5V input (USB). This is being done using a boost converter that can be adjusted using a potentiometer to get the correct 9V output. In order to make sure that the design will work the value of current from the output needs to be tested to ensure that there is enough input current at the correct voltages to power all the devices in the system at the same time. This max current draw is at the time when the lock opens after a correct set of inputs to the correct is sent. In order to understand if the power system will be able to be validated given the set of interfaces it was important to understand how much current is needed and how much is available for use. As the interfaces are the different aspects of the design that need specific amounts of current. Nominal is the normal use age and max is the peak or maximum they will use. The power system must be able to supply peak/max current to all devices if they need as the lock needs be be able to sink a lot of current to open and not cut power to the micro-controller during that time

4.7.4 Interface Validation

Interface Property	**Why is this interface this value?**	**Why do you know that your design details for this block above meet or exceed each property?**
otsd_pwr_systm_dcpwr : Input		
Other: Imax >= 1A	**This is an interface value to ensure that the max current being sunk by the power system is less than supplied**	Using a computer with USB 3.0 + will supply 1A .
Vmax: 5.0V	**This property is to show that the max input voltage to the system needs to be 5V**	The USB protocol asks for power systems to be 5V +-3% [2]
pwr_systm_brwn-ot_crct_dcpwr: Output		
Inominal: 10uA	**Locks nominal current drain**	Measured lock nominal value
Ipeak: 460mA	**The current being outputted will be the current drawn from the brown out circuit**	VP2206N3-G-P003 [1] Vdss: -60 V (drain-source breakdown voltage) Id max (continuous) = -640mA will have 0.74 (W) power dissipation the tested value for max current draw
Vmax: 9V	**The output will be between 0 and 9V going into the brown-out circuit**	VP2206N3-G-P003 [1] Vdss: -60 V (drain-source breakdown voltage) Id < -6 (A) it can supply around -6A on the typical performance curve (the lock will sink a lot less current than this)
Vmin: 0V	**The output will be between 0 and 9V going into the brown-out circuit**	Will be 0V in off condition
pwr_systm_mcrcntrllr_dcpwr: Output		
Inominal: 80mA	**Because it will show the nominal current needed to use the micro-controller**	Teensy 4.0 consumes approximately 100 mA [1]
Ipeak: 126mA	**This shows the max current needed to run the micro-controller**	peak current drain of Teensy 4.1 The value is a tested condition running the code for the brownout circuit
Vnomial: 5V	**This shows the nominal voltage is needed to use the micro-controller**	5V for the input voltage is the value being used from the 5V on the USB
Vmin: 0V	**This is the value that will turn off the micro-controller**	Will be 0V in off condition

Table 1.18: Interface Validation

4.7.5 Verification Process

1. Testing the voltage and current values from the USB This will use an electronic multi-meter and will be used to record the values of the voltage and current coming from the USB

2. Testing the current and voltage values for the micro-controller This is done by measuring the current and voltage after it has been connected to the USB

3. Testing the current and voltage values for the 9V output going to the brown-out circuit This will be done by measuring the current and voltage after the USB is connected to the micro-controller and the USB is connected to the boost converter and the brown-out circuit is under load

4.7.6 References and File Links

References

1. https://www.pjrc.com/store/teensy41.html

2. https://www.ti.com/sc/docs/products/msp/intrface/usb/pwrdist.pdf

3. https://www.olimex.com/Products/Breadboarding/BB-PWR-3608/resources/MT3608.pdf

4.7.7 Revision Table

Date	Name and Update
3/10/2023	Ryan Ostroff: Created draft and Initial Content
3/12/2023	Ryan Ostroff: Added to design section and validation
5/11/2023	Neal Gardella: Updated section and fixed broken links
5/14/2023	Ryan Ostroff: Added new references and created the interface validation table
5/14/2023	Ryan Ostroff: Added edits to design and interface validation sections along with deleting an outdated property

Table 1.19: Revision Table 4

4.8 Brown-out Circuit

4.8.1 Description

Block Champion: Ryan Ostroff

The brown-out circuit is responsible for supplying power to the external lock. The brown-out circuit is an intermediary block that is connected between the micro-controller and the lock. Its purpose is to allow the micro-controller to directly control the power supplied to the lock. The micro-controller must be able to turn on and off the power to the lock at any point during the lock opening attempt.

The circuit itself is composed of resistors and transistors. Two digital signals from the microcontroller are inputs to the brown-out circuit. One input control the power supplied to the lock itself. Another input discharges the lock's power supply quickly to remove any residual charge and ensure a fast turn-off.

In our attempt to get past the lock's security measures this block acts as a way for our circuit to turn-off the lock by removing the power supply to the lock at any time and be able to reboot, or reapply power, to the lock it if necessary. The lock has a security feature which prevents multiple wrong attempts of the passcode in a row. If five wrong passwords are attempted the lock will enter lock-out mode. The goal of this block is to prevent the lock from ever entering lock-out mode. If the power to the lock can be turned-off directly after the fourth wrong password attempt, the lock will not be able to save the wrong attempt counter into memory. This means that this block will never allow the lock to never enter into lock-out mode if the power to the lock is cut directly after the fourth wrong password attempt.

4.8.2 Design

The design of the brown-out circuit is very similar to a level shifter, as in its simplest form it is just taking the logic signal from the microprocessor and creating another voltage at a different level to supply power to the lock. The input to the circuit is a 3.3 V digital logic signal from the microcontroller. The output of this circuit is the 9V power supply to the lock. Included in the design of this block are four resistors, two NPN transistors, and one P-channel MOSFET. They each have a different function in this design and their function will be explained below.

