

ECE 445
SENIOR DESIGN LABORATORY
FINAL REPORT

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

May 1, 2024

Abstract

The following report discusses the process of building a device that stores and tracks inventory. It will go over the motivation for creating the module, design process, building, testing, and verification of functionality. It will also draw conclusions on the success or failure of the project and how the project could be continued and improved upon.

Contents

1	Introduction	1
1.1	Problem	1
1.2	Solution	1
2	Design	2
2.1	Design Procedure	3
2.2	Design Details	3
2.2.1	Locking Subsystem	3
2.2.2	User Interface Subsystem	4
2.2.3	Sensor Subsystem	5
2.2.4	Power Subsystem	5
2.2.5	Control Subsystem	7
3	Verification	9
3.1	Locking System	9
3.2	User Interface System	9
3.2.1	LCD Display	9
3.2.2	RFID Module	12
3.2.3	Realtime Database and Webpage	12
3.3	Sensor System	13
3.4	Power System	15
3.5	Control System	15
3.5.1	Weight to Count Calculation	16
3.5.2	ESP32 WiFi Connection to Database Update	17
3.5.3	User-Access Control	17
3.5.4	Theft Alarm System	17
4	Costs	19
4.1	Cost Analysis	19
4.1.1	Labor of Costs	19
4.1.2	Cost of Parts	19
4.2	Total Cost	19
5	Conclusion	20

1 Introduction

1.1 Problem

Inventory tracking is an essential process in all types of applications ranging from small businesses to large industries. 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.

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.

2 Design

The system is powered by a 120 V AC to 12 V barrel jack connector. The ESP32-S3 microcontroller provides 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 hold items and each is held shut by a 12V solenoid lock. For our demonstration, the microcontroller development board powered the other components of the project. A buck converter and voltage regulator will be used to step down the input voltage for these components in a practical application. Load cells are in the boxes to measure the weight of the items in each box. Changes in this weight 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 allows users to scan in and unlock the drawers they have access to. When a user scans in, the microcontroller allows 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 communicates 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.

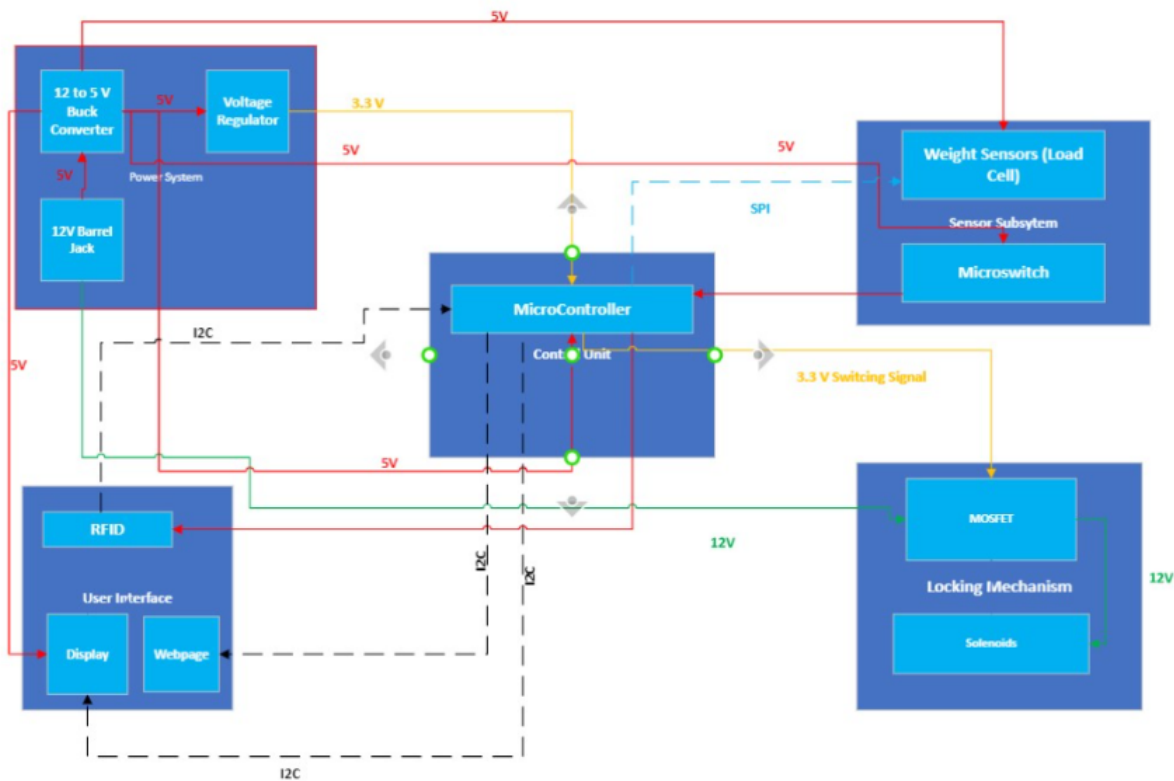


Figure 1: Initial Block Diagram

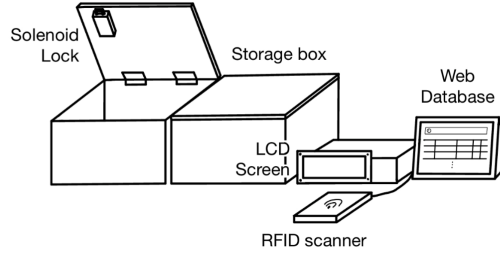


Figure 2: Physical Design Sketch

2.1 Design Procedure

For the physical design of the inventory tracker, it was best to keep it simple and it was composed of two boxes where each box tracked a specific item. The load cells are mounted at the bottom of the boxes to continuously record the weight of the items inside. A micro-switch is placed on the sides of the two boxes which determine the status of the boxes to trigger the alarm subsystem when opened without access. The solenoid locks are placed on the top of the box to allow it to close when prompted. Finally, the User Interface subsystem, which consists of RFID scanner, an LCD to display which box has been unlocked, and the web database to record the activity of the inventory tracker were all placed besides the box. as shown in Figure 2.

