Outside The Box – The “Tap” Box

Yusuf Cetin, Zachary Hogan, Will Sivolella, Fayzan Rauf

December 13, 2022

Capstone Design ECE 4440 / ECE4991

Advisor: Harry C. Powell Jr.

Signatures

Statement of work:

Yusuf:

My responsibilities in this project dealt mainly with devising and fine-tuning the piezoelectric sensor filtering circuit design, designing the PCB iterations, and final functional electronics testing. The piezo filtering circuit was heavily modified and renewed as the semester progressed while testing the sensors to determine proper required gain as an eventual input to the TI-MSP432 [1] microcontroller. I tested these circuit modifications with breadboard prototypes and the National Instruments Virtual Bench to verify functionality.

I also aided in the process of fitting all the components into the physical box as well as integration with the software and hardware aspects for the end-user's full functional experience. Debugging hardware was a big part of my job as many PCB issues arose unexpectedly, requiring various compromises given time and resource constraints.

Zachary:

My responsibilities in this project primarily pertained to the development of control and state signals on the TI-MSP432 microcontroller, programming the front-end and back-end of the web application, and programming the networking code to handle payloads to a MQTT [3] broker. In programming the TI-MSP432, I wrote an analog-to-digital conversion (ADC) function to read the battery life of the rechargeable batteries, a pulse-width modulation (PWM) signal to control the servo [6]'s directional motion, and I wrote a finite-state machine (FSM) to control the appropriate states and output functions of the TI-MSP432. I programmed the web application in Python [13], HTML [14], CSS [15], and JavaScript [16] to read battery life, to reset the tap sequence stored in memory, and to record the date and times of MSP resets.

For networking, I programmed a TI-MSP432 and TI-CC3120 [4] stack to subscribe and publish to a MQTT [3] broker as well as handle control signals to and from the TI-MSP432 [1] connected to the PCB. I verified MQTT [3] broker connections using a program called Putty [7], which enabled me to verify connections to MQTT [3] port 1083. I also worked on creating and verifying the network architecture, which was supposed to include a single MSP432 that ran a mounted CC3120 [4]. However, due to unexpected stack allocation limitations, a second MSP432 [1] was employed to mount the CC3120 [4] while the other mounted the PCB. Secondarily, I assisted with the electrical design of the battery monitoring circuit and in electrical testing and debugging throughout the semester. This included measuring signals input to the port pins from the piezoelectric sensors and the CC3120 [4] network communications on an NI testbench [2] to ensure that all incoming signals were correct.

Fayzan:

When the semester started, I was first tasked with configuring the filter. After we decided what frequency range to use, we determined that it would be most appropriate to make a bandpass filter. I created a bandpass filter with a Sallen-Key based design. Given the frequency range, I also found the corresponding component values for the resistors and capacitors in the filter. Another task I found myself doing as time went on was to mitigate the high-power consumption that was coming from the bandpass filter. To remedy this problem, I had to find new operational amplifiers that were low power. In other words, I had to find operational amplifiers that drew the lowest amount of current. The operational amplifier that I ended up using was the TLV9004 [5] with a low input bias current of 5pA and a unity gain bandwidth of 1 MHz.

Secondarily, I assisted my teammates in other aspects of the project as well. I assisted in the development of the C code that Zachary spearheaded to get the piezoelectric sensors to detect knocks, I assisted in the development of the Web Application that Zachary spearheaded also, and I assisted in the creation of the physical box enclosure that Will spearheaded.

Will:

My contributions to the “Tap” Box primarily consisted of the physical and mechanical aspects of the project as well as hardware and analog circuit aspects. With regards to the physical side of the project, I designed the physical box in Adobe Illustrator and then utilized the FabLab laser cutters located at The University of Virginia School of Architecture to cut the physical pieces of ¼'’ plywood. After assembling the pieces into the full enclosure, I designed and created the physical locking mechanism that is compatible with the FEETECH FS90R Micro Continuous Rotation Servo [6]. I 3D printed pieces that attached the servo to the interior from of the box. I also 3D printed another piece behind the door of the box that the servo arm would latch to while the door is locked.

With regards to hardware, I contributed to the circuit design revolving around servo functionality and protection as well as the battery charger. I contributed to the connection of the servo to the TI-MSP432 output signal and added protection circuitry for the servo. I also added a battery charging chip [10] and created circuit protection for the chip. My contributions also included aiding Yusuf in the PCB soldering and testing by utilizing soldering tools as well as the National Intruments Virtual Bench and multimeters.

Table of Contents

Contents

Table of Figures

(This should list the page of each figure used in your document, including the full caption.) Word has tools to help you do this very easily)

Abstract

Security is paramount when it comes to our valuables. That is why our group set out to create an inconspicuous enclosure that can only be unlocked with a certain sequence of knocking patterns on its surface. By 1) taking the shape of a tissue box, which is an everyday, unassuming object, and 2) requiring a unique unlocking mechanism, the chances of unwanted access to the box are minimized. The box contains two piezo sensors, ensuring an extra layer of security, a Wi-Fi connected web application, and a rechargeable battery powering all the electronics. The box utilizes a servo motor lock to control the unlock and lock state of the system. A given user inputs the correct sequence of knocks, upon which the box's door opens, allowing access inside.

Background

Why We Chose this Project

This project was chosen as a derivation and improvement of our previous ideas. Initially, we planned to make an unlocking mechanism that implemented an arm that was going to be attached to a joystick. There would have been a "secret handshake" performed with the arm that when completed, it would unlock the enclosure. Our new idea, the "Tap Box," is a variation on that idea. We chose the "Tap Box" to provide a more covert design using piezoelectric sensors; this way the input from the user is virtually hidden from other onlookers. For practical use, we chose this project to provide a novel and distinct authentication system from traditional security systems (door locks, numeric keypads, and touch ID) where the method of input is apparent and thus more susceptible to attacks. From a technical standpoint, we chose this project because of the course-relevant applications involved: a piezoelectric input, a microprocessor, a solenoid lock, and an extra layer of security and access provided via Wi-Fi and a web application.

Similar Projects

After searching online for similar projects and research, we found that research on piezoelectrics is extensive, with many research applications pertaining to renewable energy conversion, such as a Piezoelectric energy harvester utilized to identity pedestrians by their gait and harvest energy from the impact from each step. Though there are no research articles that pertain to the scope of the "Tap Box," we did find that there are several hobbyist projects involving similar ideas. Many of the sources from create.arduino.cc to other hobbyist projects simply involve a single application incorporating a piezoelectric sensor to read a knock above a voltage threshold, and then either lighting an LED or reading and logging that a knock has occurred through an Arduino.

The most similar hobbyist project is a secret knock activated dresser drawer lock that reads the delays between detectable knocks above threshold, and it then compares the delays to the stored memory sequence in the microcontroller. If the lock sequence is correct, the solenoid retracts; if the sequence is incorrect, the solenoid remains extended, and the dresser door remains locked.

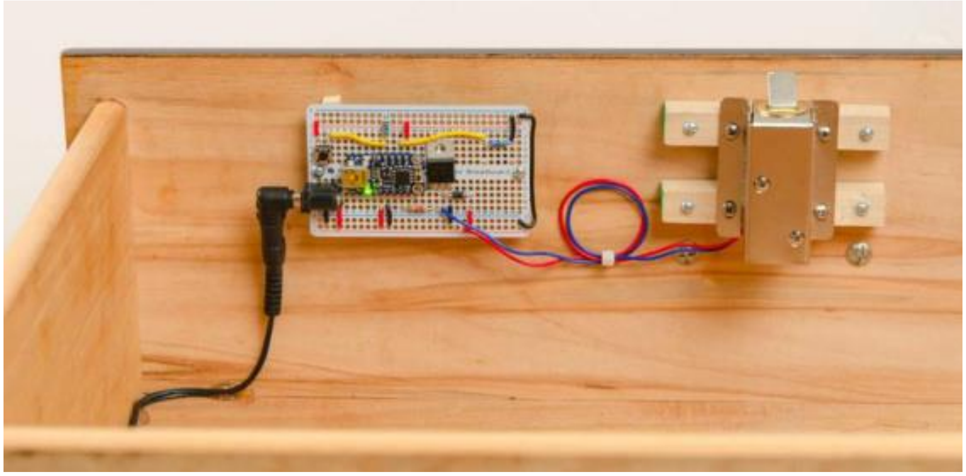


Figure 1 - Physical Layout of the Secret Knock Activated Drawer Lock [2]

Web Address*:* secret-knock-activated-drawer-lock.pdf (adafruit.com)

This design uses a breadboard, a lock-style solenoid, a single piezo buzzer, 8xAA battery pack for power, an Adafruit trinket mini-microcontroller to handle analog-to-digital readings of the voltage induced from the piezo buzzer.

Our Spin on the Project

What separates the "Tap Box" from other unlocking mechanisms derived from a piezo sensor input is the incorporation of an additional authentication piezo sensor, rechargeable batteries, input filtering, and Wi-Fi communications with an app to obtain battery readings, reset the knock sequence, and unlock in case the user forgets the knock sequence. One significant dilemma that is easy to see with other projects is the use of a single piezoelectric sensor. Though this ensures that the knock sequence matches within a certain margin of error, the device does not deter instances where the user may be listened to or recorded by an ill-intentioned third party. In effort to thwart these infiltrations, the "Tap Box" will use a second piezo electric sensor placed on the opposite side of the first sensor in the box; this will mask where precisely the box should be tapped or knocked during a given sequence. A second feature of the "Tap Box" is the use of rechargeable batteries to help isolate the design. Given that the box should not be connected to any conspicuous external power chords, the use of rechargeable batteries would be of significant benefit as the batteries could be recharged only when needed through a camouflaged input. A third benefit of the design of the "Tap Box" over other designs, specifically the Secret Knock Activated Lock, is the incorporation of a bandpass filter. The

Secret Knock Activated Lock simply reads in any piezo sensor input actuation above threshold as a “knock,” but with this limited capability, other disturbances such as sound, pressure, and even unintentional thumping of the box could potentially cause it to unlock.

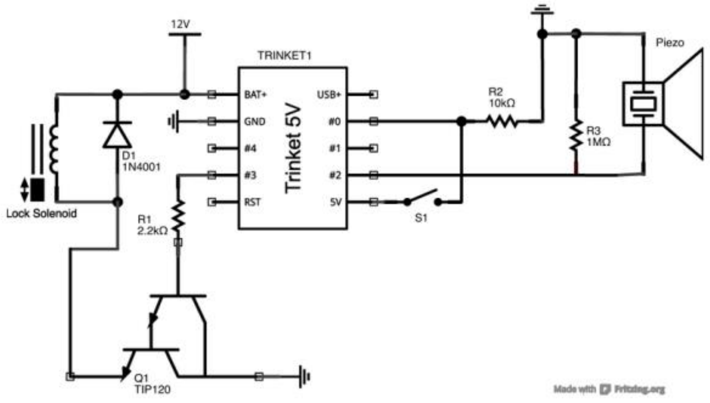


Figure 2 - ***High-level schematic of the Secret Knock Activated Lock with no filters [2].***

Web Address*:* secret-knock-activated-drawer-lock.pdf (adafruit.com)

By incorporating a bandpass filter for a specific “knock” frequency, this will make the device more difficult to hack into, and it will make the device less susceptible to unlocking due to irrelevant external stimuli. The fourth feature of the Tap Box is the incorporation of a Wi-Fi-connected application that can communicate with the MSP432 [1]. This will read the remaining battery life, will allow for a knock sequence reset, and will allow for a last resort unlock. The reason for incorporating the Wi-Fi-connected application is to reduce the potential failure of the product due to likely user errors; additionally, the app provides more information to the user and allows for a reset or unlock without the user having to manually break into the box and damage the device.

Lastly, the box itself will be disguised as a tissue box to add another level of discreteness. This design choice is because the box is intended to blend in with everyday objects, so the unwanted users won’t recognize that box’s purpose is that of a safe.

Background From Previous Coursework

This project relies heavily on concepts and techniques learned from previous classes. Topics from FUN I-III (ECE 2630, ECE 2660 and ECE 3750) will prove to be useful for the entire circuit in determining power consumption, nodal analysis, signals processing and amplification, PCB layout, and simulations and testing. Topics discussed in EAR I, EAR II, and Introduction to Embedded Computer Systems (ECE 3501, ECE 3502 and ECE 3430) will be useful in programming the MSP432 to handle interrupts, implement timers, enable and disable port pins, read wireless communications, maintain correct state (FSM implementation), and use of effective debugging tools in Code Composer Studio [10]. Furthermore, the Wi-Fi-connected application will draw strongly from coursework in Advanced Software Development (CS 3240) and Computer Networks (CS 4457) to ensure that data is communicated and parsed correctly, and to help in development of the application site.

Physical Constraints

Design Constraints

CPU Limitations

Our group used the Texas Instruments MSP432 microcontroller as it was readily available, and its operation was familiar to all team members from previous courses. It supports a clock speed of 48 MHz, allowing us to capture piezoelectric sensor signals responsively. The TI-MSP432 [1] also met our needs for GPIO pins, including an ADC pin which enabled our project to interface smoothly with peripherals.

Software Availability

Code Composer Studio (CCS) was our IDE of choice when programming in C. This allowed for simple compatibility with the TI-MSP432 microcontroller. For programming the web application, Python was used with a Django framework [Django], styled with HTML and CSS all through Microsoft's Visual Studio Code [X]. Lastly, KiCAD [X] is the free circuit and board simulation software that was used to create our schematics and PCB.

Manufacturing Limitations

The manufacturer of our PCB, Advanced Circuits, had a size constraint of 60 square inches for the board. In addition, the size of the box enclosure had to resemble a tissue box, meaning that all the components had to fit in a 6"x6"x6" wooden cube. This meant a tight fit would be required and proper orientation of components within the box needed to allow for the maximum utilization of space.

Cost Constraints

Most of the components that we needed for the project were readily available from either the NI Lab, DigiKey, or Mouser. The remaining parts were sourced from other vendors, such as Amazon. As per the outlines of the class, we were allotted a $500 budget. The costs of the project were not a significant constraint, as we ordered excess parts and had a portion of the budget remaining in the end. Parts that went into the actual final product would be significantly cheaper than what our group spent in total.

Tools Employed

Hardware

When it came to designing and routing the PCB board, we used the KiCAD simulation and design tool. Testing was performed with National Instruments' Virtual Bench. KiCAD was also used to create board schematics and footprints for some components. It was also used for routing and designing the circuit board. Furthermore, the FreeDFM service from Advanced Circuits was used to check the PCB for any errors and to make sure that the board was ready to be manufactured. The components were assembled onto the PCB by 3W Electronics. FreeCAD was used to make models of a mechanism that would attach the servo to the inside of the box as well as a lock mechanism. These 3D models were later printed.

Firmware

The firmware was written in C via Texas Instruments' IDE, Code Composer Studio. When testing this firmware, the driver library of the MSP432 was used, and other existing libraries.

Software

When it came to managing version control of the software and firmware, Git and GitHub were used. The web application was written using the Python language and was written in Microsoft's Visual Studio Code ID

Societal Impact Constraints

Environmental Impact

3D wooden printed parts made use of a 3D printer which is a better alternative to manufacturing. This way we were lowering waste, energy usage, and need for space compared to traditional factory processes. However, it is likely that the manufacturing of the PCB runs in a traditional factory-based setting. Usually, these types of settings do not use renewable energy. If it is indeed true that this is also one of those settings, the class's need for PCBs had an adverse effect on the environment. All that being said, the wood can still be recycled for future use, and the PCB and battery can also be recycled at proper facilities.

Sustainability

Our product is more environmentally sustainable than a plastic design, which would take many more years to decompose naturally than for the wood to be upcycled.

Health and Safety

This is a relatively low-risk project in terms of health and safety. The only real significant precaution to take is to make sure the rechargeable battery and battery charging circuit (which are in the box) are not put in a position to be overheated. As long as the Tap Box is put in a setting that is decently close to room temperature, there is minimal risk for fire.

Ethical, Social, and Economic Concerns

In this section you should address how your project might affect society, both from a human interaction perspective as well as an economic one. You should consider issues such as privacy, security, or how devices such as yours might influence society both for good as well as not. For example, how would a system that employs robots affect human employment opportunities? What are the ethics of automated weapons systems? How would your device affect those who might be economically disadvantaged? (You get the idea here, I hope!)

Our project is a wooden box that houses valuables and can only be opened when a correct knocking sequence is input. In terms of human interactions, this project serves as a safety measure for those who want to guard valuables from theft. This project as a concept can easily be made for large scale production by a company. Doing so would allow for monetary profit, and people en masse would be able to protect their belongings.

External Considerations

External Standards

- IPC Standards for PCB Design
- SMD Component Packages – JEDEC
- Board Manufacture, IPC
- NEMA Standards for AC Adaptors for charging the battery

The project will have to meet IPC standards particularly for PCB layouts. The design must meet IPC-2221 standards, which covers practically all aspects of PCB design [17]. IPC-2221 code specifies conductor clearance, impedance control, interconnections, drill sizes, conductor spacing and other guidelines, which will all have to be met by the project [17]

1. IPC Standards for PCB Design - IPC standards are the general requirements for the design of printed boards. IPC-2221A standardizes track and part spacings. IPC-A600J sets the standards for what is accepted for printed PCB boards, including material, holes, plating.

