Federal University of Technology of Paraná – UTFPR
Academic Department of Electronics – DAELN
Academic Department of Informatics – DAINF
Computer Engineering
Integration Workshop 3 (EEX23) – S71 – 2024/1

Technical Report
Hectoliter Scale

Ellejane Camila Marques Ferreira dos Santos ellejeane@alunos.utfpr.edu.br
Franziska Romani Furtado franziska@alunos.utfpr.edu.br
Guilherme Saito guilhermesaito.2001@alunos.utfpr.edu.br
João Pedro da Matta Galera da Silva joaosilva.2016@alunos.utfpr.edu.br
Mariana Gomes da Luz marianagomesluz@alunos.utfpr.edu.br
Thiago Henrique Xavier Medeiros thiagomedeiros@alunos.utfpr.edu.br

June 2024

Abstract

The Brazilian state that produces most wheat is Paraná [1]. In this context selling wheat is a big part of Paraná's economy. To sell wheat of the harvest the wheat has to be classified via a sample. Many parameters can be used to separate a higher-quality wheat from a lower one. The one that was focused on in the project was the hectoliter weight, which is a density measurement. To acquire this parameter from a sample it is needed to weight a defined volume of the sample taken from the harvest. The process of acquiring the hectoliter weight by hand is very long and vulnerable. The team looked at the vulnerability and complexity of acquiring the hectoliter weight of a wheat sample and developed a solution. The system developed automated the acquiring of the hectoliter weight provided a sample and safe keeping of measurement information.

1 Introduction

In the agroeconomic industry, the measurement of hectoliter weight is used to classify the quality of grains. Since the price is based on the quality this parameter is important[2]. Hectoliter weight is the amount of kilograms in 100 liters of wheat, so it is a measurement of density[3] The most recent video we could find of the measurement process[4] as an example was very complex and required lots of manual work from the measurement professional taking the hectoliter weight of the sample. While searching for other methods of measurements on the internet we found several indicating a system of 3 stages[5]. Looking at these examples, we better understand the process and how to make it automatic. The manual way of acquiring the hectoliter weight is to get a sample from a harvest, or part of it, one sample of 1kg for every 30 tons stored for

Hectoliter Weight	Classification
100kg/hl or more	Too Heavy
75kg/hl or more	Good
65kg/hl or less	Light
50kg/hl or less	Too Light

Table 1: Example of Table of Hectoliter Weight for Wheat Classification

example. From this sample, the process of acquiring will be used to get the values for the whole wheat harvest. First, the sample is poured into a cup-shaped container, that has a divisor between the upper part and the lower part. The divisor is then taken out, letting all the grains previously poured into it fall into the lower part of the recipient. The divisor is then placed again as to close all the grains currently in the bottom part of the container. The grains from the upper part are removed so that the grains that occupy only the lower part of the container are left. This volume is already known to the user, and it is used on the calculations. It is important that the volume is correct, since it is based on it that the hectoliter weight is calculated. The bottom container is then weighted and the hectoliter weight is acquired by calculating the density of the grains by dividing the weight by the already known volume of the bottom part of the container. The hectoliter weight of that sample is then compared to others using a table, like shown in Table 1, to classify the hectoliter weight properly. In Table 1 it is seen that there is a hectoliter weight range in which the wheat is considered of good quality. A high hectoliter weight is usually related to a higher quality[6], but if it is very high this points to a grain that the sample has other things besides the grain on it. On the other hand, if it is shown that the wheat has a very low hectoliter weight, this indicates a high porosity grain that equates to a lower quality grain. We saw some areas that could use improvement, one of them being the automation of the process itself, and the other the accountability of the measurement professional for errors or fraud.

In the process is easy to see that while the user is acquiring the hectoliter weight in the manual way the apparatus could very easily be managed in a wrong way leading to wrong measurements. The person could put not enough sample to fill the tube this way not getting the correct volume of the sample, for example, this would mean a lower weight and a lower Hectoliter Weight. Someone could temper the scale, by adding something to the grains increasing the weight on the scale, leading to a false higher hectoliter weight. By automizing the procedure, using an IoT(Internet of Things) approach, with sensors to warn for errors, and a database to obtain and store directly the measurements, the security against such tampering/errors can be greatly improved.

1.1 Overview

The Hectoliter Scale system, composed by the Android App, Cloud Server, and the machine itself, is designed to improve repeatability, prevent fraud, and reduce human interaction in the hectoliter weight measurement process. Figure 1 shows the system‘s Block Diagram with each part’s main components.

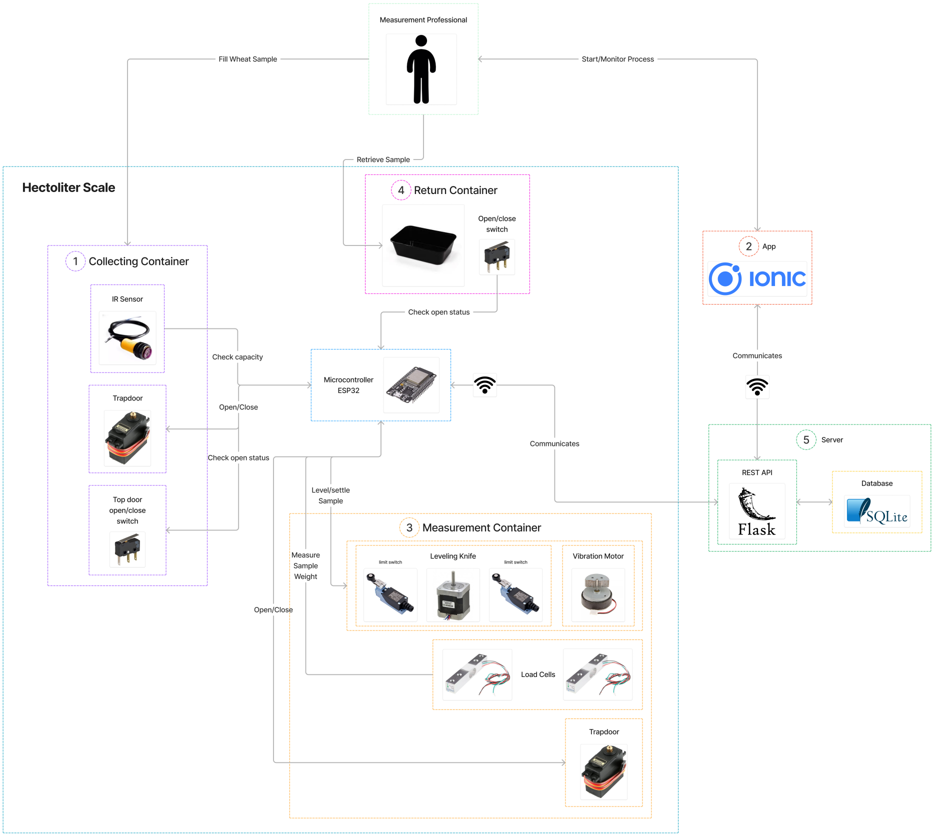

Figure 1: Block Diagram

To better visualize the following explanation a photo of the machine with pointers to the main components mentioned is presented in Figure 2

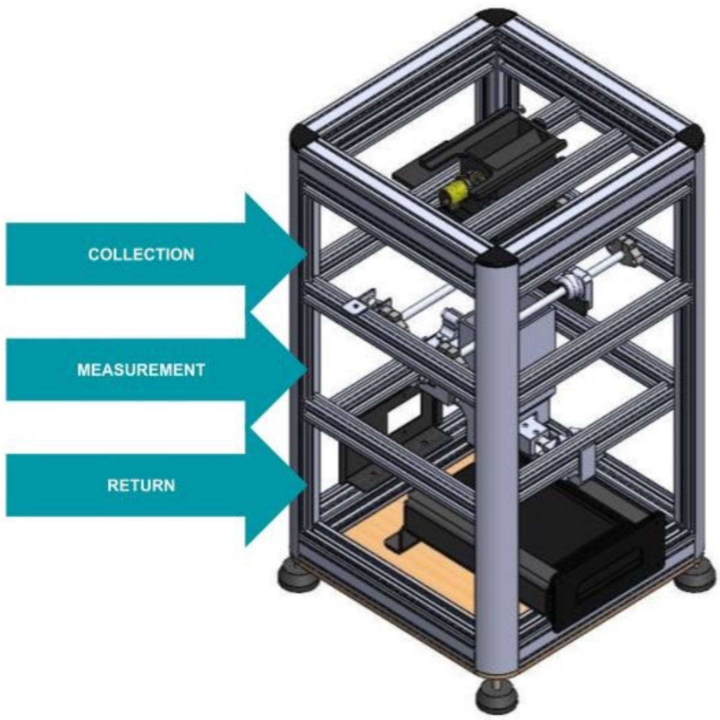

Figure 2: Photo of Machine with its Components Labeled

To start the process, the measurement professional will pout the wheat grains sample into the machine. From the opening they will fall into the Collecting Container. The user will then make a request on their phone to start the process via the custom-developed Android APP. The microcontroller (an ESP32) that controls the machine is alwasys checking the cloud based servers for a new measurement request. If it identifies a new request for a measurement process it begins to check if the machine is ready. First of all it checks if the Collection Container door is closed, since it is required for it to not be open to begin the process. The doors position is known by a end switch that is pressed when the door is fully closed.

