ECE 445, Senior Design

Team 8:
Ernesto O Marquez
Max Armbruster
Samarth Jain

TA: Yichi Zhang

Secure Smart Locker for Doorstep Delivery

Fall 2020

1. Introduction

1.1 Problem and Solution Overview:

Package theft is a real issue that several of our group's members have dealt with personally. You might receive a delivery notification from USPS or Amazon only to find that the box has disappeared within the hour it took for you to retrieve it. This is especially frustrating because there isn't much that delivery service providers can do about the issue if you didn't purchase insurance.

Our solution is to design a secure smart locker to be placed on one's doorstep. Our app will enable the owner to create a new, completely random numeric code that he or she can share with the delivery service. The delivery person can then key in that code on a numpad, after which the lock will open up. Once the package is placed inside and the door is closed, the locker will use the owner's WiFi to notify him/her of the delivery. The owner will interact with the locker using a mobile app that we develop. The locker will also function in the reverse direction, allowing one to stage packages for pickup.

The current global pandemic has moved many people's shopping habits online as e-tailing is much safer than retailing. Digital Commerce 360 put together a good analysis of data released by the US Department of Commerce, which indicates that online spending over the first 6 months of 2020 is up over 30% from the same period in 2019 [1]. The result is many more packages sitting idle on doorsteps, vulnerable to package thieves. Your best bet is to order to something like an Amazon Locker or P.O. Box, but those add a layer of inconvenience to "home" delivery and are essentially just as COVID-risky as retail is. Our solution intends to keep home delivery easy and safe.

1.3 High-level Requirements List:

- Our locker should use a number pad to take a passcode input from the delivery driver, unlocking temporarily when the correct passcode is given. The passcode will be randomized, and will last only as long as the delivery is "in progress". Once the delivery code is no longer needed, the box will not open with that code in the future.
- Our locker should use a speaker to give an audible indicator when the door is left open, and also double as an alarm when the locker is unplugged from the wall or when the motion sensor detects malicious activity. The alarm should trigger within 3 seconds of the malicious activity taking place to give the owner adequate time to hear and respond to it.
- Our solution should include a mobile app that allows the owner to remotely unlock the container and receive delivery/pickup notifications. The notification should arrive no more than 10 minutes after the delivery/pickup in order to give the owner accurate information regarding their package's status. Remote unlocking should take no more than 5 seconds in order to keep this feature convenient.

2. Design

2.1 Block Diagram:

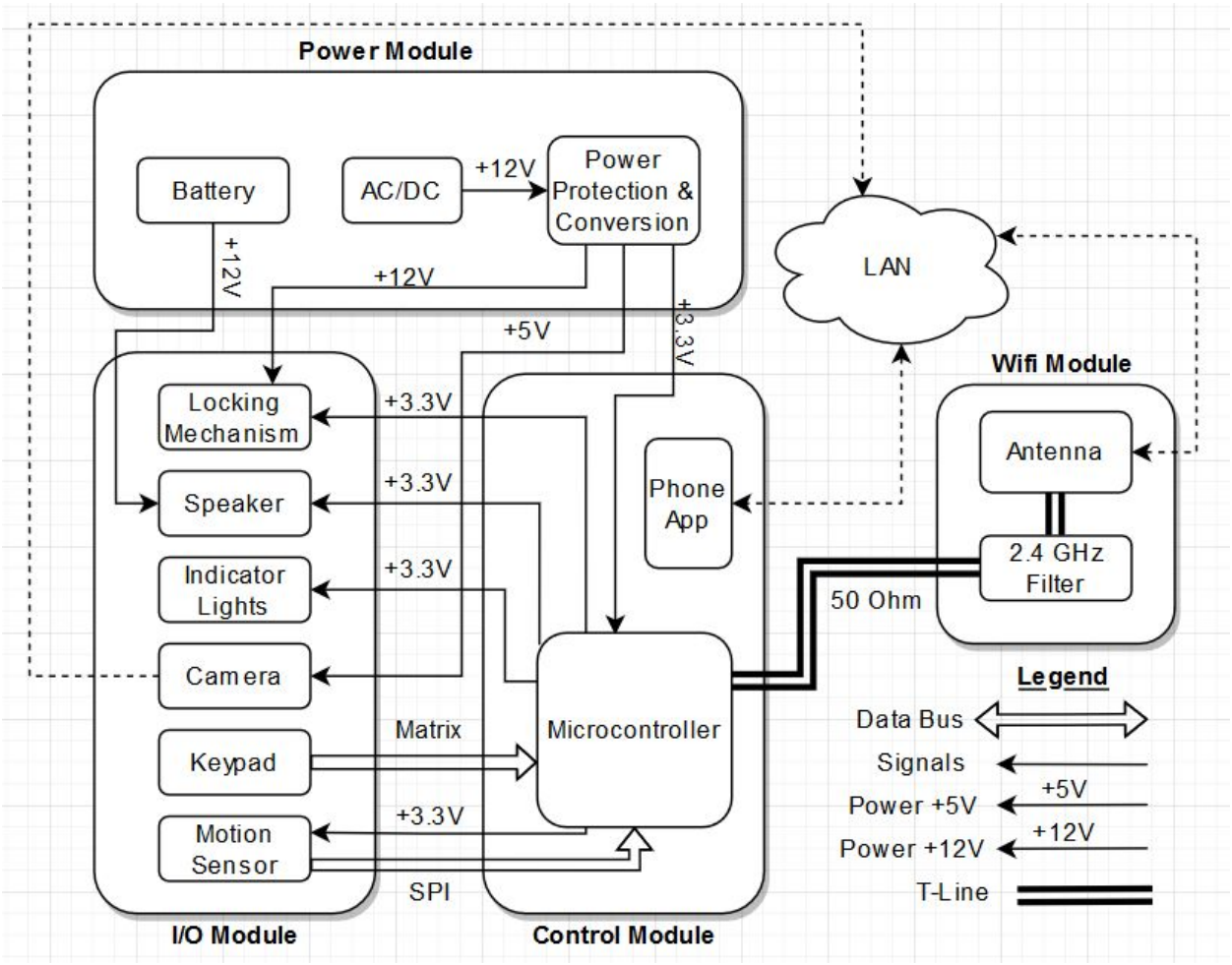


Figure 1: Block diagram of our project

Figure 1 shows our system design diagram that will contain four major modules: the Wifi module, the Power module, the I/O Module and the Control module.

