ECE 445

Senior Design Laboratory

Design Document

Inventory Tracker

Team 46

Sara Alabbadi (saraa6@illinois.edu)

Alex Buchheit (alexwb2@illinois.edu)

Sooha Ryu (soohar2@illinois.edu)

TA: Jason Zhang
Professor: Arne Fliflet

Abstract

This document provides details on the proposed design and components of an inventory tracking system. It includes a proposed schedule for the project construction and testing and total estimated cost of the project.

Table of Contents

1 Introduction

1.1 Problem

Inventory tracking is an essential process in all types of applications ranging from small businesses to larger ones. It helps business owners keep track of their supplies and optimize their inventory levels based on demand. In addition, controlling the inventory could increase profitability when performed accurately. The process of tracking inventory is often done manually as it helps reduce cost, but this results in inaccurate results and discrepancies in the available supply due to human error. Moreover, it can be burdensome and time-consuming as it often becomes repetitive which leads to inefficiency. As a result, automation has revolutionized the inventory tracking process. Although automating inventory tracking has been applied in several industries, small startups, and business owners are still required to manually process their inventory due to the high cost of automation. Therefore, the need to shift to automating the process for a sustainable solution with less cost is required. Moreover, manual tracking is not only unreliable but it also poses a threat to the inventory as it remains unsecure at all times.

1.2 Solution

One proposed solution to the problem of asset tracking in small-owned businesses is a low-cost fully automated inventory management tracking system. This system would use an RFID scanner to scan an ID and unlock supply boxes to give users access. For additional security, it will allow users to only access the inventory they are authorized to. The user would then be able to check the supplies available to them. The system would be connected to a web database that would display the stock of each item, what items have been checked out by which users, and how many items have been checked out. Moreover, the system would allow a supervisor to access all supply boxes, restock products, and update the total in the database. This would also help the user in visualizing the stock of inventory to see which items are in demand and thus increase efficiency and productivity.

Moreover, the system would alarm the user if unauthorized access had been detected. This would be established through a locking mechanism for the boxes. The boxes would be locked with a magnet and current-carrying wire to hold them shut. Once a user scans their RFID card, only the boxes they have access to will unlock and a message will appear in the display to mark which box has been unlocked. Finally, if a box is opened forcefully, this will alarm the user through the database that an unauthorized person has opened it.

1.3 Visual Aid

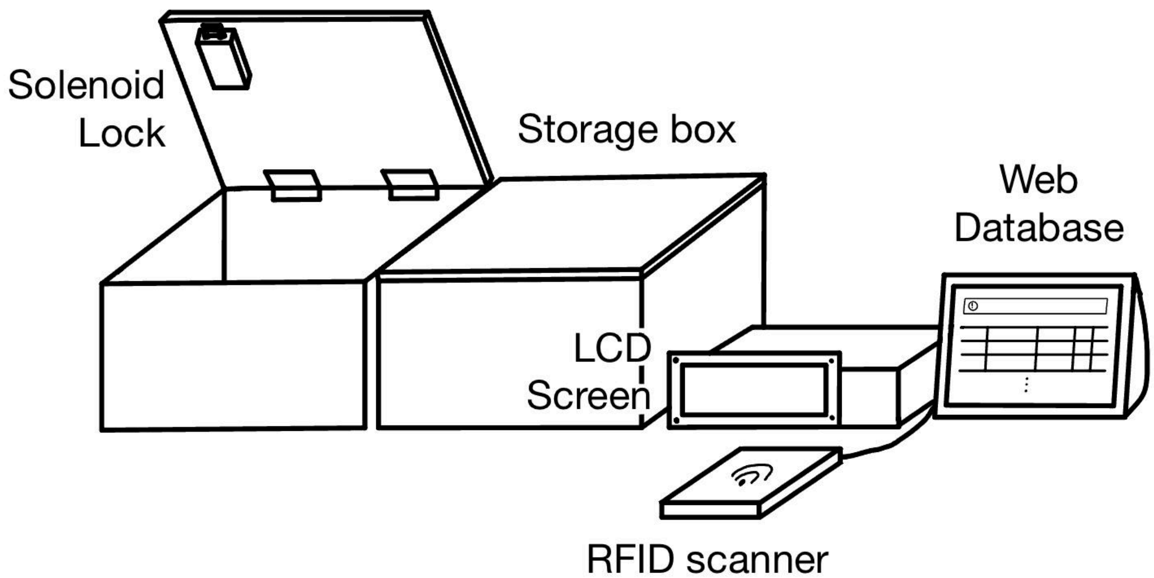

Fig. 1: Physical Design Sketch

1.4 High-Level Requirements

1. When the user scans their RFID card, the solenoids on the boxes the user has access to should unlock within 5 seconds and the LCD should display the boxes the user has access to in the same timeframe.
2. When the user takes an item out of a box, the web database should display the item taken out and what the user took out within 5 seconds. The webpage should always display the number of items in stock and keep track of items checked out and who checked them out.
3. The alarm system should be triggered when a locked box is opened. Within 5 seconds of the locked box being opened, both the LCD and web database should display that someone has tried to open a locked box.