If the Collection container door is closed the machine has its first error checking stage, that is checking if the wheat poured into it was enough to take the hectoliter weight correctly. This checking is made trough an IR sensor, that checks if the level of the wheat on the Collection container corresponds to the volume needed for the process to go as intended.

When the process starts the first LED of the bottom line is on to indicate that the sample is on the Collection Container. The wheat sample is then released from the collection container to the measurement container. This release is done by a trapdoor mechanism composed of a servo motor and gears that opens the bottom panel of the Collection Container, making the grains fall to the Measurement Container.

If the servo gets stuck the potentiometer attached to it will indicate to the ESP32 that the movement was not done, this is another error checked in the proccess.

In the Measurement Container the grains need to be leveled to its border in order for the volume to be completly and fully complete without any overfill.

On this next phase the sample is on the Measurement Container, and the yelow LED is turned on to show that the sample is in this part of the process.To make the grains settle and not overfill the Measurement Container the machine has two mechanisms.

The first tool to achieve a leveled and correctly filled Me a vibration motor that vibrates the container to remove empty spaces between the grains inside it.

The seccond mechanism that asssists on the volume of the grains being correct is a leveling knife, attached to a belt that moves horizontally with the help of a stepper motor.

This knife passes on the border of the Measurement Container to remove any excess grains that overfill the container to fall to the Return Container.

The knife has 2 end swtiches so that when the knife gets to either end they indicate that it has reached it. If it doesn't reach it it indicates that the step motor or knife has gotten stuck, this way the ESP32 knows to raise that this has happened during the process.

The weighing process starts when the leveling is complete, and the volume of wheat is correct inside the Measurement Container.

Two load cells attached underneath the container measure the weight of its contents.

The microcontroller collects the weight data and calculates the hectoliter weight of the sample by dividing the weight by the volume of the Measurement Container.

It then opers the Measurement Container's trapdoor (a similar mechanism to the previous container's trapdoor), making the sampled wheat fall to the Return Container.

The ESP32 then sends all the measruement process information, including who started the process and errors, if they occur, to the server and they are saved in the database.

When the ESP32 is able to send the measurement the green LED turns on to indicate to the user that the process has ended and that the Return Container can be retrieved.

If the Return Container is taken out during the process it raises an error. This is done in a similar way as the Collection Container door, with a end switch on the end of the Return Container's rails. If it's not inside the Hectoliter Scale the process won't start and the app will notify the operator to put it back on the machine, like the Collection Container door.

In addition to starting the process, in the app, the user can see the history of measurements with the operator, the hectoliter weight, and any errors that happened during each process. The server is composed of a REST API that acts as a middleman between the app, the machine, and the database. It verifies the users, manages the processes running, and retrieves the measurement history.

2 Project Specification

We attempted to create a project based on problems2.1, that turned in the needs of the client2.2 and then requirements2.3 for the system that we were going to plan.

2.1 Client Problems

The following problems affected the process that we took into account while developing the system.

- **CP01**: The company needs to obtain the hectoliter weight of wheat for sale, as failure to classify per hectoliter weight can result in incorrect sales, lack of trust in the company, and decreased sales.
- **CP02**: The company cannot damage the sample of wheat used, because must be kept for audits, if the classification cannot be redone with the same sample in the audit the company could lose its license.
- **CP03**: The company needs to save the measured hectoliter weight of the sample for future audits, to have information about the classification for which that harvest was sold, and to have a basis for comparison during the audit.
- **CP04**: The company hopes to reduce the number of banal errors, such as calculation errors, so that obtaining the weight does not require so much diligence on the part of the measurements professional, thus reducing the need for hard work and the necessary technical knowledge.
- **CP05**: The company hopes to keep the time comparable to that of the manual process so that measurements do not take more time than they currently do.
- **CP06**: The company needs to weigh several harvest samples during the day.
- **CP07**: The company hopes to reduce unreported cases of undue human interference in the sample measurement process, and fraud so that there is a record and that management can have more control over its measurement professionals

2.2 Client Needs

The following necessities were taken out of the problems that were caused by the client's problems.

- **CN01**: The process manager needs a system to obtain the hectoliter weight of each sample.
- **CN02**: The process manager needs a system that returns the undamaged sample that was used in the process to the meter.
- **CN03**: The measurement manager needs a system that allows information on measurements already taken to be viewed.
- **CN04**: The measurement professional needs a system that performs calculations automatically and consistently.
- **CN05**: The measurement professional needs a system that takes no more than 5 minutes per sample measurement.
- **CN06**: The process manager needs a system that records any interference that occurs in the measurement process of a sample.

2.3 Requirements

In the website of the project [7] it is detailed what is the main area of the requirement and what other requirements are related to it. In this document, only the itemized version of the requirements will be presented.

3 Development

The project's development process will be described in this section, with technical details and related difficulties. Extra details can be found on the project's blog[8].

3.1 Mechanical

The first part of the development was the design and assembly of the mechanical structure of the project. It started by creating the CAD project using SolidWorks, with it the group was able to create all the necessary diagrams and a structure that fully encapsulated and sealed all the motors, sensors, and actuators needed for the system to work as specified by the requirements. On Figure 3 the 3D view of the final mechanical design made in Solid Works is presented.

FR01: The system must have information LEDs (ON/OFF, WIFI, Process progress) for the user on the front of the machine, so that the user can see it from 1m away.
FR02: The machine must have 3 containers, one for each stage of the weighing process (Collection of the sample; Measurement of the sample; and Returning of the sample)
FR03: The collection container must have a volume of 450ml +/-50ml.
FR04: The machine must detect if the door of the collecting container is opened or closed.
FR05: The collecting container door must have protection, a deflector, so that hard objects bigger than 4cm x 11cm x 3cm can not pass through the opening of the Collection Container.
FR06: The machine must detect if the wheat put in the collection container is at a level of 350 ml +/- 120ml
FR07: The measuring container must have a volume of 200ml +/- 20ml
FR08: The return container must be at least 600ml +/- 20ml to fit the entire sample placed in the machine.
FR09: The returning container must be removable.
FR10: The machine must detect if the return container is on the machine.
FR11: The energy supply must come from an outlet power(127V) that will turn on the power supply of 12V that feeds the internal mechanisms of the machine.
FR12: The machine must detect if the motors move to the set position when requested within 30s.
FR13: The machine must level the sample on the measurement container. (The sample can't be more than 1cm higher than the border when measuring the weight)
FR14: The machine must measure the sample inside the measurement container's weight, with 5g precision.
FR15: The machine must have a cooler that will be on while the power is on.
FR16: The machine must not continue a measurement process when the power is reconnected.
FR17: The machine must calculate the hectoliter weight and attempt to send all measurement information to the database.
FR18: The process must not start if the collecting container door is not closed, or if the return container is not on the machine.
FR19: If the microcontroller has an internet connection and is not working on a measurement, it must query the database for new measurement requests, every 20s or less.
FR20: If any errors are detected in the measurement process it must be in the measurement information. (If the collecting container door is opened during the process; If there wasn't enough wheat put on the machine for the measurement process to occur correctly; If the servo motors gotten stuck for more than 20 seconds; If the step motor hasn't been able to finish its movement for more than 30 seconds)
FR21: The app must authenticate the user before letting them access the machine's information and function.
FR22: The app should display a wait screen while the process is carried out.
FR23: The app should display, at the end of the process, all information about the measurement.
FR24: The app must have a button to request the start of a process.
FR25: The app must have a history screen, that shows all previous measurement results.
FR26: The app must allow the user to filter the history screen data by dates and display a list of measurements within that period.
FR27: The app's start button must only work if the machine is available (has returned the measurement information from the previous request, if there was one).