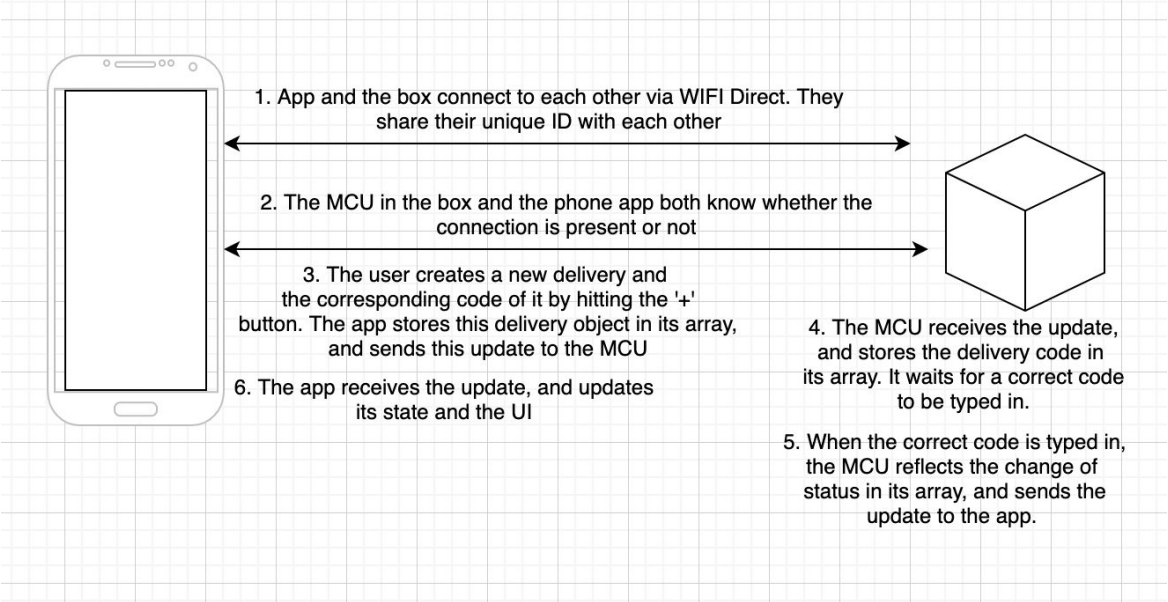


Figure 2: visualization of the handshake and the communication between the device and the app