2 Design

2.1 Physical Design

The system will be powered by a 120 V AC to 12 V barrel jack connector. The ESP32 microcontroller will provide a Wifi/Bluetooth module, which will connect to a webpage to monitor parts that have been checked out and the total stock of each item. Two boxes will hold items and each will be held shut by a 12V solenoid lock. A buck converter and voltage regulator will be used to step down the input voltage for the load cells and microcontroller. Load cells will be in the boxes to measure the weight of the items in each box. Changes in this weight will determine if an item is checked out. The microcontroller will monitor this weight and communicate with the webpage when an item is checked out. An RFID module will allow users to scan in and unlock the drawers they have access to. When a user scans in, the microcontroller will allow power to be sent to the corresponding solenoid, unlocking the box the user has access to. An LCD screen will display to the user which boxes they have access to so they know which box to open. When this user takes an item out of the box, the microcontroller will communicate which part and what the user took to the webpage and the webpage will display this information and the total stock of all parts. A microswitch will also be placed onto both boxes to indicate the status of the box, closed or open. If the microcontroller detects it to be open by an unauthorized person, then it should trigger the database to alarm the user.

2.1.1 Schematic

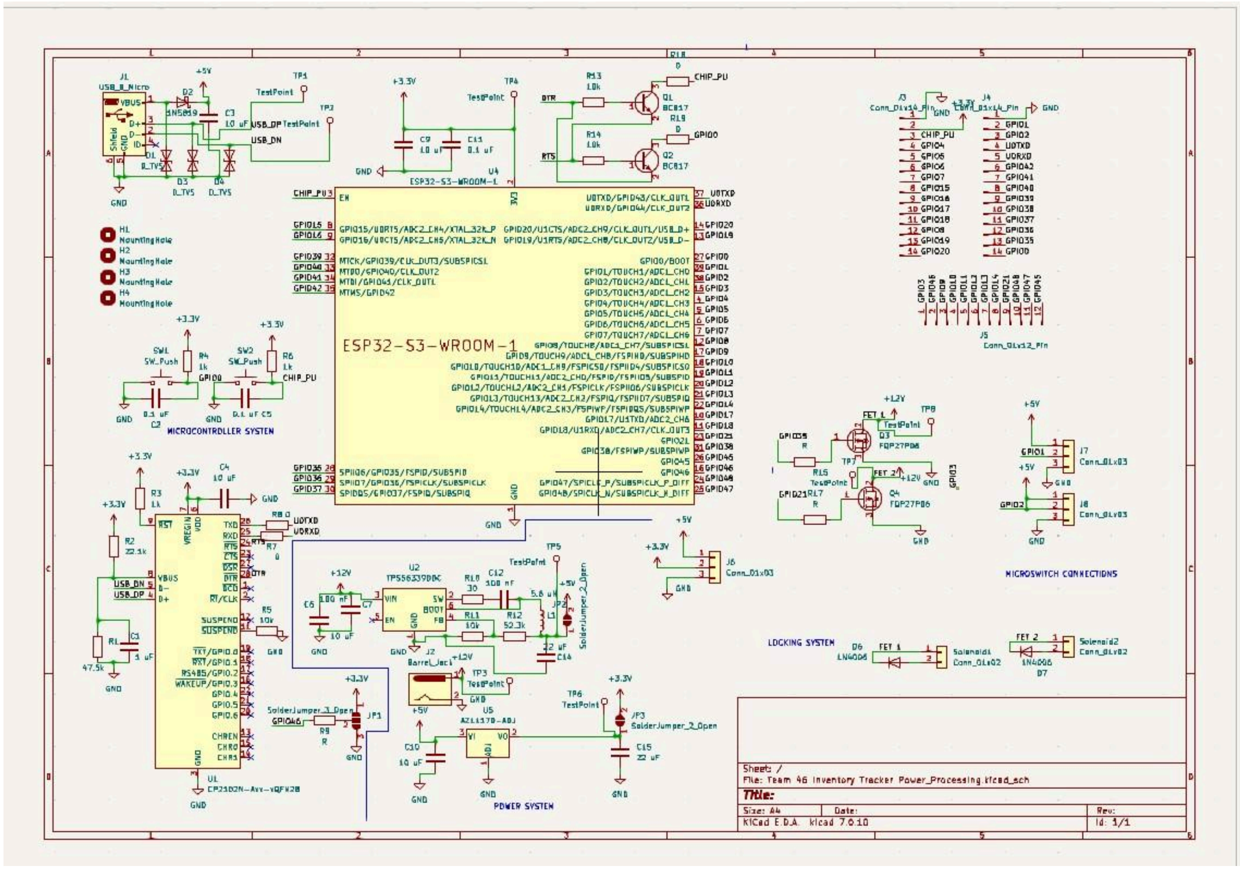

Fig. 1: PCB Schematic - Power, Solenoids, and Microcontroller

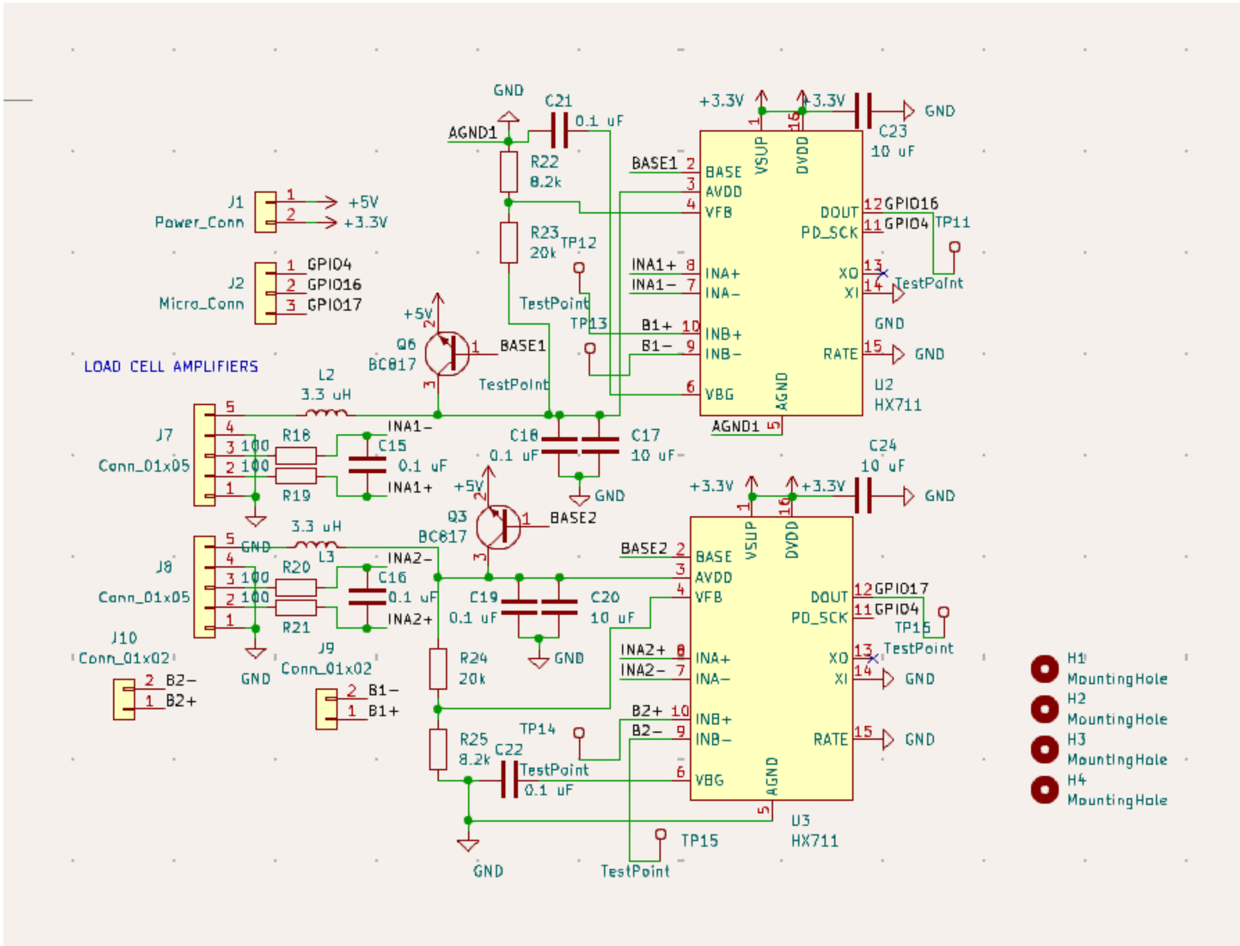

Fig. 2: PCB Schematic - Load Cell Amplifiers

2.2 Block Diagram

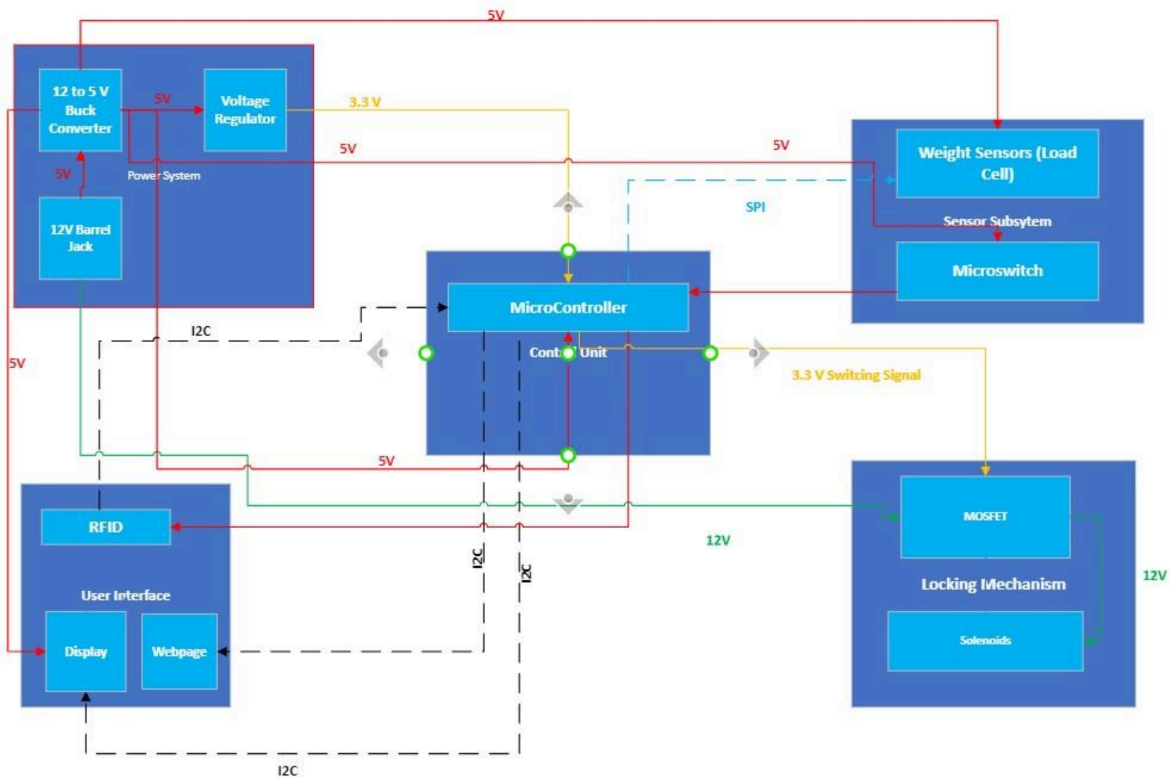

Fig. 3: Subsystem Block Diagram

2.3 Functional Overview

2.3.1 Power Subsystem

The power subsystem is responsible for providing power to all of the components of the project. The system is expected to be powered by a 120V AC to 12 V DC barrel jack connector, which will output 12 V. At normal operation, the system should draw around 2.12 W, and at maximum load, which is when both solenoids are being unlocked, it should draw about 11.02W. The solenoids need 9-12 V for power and when unlocking they each will draw 350 mA. A buck converter will be used to step down 12 V to 5 V. The planned components are an AZ1117 5 V to 3.3 V voltage regulator, a TPS56339 boost converter, and a 120 V AC to 12 V barrel jack plug and cable. Every component should be powered by 5 V, except for the solenoids, microcontroller, and load cell amplifiers. The microcontroller and amplifiers will be powered with 3.3 V and the solenoids will be powered with 12 V.