2. SMD Component Packages - Surface Mount Device (SMD) components conform to industry standards outlined by Surface Mount Technology (SMT) packages. JEDEC is the leading standardization entity for the size specifications for SMT packages.

3. Embedded C Coding Standard - The Embedded C Coding Standard by Michael Barr was used to facilitate the software development process and avoid bugs.

Intellectual Property Issues

- https://patents.google.com/patent/US20070159297

In this patent, the patent describes a “secure lock box system” which implements a key fob device to gain access to its contents. The device is also IoT-enabled, for the purpose of logging unauthorized and authorized accesses, relayed [8] to a server and ultimately the user. Curiously, this system also has an anti-tampering feature which utilizes a piezoelectric sensor, and a microprocessor, "which is awakened from a low power condition by the signal", overlapping with our power consumption concerns as well. Despite these similarities, our project would still be patentable since the premise of our system relies on piezoelectric sensor authentication rather than a key fob. The "Tap Box", therefore, employs a fundamentally different lock and unlock mechanism than what this patent presents.

- https://patents.google.com/patent/US20180165637A1

In this patent, although not a tangible product, the IoT security aspects of this patent overlaps with our project's Wi-Fi capabilities. The system in this patent has a remote-controlled electronic lock device that can prevent access to a door at a specific location. There is a notification function with the system connected via Wi-Fi that lets the user know if a package has been delivered. The differentiation between the Tap Box's Wi-Fi capabilities and the capability of this patented system is that the Tap Box uses Wi-Fi to connect to a module in the enclosure that "communicates" with the Web App. The Web App displays the battery level of the rechargeable battery, allows the knocking sequence "password" to be reset, and shows the previous times the box was unlocked. Therefore, there is plenty of differentiation between the Tap Box and this patent.

- https://patents.google.com/patent/US9536359B1

In this patent, a compact "cam lock" is used to secure cabinets, drawers, sliding doors, lockers, mailboxes and other door type applications and is compact in size. The lock itself is described as "long, narrow, and low in profile". The similarity between the Tap Box and this patent is that the Tap Box makes use of a low-profile lock as well in the form of the servo [6]. However, that is also where the major difference is as well. This patent asserts the invention of a new type of lock that has a unique design. The Tap Box does not assert to have a whole new style of lock that is unique. Furthermore, the designs of the locks are completely different. In contrast to the long and narrow cam lock in the patent, the servo [6] is a small rotating bar. Although similar in function, both are much different in form and execution.

Detailed Technical Description of Project

- Hardware
  - Linear Regulator
  - Piezoelectric Sensor Filtering Circuit
  - Battery charging chip
  - Battery Monitoring Circuit
  - Servo Control
  - Board Layout
- Software
  - FSM

    - Servo PWM Control
    -
- Physical Box
    - Design Overview
    - CAD Design and Modeling Process
    - Construction and Assembly Process

Block Diagram

In Figure 3, the overall block diagram of our project is graphically represented. The red indicates the enclosure, yellow indicates physical inputs to the microcontroller, light blue shows designed electronic components, green is our microcontroller, the MSP432, and purple is the mobile app, which is an output of the microcontroller. The direction of the arrows indicates flow of information, I.e. inputs and outputs.

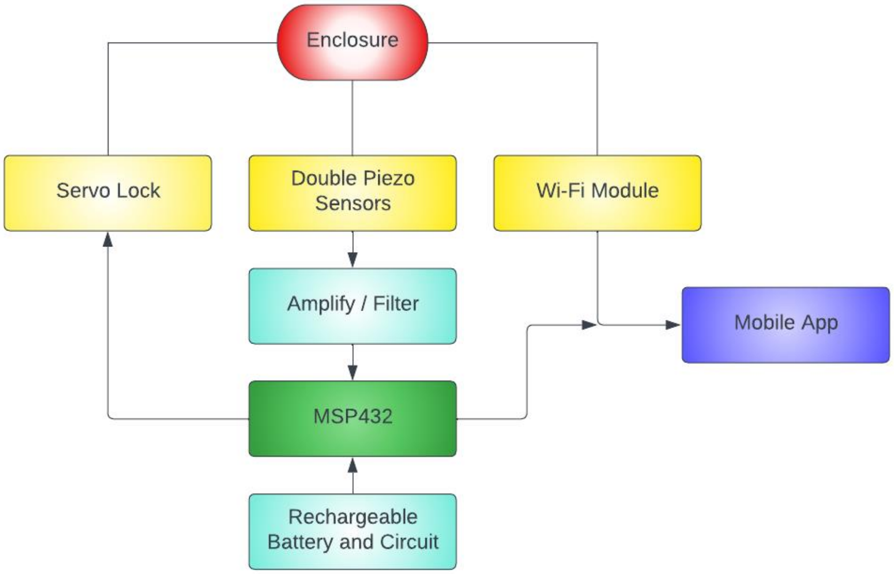


Figure 33 - Overall Functional Block Diagram

Hardware

The main components of the hardware consist of power, piezoelectric sensor filtering, and the battery charging and monitoring circuits. Because minimizing power consumption as much as possible is vital for a battery-powered project like ours, we took great care in only powering components when necessary and used the lowest power consuming parts that were available.

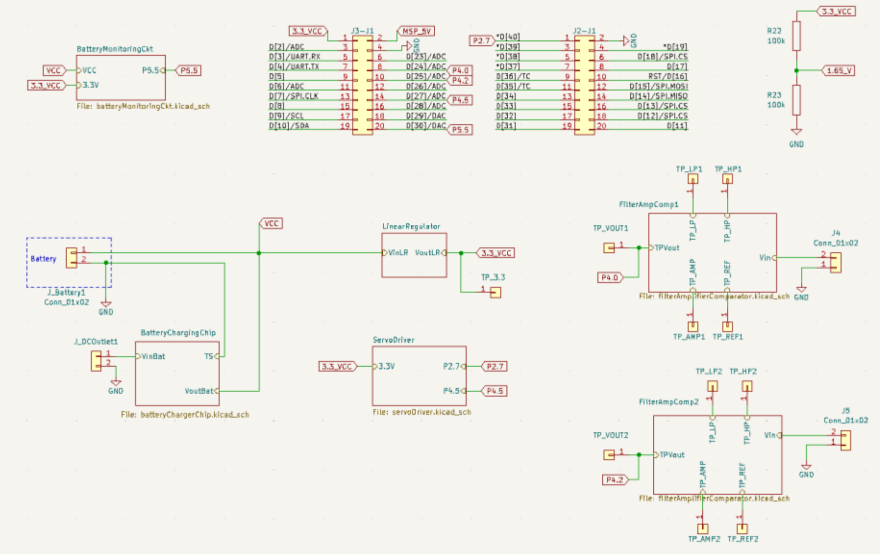


Figure 44 - Top Level Schematic

Linear Regulator

The source of power for all the components in our project was conveniently 3.3V, meaning that a single regulator would be sufficient. The power requirement was to step down our battery voltage of 9.6V to 3.3V and given that our components would be drawing minimal current during operation as well as the unwanted complexity of a switching regulator, we chose a linear regulator over a switching regulator model. Following the BA33BC0T regulator's datasheet, we arranged the schematic as shown in Figure 5.

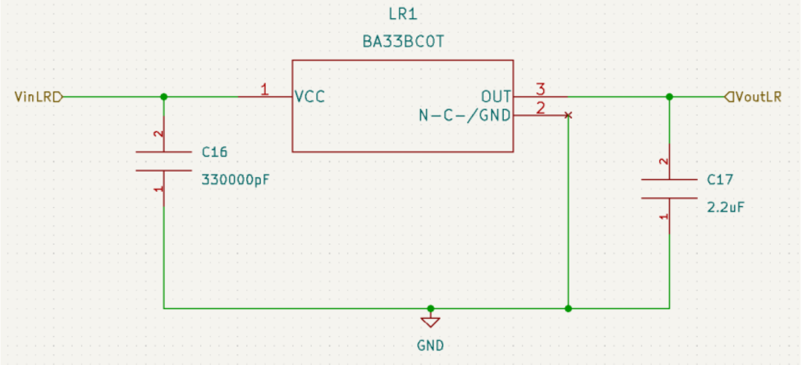
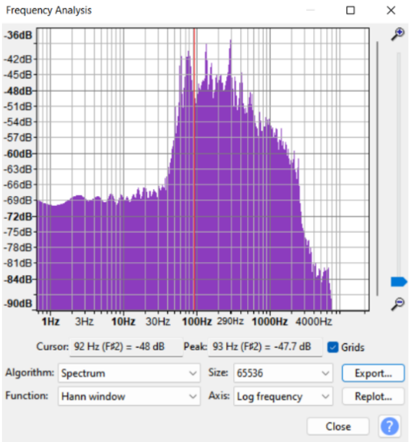


Figure 55 - Linear Regulator [9]

Piezoelectric Sensor Filtering Circuit

The piezoelectric sensors are a vital part of this project, as they provide the inputs given from a user's "knock". To serve as a proper signal to the microcontroller, however, filtering and amplification of the raw piezo signal was necessary. Since a "knock" on wood will have a characteristic frequency range associated with it, we determined this frequency range through testing as shown in Figure 6, a range of about 50Hz – 2kHz. We then designed a bandpass filter that allowed signals in this frequency range to pass, but all other frequencies to be rejected. This was possible using high and low pass blocks with operational amplifiers cascaded in the Sallen-Key architecture. Per TI documentation, component values were determined, and the filter completed as shown in the KiCAD schematic in Figure 7.



Cursor: 92 Hz (F♯2) = -48 dB Peak: 93 Hz (F♯2) = -47.7 dB ☑ Grids

Algorithm: Spectrum Size: 65536

Function: Hann window Axis: Log frequency

Figure 66 - Frequency Spectrum of Test Knocks

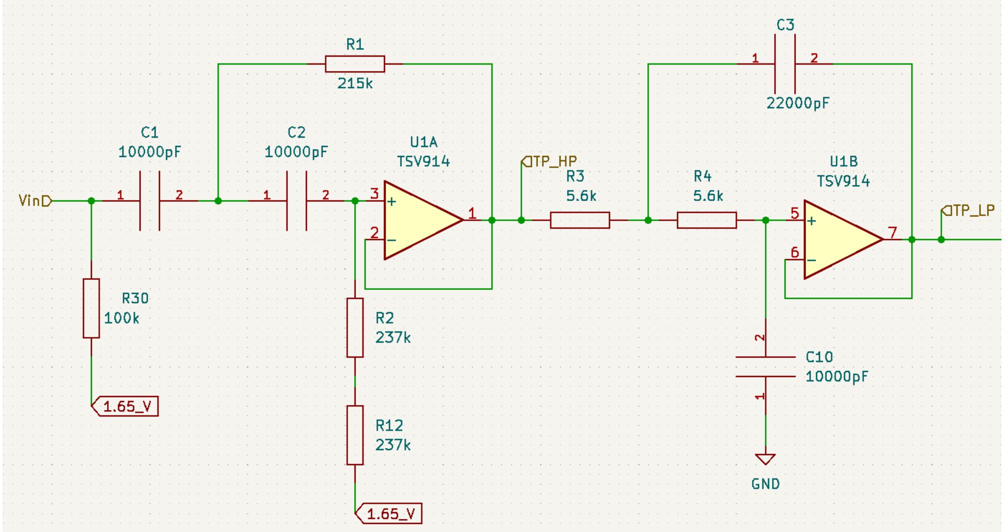


Figure 77 - Bandpass Filter

An important component is the pull-down resistor, R30 in Figure 7, which accounts for the capacitive nature of the piezo sensor via impedance matching, making it a signal that the op-amps can have no trouble with. Additionally, the input signal is referenced to 1.65V, half of the op-amp supply voltage, since the negative portion of the raw signal would be lost if ground was used. The op-amp positive rails are set to 3.3V and negative rails set to ground, 0V.

After filtering, amplification was required as the piezo signal's voltage amplitude was far too small to be detectable by the microcontroller, about 0.3V-0.4V during testing. Therefore, a simple non-inverting amplifier was configured with a gain of about 5 along with a DC-blocking capacitor to ground, schematic shown in Figure 8.

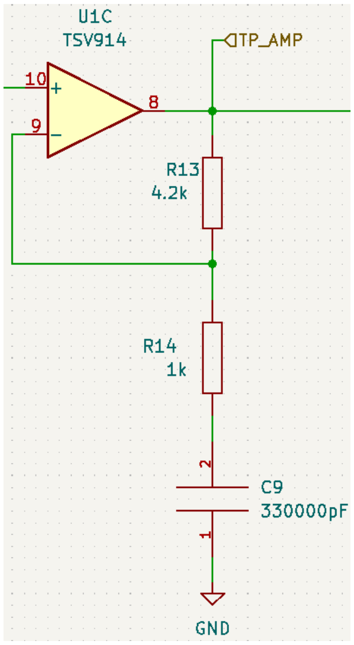


Figure 88 - Non-Inverting Amplifier Schematic

After filtering and amplification of the piezoelectric signal, a comparator circuit was configured in order to output a high 3.3V when no signal is present and a low 0V when there is a signal to the MSP432. To accomplish this behavior, a voltage divider bridged the inverting and non-inverting inputs of an op-amp, while an adequately sized capacitor in parallel provided a large enough RC time constant for the signal to be captured, swinging the output of the op-amp to the negative rail as shown in the schematic in Figure 9.

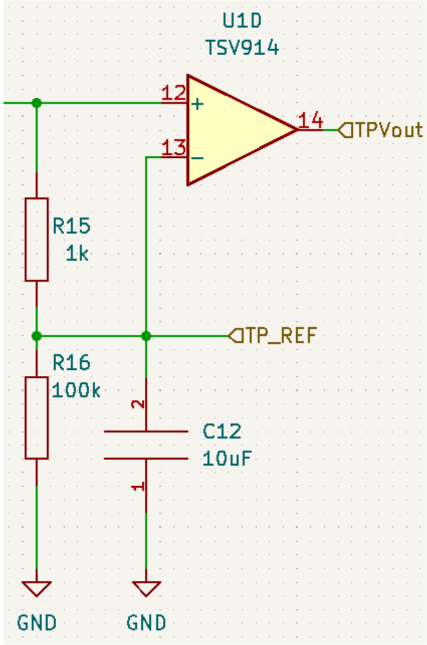


Figure 99 - Comparator Circuit

An overall schematic of the entire piezoelectric signal processing circuit is shown in Figure 10, with the positive rail of the op-amps powered by 3.3V and a DC-coupled capacitor, and the negative rail connected to ground.

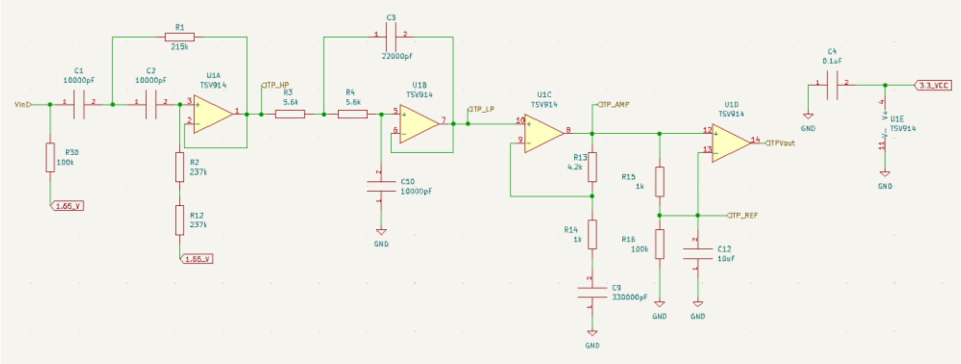


Figure 1010 - Piezoelectric Sensor Filtering Circuit

Battery charging chip [10]

To provide a robust, portable system to a user, our product had to include an internal charging capability. Luckily, such chips for Ni-MH rechargeable batteries exist from electronics suppliers and are configurable with a few passive components. The chip that was chosen, the BQ25172DSGR supported our 4-cell battery and had very low power consumption. Per carefully reading the datasheet as shown in Figure 11, we were able to assemble the chip in right fashion to suit our battery's needs. The schematic we derived is shown in Figure 12.