Table 2: Functional Requirements List

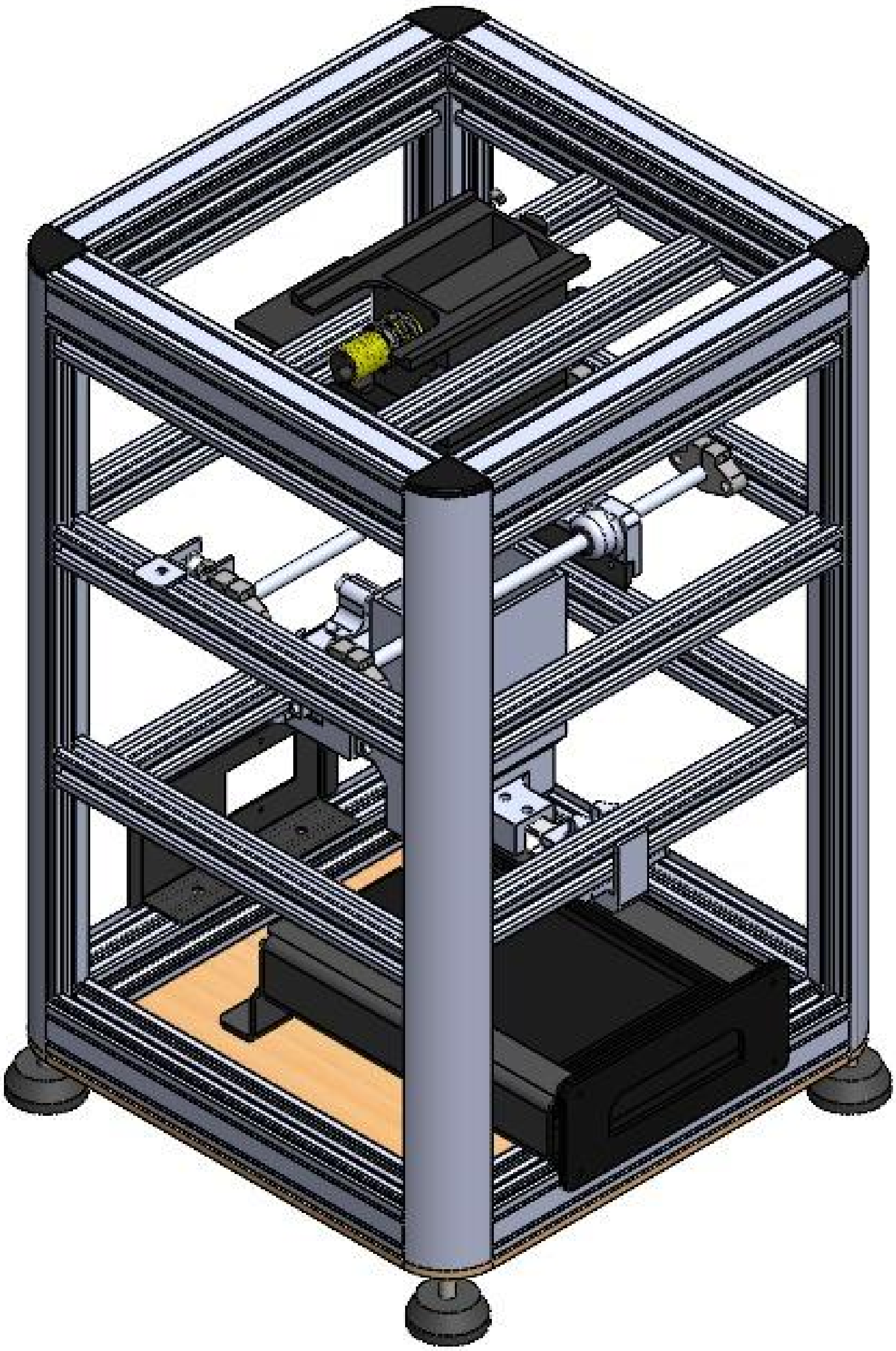

Figure 3: Isometric View of the 3D Design

The structure was made using 20mm x 20 mm aluminum profiles, angle brackets, screws, and a wooden base. To enclose the machine acrylic sheets were fitted to the notches of the aluminum profiles. Internally more aluminum profiles were used to create a base for fixing the 3D-printed containers and parts. The mechanical project was divided into several parts which will be described in the following subsections.

3.1.1 Collection Container

The collection container, a 3D-printed assembly, and the entry point of wheat samples can be summarized in 4 components a door on top, the deflector, the sample container, and a trapdoor on the bottom. The collection container itself is an open box that doesn't have the top or bottom. The four walls enclose a volume of about 450ml, and this container is responsible for getting the whole sample dumped by the system user. Figure 4 presents the Hectoliter Scale with only the Collection Container mounted on the Structure.

Figure 4: Collection Container

At the top of the collection container, there is a slidable door that contains the sample already put in the collection container. The sliding door has two main positions, fully open (Figure 5), and fully closed (Figure 6).

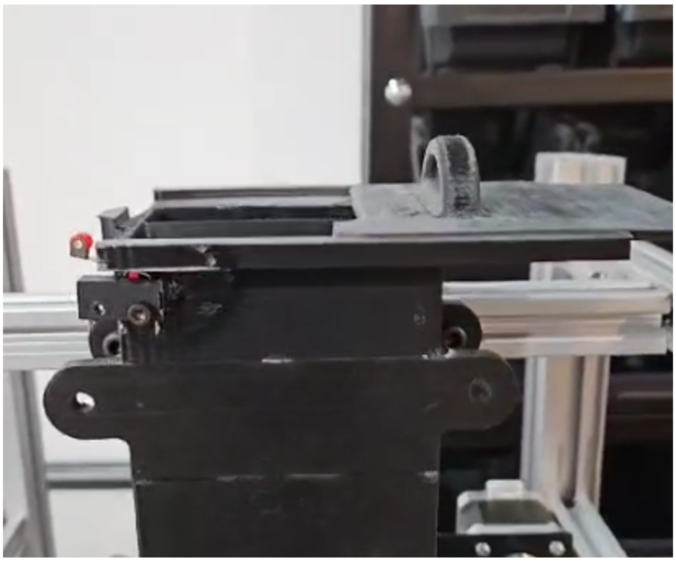

Figure 5: Sliding Door Fully Open

Figure 6: Sliding Door Fully Closed

A limit switch sensor behind the sliding door is used to detect if the collection container is fully closed (Figure 7).

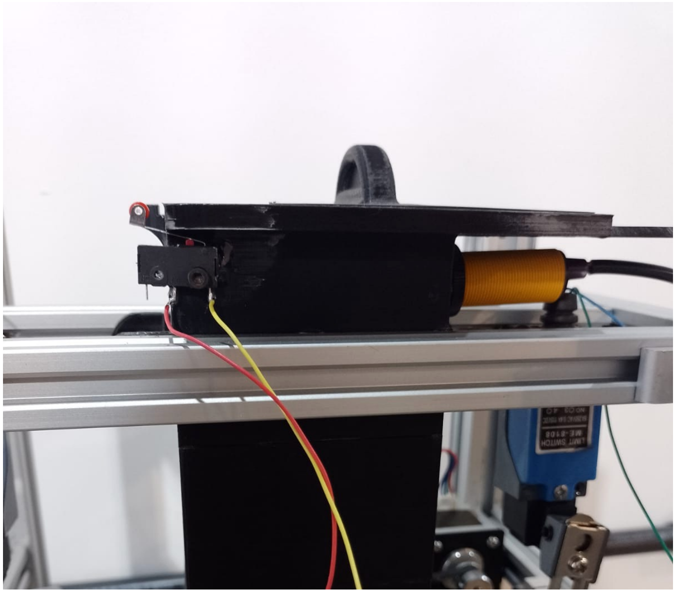

Figure 7: Limit Switch Sensor of the Collection Container Sliding Door

The collection container has a deflector right under the collecting container door. This deflector blocks people from putting their hands or other objects in the machine (Figure 8).

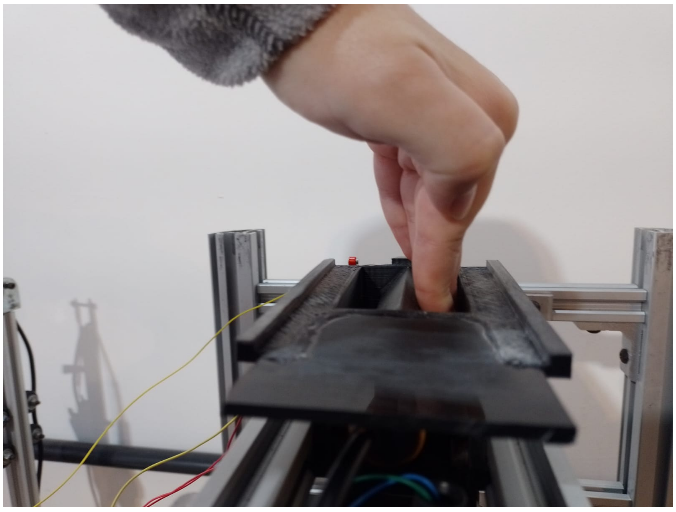

Figure 8: Deflector

Just under the deflector, an IR sensor was built into the collection container (Figure 9). This sensor was added to trigger when the sample level gets to its height, this way it is known when the sample volume put in the container was enough to proceed with the process correctly.