Requirements	Verification
The voltage regulator should output between 3.0 and 3.6 V, which is the range needed to power the microcontroller.	- An oscilloscope probe should be used to measure the voltage between the regulator output and ground.
The buck converter should output between 4.75 and 5.5 V.	- An oscilloscope probe should be used to measure the voltage across the converter output and ground. - With the system powered on and the probes connected as described above, scan the RFID module to unlock both solenoids and use the oscilloscope to monitor the voltage of the converter.

Table 1: Power Subsystem Requirements & Verifications

2.3.2 Sensor Subsystem

The sensor subsystem is responsible for sensing the different weights of the items tracked and accurately distinguishing between them. We will implement the weight scaling by using two load cells mounted at the bottom of the boxes. Due to the load cells having a few volts as an output, an amplifier will be needed for them to serve as the ADC convertor. The HX711 Load Cell Amplifier will be used and then it will be connected to the ESP32- Microcontroller.

Requirements	Verification
The load cells should accurately measure the weight of the items between 0-50 grams. The sum of the items in the box should be 0-300 grams. It should report the weight with an approximated error of about 5% than its real weight.	- Take an item and weigh it with an accurate weight scale and record its weight. - Next, connect the load cell to the HX711 Amplifier. Connect the Red wire from the load cell to E+ on the amplifier, Black Wire from the load cell to E-, white wire from the load cell to A-, and green wire from the load cell to A+. - Connect the HX711 to the microcontroller. GND to GND, DT to any Data Pin available on the microcontroller, SCK to SCK and VCC to 3.3V pin on the

	ESP32 - After building this circuit, put the same item that was weighed earlier on the load cell and observe the measured weight. - If the measured weight is satisfactory (about +/- 5% in accuracy), this step is completed. - Else, the load cell needs to be recalibrated.
The load cell should output the weights of the items between 0 - 50 g with a maximum deviation of 3% between each reading for the same item.	- After connecting the load cell to the HX711 and the microcontroller like mentioned above, weigh an item with a scale to record its weight. - Place the item on the load cell and wait about 1 second for the load cell to record the data. - Keep the same object placed while the load cell is still logging the data until about 5 readings. - Each reading should not deviate by more than 3% from the other readings and it should be about +/- 5% to its reference weight.

Table 2: Sensor Subsystem Requirements & Verifications

2.3.3 Control Subsystem

The Control System is the main subsystem that connects all the different components of the design together. It is responsible for receiving the weights of the different items from the weight sensors. When a user checks out an item, based on its weight, the microcontroller should send the updates with the items checked out to a web database. We will use the ESP32 microcontroller which has a built-in WIFI module to regularly update the web database when a user checks out an item. Moreover, the microcontroller should also communicate with the RFID Module to determine which box to unlock based on the specified access. This will be done through the communication between the ESP32, the solenoid locks, and the RFID Module. Finally, the microcontroller should determine when the box is opened by receiving a signal from the microswitch. If it was an unauthorized access, then it will trigger the database to alarm the user.

Requirements	Verification

Correctly interpret the weights of the items checked out with their corresponding name.	- When the box is open, take out the items and make sure that the correct item is displayed on the web database. - Similarly, return an item into the box and the web database should also update successfully. - This should be done with all items in the box to make sure that the tracking is accurate and successful.
Send the data to the web database within 5 seconds.	- Close the lid of the box and scan the ID again to confirm that there is no more need for items checked out. - Check the web database and it should be updated to the most recent version with the ID of the user, Items Checked Out, and the number of items checked out, and the initial time of the ID-scanning and the final time of the scan. - This process should take no more than 5 seconds.
Differentiate between the users' scanned ID to only unlock the authorized boxes by controlling the switching signal to the solenoid.	- Connect the microcontroller to the solenoid lock. - Have a user who only has authorized access to one box scan their ID onto the RFID, and make sure that only the specified box will unlock within 3-5 seconds. - Have a user who has access to both boxes scan their ID, and make sure that both boxes unlock.

Send an alarm to the web database when triggered by the FSR.	- This should be verified in these two scenarios:
	    - Scan an ID of a person who has authorized access to Box A and not Box B. After Box A is unlocked, try opening Box B by pushing into the lid or applying a strong push on the lid.
	    - The microcontroller should send a signal to the web database to alarm the user of unauthorized access.
	    - The second scenario is if a person does not scan their ID, and still tries to access the locked

	inventory. Don't scan your ID, and try pushing into the box to open it. - Similarly, the microcontroller should send a signal to the web database to alarm the user of unauthorized access.