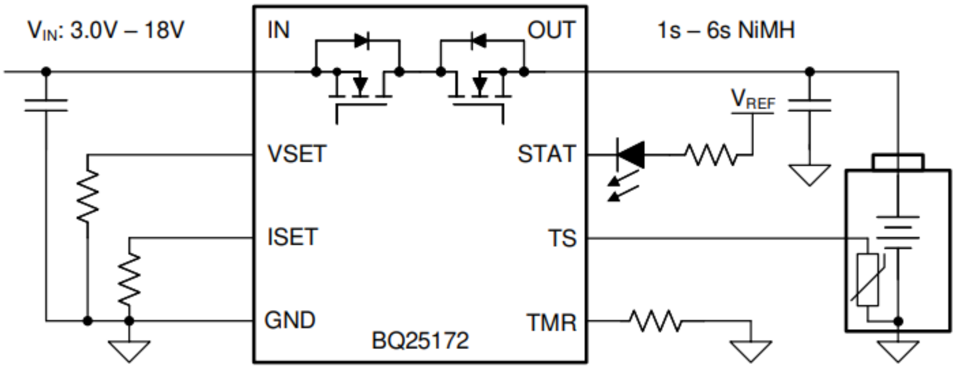


Figure 8-1. BQ25172 Simple Schematic

Figure 1111 - Battery charging chip [10] Example Configuration

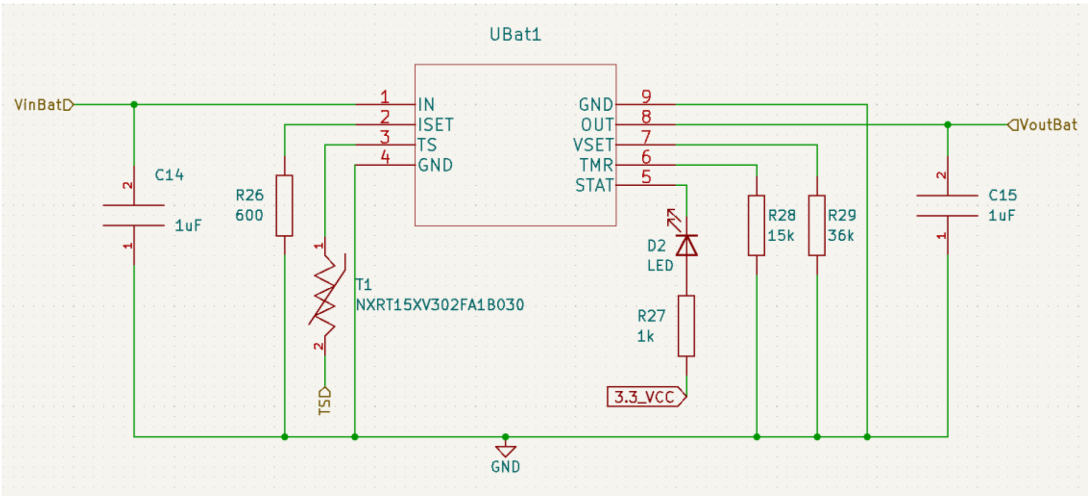


Figure 1212 - Battery charging chip [10] Circuit

Battery Monitoring Circuit

For user functionality and ensuring that the battery-powered box doesn't run out of battery, a mechanism for checking the amount of power left before shutdown was an important aspect of the project that we had to consider. This feature underwent several iterations but had the same idea at its core in all its revisions. Of course, 9.6V cannot be directly connected to the MSP432 at any time, due to the datasheet's specifications. Therefore, a simple voltage divider to reduce the battery's voltage to a maximum of about 3.3V connected to the MSP432's analog to digital (ADC) pin would allow us to accurately "measure" the battery's power level. Because of power constraints, however, the power dissipation across low value resistors would cause fast drainage of the battery. The immediate thought would be to increase the resistor values so high that power consumption wouldn't be an issue and the resistive divider could remain connected continuously. The tradeoff with this option is that the MSP432's maximum input impedance would conflict with such a high impedance seen coming in.

For this reason, low-value resistors with MOSFETs switching the divider on and off periodically were chosen as the first option. The idea behind using MOSFETs was to turn "on" the resistive divider via a control pin coming from the MSP432, and the output would feed into the ADC pin. This would happen every 30 seconds or so, to minimize power dissipation across the resistors. However, this option became unusable upon testing. The threshold voltage of the N-channel MOSFET was too high to be switched by the MSP432. Thus, bipolar-junction transistors were seen as the next possible option as shown in Figure 13, upon which the switch idea was scrapped and a different approach was taken.

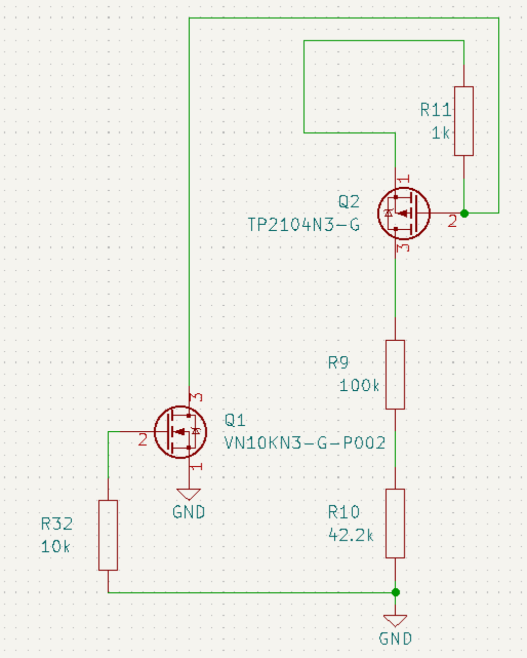


Figure 1313 - BJT Driven Resistive Divider for Battery Monitoring Prototype

Because of the complexity and uncertainty of MOSFETs and BJTs as switches for the resistive divider, the next idea simplified the circuit into just the voltage divider. It is true that by selecting high value resistors, the impedance going into the ADC pin would render the voltage level useless, however, all that is needed is an op-amp with unity gain to mitigate this issue. Since op-amps have very high input impedance and very low output impedance, simply passing the voltage through a unity gain buffer solves this issue and allows for a simpler, more elegant solution to monitoring the battery level, the schematic shown in Figure 14.

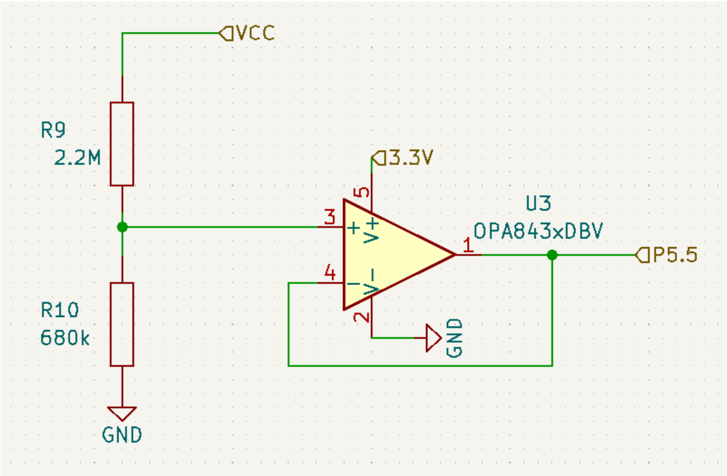


Figure 1414 - Final Iteration of Battery Monitoring Circuit

Servo Control

Staying consistent with our project's theme of minimizing power consumption at every point, our locking and unlocking mechanism was no exception. Until about halfway through our design process, our locking mechanism centered around the use of a small solenoid which we would drive with a transistor. However, upon receiving the solenoid and examining it in person, our expectations were not met, and a different approach was rapidly adopted.

Servo [6] motors are frequently used in electronics projects and present a simple method of physical rotational control. All that is required is a power source and a PWM signal to have it spin to a desired angle. This meant we could use the servo [6] with 3.3V and an output from the MSP432, making integration into our project rather seamless. For power consumption, we surmised that the servo [6] would draw 800mA of current in the worst case, for example, if the user was intentionally attempting to harm the system. In normal operation, the servo [6] draws 130mA, which is why we employed an electronic switch in the form of a solid-state relay [8] (SSR) in order to control when the servo [6] receives power, shown in Figure 14. When an unlock or lock is desired, the relay [8] is switched on with pin 4.5 on the MSP432, and PWM-controlled by pin 2.7. When pin 4.5, the input to the SSR is set high, the output pins are connected, so the servo [6] receives its 3.3V required for operation. R11's value, the input resistor, was determined via the SSR's datasheet.

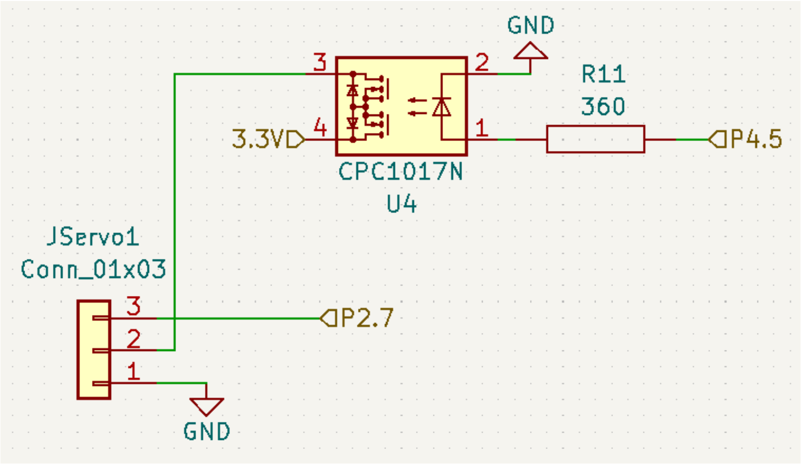


Figure 1515 - Servo [6] Driver with Solid-State Relay [8]

Board Layout

All of the schematics and PCB layout work was conducted in KiCAD software, enabling many useful features at the designer's expense. To properly interface our board with the MSP432, we imported the NI 40-pin header arrangement into KiCAD so that spacing, and connections would be consistent and accurate with the standard and the fit onto the microcontroller would be perfect. Our PCB underwent one revision to account for circuit rearrangements, the final revision shown in Figure 16.

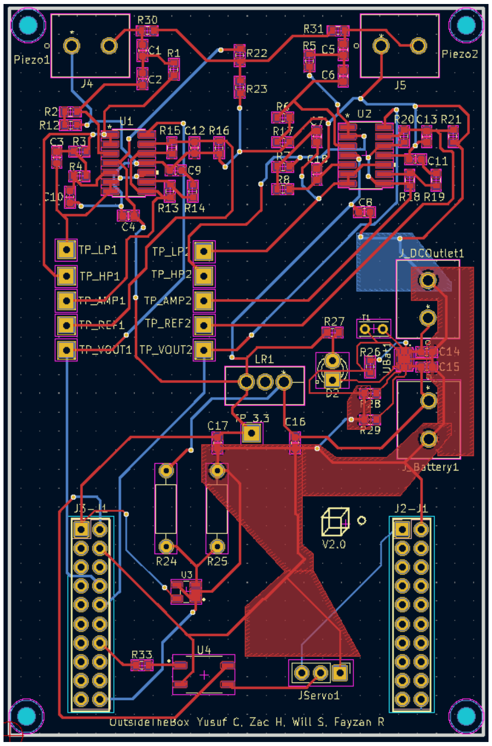


Figure 1616 - Final Revision of PCB

Surface mount components were shipped off to 3W Electronics, while the rest were hand soldered by our team.

Software

FSM

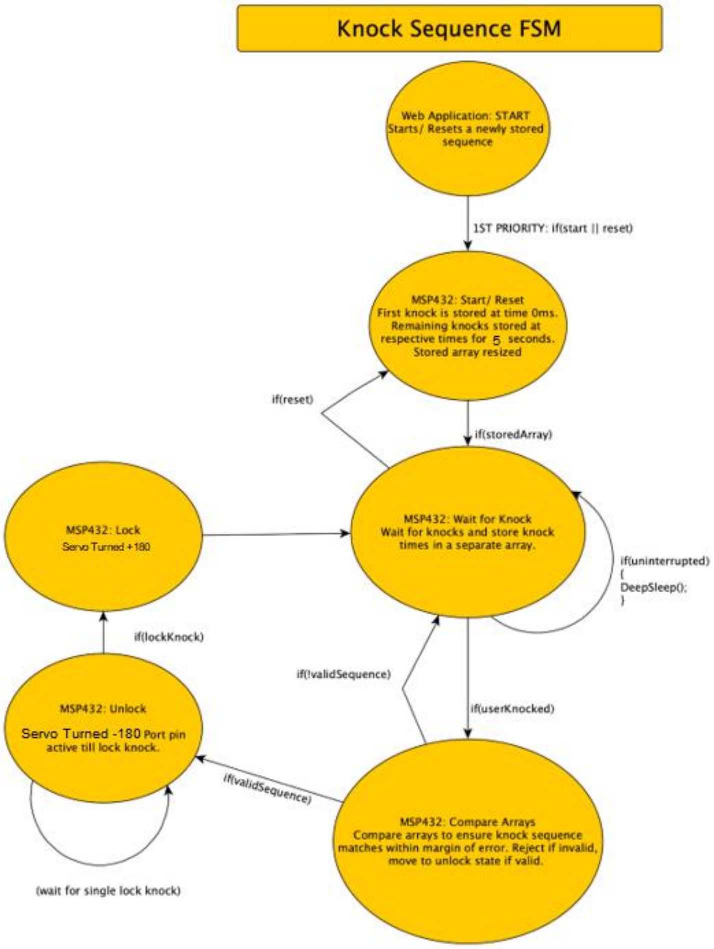
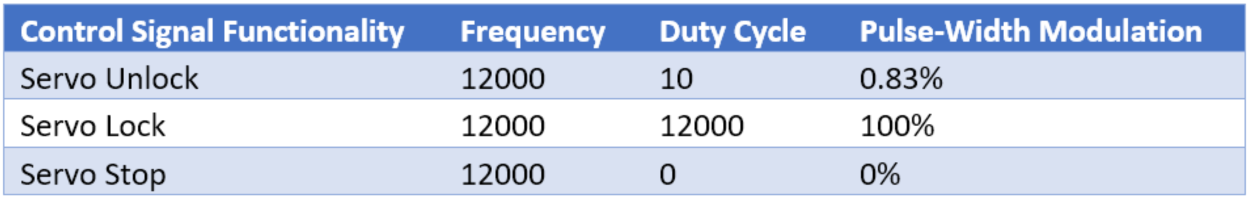

Knock Sequence FSM
Web Application: START
Starts/ Resets a newly stored sequence
1ST PRIORITY: if(start || reset)
MSP432: Start/ Reset
First knock is stored at time 0ms.
Remaining knocks stored at respective times for 5 seconds.
Stored array resized
if(reset)
if(storedArray)
MSP432: Lock
Servo Turned +180
MSP432: Wait for Knock
Wait for knocks and store knock times in a separate array.
if(uninterrupted)
{
DeepSleep();
}
if(lockKnock)
if(!validSequence)
if(userKnocked)
MSP432: Unlock
Servo Turned -180 Port pin active till lock knock.
if(validSequence)
MSP432: Compare Arrays
Compare arrays to ensure knock sequence matches within margin of error. Reject if invalid, move to unlock state if valid.
(wait for single lock knock)

Figure 1717 - FSM Implementation

The state diagram in Figure 17 depicts the primary states that the MSP432 connected to the PCB will execute at any given time. The first state is the start state, which is the first state that the program will execute. From here, the program looks for a knock or tap interrupt from one of two port pins on the MSP432 that correspond to the piezoelectric sensors. Upon a knock interrupt, the knock sequence array will store 2 to15 knocks for a time of 5 seconds. If only one knock is detected, the state will remain in the start state. If the knock sequence array is already stored in memory, it compares the newly input array with what is already stored in memory. The microcontroller then determines if the number of knocks matches the number of knocks stored in memory, and then it ensures that each knock happened within a 500ms margin of error at each respective index of the knock sequence storage array.

Authentication will happen when both arrays match within this margin of error; this will cause the box to unlock. The box will unlock by calling the ADC unlock function detailed below in *Servo [6] PWM Control* for 500ms, which executes a counterclockwise motion for 500 ms. After this, the user must enter the knock sequence stored in memory again in order to lock the box at which point the FSM executes the clockwise PWM function for 500ms at which point the FSM will return to the start state. If the state is either in start or in the listen-for-user-input mode and a reset occurs from the web application via MQTT [3] broker, then the stored and user input arrays are reset to zero and the timer resets to 0 from the start state.

Servo [6] PWM Control

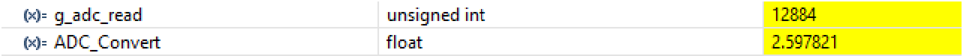


Control Signal Functionality	Frequency	Duty Cycle	Pulse-Width Modulation
Servo Unlock	12000	10	0.83%
Servo Lock	12000	12000	100%
Servo Stop	12000	0	0%

Figure 1818 - PWM Table

The Pulse-Width Modulation system is set on PWM port pin P2.7 to control the box locking and unlocking servo [6] directions. The PWM signal is generated by first using the low-speed subsystem master clock (SMCLK) of the MSP432 that has a frequency of 12MHz [1]. With the SMCLK, a clock divider of 1,000was used to reduce the frequency to 12 kHz, and the fourth Timer A Capture Control Register of the MSP432 was used to trigger interrupts at this frequency. Chosen characteristics for the duty and frequencies are displayed in the table above. These values were determined according to guidelines in servo datasheet [6] where the duty cycle for counter-clockwise servo direction (unlocking) was determined and configured to require a low PWM signal of PWM = 0.83% and the clockwise servo [6] direction (locking) was determined and configured to be high at PWM = 100% . To stop the servo , a duty cycle of 0% , and a consequent 0% PWM was implemented. In the FSM, the servo is powered using GPIO port pin P4.5 when an input sequence matches the stored sequences stored in mem ory, and P4.5 is set low when the box is finished unlocking for a turn that lasts for 500 ms. Likewise, P4.5 is only active for the box locking state, which lasts for 500ms. By setting this P4.5 active only for 500ms, this saves a significant amount of battery power, which is one of our team's greatest constraints.