2.2 Physical Design:

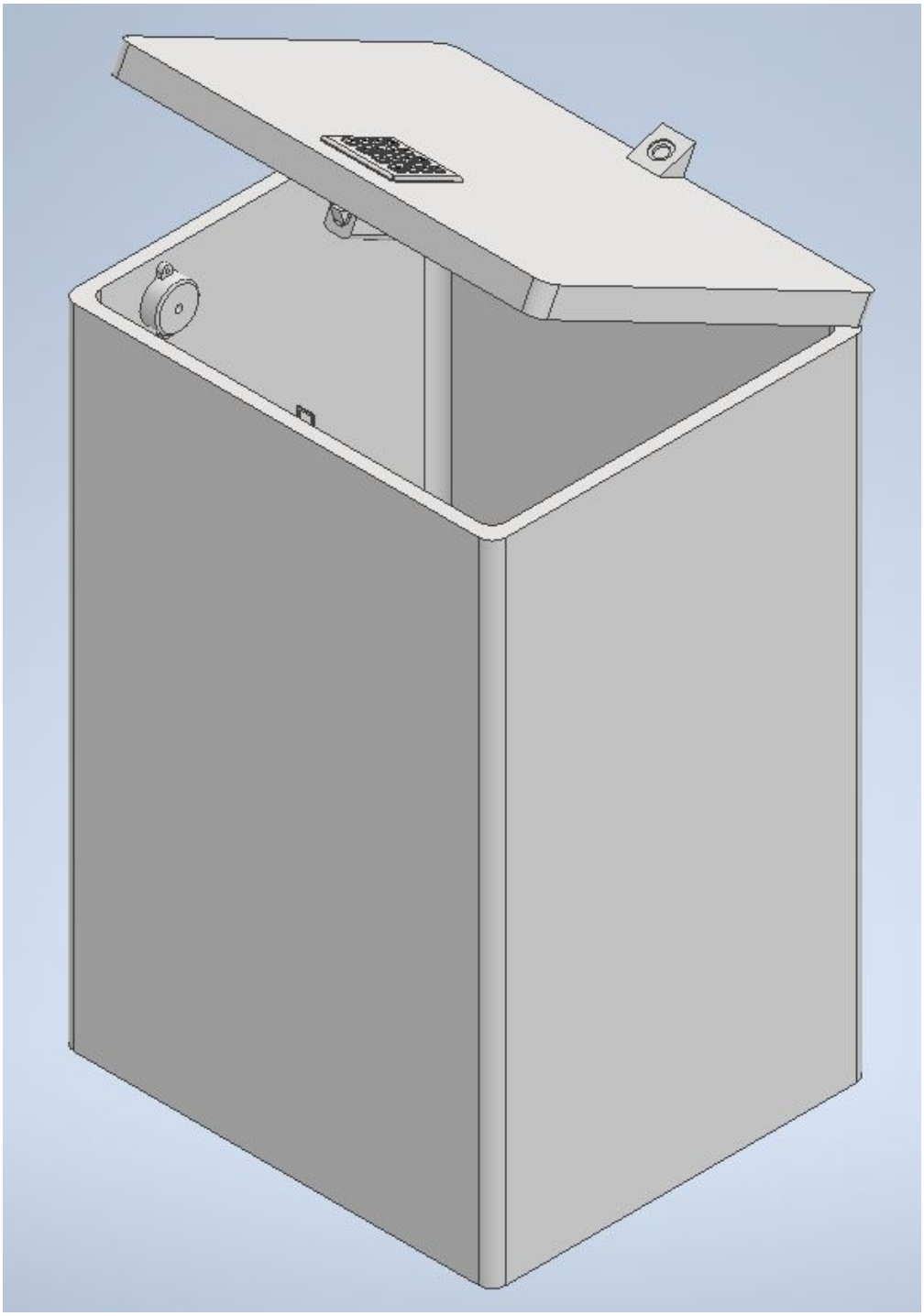


Figure 3: Isometric view of our Smart Locker. The speaker can be seen mounted to the inside of the locker (front-left side).

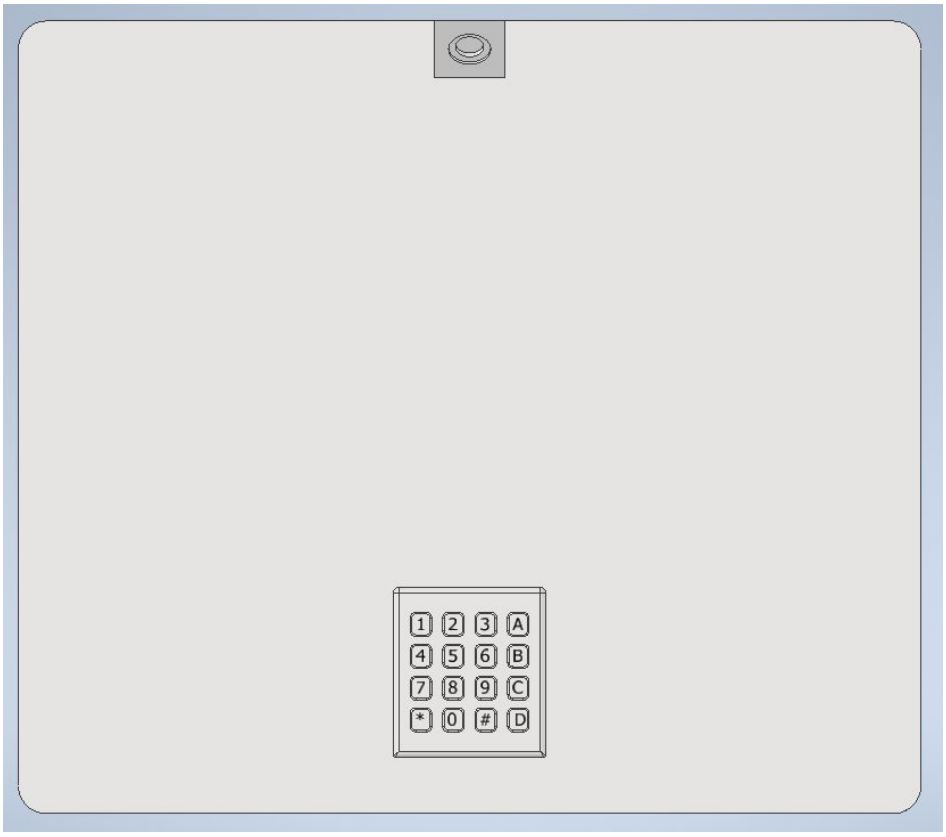


Figure 4: Top view of our Locker's lid. The camera mount can be seen near the top while the keypad is mounted near the bottom.

Our locker will have a rectangular footprint sized roughly 14"x16", and will be 2-3 feet tall. We want the longest dimension of our locker to be height, so as to take up the least amount of space on one's doorstep. Both the number pad and the camera will be mounted to the locker's exterior, while the electronics will be housed in a compartment inside the locker, with the speaker mounted to the inside wall of the locker. This container will be at least IP 65 rated, protecting the electronics from low-pressure water jets and dust, so as to withstand the elements.

2.3 Subsystems

2.2.1 Wifi Module

- Antenna: We will need an Antenna for the Wifi Module. The antenna will be used as a transceiver for the communication between our microcontroller and the owner's home Wifi, allowing our mobile app to interact with the locker. We will use the antenna recommended to us in our MCU's datasheet.

- 2.4GHz Filter: We require a filter to isolate the 2.4GHz Wifi signal from the antenna. We will use the filter recommended to us in our MCU's datasheet. To connect the antenna to the RF input on our MCU we will need to design a microstrip transmission line with a 50-Ohm impedance at 2.4GHz.

Requirement	Verification
1. Antenna will be needed for MCU to communicate wirelessly 2. Filter will be needed for MCU to filter unwanted frequencies for proper reading through antenna	1. Follow MCU datasheet for connection instructions with AH316M245001-T component 2. Follow MCU datasheet for connection instructions with DEA202450BT-1294C1-H component

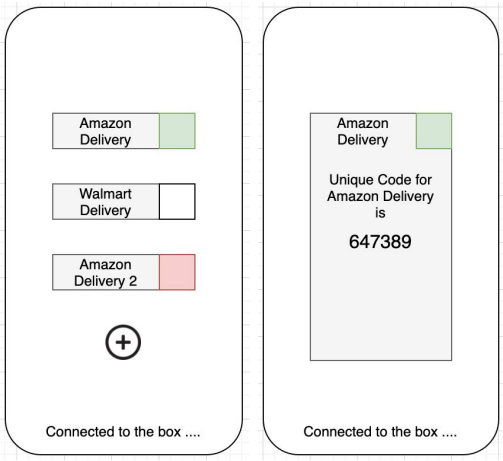
2.2.2 Control Module

- Microcontroller: The microcontroller is responsible for reading sensor inputs (and user input over Wifi), interpreting those inputs, then controlling certain outputs like the locking mechanism, lights, and speaker. The microcontroller will also send data over Wifi to the mobile app, such as the camera feed and delivery/pickup notifications. We will use the TI CC3200, which supports all of the communication we plan to use for our peripherals.