Next, the PCB design procedure began by attempting to replicate the ESP32-S3 development board on a PCB. This was done by using a GitHub repository with KiCAD libraries for the ESP32 processors and schematics for each development board [4]. Also added to the PCB are solenoid driver circuits, microswitch connections, and power electronics to step down 12 V to 3.3 V and 5 V to power all other modules. For the demonstration we were not able to use this because we had difficulty getting the PCB to work, so we used the 3.3 V and 5 V pins on the development board. Parts were ordered through Digikey, Mouser, or checked out of the ECE 445 Lab.

2.2 Design Details

2.2.1 Locking Subsystem

The locking subsystem is two solenoid driver circuits, one for each box. A solenoid is connected in parallel to a reverse bias diode. The diode allows the solenoid to dissipate current, so there is no voltage spike when power is disconnected. The solenoid and diode are connected in series to an IRF510 power MOSFET, which is connected to 12 V power. The MOSFET's gate threshold voltage is 4 V [10]. Since the ESP32 can only output 3.3 V [4], a gate driver was needed. We used a MIC4424 [15]. It can take 3.3 V input and output a 12 V signal [5].

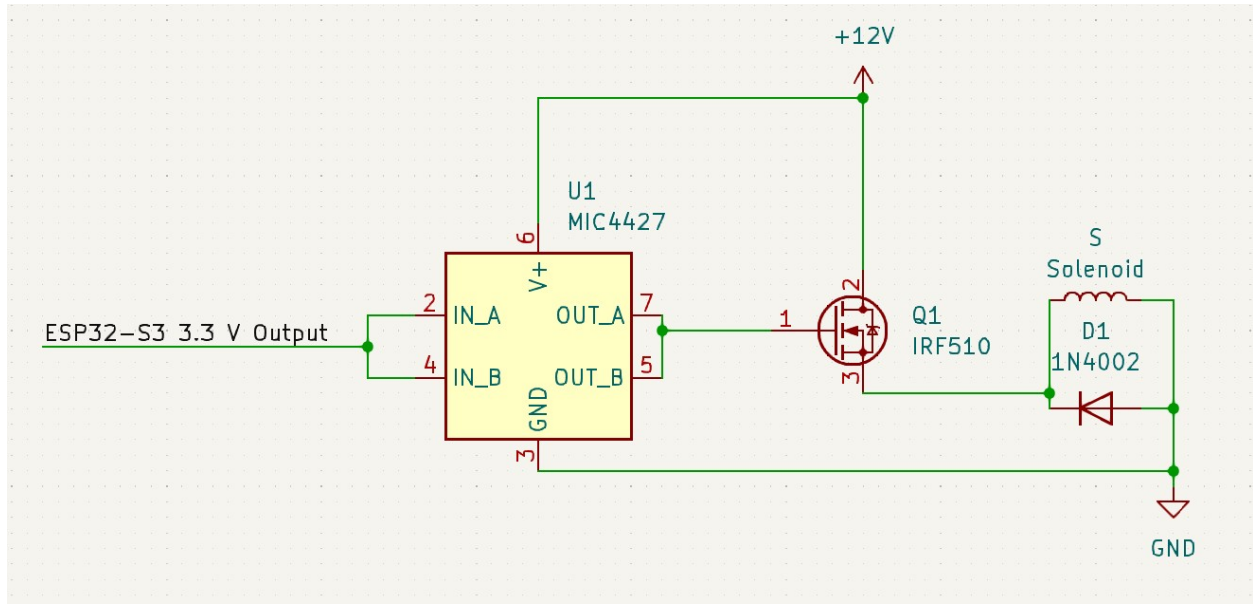


Figure 3: Schematic of Driver Circuit

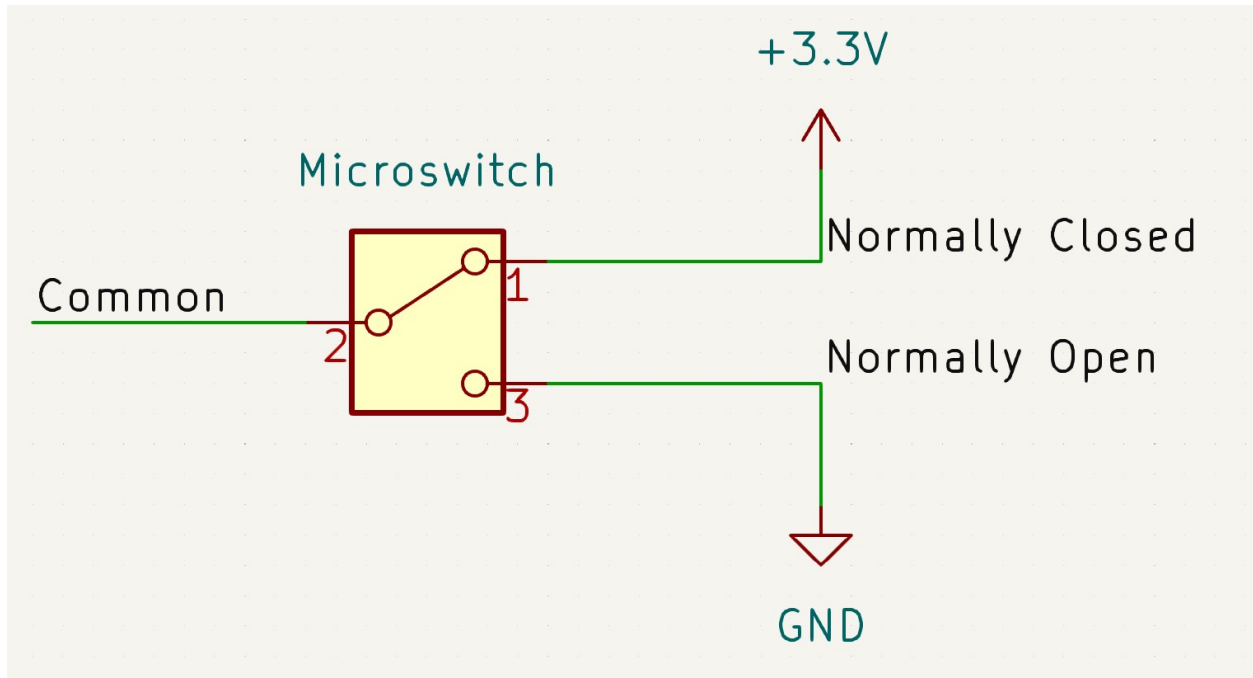


Figure 4: Microswitch Schematic

2.2.2 User Interface Subsystem

The user interface subsystem consists of several components: an RFID scanner, an LCD display, a real-time database, and a web page displaying the database. The RFID module will allow users to access the selection of boxes they have been given access to through their RFID card. The module we used is the RC522. The pinout is shown in Figure 4. It

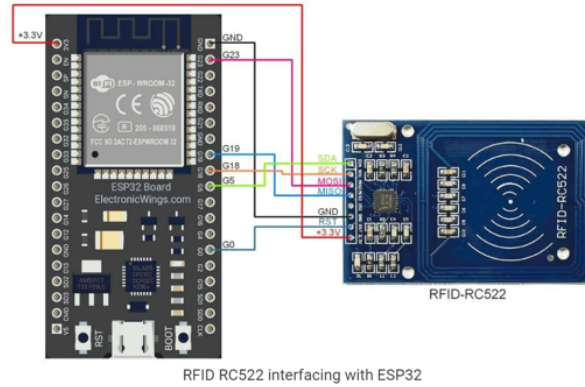