ADC Conversion



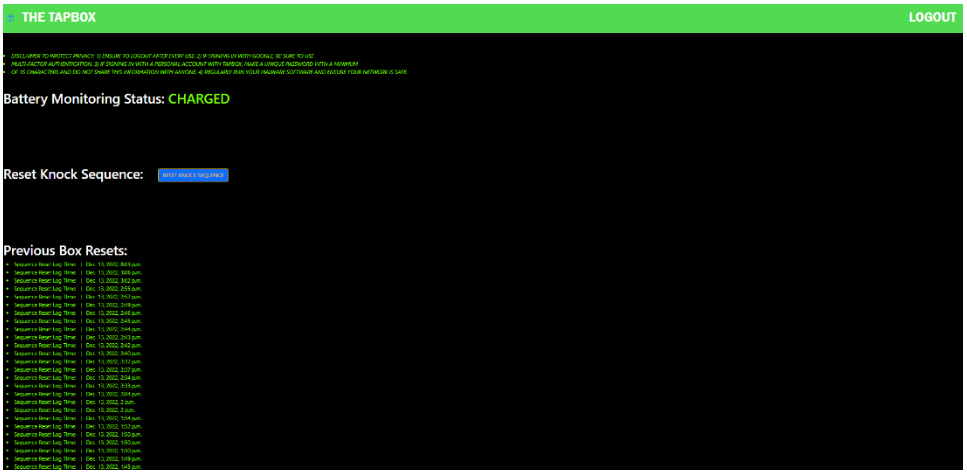
```
(x)= g_adc_read          unsigned int          12884
(x)= ADC_Convert         float                 2.597821
```

Figure 1919 - ADC Conversion in CCS

The ADC Conversion is configured using the ADC-enabled port pin P5.5 on the MSP432 connected to the PCB. Furthermore, the reference voltage is set to VCC, which is nominally measured at 2.597V after the incoming battery power is regulated from the PCB voltage regulator [9] and volage divider. Port pin 5.5 is connected to the MSP432's ADC14 module, which contains a 14-bit register that measures voltage on the pin proportionally to the ADC's 14-bit maximum value of 16383$_{10}$. For instance, if the voltage on the port pin writes the ADC14 registers to 1111 1111 1111 11$_2$ (16383$_{10}$), this indicates that the voltage on the pin is 3.3V. The

ADC code is configured in main.c to send a high GPIO signal every 10000 ms from port pin P4.4 to the CC3120 with MSP432 unit if the measured voltage is above 2.4V to indicate there is sufficient battery life. Likewise, if the voltage measured at the ADC port pin is below 2.4V, the port pin connecting to the MSP432 with CC3120 booster pack is set low every 10000 ms to indicate that the battery needs charging. An image of the ADC output is displayed in decimal as ADC_Convert of when the voltage was measured at our expected input maximum in Figure 19.

Web Application



- DISCLAIMER TO PROTECT PRIVACY: 1) ENSURE TO LOGOUT AFTER EVERY USE. 2) IF SIGNING IN WITH GOOGLE, BE SURE TO USE
- MULTI-FACTOR AUTHENTICATION. 3) IF SIGNING IN WITH A PERSONAL ACCOUNT WITH TAPBOX, MAKE A UNIQUE PASSWORD WITH A MINIMUM
- OF 15 CHARACTERS AND DO NOT SHARE THIS INFORMATION WITH ANYONE. 4) REGULARLY RUN YOUR MALWARE SOFTWARE AND ENSURE YOUR NETWORK IS SAFE

Battery Monitoring Status: CHARGED

Reset Knock Sequence: RESET KNOCK SEQUENCE

Previous Box Resets:

- Sequence Reset Log Time | Dec. 13, 2022, 3:03 p.m.
- Sequence Reset Log Time | Dec. 13, 2022, 3:06 p.m.
- Sequence Reset Log Time | Dec. 13, 2022, 3:02 p.m.
- Sequence Reset Log Time | Dec. 13, 2022, 2:55 p.m.
- Sequence Reset Log Time | Dec. 13, 2022, 2:52 p.m.
- Sequence Reset Log Time | Dec. 13, 2022, 2:49 p.m.
- Sequence Reset Log Time | Dec. 13, 2022, 2:46 p.m.
- Sequence Reset Log Time | Dec. 13, 2022, 2:45 p.m.
- Sequence Reset Log Time | Dec. 13, 2022, 2:44 p.m.
- Sequence Reset Log Time | Dec. 13, 2022, 2:43 p.m.
- Sequence Reset Log Time | Dec. 13, 2022, 2:42 p.m.
- Sequence Reset Log Time | Dec. 13, 2022, 2:40 p.m.
- Sequence Reset Log Time | Dec. 13, 2022, 2:37 p.m.
- Sequence Reset Log Time | Dec. 13, 2022, 2:37 p.m.
- Sequence Reset Log Time | Dec. 13, 2022, 2:34 p.m.
- Sequence Reset Log Time | Dec. 13, 2022, 2:33 p.m.
- Sequence Reset Log Time | Dec. 13, 2022, 2:04 p.m.
- Sequence Reset Log Time | Dec. 13, 2022, 2 p.m.
- Sequence Reset Log Time | Dec. 13, 2022, 2 p.m.
- Sequence Reset Log Time | Dec. 13, 2022, 1:54 p.m.
- Sequence Reset Log Time | Dec. 13, 2022, 1:52 p.m.
- Sequence Reset Log Time | Dec. 13, 2022, 1:50 p.m.
- Sequence Reset Log Time | Dec. 13, 2022, 1:50 p.m.
- Sequence Reset Log Time | Dec. 13, 2022, 1:50 p.m.
- Sequence Reset Log Time | Dec. 13, 2022, 1:49 p.m.
- Sequence Reset Log Time | Dec. 13, 2022, 1:45 p.m.

Figure 2020 - The Tap Box web application with 3 functions: battery monitoring status, reset, and reset logs

The web application (formerly hosted on https://thetapbox.herokuapp.com/ before expiration of access) was written in Python with Django frameworks and was styled with HTML5, CSS, and JavaScript [13][14][15][16]. The web application uses a Model-View-Controller software architecture for storing database entries, generating HTML templates, and redirecting URL requests respectively. The first page is a login page that is backed by Google Authentication that only allows the user to access the main page shown in Figure 20 after successful authentication. The main page shown in Figure 20 has three primary features. The first feature shown in Figure 20 is the "Battery Monitoring Status" that reads "CHARGED" if the MQTT client on the web application detects a new subscription from the payload of the MQTT broker reads that the battery is "charged." Similarly, the "Battery Monitoring Status" reads "NEEDS CHARGING" if the MQTT [3] client on the web application detects a new subscription from the payload of the MQTT broker that states that the battery "needs charging." The second feature is the knock sequence reset that uses a bootstrap button that uses the HTTP POST method as soon as it is pressed. When the button is pressed, the MQTT client on the web application publishes a payload to the MQTT broker that contains a char-pointer array of the word "RESET." The third feature is a set of reset logs that indicate to the user of the reset history; this helps the user to recognize if the box was reset at a date and time the user does not remember. The battery status and reset log entries depicted in Figure 20 are currently stored on a locally hosted SQLlite database, which are updated upon each MQTT [3] topic subscription and reset button press, respectively.

Networking

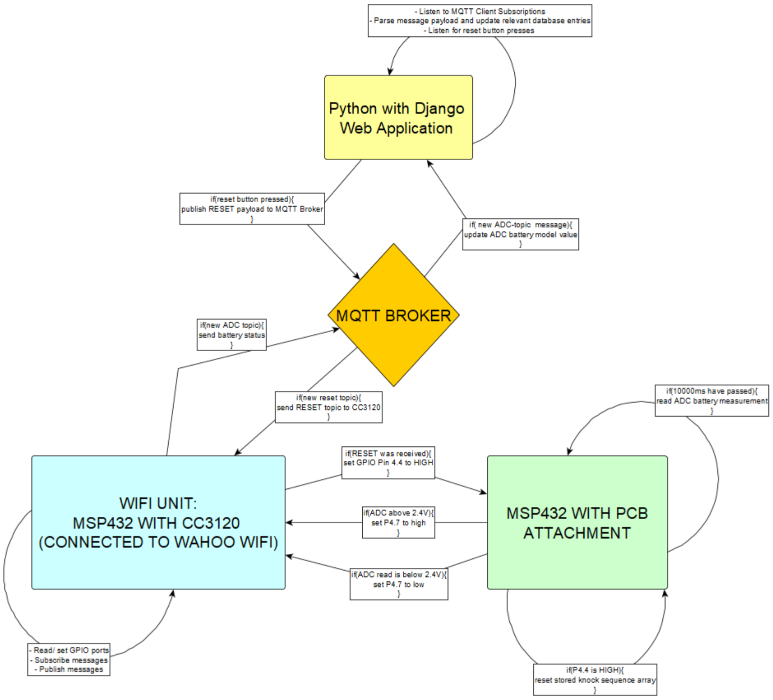

- Listen to MQTT Client Subscriptions
- Parse message payload and update relevant database entries
- Listen for reset button presses
Python with Django Web Application
if(reset button pressed){ publish RESET payload to MQTT Broker }
if(new ADC-topic message){ update ADC battery model value }
MQTT BROKER
if(new ADC topic){ send battery status }
if(new reset topic){ send RESET topic to CC3120 }
if(10000ms have passed){ read ADC battery measurement }
if(RESET was received){ set GPIO Pin 4.4 to HIGH }
WIFI UNIT: MSP432 WITH CC3120 (CONNECTED TO WAHOO WIFI)
if(ADC above 2.4V){ set P4.7 to high }
MSP432 WITH PCB ATTACHMENT
if(ADC read is below 2.4V){ set P4.7 to low }
- Read/ set GPIO ports
- Subscribe messages
- Publish messages
if(P4.4 is HIGH){ reset stored knock sequence array }

Figure 21 - Network architecture involving MQTT [3] broker between the CC3120 [4]and the Web Application

Figure 21illustrates the flow diagram of network traffic and inter-device control logic to keep the Tap Box fully functional. The CC3120 is primarily responsible for acting as a MQTT client for sending and receiving MQTT payload. If a new topic message is received signaling that the web application's reset button has been pressed, the MSP432 connected to the CC3120 unit signals to the MSP432 connected to the PCB to reset the stored knock sequence array in memory. Furthermore, the MSP432 connected to the PCB reads the battery status from the ADC functions in ADC.c, and then signals to the CC3120 if it should publish a payload to the MQTT broker that the battery status is high or low.

Physical Box

As stated, the goal of the box is to be a discreet and secure enclosure device. In order to make the inconspicuous we disguised it as a tissue box even though its true purpose is to act as a safe. The guise of a tissue box is an extra level of security to keep out unwanted users.

Design Overview

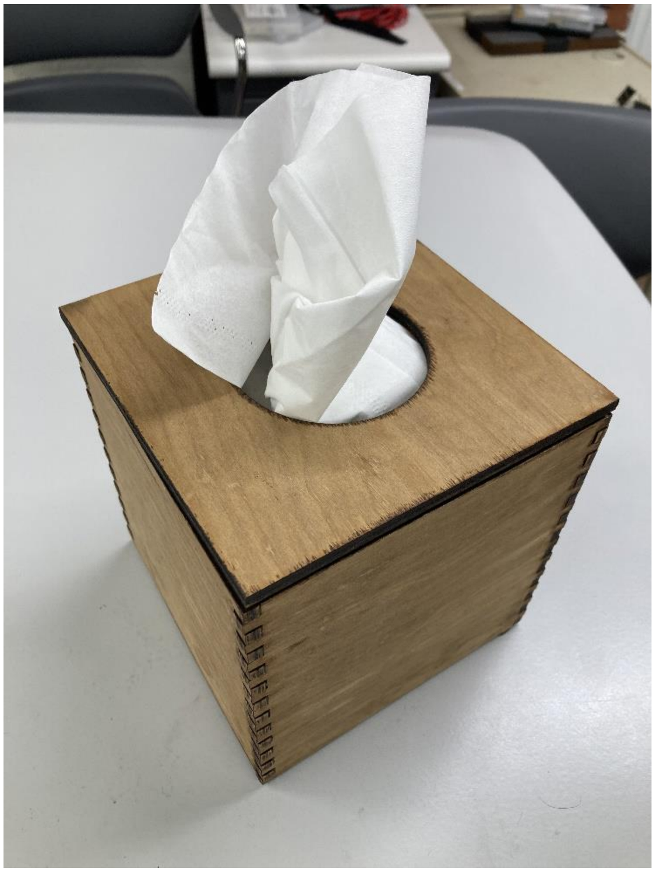


Figure 2222 - Isometric View of the Front of the Box Enclosure

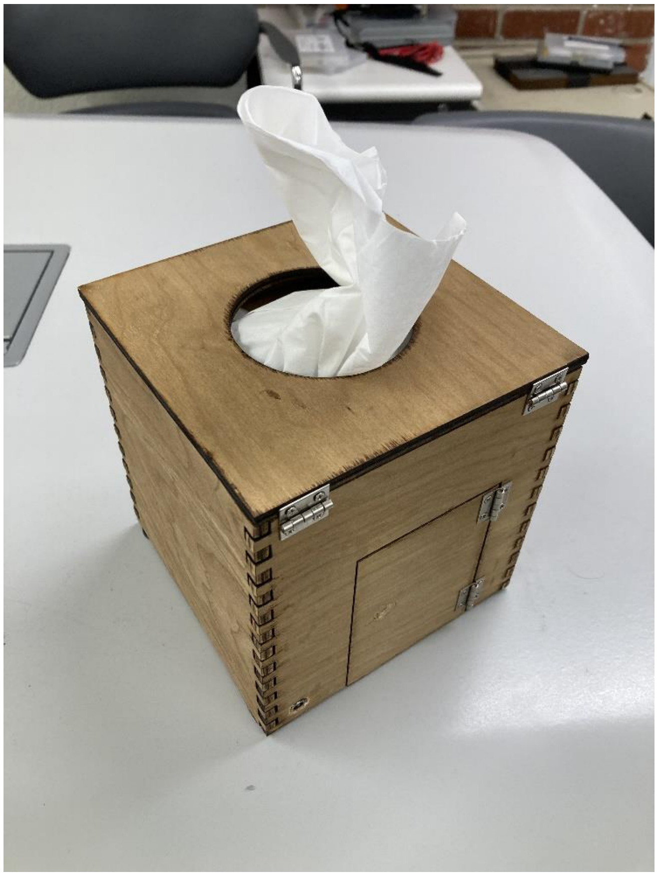


Figure 2323 - Isometric View of the Back of the Box Enclosure

Figure 22 is labeled as the “front” of the box while Figure 23 is labeled as the “back” of our box so that the user knows how to orient to box in order to maximize its ability to disguise itself. The user should most definitely place the box so that the “front” is facing towards the room so that it appears as just an ordinary tissue box and does not look out of place. If the door and barrel jack (bottom left, next to the door in Figure 23) are facing the room, it may arouse suspicion to unwanted users of the box and compromise security.

In order to aid the disguise of the box, we made it so that it has the dimensions of a typical tissue box: 6’’ by 6’’ by 6’’. As shown in Figures 22 and 23, the box has multiple layers, each of which serves a particular purpose. The top layer has a hole for tissues to go through and aid in the appearance of a tissue box as well as hinges to lift the top layer and reload tissues into the box. There is 1.25’’ of depth utilized as space for the tissues and then a 0.25’’ layer of plywood, which is parallel to the top and bottom pieces of the box, installed to separate the internal tissue compartment from the storage space. Thus, there is an approximately 5.5’’ by 5.5’’ by 4.25’’ space for the user to store items while avoiding the hardware. The TI-MSP432, CC3120 and PCB are connected to each other in a stacked formation and screwed into the back wall of the storage space. Also, the battery pack is attached to the bottom side of the middle layer of plywood using Velcro. There is also horizontal space for the servo to attach behind the back wall and next to the door. Next, there is space for the device the servo can latch to and lock the door behind the door. Lastly, there a small hole drilled into the bottom left-hand corner of the back of the box in order to make space for the barrel jack used as a port to charge the battery when low power is detected.

CAD Design and Modeling Process

The box was modeled using the software FreeCAD to help us visualize the final product of our design and organize the different subsystems of our project. The first step in the CAD modeling and design process was creating the different walls and surfaces. The top piece is 6’’ by 6’’ with a hole directly in the center that is 1.5'' in radius. The top piece is also 0.25’’ thick which goes for all the pieces since we were working with a 0.25’’ thick piece of plywood. The sides of the box are 5.75’’ tall and 6’’ wide (different from wood piece used) so that when the top piece is stacked on top, the box is 6’’ tall due to the 0.25’’ thickness of the top piece. The front and back pieces of the box are 5.75’’ tall for the same reason as the side pieces and 5.5'' inches wide (different from wood piece used) because the top and back pieces are placed between the side pieces which are each ¼’’ thick totaling the desired 6’’ width for the box. The bottom piece and middle (separates tissues compartment from storage space) piece is 5.5’’ square so that it can fit perfectly between the front, back and two side pieces. We also modeled the door, which is 4’’ by 4’’ as well as creating a 4’’ by 4’’ space in the back piece of the box for the door to occupy.

After creating all the surface pieces, we then modeled the stack of the TI-MSP432, CC3120 and PCB to the correct dimensions.