Table 3: Control Subsystem Requirements & Verifications

2.3.4 User Interface Subsystem

The user interface consists of the RFID module and a display. The RFID module will allow users to access the selection of boxes they have been given access to through their RFID card. It will communicate with the microcontroller, which will communicate with the locking system for successful operation. A web database consisting of items checked out by the user, along with the current stock of items, will be visible on a web page.

Requirements	Verification
The web database should be accurately updated with the user's information with the items they checked out	- Scan the iCard of the student on the RFID scanner and wait for the box to unlock, then correctly update the item checked out with the student's information. - The updated database will be pulled up and using SQL queries to correctly maintain the item stock.
The web database should be accurately updated with the user's information with the items they returned	- Scan the iCard of the student on the RFID scanner and wait for the box to unlock, then correctly update the item returned with the student's information. - The database should be pulled up and updated using SQL queries to correctly maintain the item stock.
There should be an alert displayed on the web when there's been suspicious activity trying to open the box when there's no access granted	- Try to open the door by force when the box is still locked. - An alert notification should be displayed on the web with the time and the box # to report suspicious activity.
RFID correctly reads the user data within 3 seconds when iCards is scanned	- RFID correctly reads and sends the user information when the iCard is scanned to the control subsystem
Display which box has been unlocked on LCD	- Scan the ID of the authorized user into the

within 3-5 seconds. Ensure that the other boxes which the user is not authorized to access are locked.	RFID scanner. - Wait for about 3 seconds and see which bx number is displayed on the screen. - LED displays the unlocked box #. After the display, try unlocking the other box and make sure it remains locked.

Table 4: User Interface Requirements & Verifications

2.3.5 Locking Mechanism Subsystem

The Locking Mechanism Subsystem is added for safety reasons and to trigger the user when unauthorized access has been determined. This subsystem consists of two solenoid locks and a microswitch which will be mounted on the sides of the box to determine the status of the box. The microswitch is powered by the 12V-5V convertor, and it would indicate when the box is opened and/or closed. In our design, the microswitch will be operating in the normally closed mode. The solenoid receives a signal from the microcontroller after a user scans their ID. The microcontroller's signal will specify which box to unlock and the solenoid will only unlock the specified ones. The solenoid locks will be connected to a MOSFET and the boost converter. The MOSFET's switching signal will be sent from the microcontroller to ensure that it is only unlocked when an authorized user tries to access it.

Requirements	Verification
Unlocks the solenoid of the correct box within 5 or fewer seconds	- Connect the solenoid to the microcontroller. - Scan the ID of a person who has access to one of the boxes. - Wait for about 5 seconds - The solenoid lock should only unlock the authorized box.
Microswitch can determine when the box is opened and/or closed	- Connect the terminal labeled G to GND of the microcontroller. - Connect the terminal labeled NC to the 5V power supply. - Connect an oscilloscope to measure the output voltage across the microswitch. - When holding the button down, the power should be cut off and when holding the button out the power should be continued.

Table 5: Locking Mechanism Subsystem Requirements & Verifications

2.4 Tolerance Analysis

One major aspect of this project that is critical to its success is the ESP32 microcontroller. The microcontroller needs to have consistent power to work properly. To power the microcontroller, the AZ1117 voltage regulator will step down from 5 V to about 3.3 V. The voltage regulator must be able to do this consistently and not overheat. According to the ESP32 datasheet [2], it can operate with an input voltage from 3.0 V to 3.6 V, so the voltage regulator must be able to stay within that range. According to the Wiki page on the ECE 445 website [1], the equation to determine the operating temperature of the voltage regulator is $T= i_{out}(v_{in}-v_{out})\theta_{ja}+T_a$ where θ_{ja} is the junction-to-ambient thermal resistance and T_a is the ambient temperature, assumed to be 25 ^oC.

According to the datasheet of the regulator [3], it can operate at temperatures up to 150 ^oC and is recommended to operate at less than 125 ^oC. It also has a junction-to-ambient thermal resistance of 100$^oC/W$. According to the ECE 445 Wiki page, the ESP32 should draw a maximum of 355 mA. Using these parameters and the equation above, assuming in the worst case we operate at the minimum 3.0 V, the maximum calculated regulator temperature is 96 ^oC, which is within the recommended operating range.