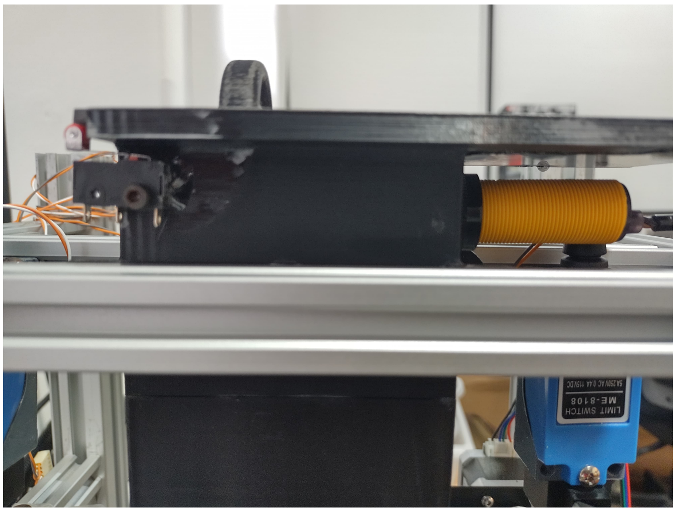

Figure 9: IR Sensor in the machine (in orange)

The bottom trapdoor of the container is moved by a servo hooked on gears that move the door from side to side. The trapdoor is presented in Figure 10 in the closed position.

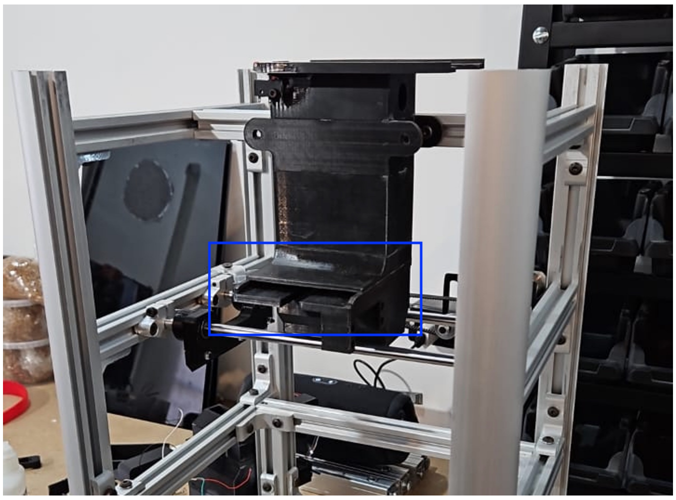

Figure 10: Trapdoor at the Bottom of Collecting Container

3.1.2 Measurement Container

The Measurement container is also a 3D-printed "box", with an open top turned to the bottom of the collection container. The bottom also has a trapdoor mechanism similar to the bottom of the collecting container, it uses a sliding door and a servo motor to move it from side to side (Figure 11).

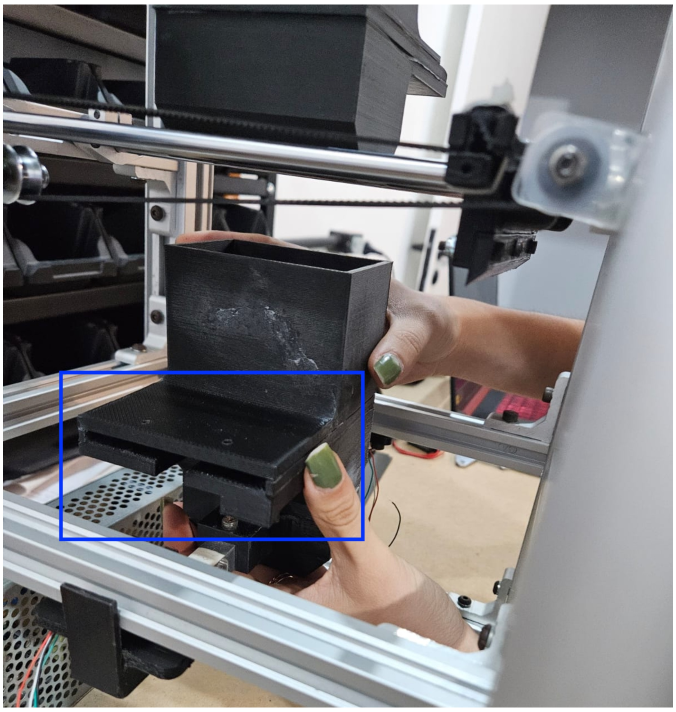

Figure 11: Trapdoor at the Bottom of Measurement Container

The other motor attached to the measurement container is the vibration motor, which is necessary to seat the sample better in the measurement container. It can be seen attached to the measurement container in its 3D printed cover in Figure 12.

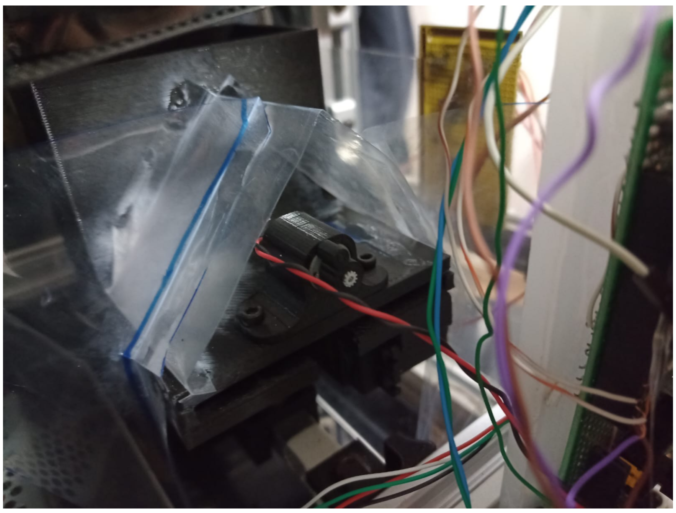

Figure 12: Vibration Motor

The last two things attached to the measurement container (the most important ones) are the load cells, which are used to obtain the weight used in calculating the hectoliter weight. They are attached to the load cells, as presented in Figure 13, and support the entirety weight of the measurement container when inside the Hectoliter Scale structure.

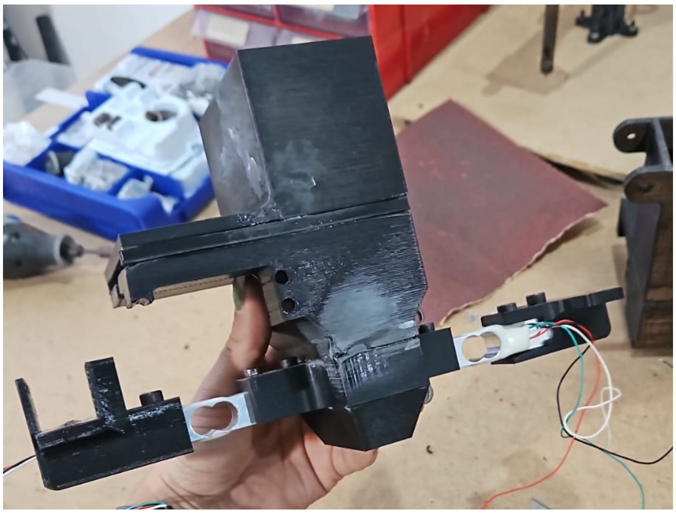

Figure 13: Load Cells with Supporting Tabs for the Measurement Container

3.1.3 Leveling Mechanism

The leveling mechanism is one of the critical parts of the system, which guarantees that the sample does not overflow the measurement container when measuring the weight. When the wheat grains fall from the collection container to the measurement container the sand cone phenomenon happens, and a conical pile is formed. This volume of wheat that is higher than the measurement container is not constant, therefore the leveling of it is important to have the correct volume of grains weighted, otherwise, the measurement would be compromised. The leveling mechanism is composed of 5 main parts, the smallest part is the 3D-printed leveling knife presented in Figure 14, which is used to pass from border to border of the measurement container.