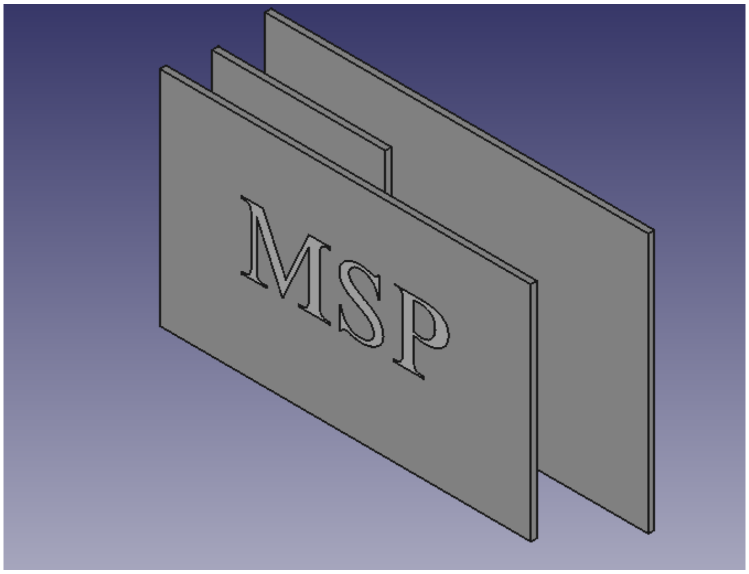


Figure 2424 - CAD Model of Board Connections

The boards are connected as in the CAD model shown in Figure 24 with the CC3120 in between the TI-MSP432 and PCB. The PCB is 3.875’’ by 2.625’’, the CC3120is 2.25’’ by 1.75’’ and the TI_MSP432 is 3.75’’ by 2.25’’ while they are all 0.0625’’ thick. They are also separated by 0.2’’ terminal blocks. Once assembled these boards are placed in the back of the box enclosure.

After modeling the boards to see how they would appear in the storage space, we modeled the servo with the servo attachment and latch that would later be 3D printed. The servo attachment is the width of the servo so that it can fit comfortably on top or on the bottom of the servo and has a small hole in order to screw the servo to the attachment. It was also important to make sure there was enough space between where the servo attaches and where the attachment connects to the wall via Gorilla Clear Max Squeeze superglue. It was also necessary to make sure the surface attaching to the wall had enough surface area to properly secure it to the wall. The servo attached to the servo attachments as shown below in Figure 25:

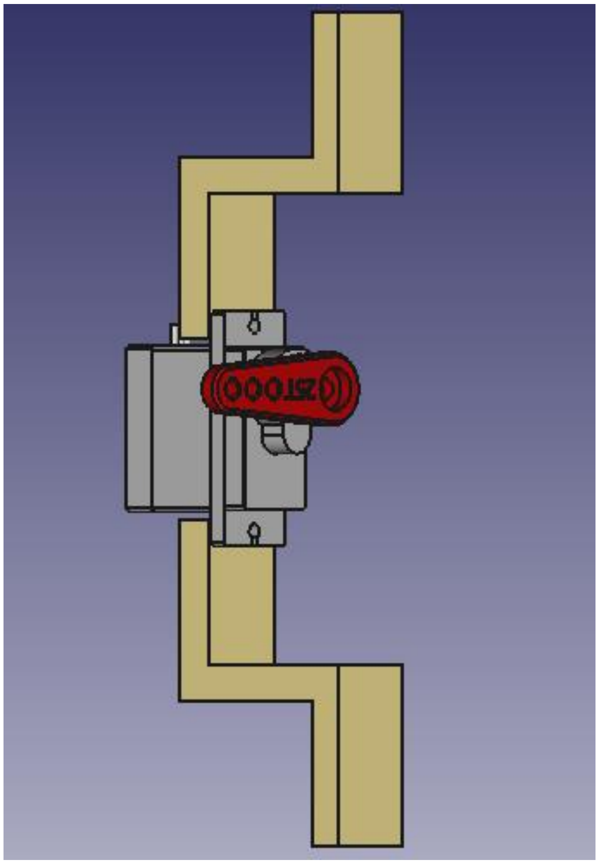


Figure 2525 - CAD Model of Servo [6] with Servo [6] Attachments

After modeling the servo attachments, it was then time to model the latch that the servo arm would go into in order to lock the box. The key for the latch was making it the proper length so that its dimensions lined up with the space between the wall and the servo arm so that the arm could properly fit. It was also important that the space for the latch to occupy was sized so that there is enough space for the arm to fit in but not enough so that the door can partially open when it is supposed to be locked.

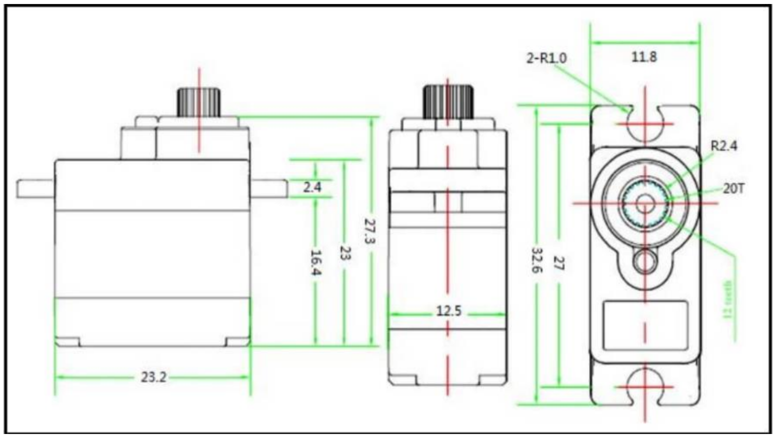


Figure 2626 - Servo Dimensions (**https://headstart.technology/index.php/product/fs90r-micro-360-degree-continuous-rotation-servo [6]/**)

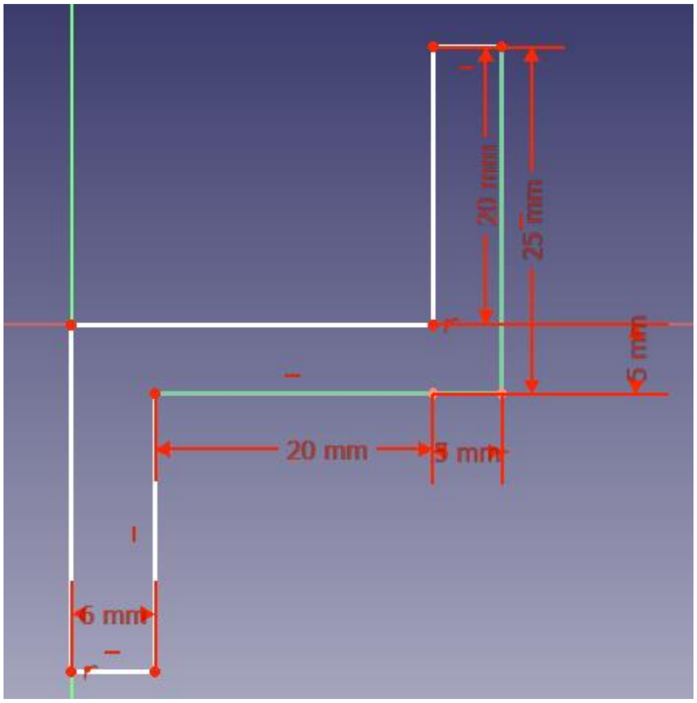


Figure 2727 - Servo [6] Attachment Dimension

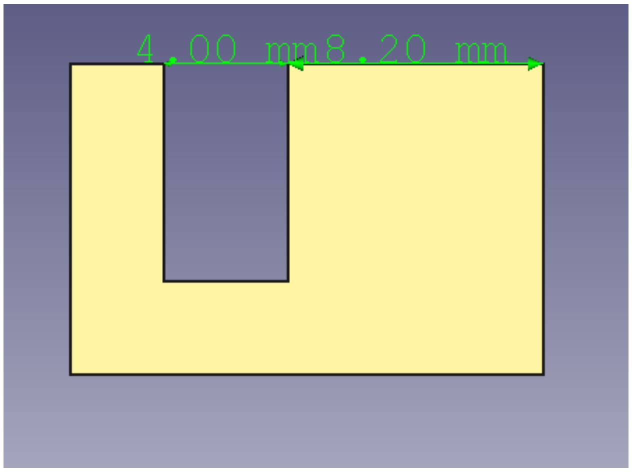


Figure 2828 - Latch Dimensions

As shown in Figures 26-28 the servo connects to the attachment 25 mm from the wall and the servo extends a total of 10.9 mm from the attachment. When the servo arm is attached the servo extends an additional 2.9 mm from the attachment making the total distance from the attachment 13.8 mm and the total distance from the wall 11.2mm. Since the servo arm is 2mm in thickness, it should occupy the space 11.2 mm to 9.2 mm from the wall. That is why the latch creates a space for the servo arm to latch that occupies the space 8.2 mm to 12.2 mm from the wall. It was important to create this margin of error to account for potential imprecision of the dimensions during the manufacturing process of the parts. Shown below in Figures 29 and 30 is the servo apparatus attached to the wall and the latch attached to the door in both the lock and unlock states.

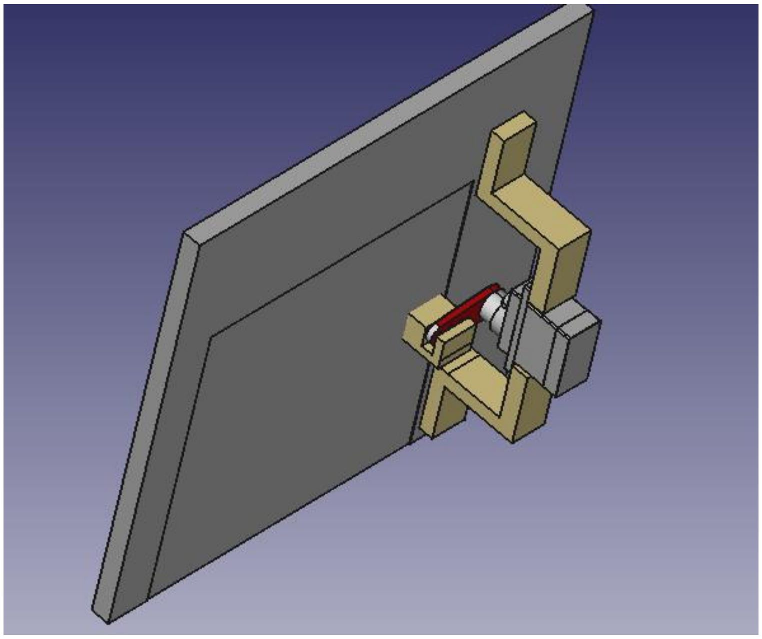


Figure 2929 - CAD Model of Box Locked

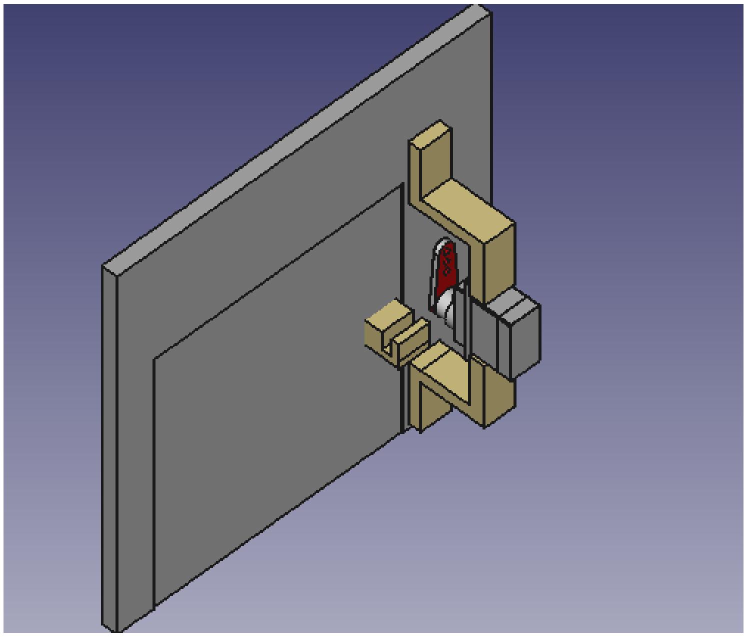


Figure 3030 - CAD Model of Box Unlocked

After we modeled the locking mechanism, we then modeled the battery pack and its attachment to the bottom of the middle piece that separates the two compartments. The battery pack was attached by Velcro. It also fits within our dimensions and is 4’’ by 2.25’’ and 0.5’’ thick. We then put everything together to create visual models of the interior of the storage space, the tissue compartment and the outside-the-box perspective all shown in Figures 31- 33.

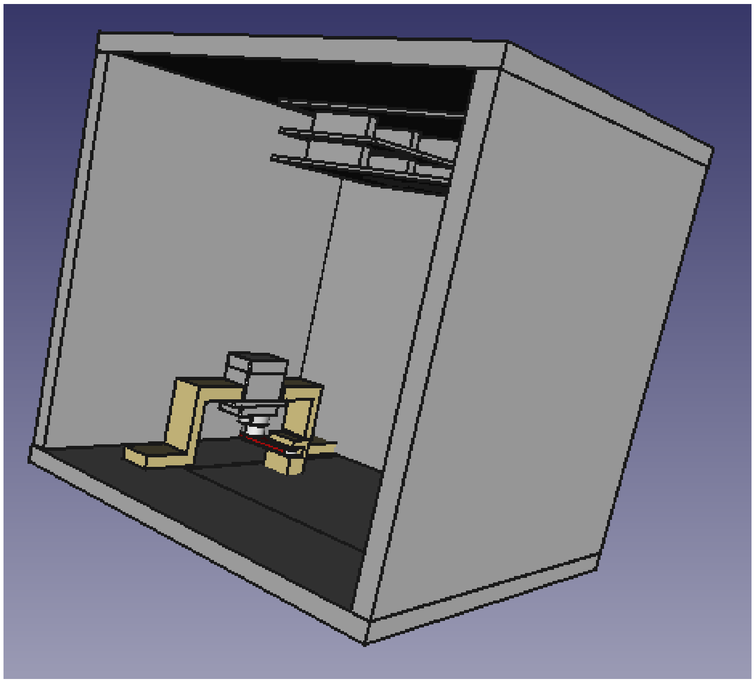


Figure 3131 - CAD Model of Storage Interior

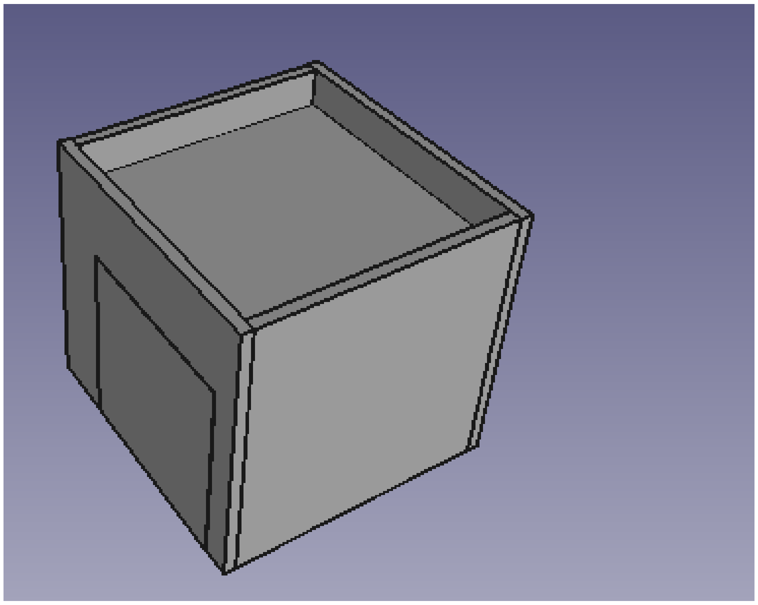


Figure 3232 - CAD Model of Tissue Compartment

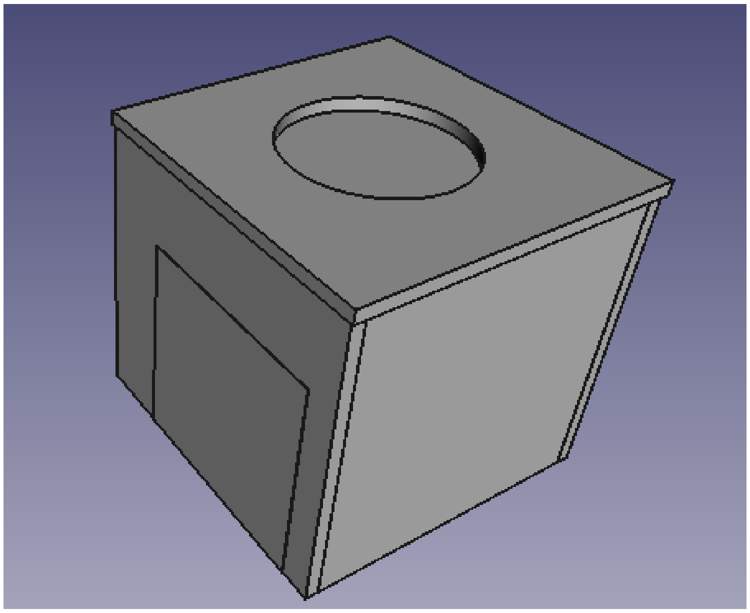


Figure 3333 - CAD Model of Box Exterior

Construction and Assembly Process

The physical manufacturing for the "Tap Box" involved 3D printing the parts for the locking mechanism as well as cutting and assembling the different parts of the box. With regards to the 3D printed parts, two servo attachments and the latch were printed from a 3D printing in Clemons Library at the University of Virginia. The servo was screwed into the servo attachments using the screws that came with the servo.