The function of the different components within the schematic will be explained in detail. There

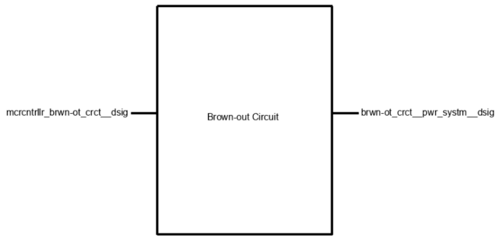


Fig. 1 : Brown-out Circuit Black Box

Figure 1.16: Brown-out Circuit Black Box Diagram

are two input signals from the micro-controller. These input signals are being modeled as digital logic signals ranging from 0 to 3.3V with 200 ns periods every 1 ms. V2 is a pulse from 0V to 3.3V (acting like a high signal from the microcontroller being sent). Initially, 3.3V is applied to R3 to create base current to Q2. R1 is a pull-up resistor that normally keeps the P-channel, Q1, off. When current flows into and turns on Q2, the gate of Q1 will be pulled to ground, turning on Q1, supplying power to R2. The lock, represented by R2, is now powered by the 9V supply. When V2 is de-asserted (0V), Q1 and Q2 will both turn off and power will be disconnected from R2 (the lock). This schematic also shows a second digital logic signal, V3, and another NPN transistor, Q3. These components are used to help dissipate any energy left inside the lock after power has been removed. An example would be when V2 is de-asserted (switching from 3.3 V to 0V) turning off Q2 and Q1. If R2 represents the lock, and the lock includes a substantial capacitance, the fastest way to lower the voltage across the load would be to create a path to ground. V3 turns-on Q3 and rapidly discharges any residual voltage that might exist across the load (R2). This will decrease the time it will take for the lock to turn off after its 9V power supply has been cut. Note that there is a risk of crow-bar current that can flow if Q1 and Q3 happen to be turned on at the same time. Although we didn't see this in the simulation, if we do see this occur with the prototype, we can quickly alter the timing of the V2 pulse to create a non-overlapping signal with V1. This means that after we deassert V3, we can wait a short delay, then assert V3, keeping this signal asserted long enough to discharge the load, then deassert V3, wait a short delay, before asserting V3 again to reboot the lock. Controlling these two signal independently gives us lots of flexibility.

R1 was chosen to be 100K. Too small a value and the resistor would waste a lot of current from the 9V supply. Too large a value and the turn-off time of Q1 would be too long. For a value of 100K, the turn-off time is approximately R * C, where R = 100K and C is the gate capacitance of Q1, which is typically 325 pF. This yields a time constant of 3.3 us, which is plenty fast enough for our application.

In order to see if the block was going to operate like it was designed, the parts needed to be put into a breadboard and input and output measurements needed to be done in order to test the prototype. This testing portion along with feedback, was the reason for the second input and the Q3 transistor, as I was not able to test with the lock but only a resistive load. I was unsure about being able to turn off a capacitive load and how long that would take. Adding the Q3 and the path for the load to have a direct connection to ground would ensure that the circuit would be able to turn off the lock within the allowed time.

4.8.3 General Validation

The Brown out circuit is being used to cut power to the lock and needs to do so fast enough to stop the lock from recording that it was an incorrect passcode. In order to complete this a signal sent from the microcontroller needs to cut power to the lock as quickly as it can. In order to see if this circuit will be able to not let the lock enter lockout mode, other portions of the project need to be completed as well. Within the code that will enter passwords into the lock the brown-out circuit will only be used when 4 other wrong attempts have been made, this also means the photo-diode circuit (which will read data from the lock and determine if a correct or incorrect password has been attempted) needs to be completed in order to know if the lock is about to enter into lock-out mode.

The real use of the brown-out circuit will not be able to be tested and finalized until the project is in a more complete state with multiple blocks interfacing with each other. That is why within the schematic and the design it is shown to have a load resistor, so that the function of the 9V power supply being turned on and off can be designed around and tested.

This block was created because there needs to be a way to not set the lock into the lock-out mode in order to create a system that would be able to open it. The work around that the lock will not be able to record the lock-out counter if the lock's power is cut fast enough was given by the project sponsor. There are multiple ways to complete this function and with multiple components. It was created with these ones because I was able to order them earlier and put together a working prototype and continue to add components along with testing. This current design could be changed with an alternate solution of using all MOSFETs, but this would only be introduced if it was shown that the BJT transistors were not operating fast enough to cut power after another block had determined that the password is incorrect.

The circuit will be later implemented in PCB which will include all the components and the connector to plug into the lock itself. This will be done to ensure that our project will be able to break the lock even when it is still inside its housing and is not opened up. The 9V battery connector that is used to connect to a 9V battery will be the output cables from the PCB.