Requirement	Verification
1. The MCU connects to the app via Wifi. 2. The MCU sends a signal to the lock only if a correct code is typed into the numpad. Otherwise, the LEDs flash. 3. The MCU triggers an "alarm", and causes the speaker to beep loudly, and the LEDs to turn blue.	1. A unique ID of the particular product's microcontroller is sent to the app, and a unique ID of the particular mobile app is sent to the microcontroller. Our program checks if the handshake happened properly, in both the places. 2. Given the pending deliveries in the memory of the MCU, if a correct code is typed in, the MCU sends a signal to the solenoid lock through its GPIO pin. Otherwise, it sends a signal to the

	LEDs to flash. 3. Assuming the rest of the system works, the box can be abruptly shaken so that the accelerometer can read values greater than our set threshold. Then, the MCU understands the situation as an “alarm”, and tells the speaker to beep loudly and the LEDs to flash blue.

- Android App: The app is the interface for the owner of the product to communicate with the smart-box. Figures 5 and 6 give a visual representation of the app.



Figures 5 and 6: A barebones visualization of the app.
Left (5): the main page, with different deliveries that are open.
Right (6): when a particular delivery is clicked, the unique information for that delivery can be seen.

Requirement	Verification
1. The app connects to the microcontroller via WiFi. 2. The app is able to add a new delivery, create a random 6 digit code for it, and give it a name. 3. The app sends the correct status data of the different deliveries. Every delivery can be thought of as a struct with two bytes for the unique delivery code, and one byte for the status of the delivery.	1. The verification is similar to that of the verification (1.) above for the microcontroller. 2. The ‘+’ button on the app creates a new delivery card, allows the user to enter the name of the delivery, and creates a pseudo-random 6 digit code that is different from any other delivery’s code. 3. The microcontroller correctly receives the data from the app. It should be

4. A change in delivery status by the MCU inside the box is reflected by the app.	verified if the data in the MCU reflects the data in the app. 4. The MCU transmits the following updates to its own state, and the state of the app: ➢ Incomplete: when the smart-box is expecting the delivery to happen. ➢ In progress: when a correct code is typed in and the lock is open. ➢ Complete: when the lock is closed, and the delivery is complete.

2.2.3 IO Module

- Numpad: Figure 7 shows our numpad schematic. The numpad will read the numeric code passed to it by the delivery person. It will transfer the information to the microcontroller to allow access to the locker door. We will use a cheap membrane keypad with a matrix output, allowing us to read the number using eight GPIO pins.

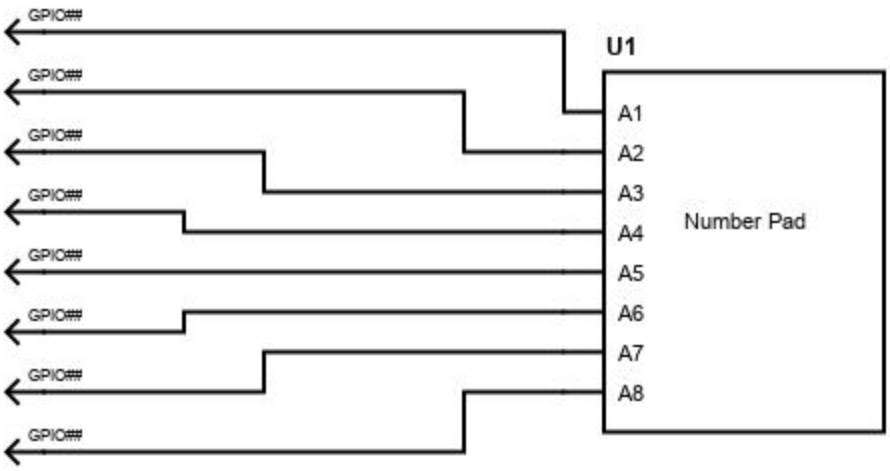


Figure 7: Schematic of the matrix GPIO output from our number pad

Requirement	Verification
1. The numpad accepts the code typed in, and sends the data to the MCU to	1. The data sent by the numpad is correctly read by the MCU if the

process.	correct GPIO connections create the correct 6 digit code in the MCU.

- Camera: Figure 8 shows the circuit schematic of our Wifi camera. The camera will be used for two purposes in security. First, the camera will be used to take pictures of an individual when the motion sensor detects unusual activity. The camera will also take a picture whenever the locker is opened, and that picture will be included in a notification sent to the owner through the mobile app. We will use a Wifi camera to send image data directly to our app for processing, which reduces the complexity of our MCU's Wifi transmission.