Next, we were tasked with cutting the parts of the box. In order to exemplify extreme precision and make the box resemble a tissue box as much as possible we decided to laser cut the pieces. The usage of the laser cutter allowed us to make a perfect cube and avoid screws.

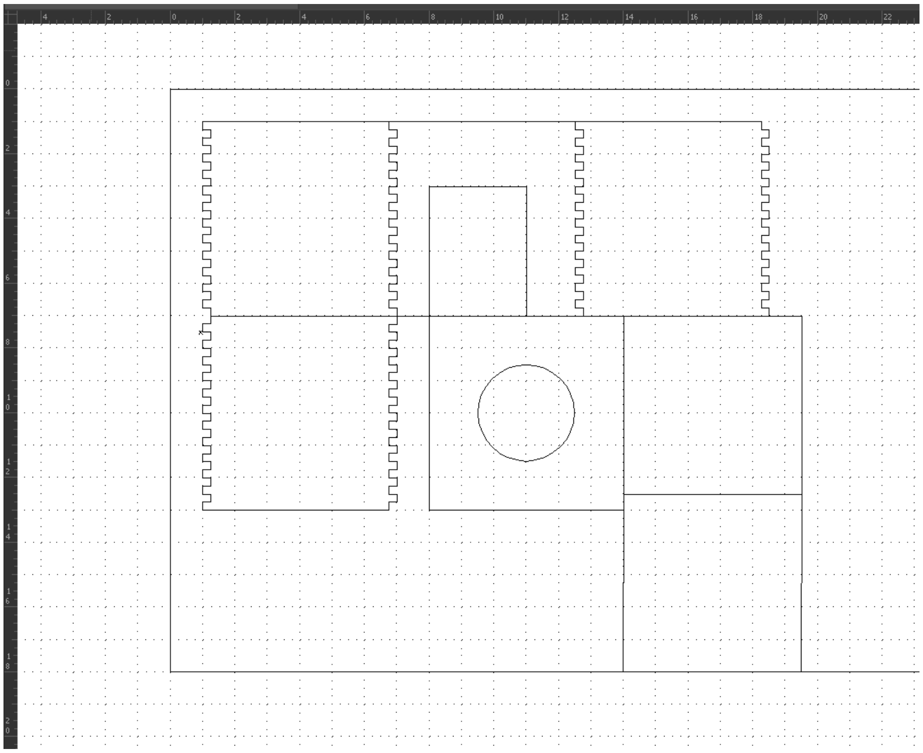


Figure 3434 - Adobe Illustrator File for Laser Cutter

Figure 34 above shows the Adobe Illustrator file that the laser cutter can read to make the appropriate cuts along the black traces. The pieces with the jig-saw-like edges are the front, back and side pieces each with the height of 5.75'' and width of 6''. The piece with the hole shown is the top part (6'' square), the two 5.5'' squares are the bottom and middle parts. Also, the door is shown on the back piece.

After placing our 0.25'' thick plywood in the laser cutter and running the laser cutter, we then had all our pieces. The back, front and sides were assembled by attaching each one together using the jig-saw-like configuration and using the Gorilla Clear Max Squeeze superglue at each jig to firmly secure the pieces together. The bottom piece was then attached via superglue and fit between the front, bottom and side pieces. We then used metal hinges to attach the door to the opening in the back of the box and used superglue to attach the latch to the back of the door. Next, the servo attachments were superglued to the inside wall next to the door in a position where the servo arm fits into the latch at the lock state.

Soon after the storage space and lock mechanism were constructed, we attached the stack of hardware to the back of the storage space and superglued the middle portion 1.25'' down from the top of the walls. Once the tissue compartment was complete, the battery was then attached to the bottom of the middle piece using Velcro. Then, we attached the top piece using hinges that face the back side of the box. Finally, we drilled a hole for the barrel jack and superglued the barrel jack to the floor of the storage space so that it was secured in the hole. Once our construction was complete, we applied wood stain to the box for aesthetic purposes. At this point from a physical standpoint, we had our final product.

Project Timeline

Below is the Gantt chart from our original proposal...

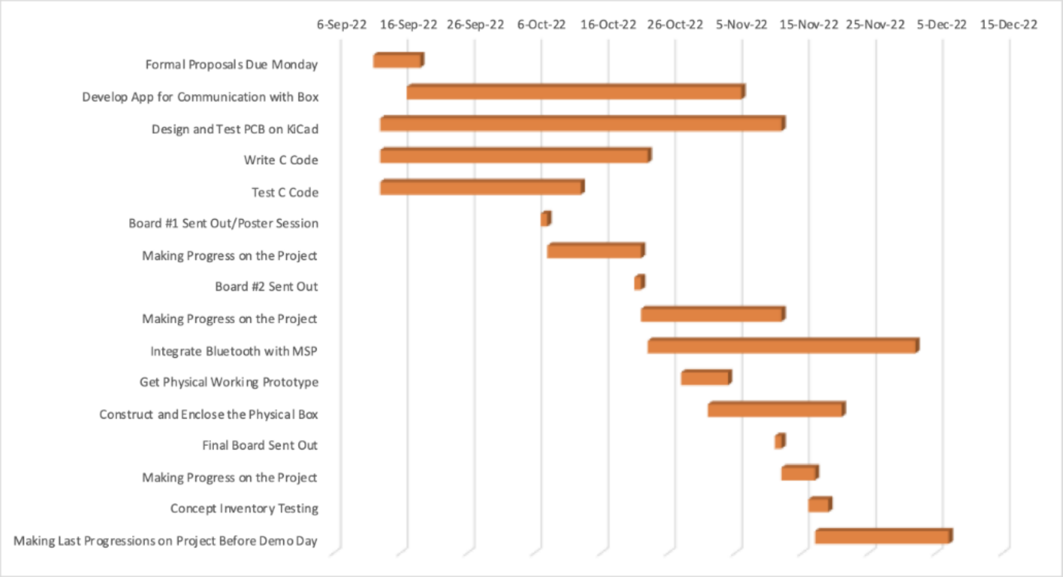


Figure 3535 – Original Gantt Chart

...and this is the updated version from the midpoint of the semester

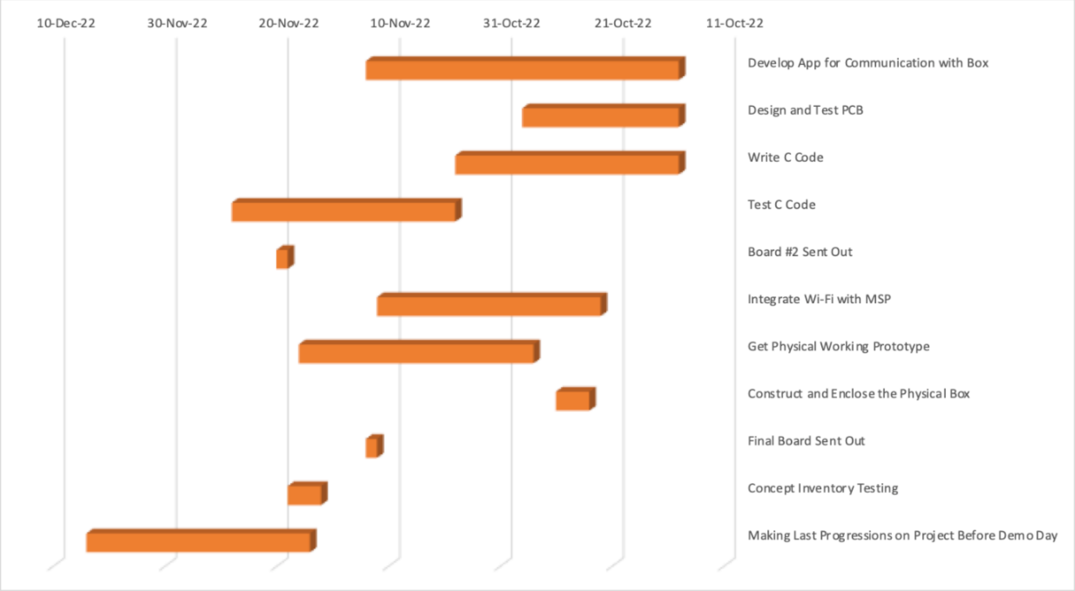


Figure 3636 – Updated Gantt Chart

Below are the tasks that were focused on by each team member. The primary tasks are the ones that are the “serial” tasks, while the secondary tasks are the ones that are the “parallel” tasks. We were able to make the secondary tasks parallel ones because those were the ones that were more effectively going to be done by multiple people. On the first day of class, we each got an idea of where the major milestones were. We were able to keep track of our progress and finish each major milestone effectively and on time.

Team Member	**Primary**	**Secondary**
Yusuf Cetin (EE)	Power distribution (buck converters to MSP432 and solenoid lock from battery), filter design	C programming to integrate devices to MSP432
Fayzan Rauf (EE)	Creation of unlocking mechanism so that it can accurately detect the knock sequences	Analyzation of amplifier/filter configuration, assist in testing the programmed C code
Will Sivolella (EE)	Working with solenoid lock: MSP432 communication with lock when knock sequence is detected, testing lock mechanism, testing code for solenoid	Building physical box, help with power distribution to solenoid lock, help with PCB design
Zachary Hogan (CPE/EE)	Writing the C programming to the MSP432 microcontroller (read input voltage from piezo sensors, write knock validation algorithm, output when solenoid lock should be contracted), developing the application interface for Wi-Fi communications	Help with PCB design, circuit simulation and testing, help with box design

Test Plan

You should show the test plan from your proposal and explain how you followed this plan or how you modified it. You should explain each of your testing procedures, and how you divided your system into testable sub modules. If testing caused a partial redesign of your device, you should explain how you arrived at that conclusion and how it influenced your redesign.

Our project, having multiple moving parts, had individual test procedures for each subsystem, under the major arches of software and hardware.

Hardware

The PCB was designed to have easy access to important nodes in the circuit with the use of test points. These pins proved to be useful and immensely eased the process of debugging when problems arose. Before even attempting to run the system as a whole, the individual circuits of the PCB were examined and tested. Upon initial inspection, missing and wrong traces were discovered, after which severing of traces and bridging of nodes was carried out. Surface mount component swaps were also unfortunately necessary, when we realized performance of the piezoelectric amplification circuit wasn’t up to expectation.

After necessary basic physical modifications were made to the PCB, the actual electronics test plan was carried out. Each subsystem was tested for proper operation as per datasheets and circuit analysis.

First, continuity was checked in order to fully ensure that no defects were produced by the board manufacturer.

Then, the BA33BC0T 3.3V linear regulator [9] was verified to produce 3.3V given a maximum of 16V input, using a voltmeter. Following this verification, the output of our battery monitoring resistive voltage divider circuit was also checked with voltmeter to ensure the battery voltage was being dropped to its calculated theoretical value. The BQ25172DSGR battery charging chip [10] was then checked to ensure that the battery was actually charging, via the indicator LED as mentioned in the datasheet’s configuration in Figure 35. Blinking indicates a fault, no LED light means charging, and a single continuous LED light means a completed charge.

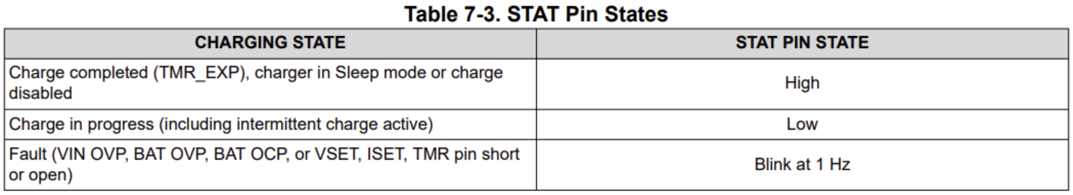


Table 7-3. STAT Pin States

CHARGING STATE	STAT PIN STATE
Charge completed (TMR_EXP), charger in Sleep mode or charge disabled	High
Charge in progress (including intermittent charge active)	Low
Fault (VIN OVP, BAT OVP, BAT OCP, or VSET, ISET, TMR pin short or open)	Blink at 1 Hz

Figure 3537 - Battery charging chip [10] Status LED Table

The most time-consuming and challenging hardware subsystem to test was the piezoelectric filter and amplifier circuits. Due to incorrect traces realized after the PCB was manufactured, manual bridging was necessary, as mentioned before. Simulating what the circuit was supposed to do via Multisim, as shown in Figure 35, represented our point of reference in the testing of our design, with the light blue square peaks representing what our circuit should output when a piezo signal is detected.

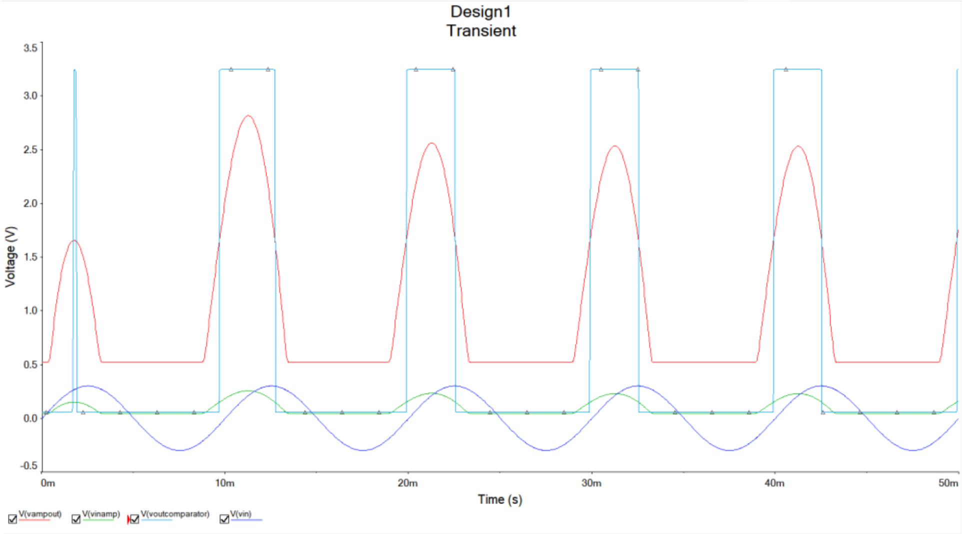
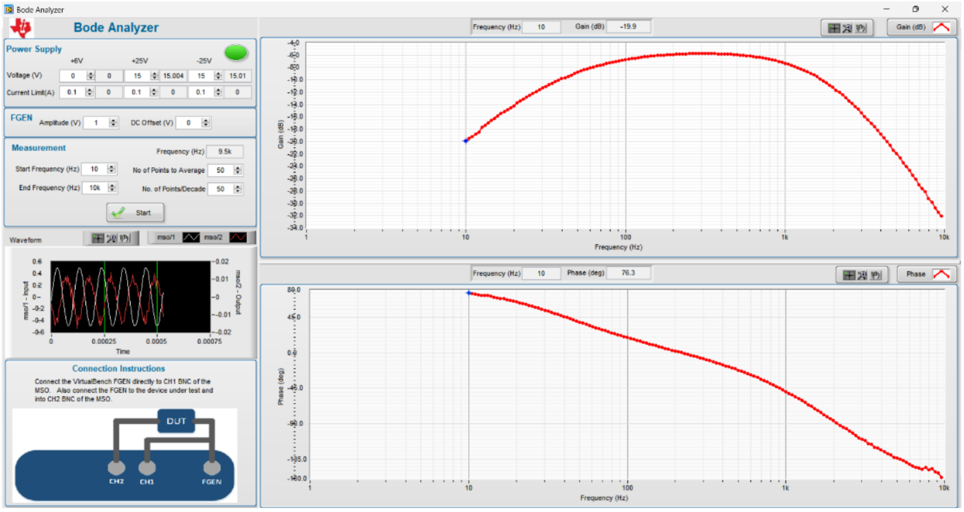


Figure 3638 - Multisim Full Circuit Simulation

The bandpass filter was built and verified using Virtual Bench software as demonstrated in Figure 36, from 50Hz to 2kHz.



Bode Analyzer

Power Supply

	+6V		+25V		-25V	
Voltage (V)	0	0	15	15.004	15	15.01
Current Limit(A)	0.1	0	0.1	0	0.1	0

FGEN Amplitude (V) 1 DC Offset (V) 0

Measurement
Frequency (Hz) 9.5k
Start Frequency (Hz) 10
No of Points to Average 50
End Frequency (Hz) 10k
No. of Points/Decade 50

Frequency (Hz) 10 Gain (dB) -19.9

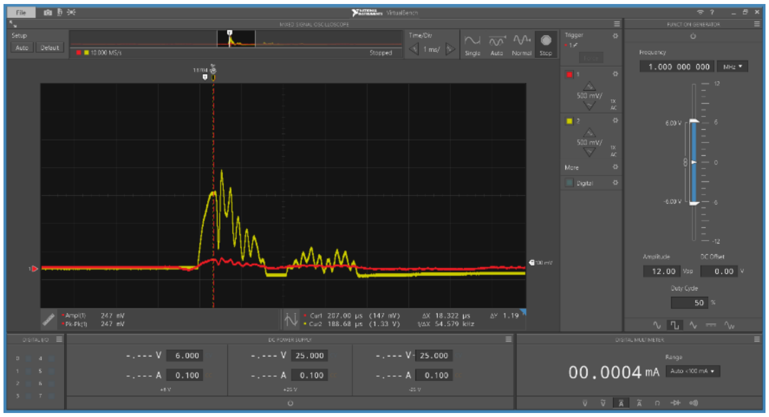
Frequency (Hz) 10 Phase (deg) 76.3

Connection Instructions

Connect the VirtualBench FGEN directly to CH1 BNC of the MSO. Also connect the FGEN to the device under test and into CH2 BNC of the MSO.