4.8.4 Interface Validation

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
brwn-ot_crct_pwr_systm_dsig : Output		
Other: Fall time < 50ms	**To ensure that the lock is turned off quick enough to stop the lock from updating its internal counter (lock-out mode)**	VP2206N3-G-P003 datasheet [1] Turn-off delay time = 50 ns (max.) Fall time = 50 ns (max.)
Vmax: 9.0V	**The output will be between 0 and 9V and the P MOSFET needs to be able to handle 9.0V**	VP2206N3-G-P003 [1] Vdss: -60 V (drain-source breakdown voltage) Id < -6 (A) it can supply around -6A on the typical performance curve (the lock will sink a lot less current than this) Id (continuous) = -640mA will have 0.74 (W) power dissipation
mcrcntrllr_brwn-ot_crct_dsig : Input		
Logic-Level: Active High	**The micro-controller will turn on the power supply to the lock with a HIGH signal and turn it off with a LOW**	Teensy 4.1 [3] 3.3 V logic (3.3V High and 0V Low) 55 digital input/output pins 2N3904 [2] Vbe(sat) = .85V (3.3V - 0.7) will be able to put 2n3904 into saturation
Vmax: 3.3V	**The npn transistor needs to be able to have around 3.3V at the base**	2N3904 (NPN) [2] iC = 200mA (max) Ic <100mA with Ib = 0.5 mA Vcb0 = 60 Vdc (max breakdown) Teensy 4.1 [3] 3.3 V output logic

Table 1.20: Interface Validation

4.8.5 Verification Process

1. Simulate electronic characteristics This is a basic simulation of the transistors and resistors to make sure the signals are correctly controlling the 9V battery
2. Built prototype on a breadboard
3. Build the circuit on the breadboard and connect the 9v and test to see if the circuit will work with a 9V battery connected.
4. Test prototype using an oscilloscope Using a resistive load, record the input and output waveforms. The waveform from the micro-controller to the circuit and the output waveform across the load.
5. Create test code to test circuit Code a scenario where a button press/timer will turn off the battery to the lock within the specific timing and then turn the lock back on.
6. Test prototype using lock/resistive and capacitive load Connect the prototype to a resistive/capacitive load in order to test functionally off offloading the load. Record the waveforms and test if they are within the timings (<50ms)

4.8.6 References and File Links

References

1. https://www.mouser.com/ProductDetail/Microchip-Technology-Atmel/VP2206N3-G?qs=NZF3WWV0eEHdOKPMQypiBA%3D%3D
2. https://www.onsemi.com/pdf/datasheet/2n3903-d.pdf

3. https://www.pjrc.com/store/teensy41.html

4.8.7 Revision Table

Date	Name and Update
2/1/2023	Ryan Ostroff: Finished draft
2/11/2023	Ryan Ostroff: Created Section and Initial Content
2/1/2023	Ryan Ostroff: Reworded design and validation along with added references
5/12/2023	Neal Gardella: Fixed broken links
5/14/2023	Ryan Ostroff: Added interface validation table along with multiple edits in design and validation

Table 1.21: Revision Table 4

5 System Verification Evidence

5.1 Universal Constraints

5.1.1 The system may not include a breadboard

The system will not include a breadboard as it will use PCBs and ribbon cable connectors to bridge the gaps between the different PCB boards. Evidence: Link

5.1.2 All connections to PCBs must use connectors:

The system will not use connectors and ribbon cable connectors have been used to bridge the gaps between the different PCB boards. Evidence: Link

5.1.3 The final system must contain a student-designed PCB:

Below is a table of the PCBs and their SMD pad values. The total exceeds the number required per the universal constraints so this meets the requirements.

Board	Number of SMD Pads
Motherboard	2
Automated Hacking	41
Brown-out	11
Power Supply	15
LED Detection	0
Total:	69

Table 1.22: Universal Constraint 2 Table

5.1.4 All power supplies in the system must be at least 65% efficient:

The USB power to the teensy 4.1 is on the teensy board and is a linear regulator that is more than 65% efficient. Evidence: Link

5.1.5 The system may be no more than 50% built from purchased ’modules’:

Below is a table that shows what modules are purchased or designed by students. Per the requirements, only 30% of the modules within the system are purchased. 70% are student created or not applicable to this requirement.

Project Blocks	Student Created
LED Detection Circuit	YES
Lock Characterization	N/A
Brown-out Circuit	YES
Power System	NO
Code Development: I/O	YES
Code Development: Keypad Simulation	YES
Keypad and Lock	N/A
Keypad SIM Circuit	YES
Microcontroller	NO
User Input Circuit	NO
Ratio:	70% are Student Designed or N/A

Table 1.23: Universal Constraint 4 Table

5.2 Requirements

5.2.1 Code Development: I/O Functionality

5.2.1.1 Project Partner Requirement: The I/O is able to interface with the electronic safe lock.

5.2.1.2 Engineering Requirement: The input system will interface with the electronic safe lock at least 9 times out of 10 tests and successfully start the hacking process.

5.2.1.3 Verification Process:

1. The I/O device will be hooked up to the microcontroller using the pin-out as specified on the PJRC website for the TFT Touchscreen utilizing the ILI9341 Controller Chip
2. The microcontroller will be flashed with the I/O code using a micro USB cable and the Arduino IDE.

3. The I/O device will prompt the user for an input that indicates what they should do (such as "Touch to begin")
4. A digital output pin will go high at a 3.3v logic level, indicating that it has started the hack.
5. The I/O device will prompt the user that it has started the hack by stating so on its interface (such as "Hacking in progress")
6. Once the hack is complete, it will indicate as such and will output the hacked pin.
7. The I/O device will prompt the user for a rerun of the hack.

5.2.1.4 Testing Evidence: Link

5.2.2 Seamless Connection to Keypad and Lock

5.2.2.1 Project Partner Requirement: Device that is able to interface with and open different electronic safe locks without knowledge of their pin

5.2.2.2 Engineering Requirement: System will interface with the external keypad and lock through only the available debug port and 9v battery port.

5.2.2.3 Verification Process: Method: Inspection
Process: The system will use a barrel jack type plug to interface with the keypad without the need to solder any wires or irreversibly cause any damage to the lock. The system will also use a 9v connector in-between the battery and an interface to access the power supply of the lock. All connections can only be made in places a regular consumer could access.
Pass Condition: The system will be considered seamless if it is able to connect to the lock and keypad through the debug port, 9v battery plug, and attach to the front facing LED without causing irreversible damage.