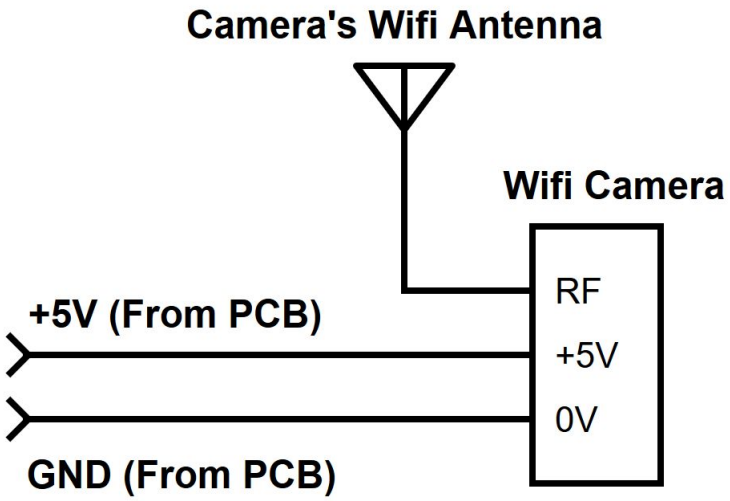


Figure 8: Schematic of our Wifi Camera, with the included antenna (separate from the antenna in our Wifi module)

Requirement	Verification
1. Need to have an antenna that can reach high frequency (2.4Ghz or 5Ghz) for internet connection and Wifi Protocols 2. It will need to have at least 720p quality 3. It will need to be powered from the Microcontroller with +5V	1. Connect to Local Wifi. 2. Test Output feed open network stream on the VLC media player app. 3. Use the multimeter and the oscilloscope to measure whether the voltage and the current going to the camera are appropriate.

- Motion Sensor: The motion sensor is to be used for security. If there is motion that is deemed *unnatural* by our system, then it will turn on the alarm and inform the owner of potential criminal activity. It sends its data to the microcontroller which will interpret the

detected activity. To accomplish this we will use an IMU with a gyroscope and accelerometer. Figure 9 shows the typical use schematic of our IMU.

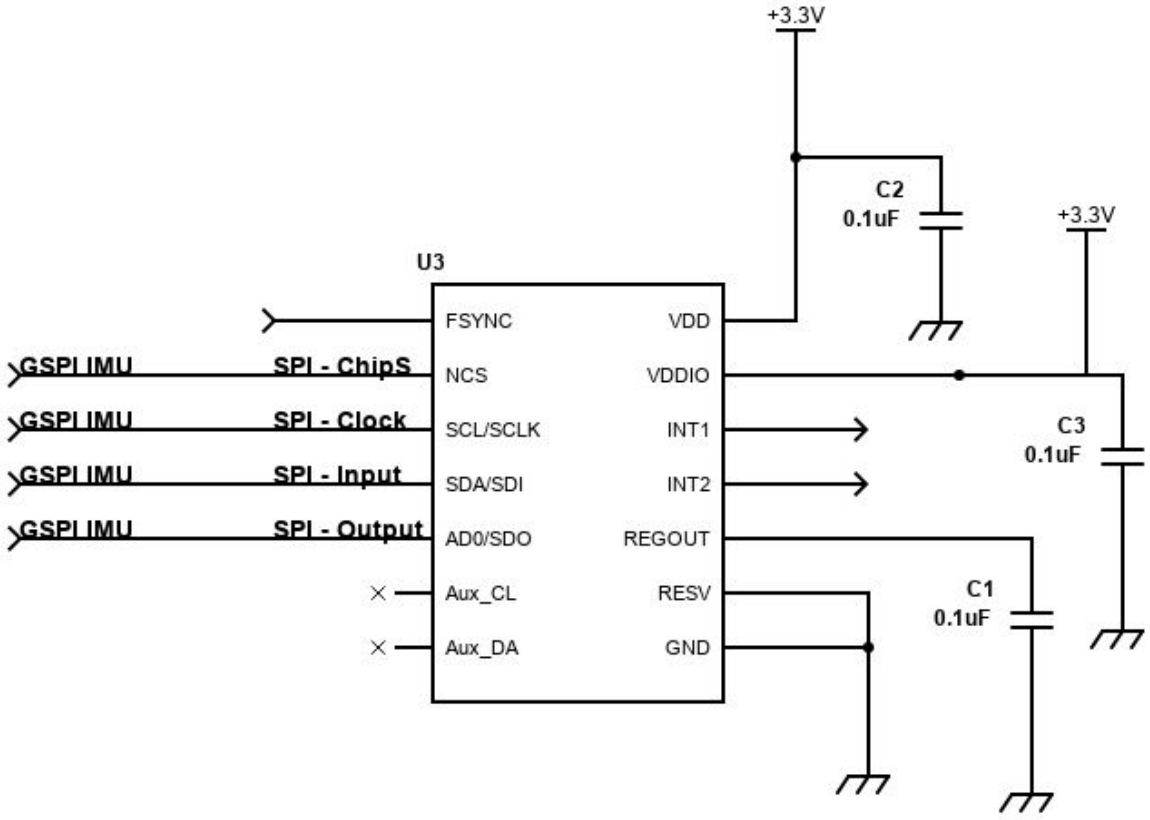


Figure 9: Typical use of our IMU

Requirement	Verification
1. The accelerometer sensor sends data in X, Y and Z axes to the MCU. The MCU checks if any of the three values pass a certain threshold.	1. The box is initially set to trigger the alarm with a low threshold for the values of the accelerometer (say ±2g). Once the alarm is triggered by the MCU and the speaker starts beeping, we will know that the accelerometer readings are being accepted by the MCU. Then, different values of the threshold will be tested to check what the ideal threshold value for the sensor should be. We will try to optimize these values so that they are sensitive to *unnatural* shaking of the box, but not so sensitive so as to cause false positives.

- Speaker: Figure 10 shows the layout of our speaker circuit. The speaker is internally driven, and can be powered with voltages from 3~24V. In the case that the power input to our locker is disconnected, the relay will connect the speaker to our 12V battery, which will cause the speaker to sound at 105dB, perfect for an alarm. If instead our board has power, the relay will connect it to an output called Door_Open coming from our MCU. This enables the MCU to drive it at 3.3V when the door is opened, sounding the speaker at a much quieter volume. We will use a 12V Piezo speaker with 2.8kHz frequency, which will sound much like what one would expect from an alarm.