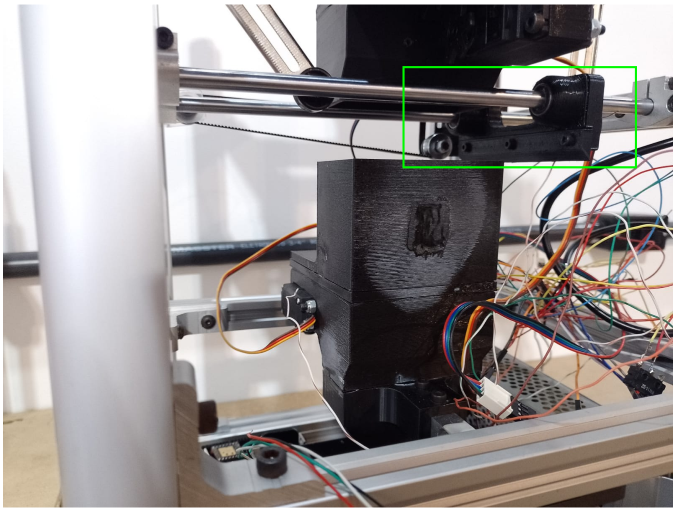

Figure 14: Leveling Knife

The leveling knife is attached to a rubber belt connected to a step motor. When the step motor turns, the leveling knife moves horizontally, leveling up the grains that are over the borders of the measurement container (Figure 15).

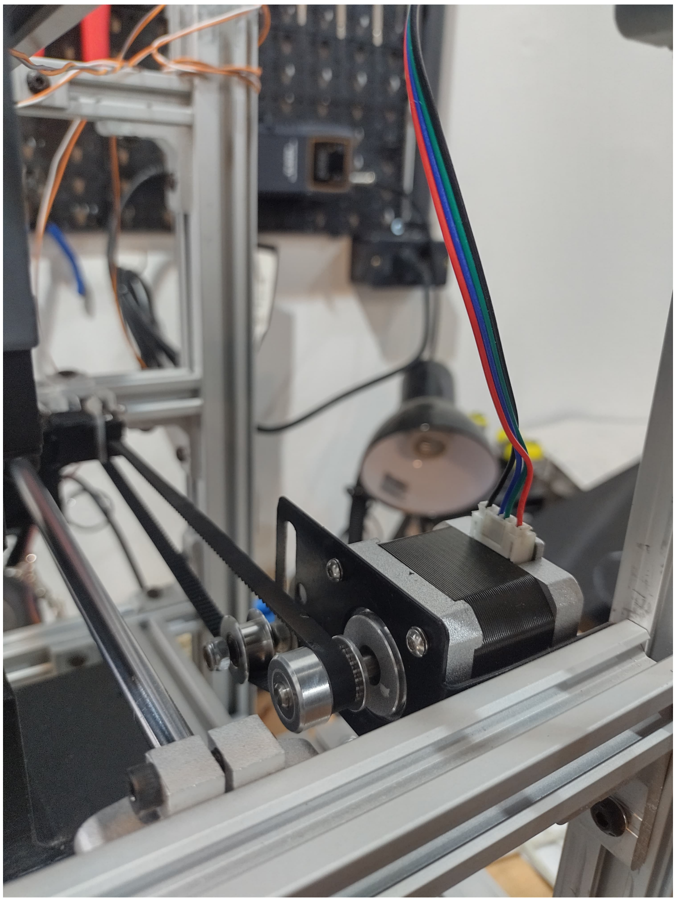

Figure 15: Belt of Step Motor

The leveling knife moves horizontally, attached to the belt. On each of the sides of the belt, there are two limit switches, that serve as sensors for the system to know that the leveling knife got to the end of the movement of each side, presented in Figure 15. The leveling mechanism has the limit switches in symmetry as can be seen in Figure 16.

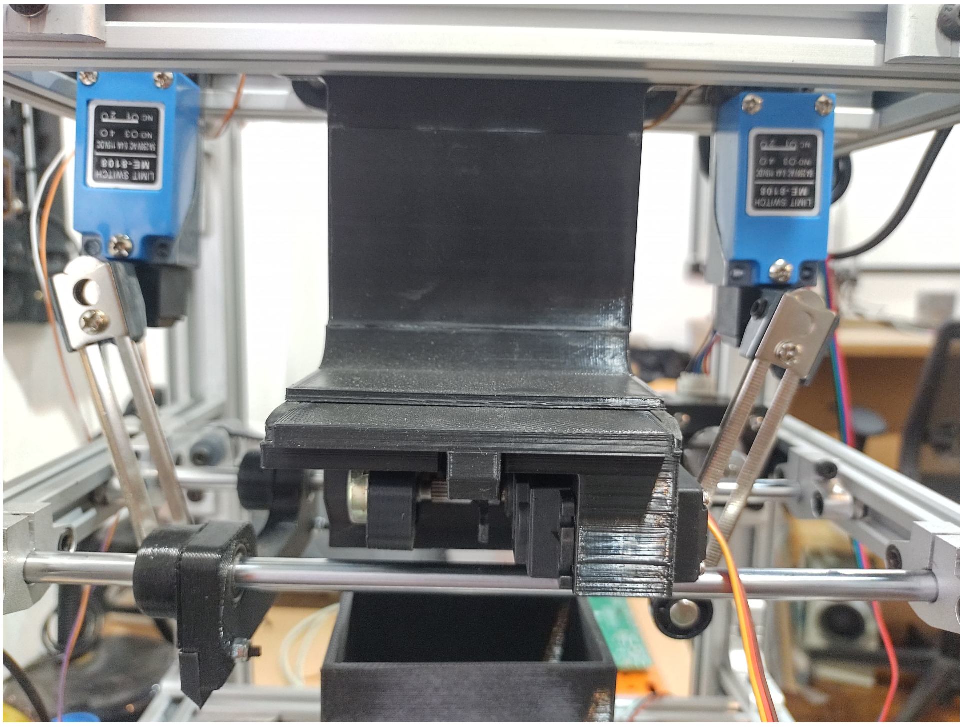

Figure 16: Structure of Leveling showing the limit Switches

As a further example, is presented the leveling knife hitting the switch, the arm of the switch can be seen in a different position in Figure 17 than that of Figure 16.

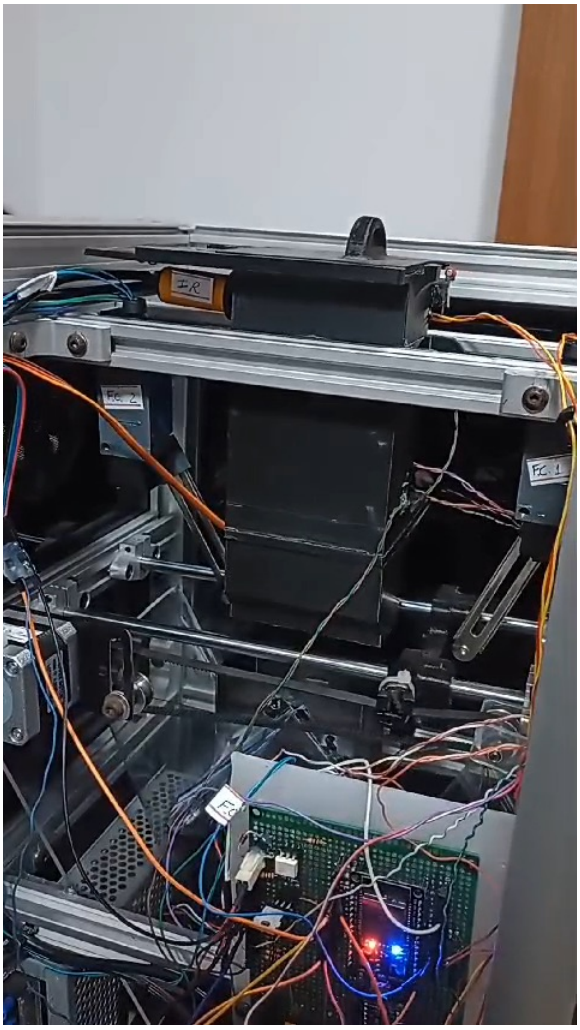

Figure 17: Leveling Knife Hitting the Limit Switch

3.1.4 Return Container

The return container is important as it recovers all the wheat put on the machine to return the sample to the user safely. The return container is composed of the return container itself that can be fully removed from the machine and was made by bending acrylic (Figure 18).

Figure 18: Return container Fully taken out of the Machine

The return containers drawer compartment has a limit switch, that is used to know if the return container is currently on the machine, as seen in Figure 19, its placement is at the end of the return container rails.