5.2.2.4 Testing Evidence: Link

5.2.3 Automated Hacking Process

5.2.3.1 Project Partner Requirement: Device that is able to interface with and open different electronic safe locks without knowledge of their pin

5.2.3.2 Engineering Requirement: System will send output signals to the external keypad and lock without the need for user input during hacking process

5.2.3.3 Verification Process: Method: Demonstration

Process: User will press down on the touch screen to start automated hacking process. The user will not need to interact with the system in anyway until the hacking process has finished. Pass Condition: Except for starting the hacking process, the lock will unlock without the user interacting with the system.

5.2.3.4 Testing Evidence: Link

5.2.4 Power system

5.2.4.1 Project Partner Requirement: The "lockbuster" will be powered using a USB cable and will use a 9v Battery to have control over the lock

5.2.4.2 Engineering Requirement: The system must be powered from USB and supply a 9V power line to the external lock.

5.2.4.3 Verification Process: Will be able to show that the power supply is from a USB connection which will power the circuity along with having control over the 9V power supply of the lock.

5.2.4.4 Testing Evidence: Link

5.2.5 Brown-out control

5.2.5.1 Project Partner Requirement: the "lockbuster" will be able to turn off the lock circuity before it enters into lockout mode, this will occur after 4 wrong lock input attempts

5.2.5.2 Engineering Requirement: The system (the "lockbuster") will cut the power to the target lock, within 300ms after 4th wrong input attempt on the target lock.

5.2.5.3 Verification Process: An oscilloscope will be connected to the output (where the system connects to the external lock) and after a full 6 input digits are sent the output on the scope will be recorded. Pass condition: the external locks power drops to 0 < 300ms

5.2.5.4 Testing Evidence: Link

5.2.6 Usability

5.2.6.1 Project Partner Requirement: The system should be able to be used by anyone.

5.2.6.2 Engineering Requirement: The system can be used and operated by at least 9 out of 10 individuals without assistance from a group member.

5.2.6.3 Verification Process: A test was used that shows that at least 10 individuals outside of the project group will be selected in order to use the I/O system of the device and operate it without a group member's assistance. A video has been linked of one of the individuals opening the lock.

Trial	Assistance Given	Successful Hack
Person 1	NO	YES
Person 2	NO	YES
Person 3	NO	YES
Person 4	NO	YES
Person 5	NO	YES
Person 6	NO	YES
Person 7	NO	YES
Person 8	NO	YES
Person 9	NO	YES
Person 10	NO	YES

Table 1.24: Usability Trial Evidence Table

5.2.6.4 Testing Evidence: Link

5.2.7 Lock Characterization

5.2.7.1 Project Partner Requirement: Characterize the timing difference between a correct and incorrect passcode

5.2.7.2 Engineering Requirement: The project must explicitly define the timing difference between a correct and incorrect input.

5.2.7.3 Verification Process:

1. Use a multimeter to measure the 9V battery that powers the lock and verify it is at least 8.6V, the lock will not function properly if the battery is depleted.
2. Connect the lock's button pad PCB to the saleae logic analyzer using the barrel jack, the trace connected to the LED, and a common ground.
3. Set the LED connected channel to digital and the barrel jack connected channel to analog, enable the highest sampling rate for both channels.

4. Start capturing on the logic analyzer and iterate through passcode combinations starting from X-1-1-1-1-1, only changing the first digit.
5. After two incorrect entries enter the correct passcode to reset the lock's internal counter of incorrect entry attempts, you will be locked out for 5 minutes after 3 consecutive incorrect attempts.
6. Using the software measure the timing difference between the falling edge of the barrel jack and the rising edge of the LED's response for each digit entry, pay special attention to the timing between the last digit entry and the response.
7. Establish an average timing delay between the correct and incorrect entries.
8. Knowing the first correct digit, iterate through again changing only the second digit (5-X-1-1-1-1).
9. Repeat steps 5 and 6.
10. Export the captures and create a table to illustrate the timing delay between correct and incorrect inputs.

5.2.7.4 Testing Evidence: Saleae captures and a corresponding document illustrating the timing delay.
Link
Link

5.2.8 Successful Lock Hacking

5.2.8.1 Project Partner Requirement: Measure the LED timing to define the difference between a correct and incorrect input.

5.2.8.2 Engineering Requirement: The system will successfully open the lock after solving for the correct passcode.

5.2.8.3 Verification Process:

1. Ensure proper connection between the photodiode leads and the transimpedance amplifier circuit.
2. Completely enclose the photodiode with a dark colored opaque material to completely insulate the photosensitive surface to ambient light.
3. Connect the output of the circuit to a DMM set to measure DC voltage.

4. Power the amplifier circuit with a 5V DC supply.
5. Measure the output of the circuit while the photodiode is covered.
6. Remove the ambient light insulation and compare the circuit's output while exposed only to ambient light displayed on the DMM.
7. Deliberately expose the photosensitive surface to light that is higher intensity than ambient (flashlight, red laser, LED) and compare the output with the earlier values.

5.2.8.4 Testing Evidence: The light detection circuit's voltage output will increase in accordance with the intensity of the light that the photodiode is exposed to, complete darkness will yield very low voltage.
Link

5.3 References and File Links

5.3.1 References

1. DEFCONConference, "DEF CON 24 - plore - side channel attacks on High Security Electronic Safe Locks," YouTube, 10-Nov-2016. [Online]. Available: https://www.youtube.com/watch?v=lXFpCV646E0. [Accessed: 16-Nov-2022].
2. "EECS Project Portal," EECS Project Submission Form | OSU. [Online]. Available: https://eecs.oregonstate.edu/capstone/submission/pages/. [Accessed: 16-Nov-2022].