Figure 5: RFID Pinout [14]

communicates with the microcontroller using I2C communication. The microcontroller will then communicate to the locking system to unlock the required boxes. Then the LCD screen displays which boxes have been successfully unlocked to let the users know which boxes they have access to once they scan their card. For inventory tracking, the Firebase Realtime database was utilized to keep track of numbers of items in each box and user access logs. The microcontroller posts all the updated data into the database over wifi network. All this information was then made visible on a web page for users and admins.

2.2.3 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. The load cells' output is a few mV, about 5-6 mV at full-load capacity. Therefore the signal needs to be amplified before interfacing with the microcontroller. A PNP transistor and an RLC circuit is used to amplify the signal. Moreover, the load cells' signals are analog, and the The HX711 IC Analog-to-Digital Converter [16] was used to interface between the microcontroller and the load cells. The circuit used for the load cells can be seen below in Figure 5.

2.2.4 Power Subsystem

The power system consists of at 12 V barrel connector, a 12 to 5 V buck converter, and a 5 V to 3.3 V voltage regulator. The barrel connector is used to power the solenoid locks. The 5 V converter was intended to power the LCD and LCA board. the 3.3 V regulator would power the RFID module and the second power input for the LCA board. The power schematic is shown in Figure 6. The converter was designed using the suggested configuration on the datasheet [12] along with the TI WeBench Circuit Designer [11], which takes input and output parameters and suggests circuit schematics and component values.