Figure 3739 - Bandpass Filter Verification

The circuit behavior at the output of the amplifier was then verified on our PCB using the Virtual Bench as demonstrated in Figure 37, with the red trace indicating the original test signal and the yellow trace showing the amplified, usable signal.



Amp(1) 247 mV
Pk-Pk(1) 247 mV

Cur1 207.00 µs (147 mV)
Cur2 188.68 µs (1.33 V)
ΔX 18.322 µs
1/ΔX 54.579 kHz
ΔY 1.19

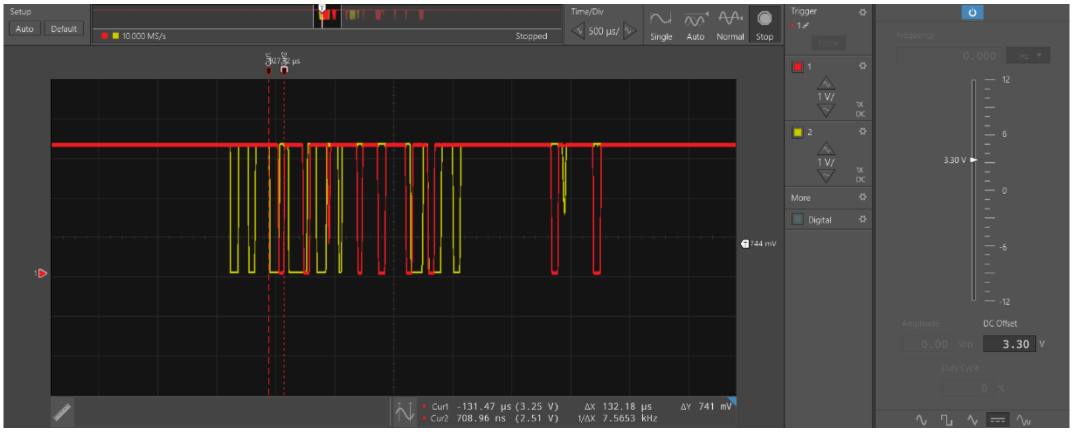
Frequency 1.000 000 000 MHz
Amplitude 12.00 Vpp
DC Offset 0.00 V
Duty Cycle 50 %

-.--- V 6.000
-.--- A 0.100
-.--- V 25.000
-.--- A 0.100
-.--- V 25.000
-.--- A 0.100

00.0004 mA
Range Auto <100 mA

Figure 3840 - Testing of Signal Amplifier

The circuit at the output of the comparator was then verified as shown in Figure 38, where the yellow trace is the output of the "knocked" piezo in question, and the red trace is the other piezo. Obviously, there are intermingling waveforms as the piezoelectric sensors aren't fully physically isolated, being in the same enclosure. However, due to the design of the software and adequately fast clock speed of the MSP432, which takes only the first input and ignores any signal after it for a time period, this issue is deemed trivial.



Cur1 -131.47 µs (3.25 V)
Cur2 708.96 ns (2.51 V)
ΔX 132.18 µs
1/ΔX 7.5653 kHz
ΔY 741 mV

DC Offset 3.30 V

Figure 3941 - Verification of Both Piezos Output

Although the results above seem straightforward and acceptable, there were many mishaps along the way. Towards the end of the project assembly process, an unexpected issue with the PCB arose, requiring extensive debugging efforts from the whole team. Because the problem was unable to be resolved, an improvised solution via breadboard was constructed to create a functioning piezoelectric sensor circuit. Although not ideal, resources and time were very strained and the team needed something that just worked. This was a great lesson in both electronics design as well as project planning in general. Making crucial deadlines and leaving plenty of room for testing and integration and overcompensating in every way is what makes a large project like this successful.

MSP Software

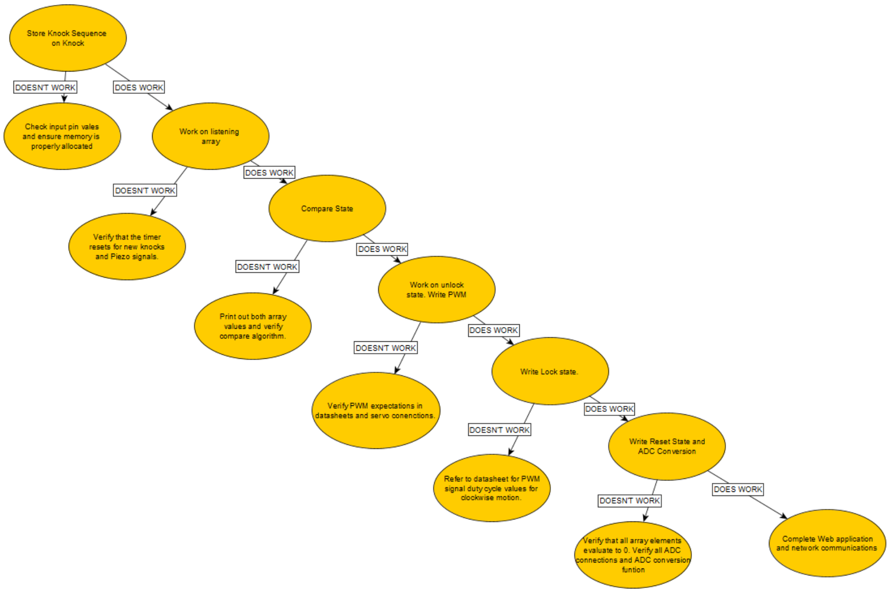

Store Knock Sequence on Knock
DOESN'T WORK
DOES WORK
Check input pin vales and ensure memory is properly allocated
Work on listening array
DOESN'T WORK
DOES WORK
Verify that the timer resets for new knocks and Piezo signals.
Compare State
DOESN'T WORK
DOES WORK
Print out both array values and verify compare algorithm.
Work on unlock state. Write PWM
DOESN'T WORK
DOES WORK
Verify PWM expectations in datasheets and servo conenctions.
Write Lock state.
DOESN'T WORK
DOES WORK
Refer to datasheet for PWM signal duty cycle values for clockwise motion.
Write Reset State and ADC Conversion
DOESN'T WORK
DOES WORK
Verify that all array elements evaluate to 0. Verify all ADC connections and ADC conversion funtion
Complete Web application and network communications

Figure 4042 - Test plan for the FSM MSP software implementation

As shown in Figure 40, the software test plan was executed one state at a time. Given the robustness of the software implementation and the number of condition variables that dictate state, a test plan was vital for the successful completion of the FSM software of the MSP432 connected to the PCB. To ensure that the FSM was implemented correctly, port pins values were measured using the NI National Instruments test bench to make sure technical values from the hardware test plan aligned with what we expected. As soon as one state's implementation was correct, we proceeded to program and debug the remaining states until all states were complete. The design was then tested with piezo inputs to ensure that knock timings and knock sequences were accurately stored in memory.

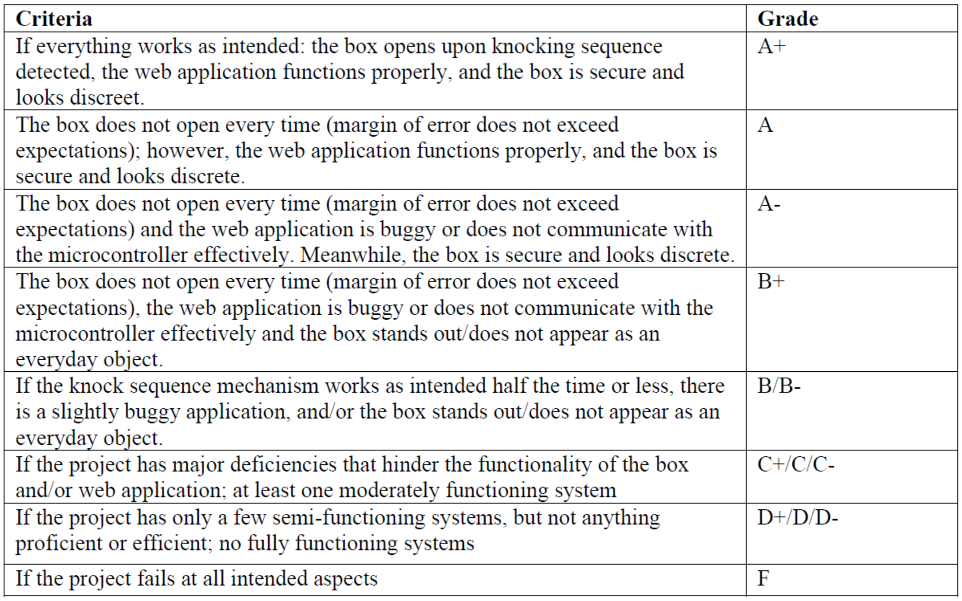
Whole System Integration Test

After all the hardware and software pieces were configured and verified to be working, integration was the next step to tie everything together into the enclosure. The whole system test included testing each functionality of our project as was intended to be experienced for the end user. Demonstrated clearly in our team video, a knock sequence was set initially as the password and then the correct and a series of incorrect sequences were inputted to the box. Then, it was verified that correct sequences unlocked, and incorrect sequences caused the box to remain in its current state.

Next, web app functionality was tested by running the server locally and clicking the "Reset" button while immediately after inputting a new knock password into the box. This password was then tested as previously mentioned. The battery status display on the web app was then verified to be the actual status of the battery via a voltmeter, which matched as expected.

Final Results

Our project exhibited full functionality at the end of the design process, despite several compromises, mainly on the hardware side. The Tap Box unlocks upon the correct password sequence, doesn't unlock upon an incorrect sequence, differentiates between knocks on each side of the box, and the web app allows for password resetting as well as battery monitoring. The physical enclosure of the box also exhibits the intended purpose of the project to be an inconspicuous household object, being a wooden tissue box shown in Figure 41. Given that the box would require further fine-tuning for the margin of error, these features all meet the criteria set forth in our proposal for a grade of A.



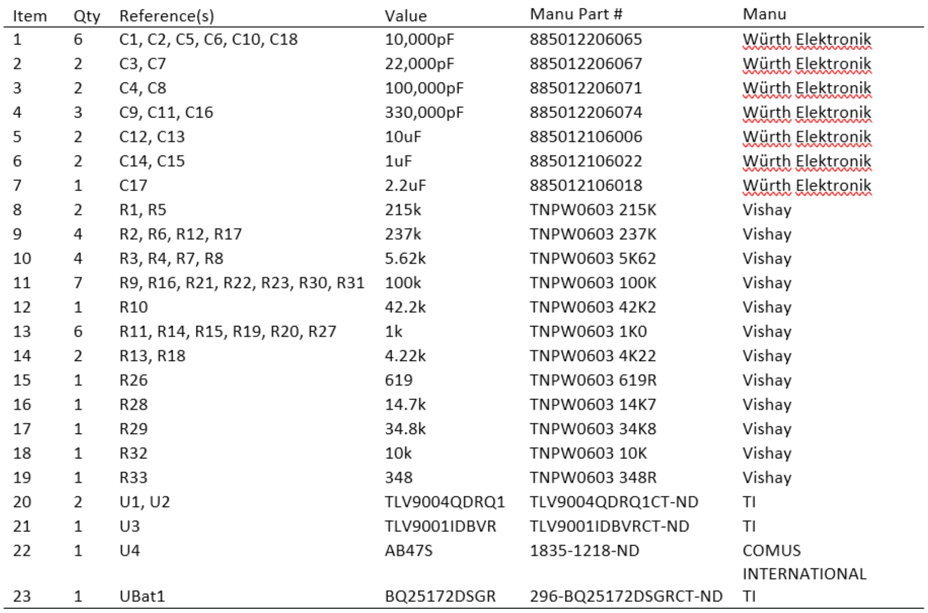
Criteria	Grade
If everything works as intended: the box opens upon knocking sequence detected, the web application functions properly, and the box is secure and looks discreet.	A+
The box does not open every time (margin of error does not exceed expectations); however, the web application functions properly, and the box is secure and looks discrete.	A
The box does not open every time (margin of error does not exceed expectations) and the web application is buggy or does not communicate with the microcontroller effectively. Meanwhile, the box is secure and looks discrete.	A-
The box does not open every time (margin of error does not exceed expectations), the web application is buggy or does not communicate with the microcontroller effectively and the box stands out/does not appear as an everyday object.	B+
If the knock sequence mechanism works as intended half the time or less, there is a slightly buggy application, and/or the box stands out/does not appear as an everyday object.	B/B-
If the project has major deficiencies that hinder the functionality of the box and/or web application; at least one moderately functioning system	C+/C/C-
If the project has only a few semi-functioning systems, but not anything proficient or efficient; no fully functioning systems	D+/D/D-
If the project fails at all intended aspects	F

Figure 4143 - Original Proposal Grading Scheme

Costs

The spreadsheet from our parts list is seen in Appendix A. Overall, it cost $477.35 to make the Tap Box. This number could have been significantly higher had we not already owned many parts, such as breadboards for preliminary testing, resistors, capacitors, etc. In order to decrease costs under mass production conditions, the 3D printed components and wooden pieces for the physical box could potentially be made via automated equipment. Furthermore, when purchasing parts from online vendors such as DigiKey and Mouser, there is a considerable discount when parts are ordered in bulk.

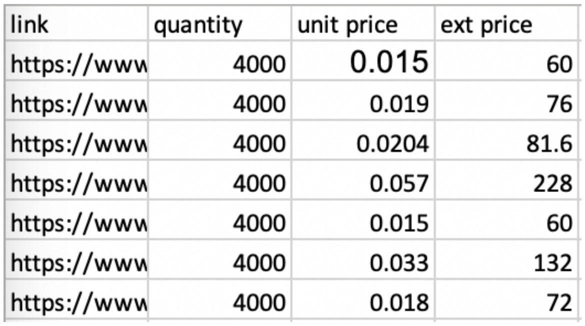
In the below figure, there is an outline of all the components that were used for the PCB design



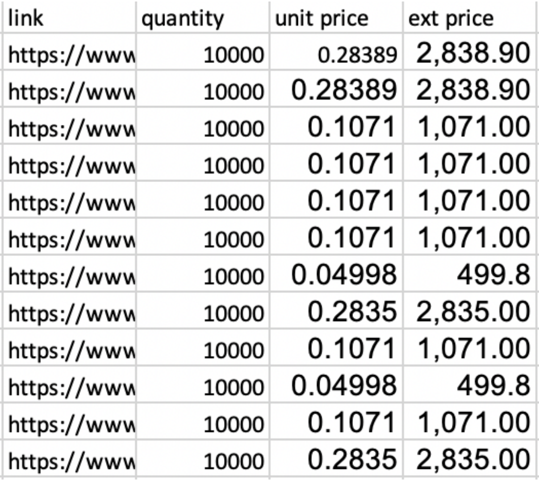
Item	Qty	Reference(s)	Value	Manu Part #	Manu
1	6	C1, C2, C5, C6, C10, C18	10,000pF	885012206065	Würth Elektronik
2	2	C3, C7	22,000pF	885012206067	Würth Elektronik
3	2	C4, C8	100,000pF	885012206071	Würth Elektronik
4	3	C9, C11, C16	330,000pF	885012206074	Würth Elektronik
5	2	C12, C13	10uF	885012106006	Würth Elektronik
6	2	C14, C15	1uF	885012106022	Würth Elektronik
7	1	C17	2.2uF	885012106018	Würth Elektronik
8	2	R1, R5	215k	TNPW0603 215K	Vishay
9	4	R2, R6, R12, R17	237k	TNPW0603 237K	Vishay
10	4	R3, R4, R7, R8	5.62k	TNPW0603 5K62	Vishay
11	7	R9, R16, R21, R22, R23, R30, R31	100k	TNPW0603 100K	Vishay
12	1	R10	42.2k	TNPW0603 42K2	Vishay
13	6	R11, R14, R15, R19, R20, R27	1k	TNPW0603 1K0	Vishay
14	2	R13, R18	4.22k	TNPW0603 4K22	Vishay
15	1	R26	619	TNPW0603 619R	Vishay
16	1	R28	14.7k	TNPW0603 14K7	Vishay
17	1	R29	34.8k	TNPW0603 34K8	Vishay
18	1	R32	10k	TNPW0603 10K	Vishay
19	1	R33	348	TNPW0603 348R	Vishay
20	2	U1, U2	TLV9004QDRQ1	TLV9004QDRQ1CT-ND	TI
21	1	U3	TLV9001IDBVR	TLV9001IDBVRCT-ND	TI
22	1	U4	AB47S	1835-1218-ND	COMUS INTERNATIONAL
23	1	UBat1	BQ25172DSGR	296-BQ25172DSGRCT-ND	TI

Figure 4244 - Bill of Materials for PCB

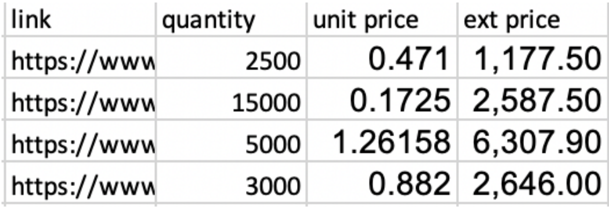
The following are the unit prices for all the components in the above order when ordered from online vendors.



link	quantity	unit price	ext price
https://www	4000	0.015	60
https://www	4000	0.019	76
https://www	4000	0.0204	81.6
https://www	4000	0.057	228
https://www	4000	0.015	60
https://www	4000	0.033	132
https://www	4000	0.018	72



link	quantity	unit price	ext price
https://www	10000	0.28389	2,838.90
https://www	10000	0.28389	2,838.90
https://www	10000	0.1071	1,071.00
https://www	10000	0.1071	1,071.00
https://www	10000	0.1071	1,071.00
https://www	10000	0.1071	1,071.00
https://www	10000	0.04998	499.8
https://www	10000	0.2835	2,835.00
https://www	10000	0.1071	1,071.00
https://www	10000	0.04998	499.8
https://www	10000	0.1071	1,071.00
https://www	10000	0.2835	2,835.00



link	quantity	unit price	ext price
https://www	2500	0.471	1,177.50
https://www	15000	0.1725	2,587.50
https://www	5000	1.26158	6,307.90
https://www	3000	0.882	2,646.00

Figure 43 – 45 Cost per unit of Parts

If the Tap Box is to be mass produced for 10,000 units, then to make one of those units would cost $204.48. This is half of the cost of making one as we did for the Capstone project.