5.3.2 File Links

Link

5.4 Revision Table

Date	Name and Update
3/10/2022	Connor Owen: Created Section and Initial Content
5/03/2023	Connor Owen: Updated tests using info from the student portal and updated testing evidence links
5/10/2023	Ryan Ostroff: Updated reference headers with all correct values per point deduction in the checkoff
5/14/2023	Ryan Ostroff: Added an edit to fix grammar/formatting

Table 1.25: Revision Table 5

6 Project Closing

6.1 Future Recommendations

6.1.1 Technical Recommendations

1. Perform a detailed analysis of the newer firmware with Dr. Immler. The newer firmware has some essence of psuedo-randomness which makes the timing analysis difficult and time consuming. Dr. Immler has dumped the new firmware and is able to analyze its vulnerabilities in detail.
   DEFCONConference, "DEF CON 24 - plore - side channel attacks on High Security Electronic Safe Locks," YouTube, 10-Nov-2016. [Online]. Available:https://www.youtube.com/watch?v=lXFpCV646E0. [Accessed: 27-Apr-2023].

2. Explore automating the information collection with the Saleae logic analyzer. Currently all information collection with the Saleae must be performed manually and iteratively which is time consuming. The user input simulation circuit is complete which allows automatic pin entry to the keypad. A script can be written for the Saleae to automatically collect traces, parse the information, and export it to a more manageable file format as having multiple sessions open in the logic software is demanding on computer memory.
   "Saleae Logic 2 Automation interface Documentation," Saleae Logic 2 Automation Interface Documentation - Saleae 1.0.6 documentation. [Online]. Available: https://saleae.github.io/logic2-automation/. [Accessed: 27-Apr-2023].

3. Integrate all the separate circuits onto a single custom-made PCB. The current project used multiple different PCBs because it was easy to test them at first, but became very hard when trying to implement them all together. Creating a single large PCB with all circuit elements on it would make the process of testing and verification a lot easier in the long run, as the first version of the PCB would show problems that could easily be fixed in the next version.

   T. Automation, "The 3 essentials of PCB design testing," Tempo, 08-Jul-2021. [Online]. Available:https://www.tempoautomation.com/blog/the-3-essentials-of-pcb-design-testing/. [Accessed: 27-Apr-2023].

4. Dr. Immler has a variety of locks available to attack. Using the foundation of knowledge and circuits provided, explore more methods of vulnerability characterization and attack a different, potentially more difficult lock that you do not already know the solution to. This should be a sufficiently challenging and thought provoking endeavor.
   F. X. Standaert, "Introduction to side-channel attacks" 01-Jan-2020. [Online]. Available:https://www.researchgate.net/publication/225852558_Introduction_to_Side-Channel_Attacks [Accessed May 14, 2023].

6.1.2 Global Impact Recommendation

1. The two global impact recommendations that can be made about this project relate to both the public health, safety, and welfare impacts, and the environmental impacts. Both of these are the most important aspects of the design impact assessment that impact the world on a global scale. For public health, safety, and welfare impacts, the biggest impact is on electronic locks and their ability to be hacked. Due to the nature of electronics, smart locks possess some inherent vulnerabilities compared to non-electronic locks and may be susceptible to hacking [1]. By hacking an electronic system and showing those vulnerabilities, you are directly impacting another group of individuals that could have your work used for malicious reasons. In this case, keeping the work confidential and safeguarded to improve the study of hardware security is key and will prevent any misuse of the technology.

2. The second impact is environmental, one that should be considered very carefully when picking materials. Since the project is based on heavy metals and will be using leaded solder, the consideration of the environment and disposal of electronics is key as not all e-waste can be recycled leading to disposal into landfills where toxins from the heavy metals can be absorbed into the soil, harming the surrounding environment and its ecosystem [2]. Choosing the correct materials and disposing of them properly will continue to help not only the environment but the personal health of the group members as well. Additionally, as the project continues to evolve and more parts are used, tested, and recycled, lowering the footprint of this project is key to preventing any lasting environmental impact.

[1]cms_app. (2021, April 6). Is a smart lock a smart idea? GEICO Living. Retrieved November 2, 2022, from https://www.geico.com/living/home/home-protection/smart-locks/
[2]“Waste its negative effects on the environment,” E. [Online]. Available: https://elytus.com/blog/e-waste-and-its-negative-effects-on-the-environment.html. [Accessed: 02-Nov-2022].

6.1.3 Teamwork Recommendation

1. This was a highly collaborative and cross-disciplinary project. Each element of the project was vastly different from the other and took a unique effort to create. Combining each block to work as desired proved to be even more difficult. In the future one potential way to ease the culmination of the project could be to develop and work on each block collaboratively as well. Each block was designed and implemented by an individual, which proved to work, but combining each block was difficult. Designing each block as a collaborative could make the inevitable integration easier.

C. Whitcomb and L. E. Whitcomb, “15 Working with Confrontation and Conflict Resolution,” in Effective interpersonal and Team Communication Skills for Engineers, New York: Wiley, 2013.