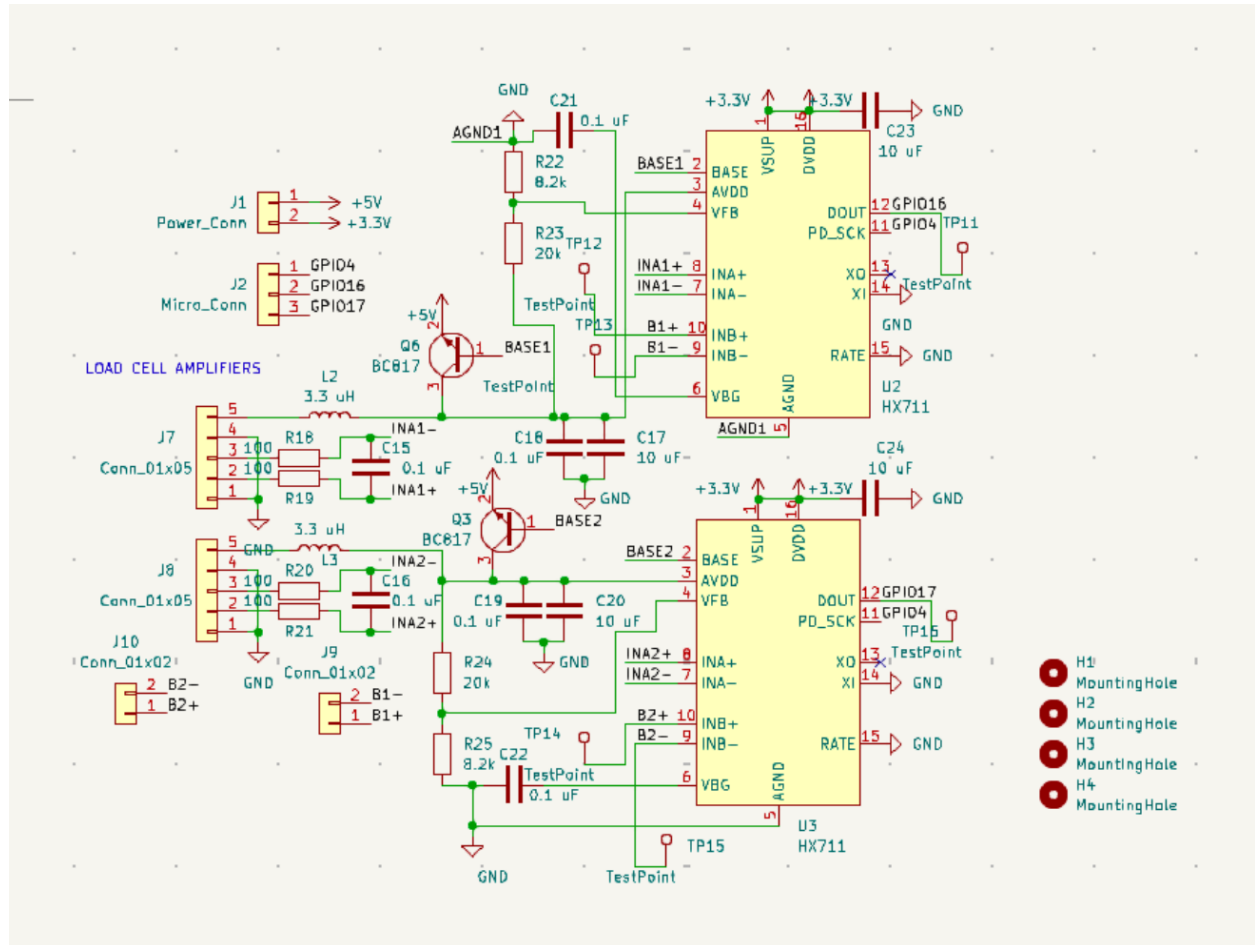


Figure 6: Load Cell Amplifier Schematic

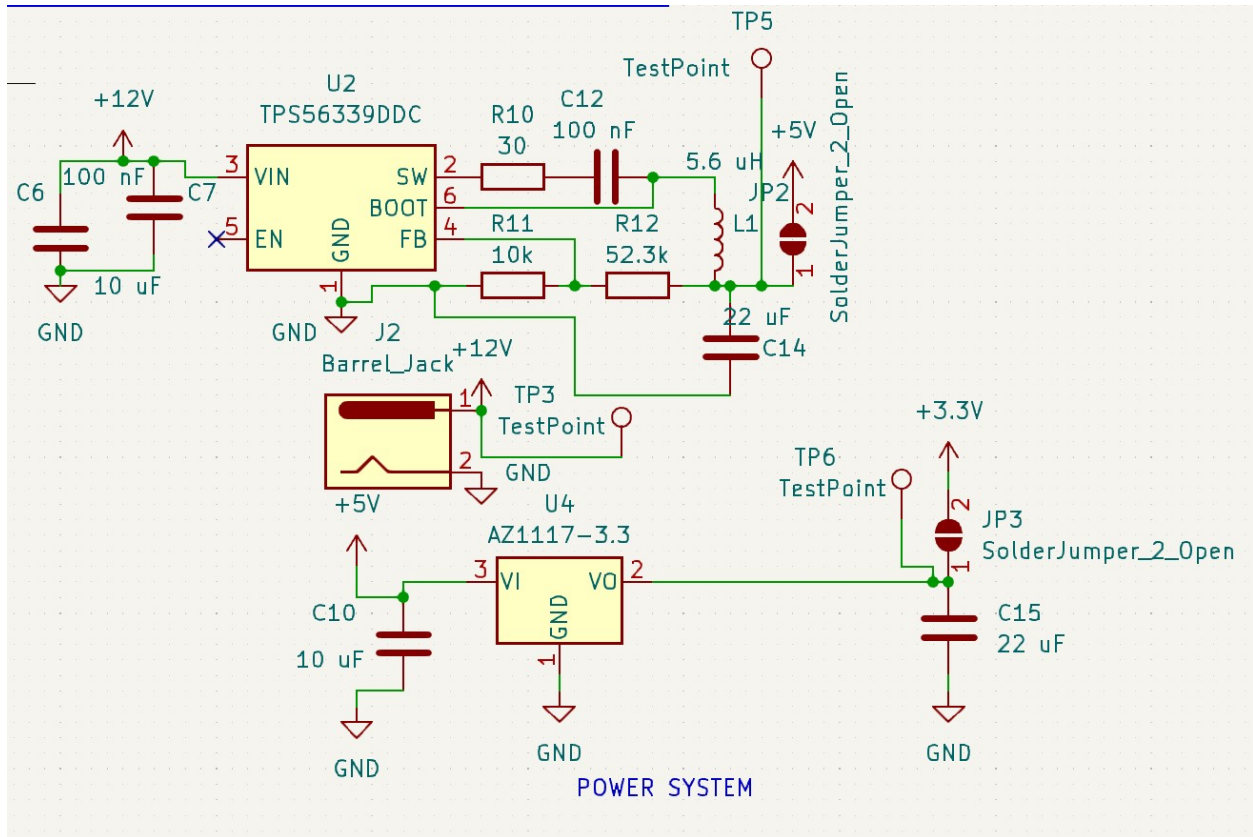


Figure 7: Power System Schematic

2.2.5 Control Subsystem

The microcontroller controls the whole locking mechanism as well as the user-access control. It also acts as a bridge between the hardware and the database with wifi connection. The esp32 receives scanned user IDs from the RFID module and weight of items inside the boxes from load cells and sends specified information to the database over wifi network to keep the database up to date. This is the flowchart representation of the program of our microcontroller:

During setup, it sets up all the subsystems controlled by the microswitch. It first sets up the load cells with defined input pins and the calibration factors for both boxes. It initializes the RFID module with all the required pins and sets solenoid pins signals to LOW. Then it configures a secure wifi and Firebase connection by signing in to the school wifi network *eduroam* and Firebase.

In the main loop, it constantly checks for a scanned RFID. Then it determines what boxes the scanned user has access to. If the current user does not have any access to the boxes, it triggered LCD to display "No Access" and goes back to the loop. If the current user has access to any of the boxes, it sends voltage signals to the MOSFET then to the solenoids to unlock the corresponding boxes. Then it waits for the user to "finish using the inventory"

3 Verification

3.1 Locking System

The locking system was tested by setting up a circuit as shown in Figure 3. Code was written to set the input signal high using the ESP32-S3. This 3.3 V signal was stepped up to 12 V using the gate driver, allowing the MOSFET to switch on and deliver power to the solenoid. The solenoids each drew about 0.41 A at 12 V when powered. The solenoid would click and the lock would retract when actuated. When power was disconnected, the lock would return to its original position.

Table 1: Locking System R & V

Requirements	Verificiation
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.

3.2 User Interface System

The user interface had several things to test. These include the LCD screen, the RFID module, realtime database, and the webpage.

3.2.1 LCD Display