Future Work

In the future, many additional features can be implemented for this project from both the software and hardware sides. Before this, however, the core functionality of the system would be prioritized, as minor changes to the PCB and Wi-Fi module would be mandatory for a more robust design.

Many features could be added to the web application, including displaying the current password for the user in case they forgot, a cleaner user interface, a log of timestamps when the box was unlocked, and timestamps of times when the piezoelectric sensors were triggered, but the box wasn’t unlocked, indicating some form of tampering. Software-wise, we would also add multiple passwords functionality, so that many users could access the box with their own passwords. Cybersecurity concerning the rise of IoT would also be considered.

Some hardware changes would also be implemented given further time and resources. A higher level of security could be achieved by adding more piezoelectric sensors to other sides of the box, making a more difficult problem for a potential hacker to crack. The physical locking mechanism would also be bolstered with a redesign and perhaps a stronger servo motor or other mechanical device.

References

[1] “MSP432P401R,” *Mixed-Signal Microcontrollers data sheet* | TI.com. [Online]. Available: https://www.ti.com/lit/ds/slas826e/slas826e.pdf [Accessed: 13-Dec-2022].

[2] “VirtualBench,” National Instruments. https://www.ni.com/enus/shop/hardware/products/virtualbench-all-in-one-instrument.html (accessed Dec. 08, 2020).

[3] “InfluxDB Cloud and MQTT,” Influxdata. [Online]. Available: https://www.influxdata.com/influxdb-cloud-and-MQTT/?utm_source=google&utm_medium=cpc&utm_campaign=2021-06-10_IIOT_MQTT_NA&utm_term=MQTT %20broker&gclid=Cj0KCQiA4uCcBhDdARIsAH5jyUm6WjZPEhKHn0ZBrEjl15SK1zeF62kv5ImyA0Bbb-X2Ye8NOZ5I3vAaAlhEEALw_wcB [Accessed: 13- Dec-2022].

[4] “CC3120,” *CC3120 [4]SimpleLink™ Wi-Fi® Network Processor, Internet-of-Things Solution for MCU Applications*. [Online]. Available: https://www.ti.com/lit/ds/symlink/cc3120.pdf?ts=1670977072025 [Accessed: 13- Dec-2022].

[5] “TLV9004 [5]QDRQ1,” DigiKey. [Online]. Available: https://www.digikey.com/en/products/detail/texas-instruments/TLV9004 [5]QDRQ1/13174037 [Accessed: 13-Dec-2022].

[6] “2820,” DigiKey. [Online]. Available: https://www.digikey.com/en/products/detail/pololu-corporation/2820/10450037?utm_adgroup=Motors%20-%20AC%2C%20DC&utm_source=google&utm_medium=cpc&utm_campaign=Shopping_Product_Motors%2C%20Solenoids%2C%20Driver%20Boards%2FModules_NEW&utm_term=&utm_content=Motors%20-%20AC%2C%20DC&gclid=Cj0KCQjwkt6aBhDKARIsAAyeLJ1Xw6kEdJrXOx_U8nD4f2HCzv4TFT2vu6daeB8M8Xuyx0soYysTFw0aAtZlEALw_wcB. [Accessed: 13-Dec-2022].

[7]“Download PuTTY,” Putty. [Online]. Available: https://www.Putty.org/ [Accessed: 13-Dec-2022].

[8] “AB47S,” DigiKey. [Online]. Available: https://www.digikey.com/en/products/detail/comus-international/AB47S/9817865 [Accessed: 13-Dec-2022].

[9] “BA33BC0T,” DigiKey. [Online]. Available: https://www.digikey.com/en/products/detail/rohm-semiconductor/BA33BC0T/722279 [Accessed: 13-Dec-2022].

[10] “BQ25172DSGR,” DigiKey. [Online]. Available: https://www.digikey.com/en/products/detail/texas-instruments/BQ25172DSGR/16585660?s=N4IgjCBcoCxgTFUBjKAzAhgGwM4FMAaEAeygG0QBmADgAZaB2CAXSIAcAXKEAZQ4CcAlgDsA5iAC%2BRMAE4mSEKkiZchEuRAwArHIaUQrEJ258hYyRIlA [Accessed: 13-Dec-2022].

[11] “PS1240P02BT,” DigiKey. [Online]. Available: https://www.digikey.com/en/products/detail/tdk-corporation/PS1240P02BT/935924 [Accessed: 13-Dec-2022].

[12] “Django download,” djangoproject. [Online]. Available: https://www.djangoproject.com/download/. [Accessed: 13- Dec-2022].

[13] https://www.python.org/

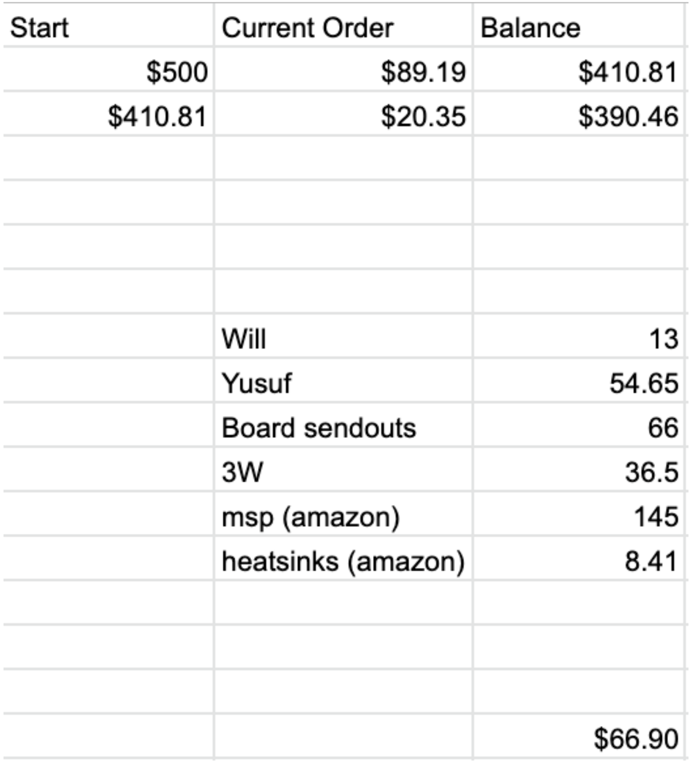
[14] https://developer.mozilla.org/en-US/docs/Web/CSS

[15] https://www.w3schools.com/html/

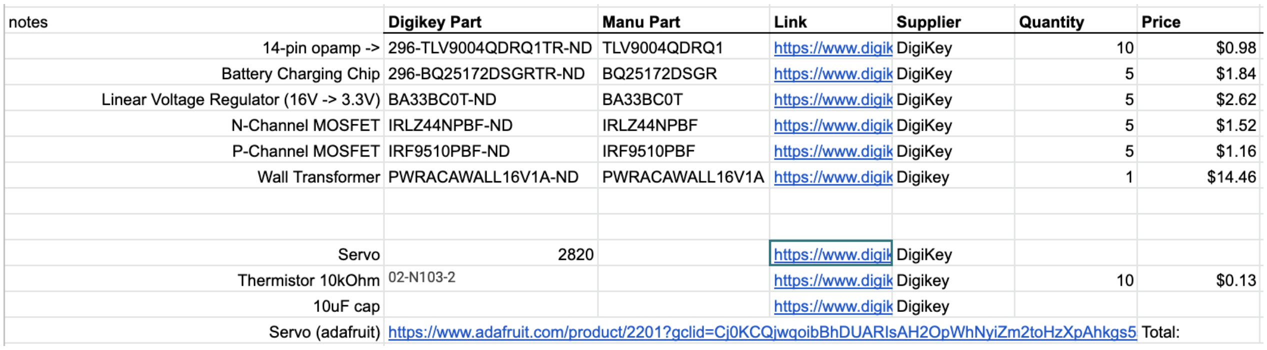
[16] https://www.w3schools.com/js/

Appendix

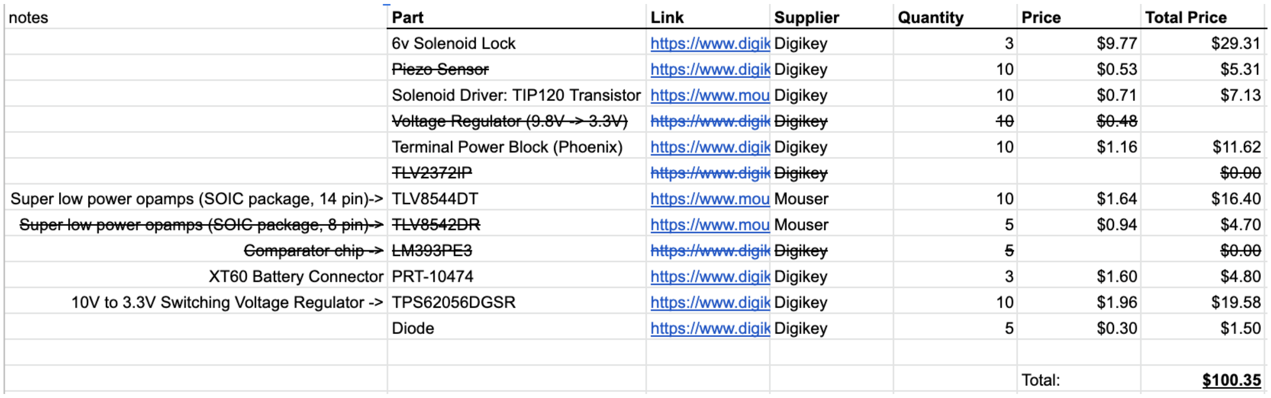
Appendix A: Budget



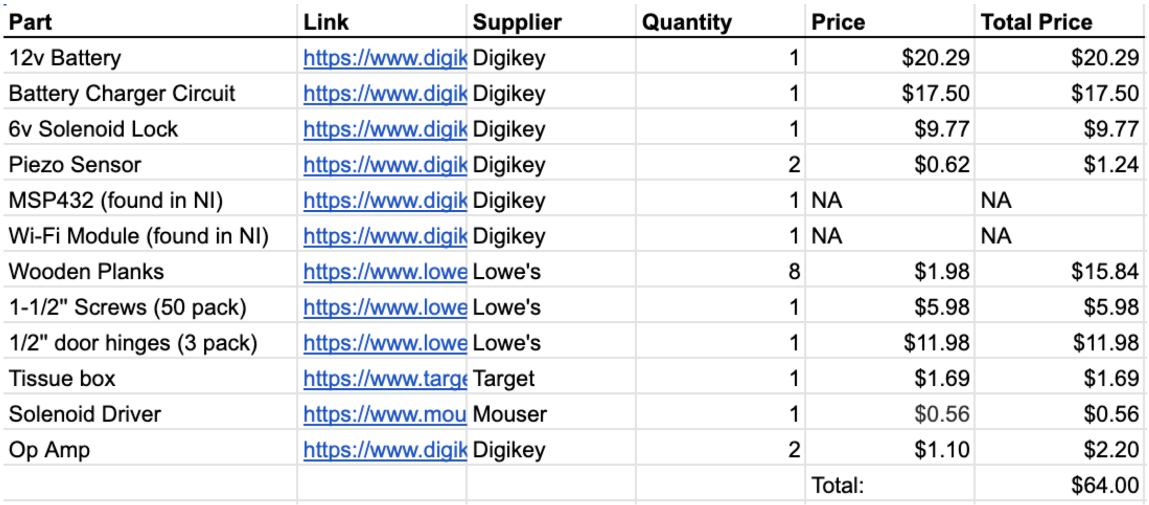
Start	Current Order	Balance
$500	$89.19	$410.81
$410.81	$20.35	$390.46
	Will	13
	Yusuf	54.65
	Board sendouts	66
	3W	36.5
	msp (amazon)	145
	heatsinks (amazon)	8.41
		$66.90



notes	Digikey Part	Manu Part	Link	Supplier	Quantity	Price
14-pin opamp ->	296-TLV9004QDRQ1TR-ND	TLV9004QDRQ1	https://www.digik	DigiKey	10	$0.98
Battery Charging Chip	296-BQ25172DSGRTR-ND	BQ25172DSGR	https://www.digik	DigiKey	5	$1.84
Linear Voltage Regulator (16V -> 3.3V)	BA33BC0T-ND	BA33BC0T	https://www.digik	DigiKey	5	$2.62
N-Channel MOSFET	IRLZ44NPBF-ND	IRLZ44NPBF	https://www.digik	DigiKey	5	$1.52
P-Channel MOSFET	IRF9510PBF-ND	IRF9510PBF	https://www.digik	DigiKey	5	$1.16
Wall Transformer	PWRACAWALL16V1A-ND	PWRACAWALL16V1A	https://www.digik	Digikey	1	$14.46
Servo	2820		https://www.digik	DigiKey		
Thermistor 10kOhm	02-N103-2		https://www.digik	Digikey	10	$0.13
10uF cap			https://www.digik	Digikey		
Servo (adafruit)	https://www.adafruit.com/product/2201?gclid=Cj0KCQjwqoibBhDUARIsAH2OpWhNyiZm2toHzXpAhkgs5					Total:



notes	Part	Link	Supplier	Quantity	Price	Total Price
	6v Solenoid Lock	https://www.digik	Digikey	3	$9.77	$29.31
	~~Piezo Sensor~~	https://www.digik	Digikey	10	$0.53	$5.31
	Solenoid Driver: TIP120 Transistor	https://www.mou	Digikey	10	$0.71	$7.13
	~~Voltage Regulator (9.8V -> 3.3V)~~	~~https://www.digik~~	~~Digikey~~	~~10~~	~~$0.48~~	
	Terminal Power Block (Phoenix)	https://www.digik	Digikey	10	$1.16	$11.62
	~~TLV2372IP~~	~~https://www.digik~~	~~Digikey~~			~~$0.00~~
Super low power opamps (SOIC package, 14 pin)->	TLV8544DT	https://www.mou	Mouser	10	$1.64	$16.40
~~Super low power opamps (SOIC package, 8 pin)->~~	~~TLV8542DR~~	https://www.mou	Mouser	5	$0.94	$4.70
~~Comparator chip ->~~	~~LM393PE3~~	~~https://www.digik~~	~~Digikey~~	~~5~~		~~$0.00~~
XT60 Battery Connector	PRT-10474	https://www.digik	Digikey	3	$1.60	$4.80
10V to 3.3V Switching Voltage Regulator ->	TPS62056DGSR	https://www.digik	Digikey	10	$1.96	$19.58
	Diode	https://www.digik	Digikey	5	$0.30	$1.50
					Total:	**$100.35**



Part	Link	Supplier	Quantity	Price	Total Price
12v Battery	https://www.digik	Digikey	1	$20.29	$20.29
Battery Charger Circuit	https://www.digik	Digikey	1	$17.50	$17.50
6v Solenoid Lock	https://www.digik	Digikey	1	$9.77	$9.77
Piezo Sensor	https://www.digik	Digikey	2	$0.62	$1.24
MSP432 (found in NI)	https://www.digik	Digikey	1	NA	NA
Wi-Fi Module (found in NI)	https://www.digik	Digikey	1	NA	NA
Wooden Planks	https://www.lowe	Lowe's	8	$1.98	$15.84
1-1/2" Screws (50 pack)	https://www.lowe	Lowe's	1	$5.98	$5.98
1/2" door hinges (3 pack)	https://www.lowe	Lowe's	1	$11.98	$11.98
Tissue box	https://www.targe	Target	1	$1.69	$1.69
Solenoid Driver	https://www.mou	Mouser	1	$0.56	$0.56
Op Amp	https://www.digik	Digikey	2	$1.10	$2.20
				Total:	$64.00