Another important aspect of our design is the load cell's accuracy of the output readings. According to TAL220 datasheet [5], the load cell should have a combined error of +/- 0.05. This means that the maximum deviation of the measured readings from the straight line drawn between readings at zero load is 0.05 at the rated capacity. Another important characteristic that the load cell should satisfy is providing consistent readings over time. The load cell's datasheet [5] specifies the repeatability of the readings to be +/- 0.03, which indicates that the same object might change weights with 0.03 discrepancy in each reading. In our design, the load cell continuously logs the weights of the items until one is removed which iterates the importance of consistent readings.

To better understand these parameters, the load cell was first calibrated in order to get accurate readings. With measurements of different objects, including but not limited to an empty water bottle, a car key, and a TV Remote, the following calibration curve was obtained.

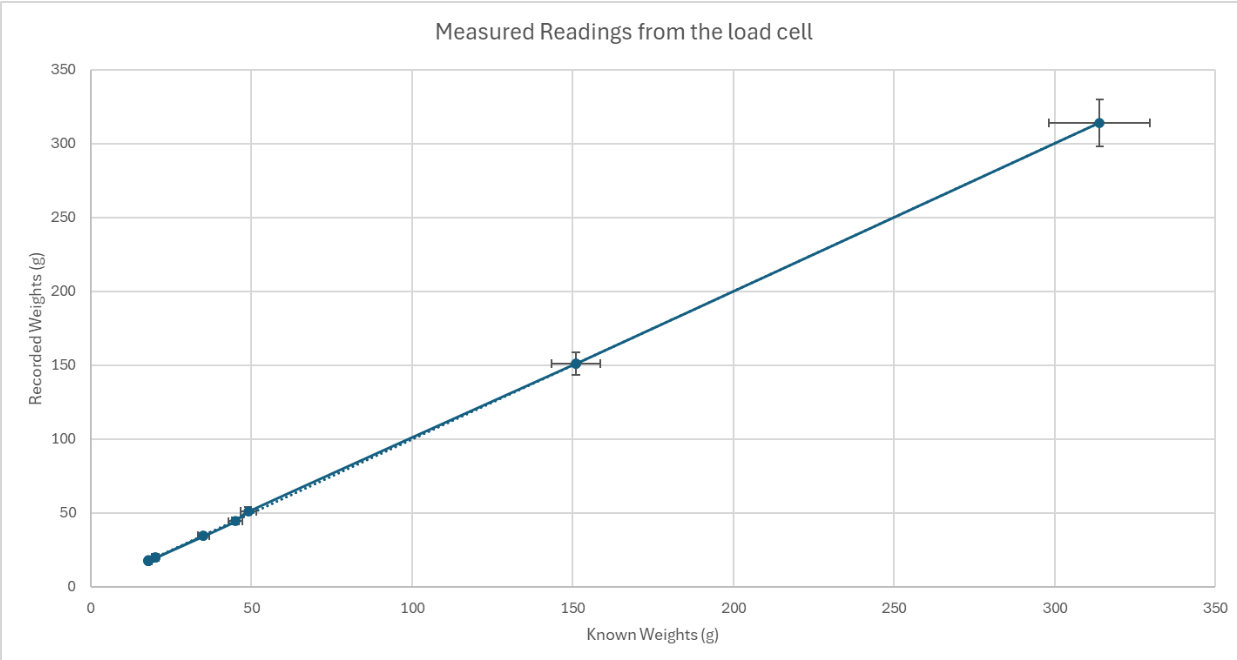

Fig 4: Calibration curve

We can see that a linear relationship exists which indicates that using the load cell for our inventory tracker is ideal. However, as shown in Figure 4, with increasing weights the error increases between the recorded weights and the measured data. Therefore, the sum of the weights placed in one box are to be limited between (0g - 300g) and the weight of each object between (1g - 50g) as that would be the most accurate range for the system to lead to precise and accurate results.

Finally, another important parameter is the creep factor, which indicates the change in the load cell signal under continuous load. With this load cell, the creep factor was +/- 0.05 for every 3 minutes [5] indicating that the recorded weights might deviate with +/- 0.05. However, knowing these limitations for our load cell indicates that over time the combined error taken into consideration with the creep factor would be about +/- 0.1.

Although this limits our design in regards to the size of the items that the users can put inside the inventory box, given that our targeted users are small startups and corporations it will still enhance their supply tracking process.