The LCD was the quickest component to test, as it needed a very small amount of code to print to the screen. Testing was done by modifying code taken from an online source [2]. This source also explained the pinout required for the display.

The pinout as explained by DeepBlueMbedded [2], the site we adapted code from to test, is as follows: pins 1, 3, and 5 are ground pins, pin 2 takes 5 V power. Pin 4 is the register selector (RS), which can be connected to any ESP32-S3 GPIO pin. Pin 6 is the enable (EN) pin and can also be connected to any GPIO pin. Pins 7-14 are data pins. We only need 4 for this application, so pins 11-14 are connected to any available GPIO pins. Pins 15 and

Table 2: User Interface R & V

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 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 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	Open the door using the provided key instead of actuating the solenoid. 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 tag is scanned	RFID correctly reads, sends the user information when the tag is scanned to the control subsystem, and prints the tag data to the serial monitor.
Display which box has been unlocked on LCD within 3-5 seconds. Ensure that the other boxes which the user is not authorized to access are locked.	Scan the ID of the authorized user into the RFID scanner. Wait for about 3 seconds and see which box 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.

16 are used to control the LCD backlight, which we did not need for this project. The final pinout is shown in Figure 9.

The code used to test the display is also adapted from the source above [2] and was done in the Arduino IDE. It needs a Liquid Crystal library. We used one available in the Arduino IDE titled "LiquidCrystal", Version 1.0.7, by Arduino, Adafruit. The code, as shown in Figure 5, initializes the LCD object. Six pins need to be identified as GPIO pins. These include RS, EN, and the 4 data pins. These are done in ascending order of LCD pinout (i.e. RS, EN, DATA 5, DATA 6, DATA 7, DATA 8). The LCD needs to be initialized using the `.begin(16, 2)` function. 16 corresponds to the number of characters in a row and 2 corresponds to the total number of rows. The cursor can be set using the `.setCursor(x,`