2. Scheduling is difficult during a three consecutive term project consisting of multiple people during their most stressful academic year. Allocate dedicated recurring meeting times, outside of lecture, as early as possible to regularly work on project objectives together and/or update each other on your most recent struggles and accomplishments. A great way to come up with good meeting times can involve using phone apps that allow the group to put in specific times they are available.
   C. S. Lessard and J. P. Lessard, “Ch. 5 Personal and Project Time Management ,” in Project Management for Engineering Design, San Rafael: Morgan amp; Claypool Publishers, 2012.

6.2 Project Artifact Summary with Links

Below are file links to all previous project documentation including schematics, code, PCB files, 3D modeling files for the enclosure, and testing data gathered during the lock analysis. Additionally, schematics and PCB drawings have been pinned below for future viewing. These can also be found within the content drive and schematic links. Please note that any blue text is a link that will lead to the relevant files.
Links:
ECE44X Shared Content Drive
Enclosure Files and Versions
Touch Screen Code
User Input Circuit Schematics and PCB
Brown-out Code
SIM Circuit Schematics and PCB
Saleae Waveforms
Timing Analysis Spreadsheets and Saleae Data
Project Code Link

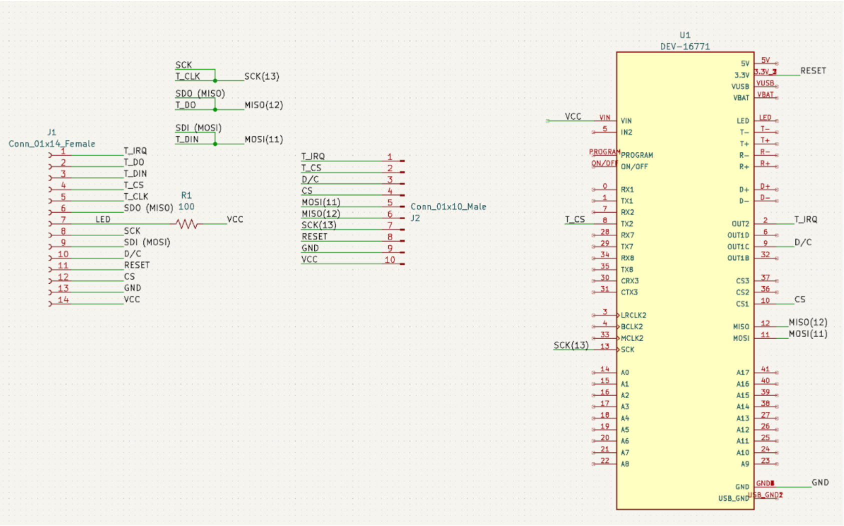


Figure 1.17: User Input Circuit Schematic

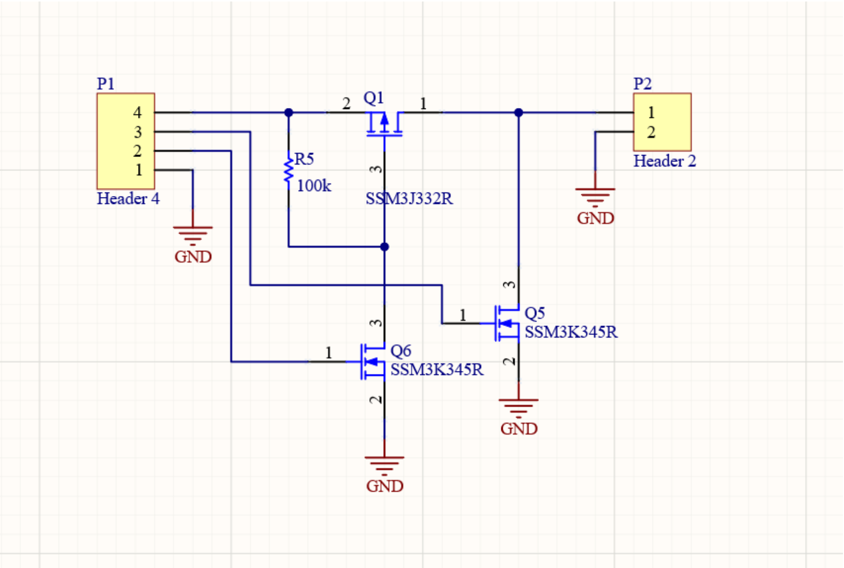


Figure 1.18: Brown-Out Schematic

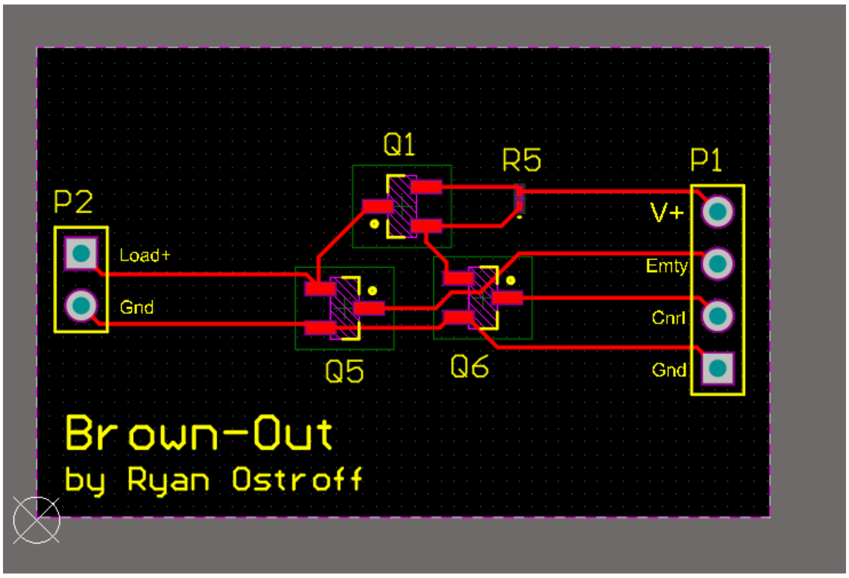


Figure 1.19: Brown-Out Pcb

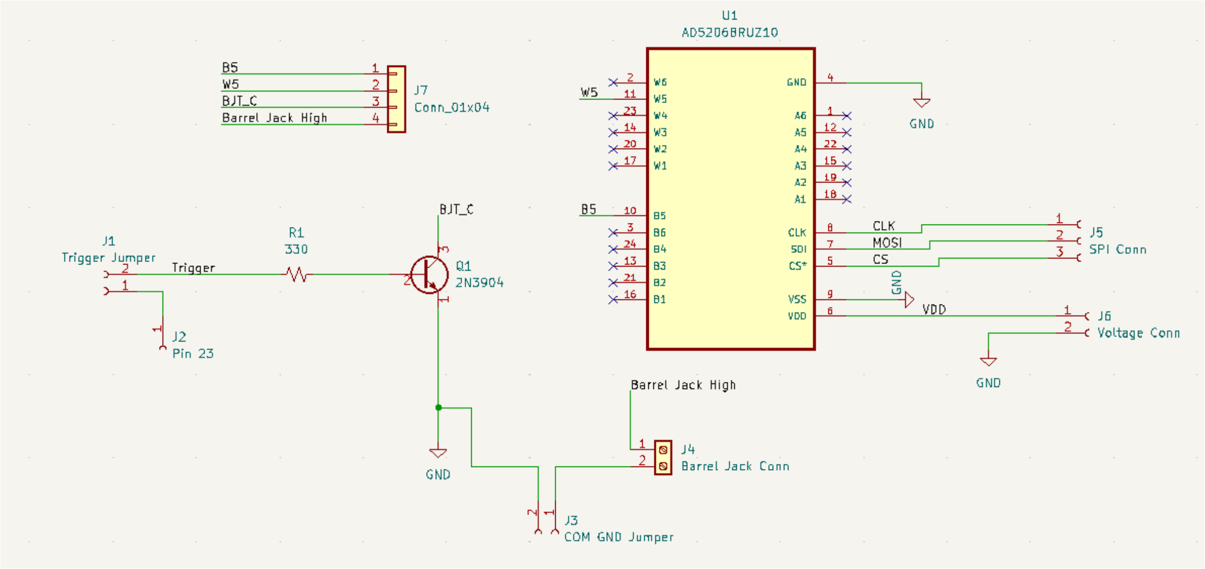


Figure 1.20: SIM Circuit Schematic

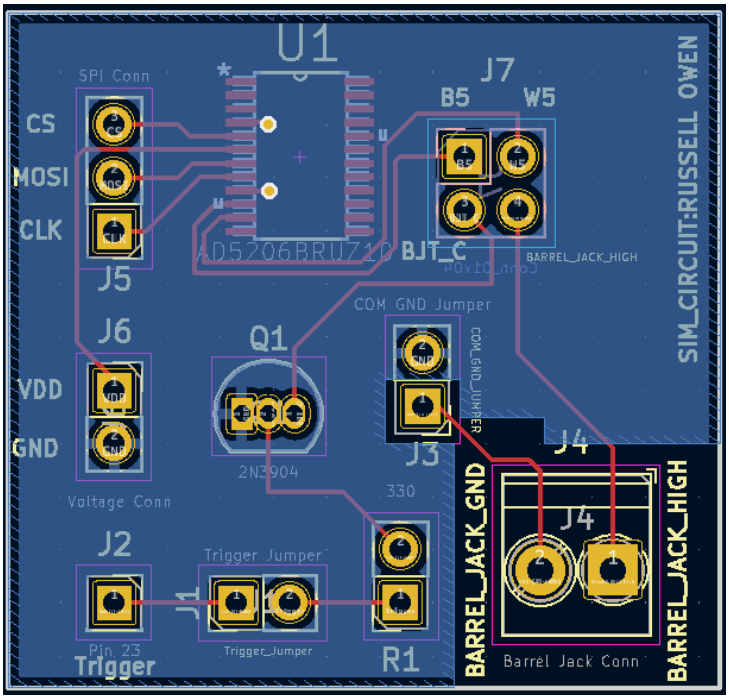
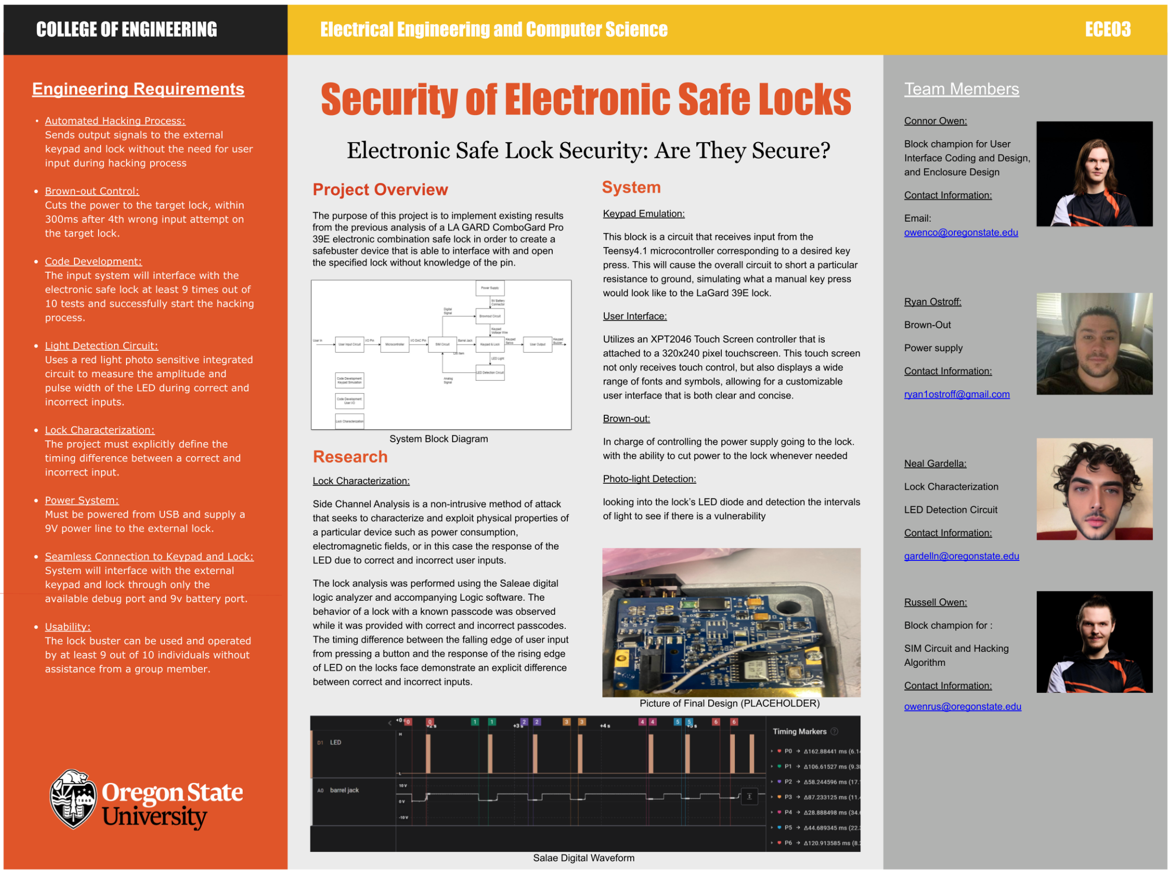