Figure 19: Limit Switch of Return Container

3.2 Hardware

3.2.1 Hardware

For the development of electronic circuits, it was studied first of all the power supply needed for all the components. Most used a 5V supply, like the ESP32 and the vibration motor, so there was a need for a 5V power supply. The load-cell drivers used a 3.3V power supply, for this, we could use the 3V3 pin that powered it. As for the step motor that requires a 12V power supply it was controlled via a stepper motor driver, that received the control signals with 5V and outputted 12V signals. For this, it was powered with 5V and 12V. After selecting components to be able to have all necessary voltage values needed to supply all the electronic components that were going to be used the power circuit was defined. Then the ESP32 was studied to make sure that the signals that were going to be had an available pin that was able to send the signal, like if there were enough PWM pins. Since there were enough pins for every need there wasn't a need to use a second ESP32. It was seen through that, besides the stepper motor that was already known that a driver would have to be used to activate it, the vibration motor would also have to have a circuit to activate it, since the drawn current was too high for the ESP32 pins to handle (max seen drawn current was 650mA). A transistor was used with an optocoupler to protect the ESP32 and be able to activate the motor.

Circuit Diagram Before making the PCB, the diagram was created for tests on the breadboard to confirm that the circuits would work as intended, as it is easier to fix mistakes on the breadboard than on the PCB. The diagram shown in Figure 20 was made after studying which pins could be used for the input and output of the ESP32, how to acquire the signals from the sensors correctly, and how to activate the motors without sending the current from the ESP32, as it would burn a pin.

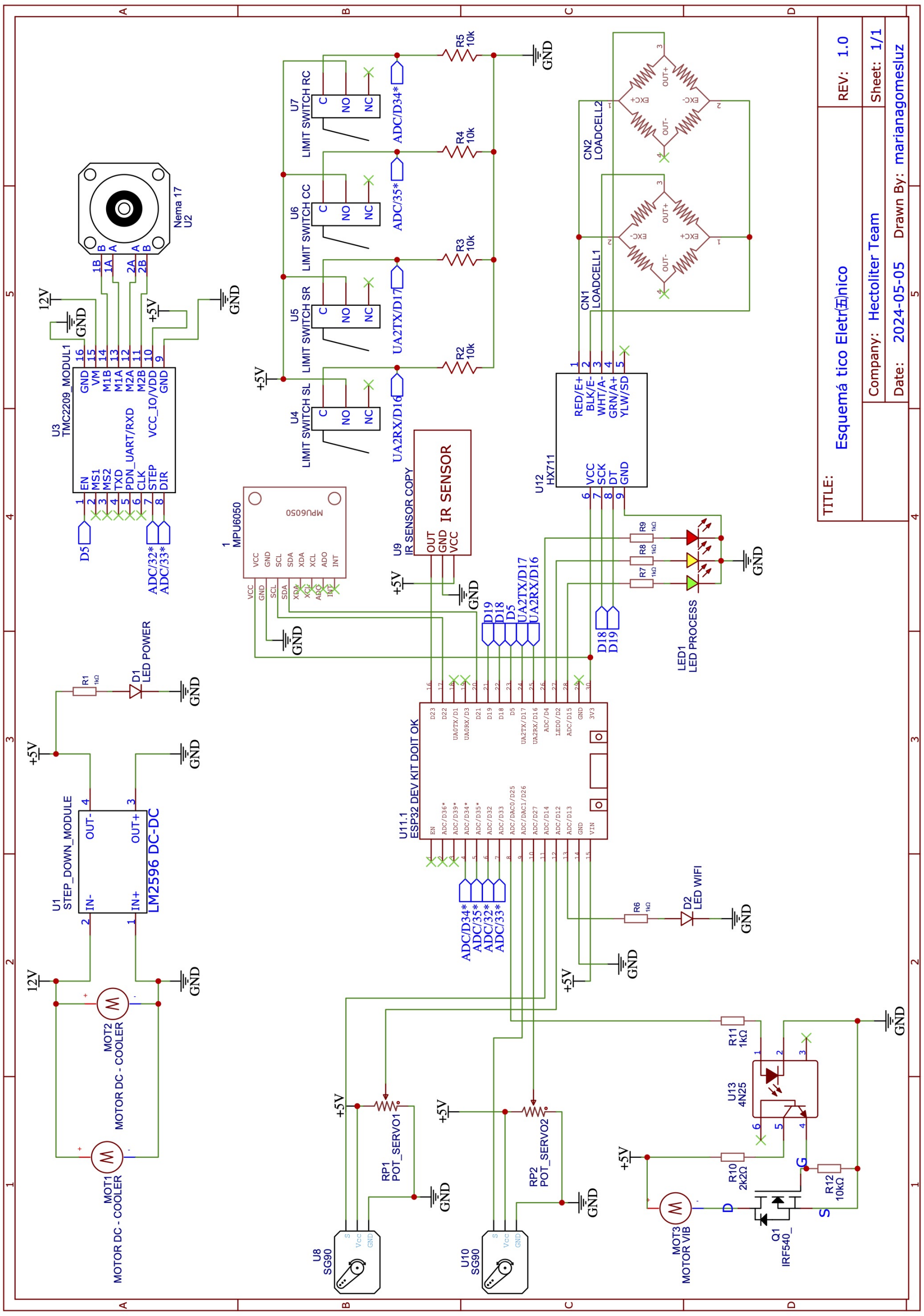

Figure 20: Electronic Schematic of All Electronic Components

Circuit Board The circuit board that all the sensors, motors, drivers, and the microcontroller ESP32 are attached to was made using a universal board and soldering.

In the PCB design, the GND and VCC have a single input line each, and jumpers were needed because each pin of the components that weren't attached to the board (like the IC, transistors, and resistors) had separated input places and couldn't be easily plugged into the board. The schematic of the board for mounting is shown in Figure 21.

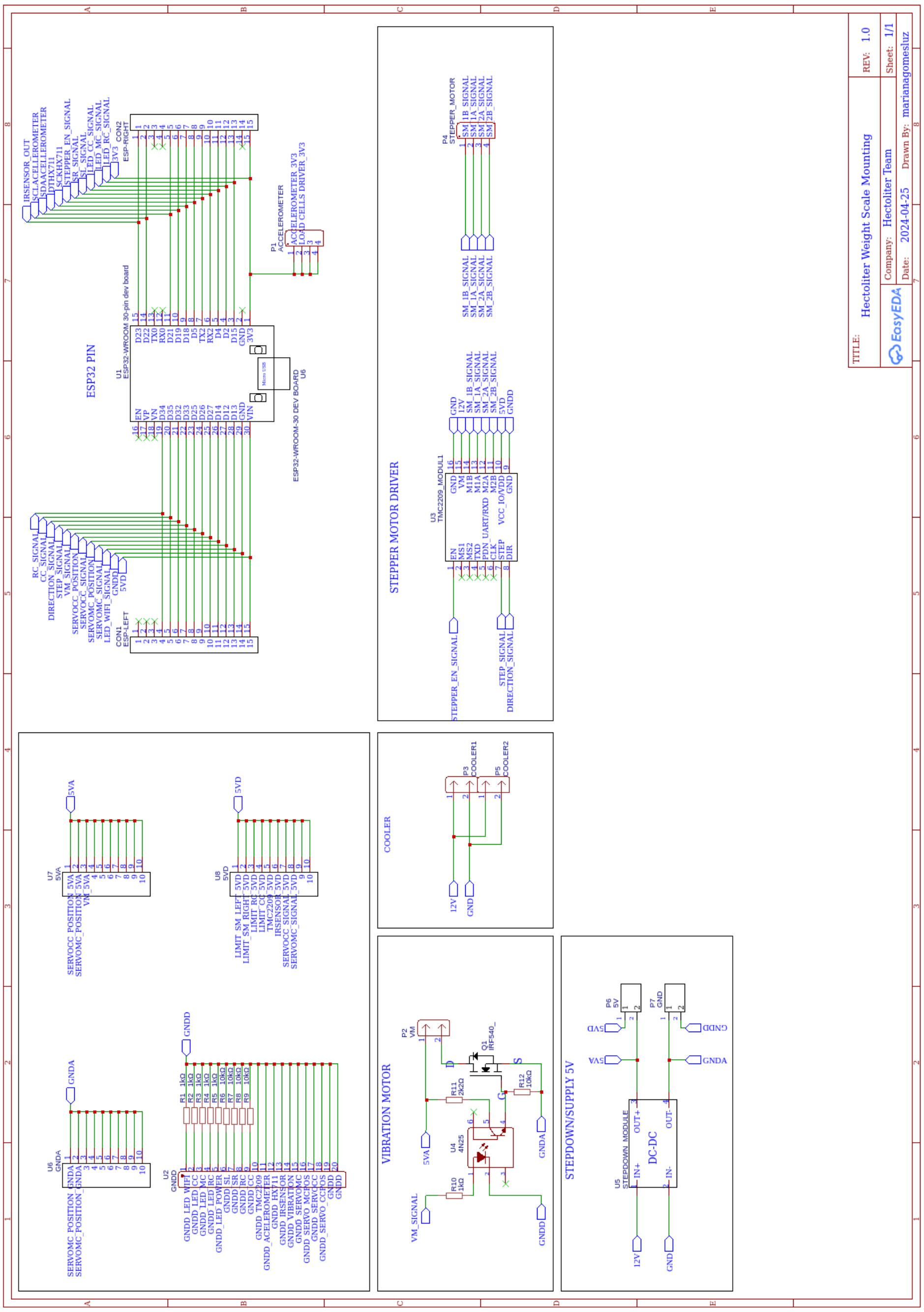

Figure 21: Mounting Schematic of Main Board

The result was a PCB that could easily be made with a universal board. The design in Figure 22 was the one transferred to the main universal board.