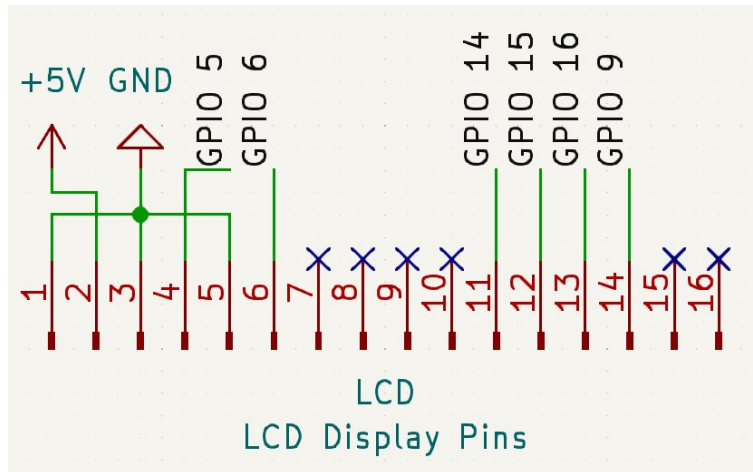


Figure 9: LCD Pinout in Final Product

```

1 #include <LiquidCrystal.h>
2
3 // Create An LCD Object. Signals: [ RS (pin 4), EN (pin 6), D4, D5, D6, D7 (pins 11-14)]
4 LiquidCrystal My_LCD(13, 12, 11, 10, 9, 8); // These are GPIO pins connected to above signals respectively
5 // PINS 1, 3, 5 and of LCD should be connected to ground
6 // With pin side facing you, it counts up 1 to 16 from right to left
7 // PIN 2 to 5 V
8 // PINS 4, 6, 11-14 (data pins 5-8) can be connected to any available GPIO pins
9 void setup()
10 {
11     // Initialize The LCD. Parameters: [ Columns, Rows ]
12     My_LCD.begin(16, 2);
13     // Clears The LCD Display
14     My_LCD.clear();
15
16     // Display The First Message In Home Position (0, 0)
17     My_LCD.print("ILL beat ISU");
18     // Set The Cursor Position To: [ Col 5, Row 1]
19     // The Next Message Will Start From The 1st Char Position in The 2nd Row
20     // Note: 1st Row Has An Index of 0, The 2nd Row Has An Index of 1
21     My_LCD.setCursor(0, 1);
22     // Display The Second Message In Position (5, 1)
23     My_LCD.print("72-69");
24 }
25
26 void loop()
27 {
28     // Do Nothing...
29 }
30

```

Figure 10: Code Used to Test LCD Display

y) function where x is the column position and y is row position. For example, (0, 0) corresponds to the first character on the first row and (0, 1) is the same position on the second row. the .print("") function prints something to the display at the cursor point. If

you want to print multiple lines, the cursor must be set for one line, print, then set for the other line, then print again. As shown in Figure 5, this test code printed "ILL beat ISU" in the top row, and "72-69" in the bottom row. When the screen needs to be cleared, the `.clear()` function is used.

3.2.2 RFID Module

Initially we used the Parallax 28440 because it was provided in the ECE 445 Lab. This module has 4 pins: Power (5V), ground, Sin, and Sout. According to the datasheet, provided on Parallax's website [6], the module communicates serially, in 8 bit strings. Sin and Sout are connected to any available GPIO pins. The code used to test this was modified from the test code on Parallax's website [6]. The code is meant to be used with an Arduino Uno. The testing involved printing the tag data to a serial monitor and once that data is found, the test is run again, but also integrating the solenoids by setting one solenoid's switching signal high when the corresponding tag's data is read. This worked using an Arduino Uno. By using three tags, we could control which solenoid unlocked: box 1, box 2, or both. The issue with this was to print to the serial monitor, we needed a Software Serial library that was compatible with Arduino, but not ESP32-S3. The ESP32-S3 has a hardware serial interface, so it is possible another library was needed, but we were unable to find one that worked.

We ordered a second RFID module, the RC522, which we were able to control with the ESP32-S3. This module communicated with the ESP32-S3 using I2C communication. This allowed the RFID to be integrated with the rest of the subsystems and it was able to control user access as described in the User Interface R & V table. It unlocked the required solenoids and displayed the proper message on the LCD screen in less than 5 seconds as described in the high level requirements and User Interface R & V.

3.2.3 Realtime Database and Webpage

There are a few things that happen on the database. The database keeps track of the item stock. When a user with some access scans in their card and when the box is opened then closed, the database will be updated with the scanned user ID, time when the box was opened, and the number of items taken out or returned (if there's any change). If there was some change in the number of items, it should update the total number of items in each box. Everything is displayed real-time on a webpage correctly, without having to refresh the page. These are all the possible scenarios considered and tested:

1. User scans their card, opens the box but does not close the box: the web screen will display an alert message.
2. User scans their card, opens the box, does not take/return items, and closes the box: The database is updated with the scanned user ID, time when the box was opened, and the number of items taken out or returned as 0.
3. User scans their card, opens the box, takes out one or more items, and closes the box: the database is updated with the scanned user ID, time when the box was opened,

and the number of items taken out.

4. User scans their card, opens the box, returns items, and closes the box: the database is updated with the scanned user ID, time when the box was opened, and the number of items returned.
5. User breaks open the box without scanning their card: the web and LCD screen display an alert message.

The webpage retrieves data from the real-time database constantly and is updated constantly as well. The page displays item stock of each inventory box and keeps track of the access logs. Each entry of the access log consists of the time the boxes are accessed, ID of the user accessed the boxes, and change in number of items in both boxes. This is verified by simulating all possible scenarios of different users checking out or returning items. In response to the data privacy issues, we did not store any sensitive user data to the database nor the webpage and we hid the user access logs to the public. To ensure that access log was only visible to administrators, it required a passcode to see the access logs.

3.3 Sensor System

The Sensor subsystem verification was essential in the design of this project. The load cells' calibration factor had to be accurate to lead to accuracy in outputted readings. According to TAL220 datasheet [16], 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 [16] 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.