3 Cost and Schedule

3.1 Cost Analysis

3.1.1 Labor Cost

Assuming the average salary/hour of an ECE graduate is $50, expecting an 8 hours of work each week for 10 weeks of project execution for a team of 3 engineers, the labor cost can be calculated as below:

50 x 8 x 10 x 3 = $12,000

3.1.2 Cost of Parts

Description	**Manufacturer**	**Part Number**	**Quantity**	**Unit Cost ($)**
Solenoid locks	Adafruit	5065	2	$7.50
Load sensor	Sparkfun Electronics	SEN-10245	2	$4.50
Force Sensitive Resistor (FSR)	Adafruit	SEN-09376	2	$3.95
RFID Read/Write Module	Parallax Inc	PARALLAX 28440	1	$49.99
TPS56339 Buck Converter	Texas Instruments	TPS56339	1	$1.00
HX711 Chip	Avia Semiconductor	HX711 SOP16	2	$0.495
LCD Display	Adafruit	181	1	$9.95
Total				**$93.83**

Table 6: Itemized List of Components and Costs

3.1.2 Total Cost

Summing up the labor cost and the cost of all parts, the total cost turns out to be **$12096.08.**

3.2 Schedule

Week	**To Be Completed**	**Assigned Member**
Week of 2/19	Complete PCB Schematic	Alex
	Acquire Parts	Sara
	Complete Sketch for the Machine Shop	Sooha
	Finish Design Doc	All
Week of 2/26	Design Review	All
	Complete PCB Design	All
	Acquire Parts	All
	PCB Review	All
Week of 3/4	Order PCB	All
	Start interfacing the Sensors with the microcontroller	Sara
	Start testing components for the Power Subsystem	Alex
	Start testing RFID with the microcontroller + Start web database	Sooha
Week of 3/18	Start testing the solenoid locks with the MOSFET and microcontroller	Sara
	Start on the identification process of the RFID and determining controlled access	Sooha
	Second Round of PCB Order	All
Week of 3/25	Test Sensor Subsystem	Sara
	Test Power Subsystem	Alex
	Test Locking Subsystem	Sara + Alex
	Test User Interface (RFID)	Sooha

	Test all these systems together	All
Week of 4/1	Begin Soldering	All
	Test PCB	All
	Begin Assembly of the system with the machine shop	All
Week of 4/8	Test PCB and the system	All
	Order 5th Round of PCB if necessary	All
Week of 4/15	Mock Demo	All
Week of 4/22	Final Demo	All
Week of 4/29	Final Presentation	All

Table 7: Assigned Schedule for Project Pt.2

4 Ethics and Safety

To ensure we work on this project ethically and safely we plan on following the IEEE Code of Ethics [4]. When issues arise between group members we will try to go through each person's point of view on the problem and try to find a good solution for everyone fairly and efficiently. We understand the necessity of bringing up problems early before they can make working together a major difficulty. Along with that, we understand that this is a learning experience and we will focus on learning throughout this project.

Regarding the privacy of data collected by the web database for user inventory tracking, we will ensure user data remains private and is only confined to the web database. The user's personal information will not be visible to the public and only necessary information will be kept. There will be no sensitive information stored in our database that can be easily used by the public to identify users (ex. names). We plan on only storing a form of identification number as user information. To further mitigate this concern, we plan to control access to the database with a PIN code so only authorized users can view the database of inventory tracking records.

Another concern is temperature regulation. The ESP32 can operate at up to 105 ºC [3] and most other components can handle up to 125. Using components above their rated temperature could cause components to burn and create safety hazards. To mitigate this, team members must determine the maximum temperature of components under the intended operating conditions and measure the temperature of components during the testing procedure. Team members must disconnect power when it is determined that components are overheating, to mitigate damage to the system and avoid potential safety hazards.

5 References

[1] ECE 445. “Linear Regulators.” https://courses.engr.illinois.edu/ece445/wiki/#/regulators/index. (accessed Feb. 20, 2024).
[2] Espressif Systems. “ESP32-S3-WROOM-1 ESP32-S3-WROOM-1U Datasheet.” Espressif.com. https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf (accessed Feb. 8, 2024).
[3] Diodes Incorporated. “AZ1117 LOW DROPOUT LINEAR REGULATOR WITH INDUSTRIAL TEMPERATURE RANGE .” https://www.diodes.com/assets/Datasheets/AZ1117I.pdf. (accessed Mar. 19, 2024).
[4] IEEE. “IEEE Code of Ethics.” IEEE.org. https://www.ieee.org/about/corporate/governance/p7-8.html (accessed Feb. 8, 2024).
[5] SparkFun. “PARALLEL BEAM LOAD CELL.” https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/TAL220M4M5Update.pdf. (accessed Feb 21, 2024).