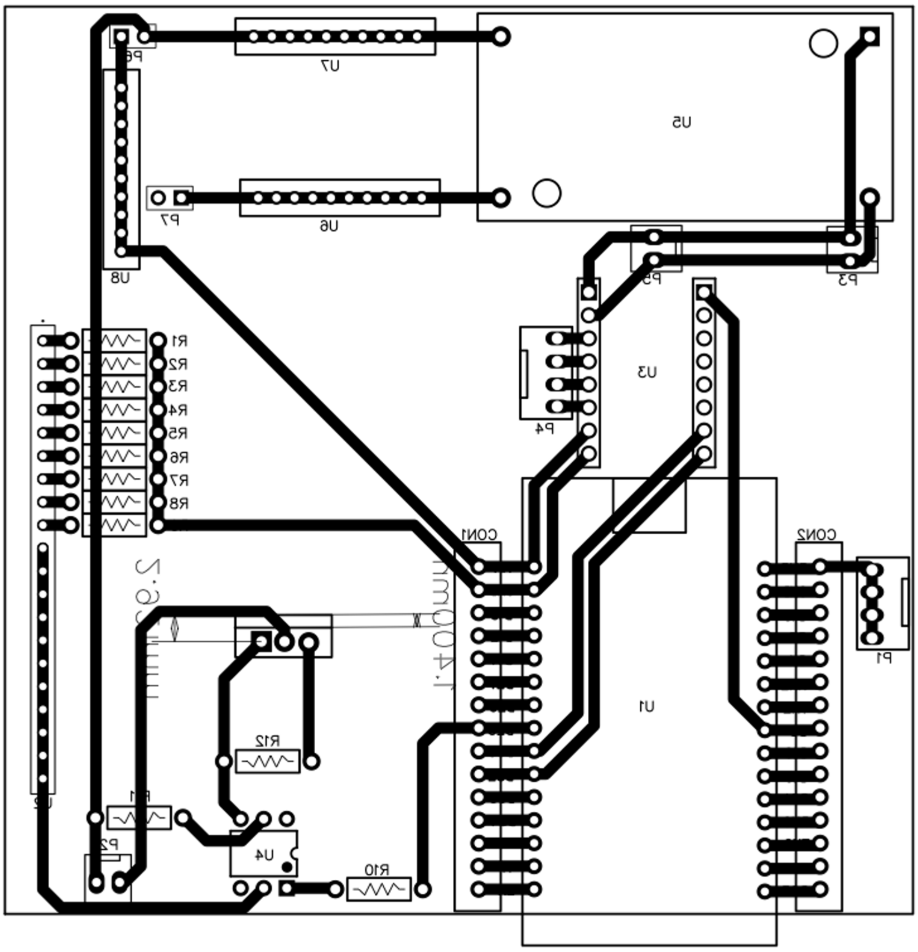

Figure 22: PCB of Main Board

One thing was let out of the main board, the circuit of the load cells because this one was best near the cells as their signals are super sensitive. The separate board for these signals only comports the entrance for the load cell signals and the HX711 module and their connections, like the schematic shown in Figure 23.

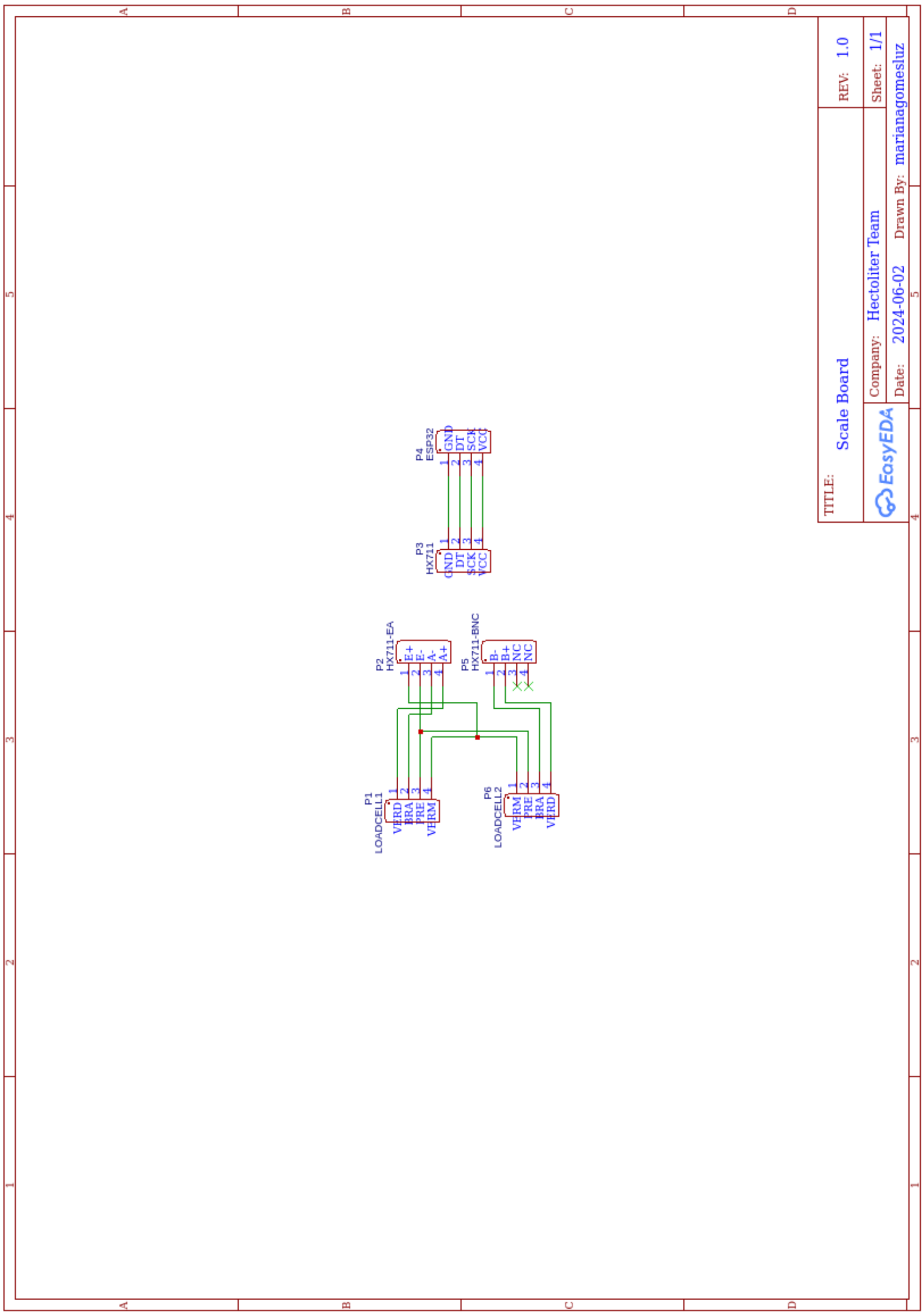

Figure 23: Schematic of Load Cells Board

The board had an easy circuit as can be seen in the PCB planned, in Figure 24.