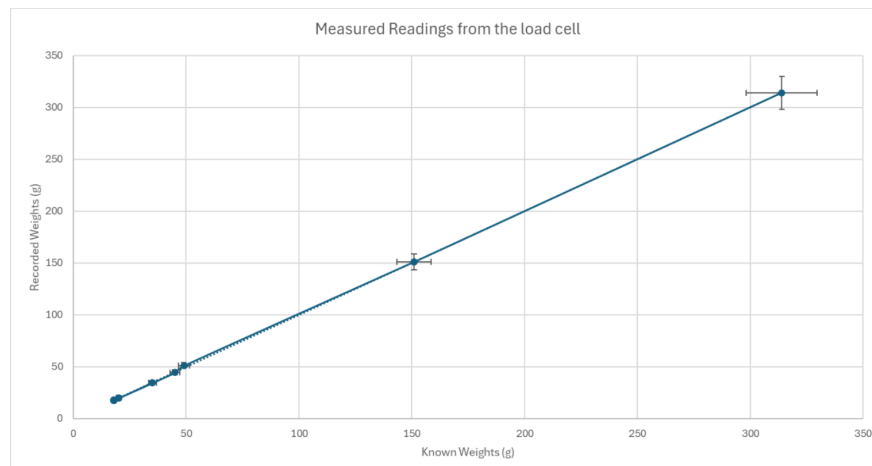


Figure 11: Load Cell Calibration

Table 3: Sensor System R & V

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.	<ul style="list-style-type: none"> - 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 re-calibrated.
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.	<ul style="list-style-type: none"> - After connecting the load cell to the HX711 and the microcontroller, 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.

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. It can be seen that a linear relationship exists which indicates that using the load cell for our inventory tracker is ideal. However, as shown in Figure 11, 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 [16] 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 .

In our design proposal, the following were the requirements and verification for the sensors' subsystem, which have been met.

3.4 Power System

The power system requirements and verification are shown in Table 3. The power system was not verified in this project. This is due to mistakes in PCB design and delays with ordering. In the demonstration, the ESP32-S3 development board was used, being powered by a micro USB cable connected to a computer. The 5 V and 3.3 V pins were used to power the various components that did not need 12 V. The barrel connector worked properly and was used to power the solenoids. The input output relationship of the buck converter is $V_2 = DV_1$ where D is the duty ratio. In future work we want to integrate this converter to the overall system, and would need to operate it at a 0.4167 duty ratio.

Table 4: Power System R & 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.

3.5 Control System

All the requirements and verification of the control system are shown in Table 5. All requirements are met to fulfill the control of the locking mechanism and user access control as well as configuring the communication with the real-time database.

Table 5: Control System R & V

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 microswitch.	To verify this, the box will be opened using a key, with the signal being sent to the MOSFET set LOW. The LCD and database should display a warning message if it works properly.

3.5.1 Weight to Count Calculation


The count of items is done in the microcontroller. Once the user finishes using the inventory boxes (closes both boxes after opening them), the microcontroller stores the last weights measured in the boxes. Using the known weights of a single item in each box, we calculate the total number of items in both boxes by dividing the weight measured by the known weight of a single item. One crucial thing to keep account was to round the

number of items to the nearest whole number. This was done through casting `float` to `int` using the typecasting function `(int)`.

For the calculation of change in number of items for access logs, we subtract the new item count from the previous item count for each box and post them to the database along with the user ID and timestamp.

3.5.2 ESP32 WiFi Connection to Database Update

Since the WiFi provided by UIUC is a WPA2 (WiFi Protected Access 2) EAP, the built-in functions in Arduino esp32 library were not sufficient. The connection could be made using functions from the latest Espressif library *esp_eap_client*. The WiFi connection was verified using the built in status signal of the esp32 library called `WiFi`. The success was checked by retrieving and printing out the IP address and the quantitative number representing strength of the connected WiFi (RSSI – Received Signal Strength Indicator) on the serial monitor.



```
15:51:35.681 -> Chip ID: 5407144
15:51:54.373 -> .
15:51:54.373 -> IP address:
15:51:54.373 -> 172.16.9.54
15:51:54.373 -> RSSI: -59
```

Figure 12: RSSI of *eduroam* wifi network from ESP32

With the secured internet connection, esp32 was able to configure a anonymous user to post data to the Firebase Realtime database within a second. This was done with *Firebase_ESP_Client* library to correctly update the Realtime database. Multiple dummy data was successfully created and updated to verify the

3.5.3 User-Access Control

The access control of the storage boxes is done through RFID scanning. When the user scans their RFID card or an RFID tag, the microcontroller should be able to send the correct signal to control the solenoids of the correct boxes. To verify that the microcontroller can handle all possible scenarios, there were 4 cases tested (Figure 13): one persona with access to both boxes, two personas with access to one box (one for box 1 and one for box 2), and one persona with no access to any of the boxes (including scanning with incorrect type of RFID cards).

3.5.4 Theft Alarm System

One variable is set to constantly monitor and post to the database once some unauthorized access is made to the inventory boxes. This was determined by constantly monitoring the status of scanned user ID and the signals from the microswitches (box status). If

	Box 1	Box 2
Case 1	o	o
Case 2	o	x
Case 3	x	o
Case 4	x	x

Figure 13: 4 cases tested for user-access control

any of the signals from the microswitches goes high while scanned user ID is -1 , the variable `trigger_alarm` is set `true`, and posted to the database instantly, while changing the LCD to display "Theft Alert". This trigger is cleared by an "admin tag," Which sets the variable to `false` and clears LCD.

4 Costs

4.1 Cost Analysis

4.1.1 Labor of Costs

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:

$$50 \times 8 \times 10 \times 3 = \$12,000$$

There's also labor done by the machine shop, making the actual inventory boxes with load cells, solenoids, and microswitches attached. Generally the engineering machine shop's rate is \$59/hr, assuming it took maximum of 5 hours to build the boxes, it comes out to be \$295.