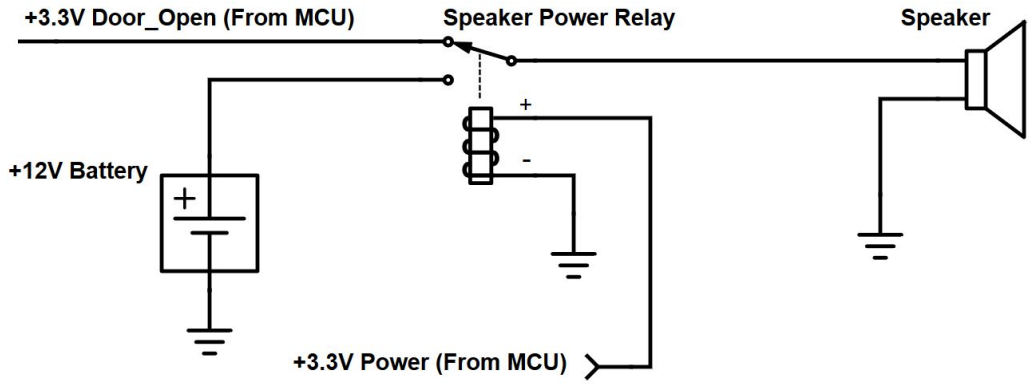


Figure 10: Circuit schematic of our speaker setup

Requirement	Verification
1. The speaker correctly produces a sound to indicate that the box is open. 2. The speaker correctly produces a much louder sound when the box is disconnected from the wall outlet.	1. A correct code is typed in and, assuming that the rest of the system works, the speaker produces a quieter sound to indicate to the delivery person that the box is open. 2. The power connection to the wall is taken out and, assuming that the rest of the system works, the speaker produces a much louder sound to indicate that the box is being tampered with.

- Indicator Lights: The indicator lights will be used as an indication for when the door of the box is locked or unlocked. The green light indicates the locker is ready to open. The red light indicates the box is locked. The blue light will be for any mechanical problems, or for the box sensor and alarm being tripped. We will use 3.3V LEDs that can be driven by our MCU.

Requirement	Verification
1. The lights show the color green to indicate that a correct code has been typed in. 2. The lights show the color blue to indicate that a correct code has not been typed in, or if the box is being tampered with. 3. The lights show the color red to indicate that the box is locked and it expects a code to be typed in.	1. A correct code is typed in and the MCU checks whether the code is in its acceptable list of delivery codes. If yes, then the MCU passes a signal from its GPIO pin to turn that LED green. 2. The backup battery connects to the lights and triggers its color to blue. 3. We can put an acceptable code, open the box and close the box. Then, we should find that the LEDs flash red again, to reflect that the box is accepting new codes again.

- Locking Mechanism: The locking mechanism will be used to keep the container locked until it is unlocked by a keycode or the owner. We will use a 12V solenoid door latch that we will control by sending a digital output from the MCU to a relay, as shown in figure 11. Figure 12 shows a bracket mounted to the interior of the locker which the lock will latch into.