Figure 1.21: SIM Circuit PCB

6.3 Presentation Materials

Link to Project Showcase



COLLEGE OF ENGINEERING

Electrical Engineering and Computer Science

ECE03

Engineering Requirements

- Automated Hacking Process:
  Sends output signals to the external keypad and lock without the need for user input during hacking process
- Brown-out Control:
  Cuts the power to the target lock, within 300ms after 4th wrong input attempt on the target lock.
- Code Development:
  The input system will interface with the electronic safe lock at least 9 times out of 10 tests and successfully start the hacking process.
- Light Detection Circuit:
  Uses a red light photo sensitive integrated circuit to measure the amplitude and pulse width of the LED during correct and incorrect inputs.
- Lock Characterization:
  The project must explicitly define the timing difference between a correct and incorrect input.
- Power System:
  Must be powered from USB and supply a 9V power line to the external lock.
- Seamless Connection to Keypad and Lock:
  System will interface with the external keypad and lock through only the available debug port and 9v battery port.
- Usability:
  The lock buster can be used and operated by at least 9 out of 10 individuals without assistance from a group member.

Oregon State University

Security of Electronic Safe Locks

Electronic Safe Lock Security: Are They Secure?

Project Overview

The purpose of this project is to implement existing results from the previous analysis of a LA GARD ComboGard Pro 39E electronic combination safe lock in order to create a safebuster device that is able to interface with and open the specified lock without knowledge of the pin.

System Block Diagram

Research

Lock Characterization:

Side Channel Analysis is a non-intrusive method of attack that seeks to characterize and exploit physical properties of a particular device such as power consumption, electromagnetic fields, or in this case the response of the LED due to correct and incorrect user inputs.

The lock analysis was performed using the Saleae digital logic analyzer and accompanying Logic software. The behavior of a lock with a known passcode was observed while it was provided with correct and incorrect passcodes. The timing difference between the falling edge of user input from pressing a button and the response of the rising edge of LED on the locks face demonstrate an explicit difference between correct and incorrect inputs.

System

Keypad Emulation:

This block is a circuit that receives input from the Teensy4.1 microcontroller corresponding to a desired key press. This will cause the overall circuit to short a particular resistance to ground, simulating what a manual key press would look like to the LaGard 39E lock.

User Interface:

Utilizes an XPT2046 Touch Screen controller that is attached to a 320x240 pixel touchscreen. This touch screen not only receives touch control, but also displays a wide range of fonts and symbols, allowing for a customizable user interface that is both clear and concise.

Brown-out:

In charge of controlling the power supply going to the lock. with the ability to cut power to the lock whenever needed

Photo-light Detection:

looking into the lock's LED diode and detection the intervals of light to see if there is a vulnerability

Picture of Final Design (PLACEHOLDER)

Salae Digital Waveform

Team Members

Connor Owen:

Block champion for User Interface Coding and Design, and Enclosure Design

Contact Information:

Email: owenco@oregonstate.edu

Ryan Ostroff:

Brown-Out

Power supply

Contact Information:

ryan1ostroff@gmail.com

Neal Gardella:

Lock Characterization

LED Detection Circuit

Contact Information:

gardelln@oregonstate.edu

Russell Owen:

Block champion for :

SIM Circuit and Hacking Algorithm

Contact Information:

owenrus@oregonstate.edu

Figure 1.22: Project Expo Poster