The total labor cost is \$12,295.

4.1.2 Cost of Parts

Table 6: Cost of Parts

Description	Manufacturer	Part Number	Quantity	Unit Cost (\$)
Solenoid locks	Adafruit	5065	2	\$7.50
Load cell	Sparkfun Electronics	4540	2	\$3.95
Microswitch	Omron Electronics	Z4613-ND	2	\$6.32
RFID Module	SunFounder	RC522	1	\$5.47
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				\$52.455

4.2 Total Cost

Summing up the labor cost and the cost of all parts, the total cost for the project is \$12347.46.

5 Conclusion

The project fulfilled all high level requirements. The solenoids unlocked when the corresponding RFID tag was scanned. The web database updated when items were added and taken out of each box. The alarm system send an alert to the database and LCD display when a box was opened without an RFID scan. However, the main PCB was not able to be used. This was caused by footprint selection errors in the first wave of ordering. This created delays in testing. When the PCB was constructed, soldering errors connecting multiple pins on the UART bridge chip caused the microcontroller circuit not to work. The PCB was able to incorporate 12 V power and the solenoid MOSFETs in the final demonstration.

This project shows how an inventory tracking system can work and how users can interface with it. Future work for this project can include adding a buzzer or other noise and text message to the administrator for the alarm system for better theft prevention. Along with that, the system can be scaled up by adding more boxes and load cells so the system can work with a larger inventory.

References

- [1] IEEE. "IEEE Code of Ethics." (2016), [Online]. Available: <https://www.ieee.org/about/corporate/governance/p7-8.html> (visited on 02/08/2020).
- [2] DeepBlueMbedded. "ESP32 LCD Display 16x2 Without I2C – Arduino Core." (2024), [Online]. Available: <https://deepbluembedded.com/esp32-lcd-display-16x2-without-i2c-arduino/> (visited on 04/24/2024).
- [3] E. Systems. "ESP32-S3Series Datasheet." (), [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-s3%5C_datasheet%5C_en.pdf (visited on 04/23/2024).
- [4] GitHub. "Espressif KiCad Library." (), [Online]. Available: <https://github.com/espressif/kicad-libraries> (visited on 04/24/2024).
- [5] M. Technology. "Dual 3A Peak Low-Side MOSFET Drivers." (), [Online]. Available: <https://ww1.microchip.com/downloads/aemDocuments/documents/APID%5C%5C%20/ProductDocuments/DataSheets/MIC4423-4-5-Dual-3A-Peak-Low-Side-MOSFET-Drivers-DS20006638.pdf> (visited on 04/23/2024).
- [6] P. Inc. "RFID Read/Write Module – Serial." (), [Online]. Available: <https://www.parallax.com/product/%5C%5C%20rfid-read-write-module-serial/> (visited on 04/24/2024).
- [7] R. N. Tutorials. "ESP32 with Load Cell and HX711 Amplifier (Digital Scale)." (), [Online]. Available: <https://randomnerdtutorials.com/esp32-load-cell-hx711/> (visited on 02/05/2024).
- [8] RepRap. "Mechanical Endstop." (), [Online]. Available: https://reprap.org/wiki/Mechanical_Endstop (visited on 02/05/2024).
- [9] SparkFun. "PARALLEL BEAM LOAD CELL." (), [Online]. Available: <https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/TAL220M4M5Update.pdf/> (visited on 01/30/2024).
- [10] V. Intertechnology. "IRF510 Power MOSFET." (), [Online]. Available: <https://www.vishay.com/docs/%5C%5C%2091015/irf510.pdf>. (visited on 04/23/2024).
- [11] T. Instruments. "WeBench Circuit Designer." (), [Online]. Available: <https://www.ti.com/tool/WEBENCH-CIRCUIT-DESIGNER> (visited on 04/23/2024).
- [12] T. Instruments. "TPS56339 4.5-V to 24-V Input, 3-A Output Synchronous Buck Converter." (), [Online]. Available: https://www.ti.com/lit/ds/symlink/tps56339.pdf?HQS=dis-dk-null-digikeymode-dsf-pf-null-ww&ts=1714417222169&ref_url=https%253A%252F%252Fwww.ti.com%252Fgeneral%252Fdocs%252Fsuppproductinfo.tsp%253FdistId%253D10%2526gotoUrl%253Dhttps%253A%252F%252Fwww.ti.com%252Fgit%252Fgpn%252Ftps56339 (visited on 04/23/2024).
- [13] D. Incorporated. "AZ1117 LOW DROPOUT LINEAR REGULATOR WITH INDUSTRIAL TEMPERATURE RANGE." (), [Online]. Available: <https://www.diodes.com/assets/Datasheets/AZ1117L.pdf> (visited on 04/23/2024).
- [14] N. Semiconductors. "MFRC522 Standard Performance MIFARE and NTAG Frontend." (), [Online]. Available: <https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf>.
- [15] M. Technologies. "Dual 3A Peak Low-Side MOSFET Drivers." (), [Online]. Available: <https://ww1.microchip.com/downloads/aemDocuments/documents/>

APID/ProductDocuments/DataSheets/MIC4423-4-5-Dual-3A-Peak-Low-Side-MOSFET-Drivers-DS20006638.pdf.

- [16] A. Semiconductor. "24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales." (), [Online]. Available: <https://www.digikey.com/htmldatasheets/production/1836471/0/0/1/HX711.pdf>.