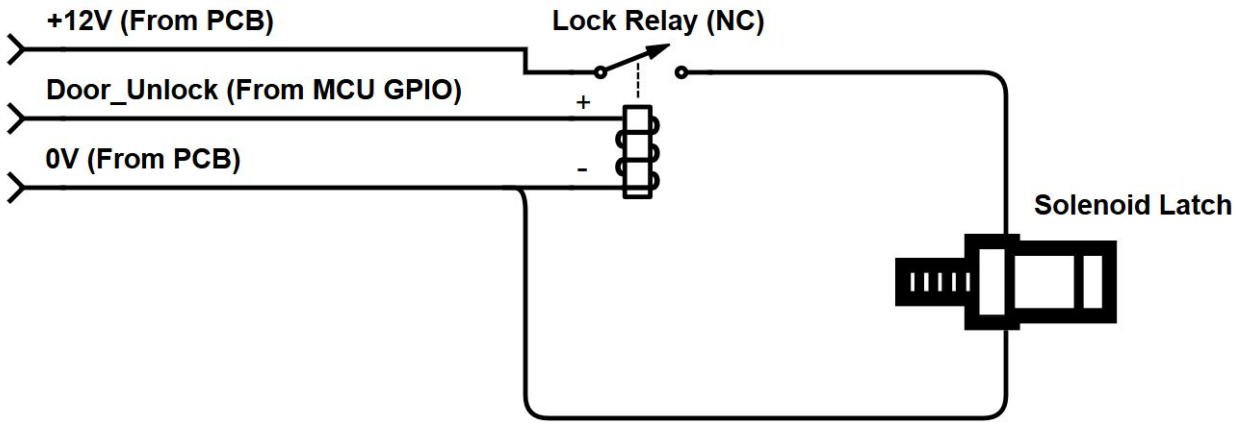


Figure 11: Circuit schematic of our locking mechanism

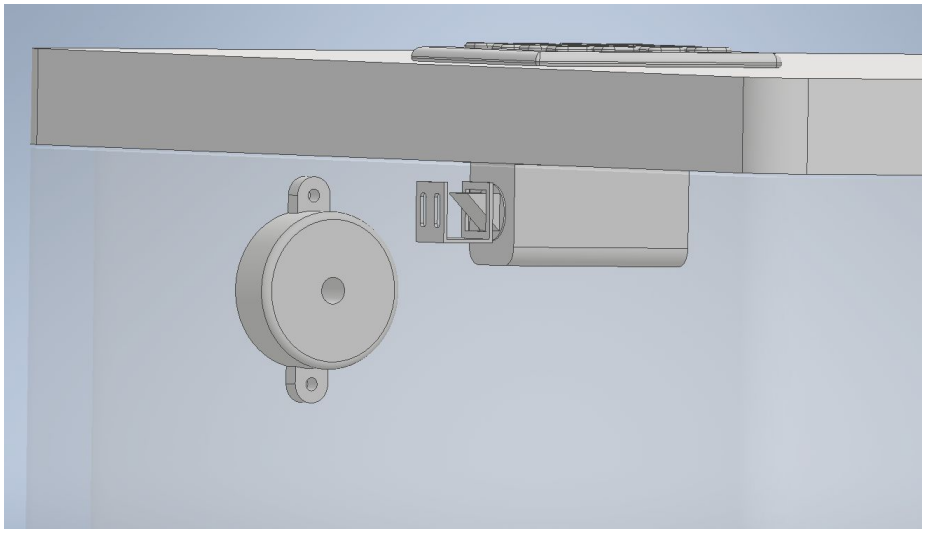


Figure 12: CAD Drawing for the locking mechanism, and attachment within the box for security reasons.

Requirement	Verification
1. It will need to be connected with a relay that toggles between ground and the power supply from a signal of the MCU.	1. Verification for requirement 2: a. Connect positive with lock relay and negative with ground. b. Connect +12 V to lock relay and signal from MCU. c. Send a small signal to the lock relay to trigger the latch to close. It should pull the locking mechanism.

2.2.4 Power Module

- Battery: We will use an external 12V rechargeable Li-ion battery. It is separate from the wall AC/DC power and will power the alarm when the wall power is disconnected. We can also use it to power other I/O devices if necessary. The battery should be accessible from the locker interior so the owner can easily remove it if they plan to unplug the locker themselves and don't want the alarm to sound.

Requirement	Verification
1. Must be connected with the Speaker 2. It will need to be stored away from the components to reduce heating.	1. Connect the positive to the positive of speaker relay and negative to ground. 2. Attachment is at the bottom of the box

- AC/DC: We will use a simple 120VAC/12VDC power adapter that we plug into the wall. This adapter will have a barrel connected to supply our PCB with +12V.

Requirement	Verification
1. Should provide constant +12V voltage output with less than 300mV of fluctuation when sampled over a minute	1. We will connect the output of the power supply to an oscilloscope and monitor the voltage fluctuation.

- Power Protection and Conversion: This circuitry will be placed on the microcontroller PCB and will be used to protect from overvoltage, overcurrent, or reverse voltage inputs. The output of the power protector will then convert our 12V input to any other required power (5V, 3.3V, etc.).

Requirement	Verification
1. Must accept a voltage input of 12V +/- 300mV 2. Must protect against overvoltage of 13V+ 3. Must protect against reverse voltage inputs of -1V or less. 4. Must protect against input currents of 5+ Amps	For each step below we will connect our input to a variable power supply and do the following, replacing the fuse each time it trips: 1. Sweep the voltage while maintaining a small current to ensure that the SCR and fuse do not trip due to these voltage inputs. 2. Continue (1) to ensure that the SCR and fuse trip before reaching 13V. 3. Sweep the voltage while our leads are connected in reverse, ensuring that the SCR and fuse trip before reaching -1V. 4. Set the voltage to 12V and slowly increase the current to ensure that the fuse trips before reaching 5A.

2.4 Tolerance Analysis:

The most critical part of the project is the security and prevention of theft of the packages as well as the smart box. Thus, the tolerance analysis of our project depends on the two most important parts:

Security of the packages:
It is critically important to have a completely new and pseudo-random security code for each delivery for the purpose of accountability. If the same code is used for multiple incomplete deliveries, then it will be impossible to put the responsibility of one package delivery on one delivery service, and the smart box will not be very secure.

The app will therefore generate a pseudo random 6 digit code, and check if there already exists an “incomplete” delivery with the same code. A six digit code can serve the purpose well, as a range of 10^6 possible combinations gives the code less chance of being duplicated, but at the time is short enough for a delivery person to type it in fast. It will be ensured that the code cannot be an *unsafe* combination, such as “000000” or “123456”.

The microcontroller will need to be in constant communication with the app to ensure that both the smart box and the app are on the same page with regards to the status of the different deliveries. The microcontroller will obviously not send a signal to the solenoid lock of the box to open if a code is typed in that does not belong to any of the incomplete deliveries.

Security of the Smart-Box:
To ensure that the Smart-Box does not get stolen, we will install a few components to prevent its theft or damage, namely a camera, indicator lights, a speaker, and a MEMS accelerometer sensor.

The accelerometer, ICM-20649 from Invensense, will be used to track the orientation of the box. In case the box is shaken very abruptly, or “unnaturally”, the sensor will trigger an alarm in the microcontroller, following which the indicator lights will flash, the speaker will beep, and the camera footage will be sent to the owner.

The accelerometer has a programmable range of ±4g, ±8g, ±16g, and ±30g, thus ensuring that our sensor will be sufficiently sensitive to pick up any shaking. It has three axes, X Y and Z, and it outputs the orientation data in each of them. If the box is kept stable, for example, then the sensors in directions X and Y will output 0g while the sensor in direction Z will output 1g.

To correctly ensure that the alarm rings upon unnatural motion, we will try different values for thresholds for the sensor readings. For example:

> if the sensor data for X, Y or Z < -10g or X, Y or Z > +10g, then the alarm will be triggered.

Additionally, a thief may just try to cut off the power supply to the microcontroller. In that case, our solution to the malicious attempt is to have a backup power supply in the form of a battery which will be connected to the speaker and the lights, as well as the camera system. The power supply of our system will switch to the backup power within a second of the main power cut, using a relay to do that.

Finally, we need to be careful about the size of our program, since the MCU has an SRAM of 256 KB that is used for both code as well as data, according to the datasheet of the MCU. Since a C or a C++ program could easily take 100s of KBs of memory [5], we need to be strict about our code usage. Our code must not have a huge memory footprint.

2.5 COVID-19 Contingency Planning:

In the event that our university transitions to online instruction due to a COVID-19 outbreak, our team will do our best to complete as much of the project as possible without a PCB or other machine shop services. Max will make a locker in his garage at home while Samarth and Ernesto will handle the software side of things, programming the MCU and developing the mobile app. We believe that we can use breadboards to test all of our electronics, although this would mean moving away from certain surface-mount components. If we are able to have a PCB made, Max also has soldering tools available at home to assemble it.

3. Cost

Cost of labor: $3\ Engineers * \frac{\$40}{hour} * \frac{20\ hours}{week} * 10\ weeks = \$24,000$

Cost Analysis:						
Part Description	**Manufacturer**	**Part Number**	**Vendor**	**Quantity**	**Unit Cost**	**Connection Type**
IMU	**TDK InvenSense**	**ICM-20649**	**Digi-Key**	**1**	**$8.69**	(SPI)
Number Pad	**Parallax Inc.**	**27899**	**Digi-Key**	**1**	**$6.50**	(I2C)
MCU	**Texas Instruments**	**CC3200**	**Texas Instrument**	**1**	**$4.50**	(n/a)
Speaker	**PUI Audio, Inc.**	**AI-4228-TF-LW140-3-R**	**Digi-Key**	**1**	**$4.08**	(GPIO)
Lock	**SparkFun Electronics**	**ROB-15324**	**Digi-Key**	**1**	**$9.95**	(GPIO)
Camera	**SparkFun Electronics**	**WRL-15124**	**Digi-Key**	**1**	**$43.75**	(Wireless)
Filter	**TDK Corporation**	**DEA202450 BT-1294C1-H**	**Digi-Key**	**1**	**$0.32**	(Wire)
Antenna	Taiyo Yuden	**AH316M245 001-T**	**Digi-Key**	**1**	**$3.98**	(Wire)

** = Datasheet Hyperlink attached to each component/part description

Total Cost = $81.77 (Bill of materials) + $24,000 (Cost of labor) = $24,081.77

4. Schedule

Schedule will consist of the assignments for the projects to be started and completed.

Schedule:	
Week	Due
9/28/20	Design Documentation
10/5/20	Last Day Revisions of DD and Machine Shop Guy
10/12/20	Finish PCB and Order
10/19/20	Testing Components
10/26/20	Finish Testing / Start Protype
11/2/20	Test and Finish ProtoType
11/9/20	Demo to Executives
11/16/20	Finish Final Bugs and Demonstrate
11/23/20	Thanksgiving Break
11/30/20	Final Presentation On Our Product
12/7/20	Final Week Wrap Up

5. Ethics and Safety

We will take the utmost care in complying with the ethics and safety standards that IEEE and ACM have laid out. To comply with the ethics and safety standards, this project will be of a high quality, and the processes and products will derive from high-quality professional work in accordance with the ACM “Professional Responsibilities” section 2.2. Our team and its individuals will take “personal and group responsibility for acquiring and maintaining professional competence” [2].

Privacy is a very critical detail to keep in mind for any project. Our project will comply with the privacy of the delivery service, by using the camera footage only to ensure that the box is not being tampered with, thus, only using the data for “legitimate ends and without violating the rights of individuals and groups”. Additionally, it will be communicated to the delivery service that they may be the subjects of our camera footage. This adheres to the ACM guidelines in the section on respecting privacy, section 1.6 [2].

Power from the battery could potentially cause a safety hazard, if care is not taken. The project will comply with the safety standards concerning the power from the battery. The Lithium Ion battery must never be charged or kept in extreme temperatures (below 32^0 F or above 130^0 F), or it may be damaged or even explode [3]. Utmost care will be taken during this project to ensure that the batteries are properly inspected and in good condition to be used, thus adhering to the IEEE code of ethics stating “to hold paramount the safety, health, and welfare of the public” [4].

Finally, the project will ensure that the numeric code generated by the app will be local to the user’s phone, and that the information will not be divulged anywhere else other than to the delivery service. Additionally, the numeric code will be randomized for every new delivery, and confidentially stored in the user’s app storage. When the delivery is done, that numeric code will be discarded, to ensure that the same code cannot reopen the box again. This will adhere to the ACM standards on confidentiality, which state that “the nature or contents of that information should not be disclosed”, according to the ACM guidelines in the section on respecting privacy, section 1.7 [2]

6. References

[1] A. Berthene, "Online merchants gain an extra $107 billion in 2020 thanks to pandemic", *Digital Commerce 360*, 2020. [Online]. Available: https://www.digitalcommerce360.com/article/coronavirus-impact-online-retail/. [Accessed: 17- Sep- 2020]

[2] "ACM Code of Ethics and Professional Conduct", acm.org, 2020. [Online]. Available: https://www.acm.org/code-of-ethics. [Accessed: 17- Sep- 2020].

[3] "Safety and Health Information Bulletins | Preventing Fire and/or Explosion Injury from Small and Wearable Lithium Battery Powered Devices | Occupational Safety and Health Administration", Osha.gov, 2020. [Online]. Available: https://www.osha.gov/dts/shib/shib011819.html. [Accessed: 17- Sep- 2020].

[4] "IEEE Code of Ethics", Ieee.org, 2020. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 17- Sep- 2020].

[5] "Why would a C executable be smaller when compared to C++ executable", Software Engineering Stack Exchange, 2019. [Online]. Available: https://softwareengineering.stackexchange.com/questions/246167/why-would-a-c-executable-be-smaller-when-compared-to-c-executable/246181#246181. [Accessed: 02- Oct- 2020].