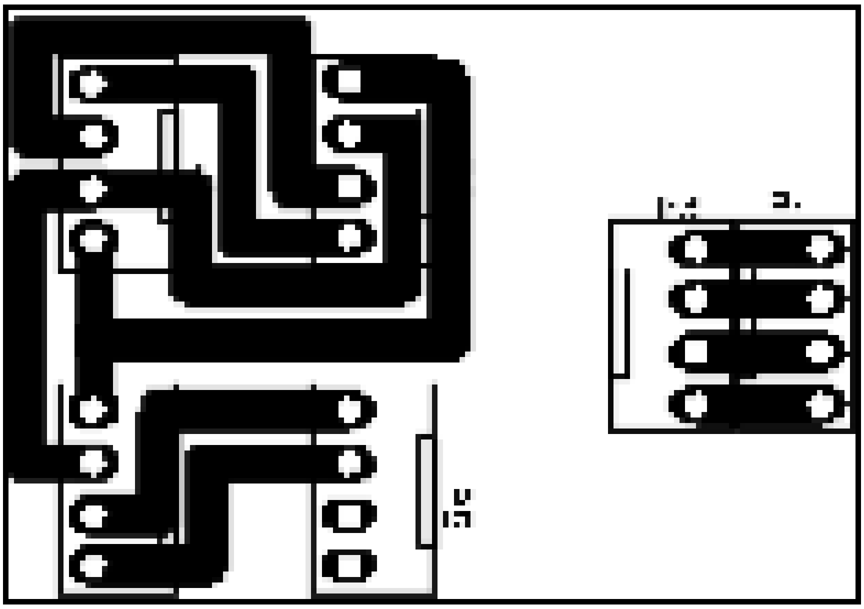

Figure 24: PCB of Load Cells Board

Mounted Boards The main board was attached to the bottom side of the inside structure of the Hectoliter Scale, as shown in Figure 25

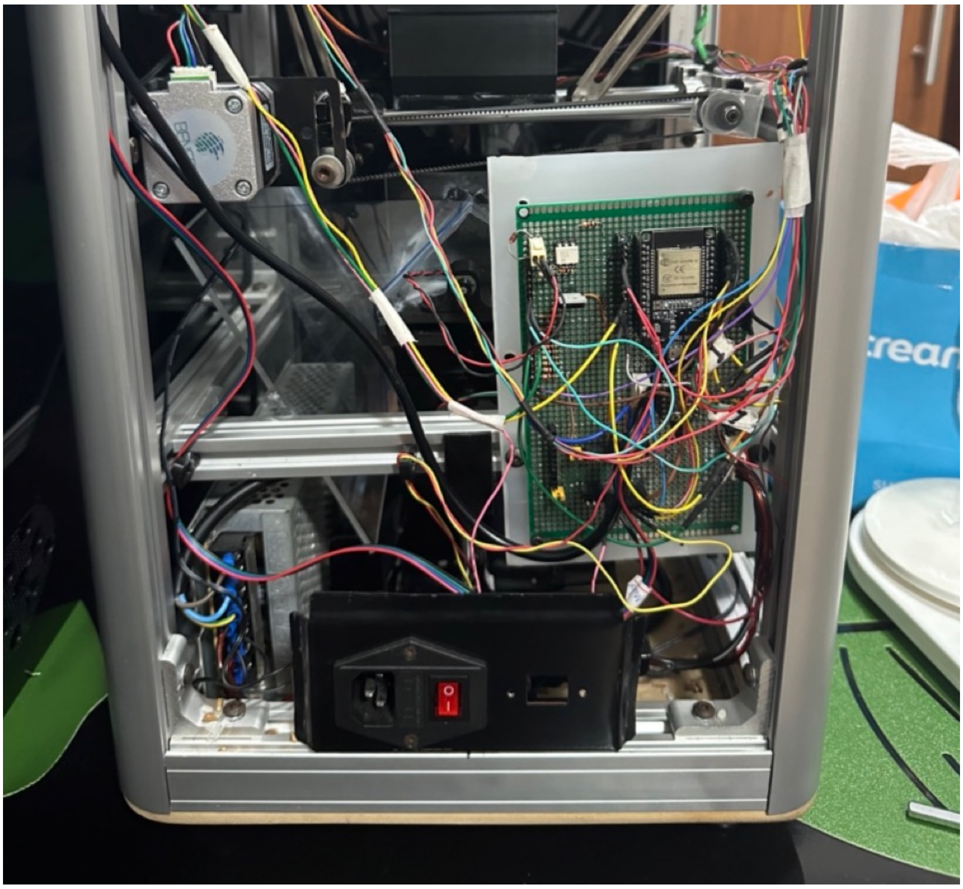

Figure 25: Main Board Attached to the Machine

The load cells board was attached as near as possible to the load, as shown in Figure 26.

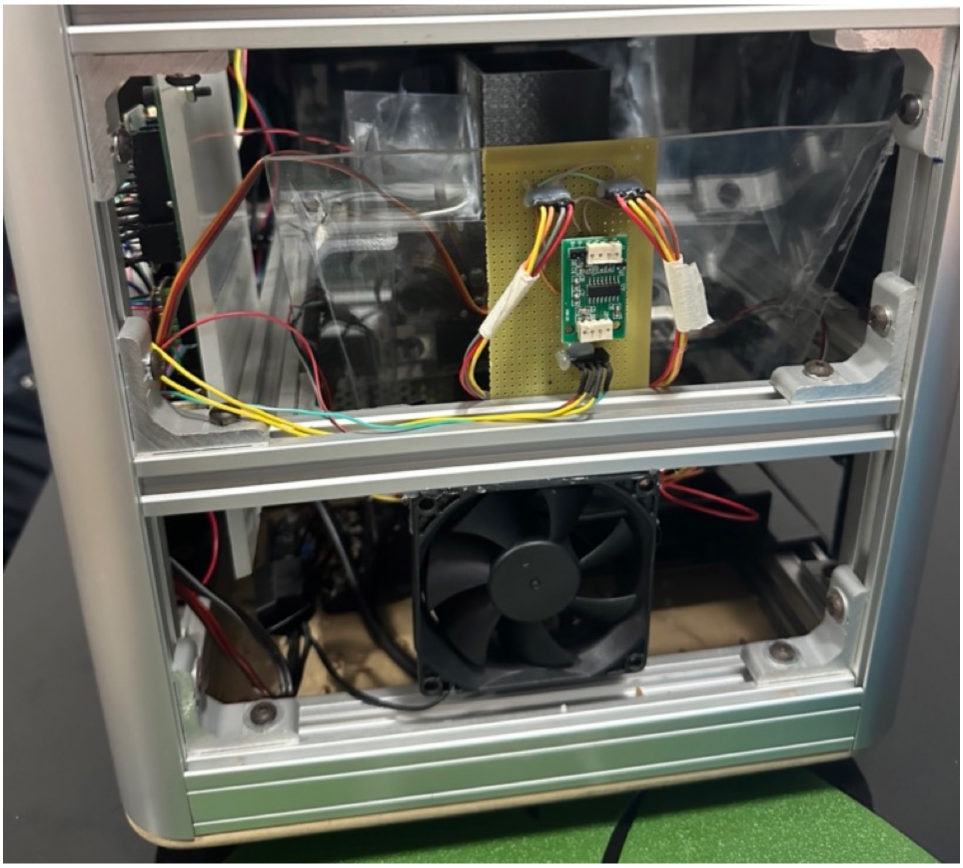

Figure 26: Load Cells Board Attached to the Machine

Finished Boards The main board had the bottom design very similar to 22, as shown in Figure 27

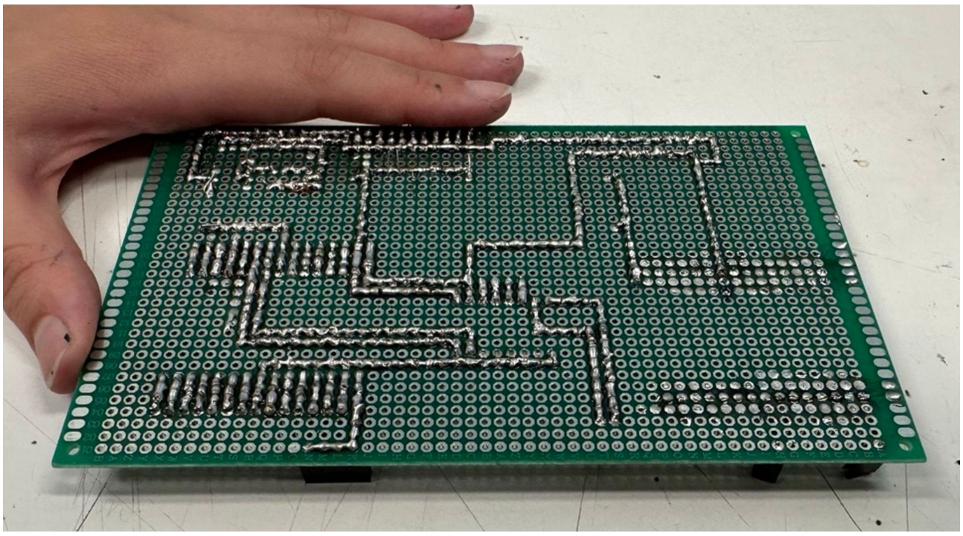

Figure 27: Photo of Main Board Bottom

The load cells board was very small and its simple design was very similar to what was planned (Figure 24) as shown in Figure 28

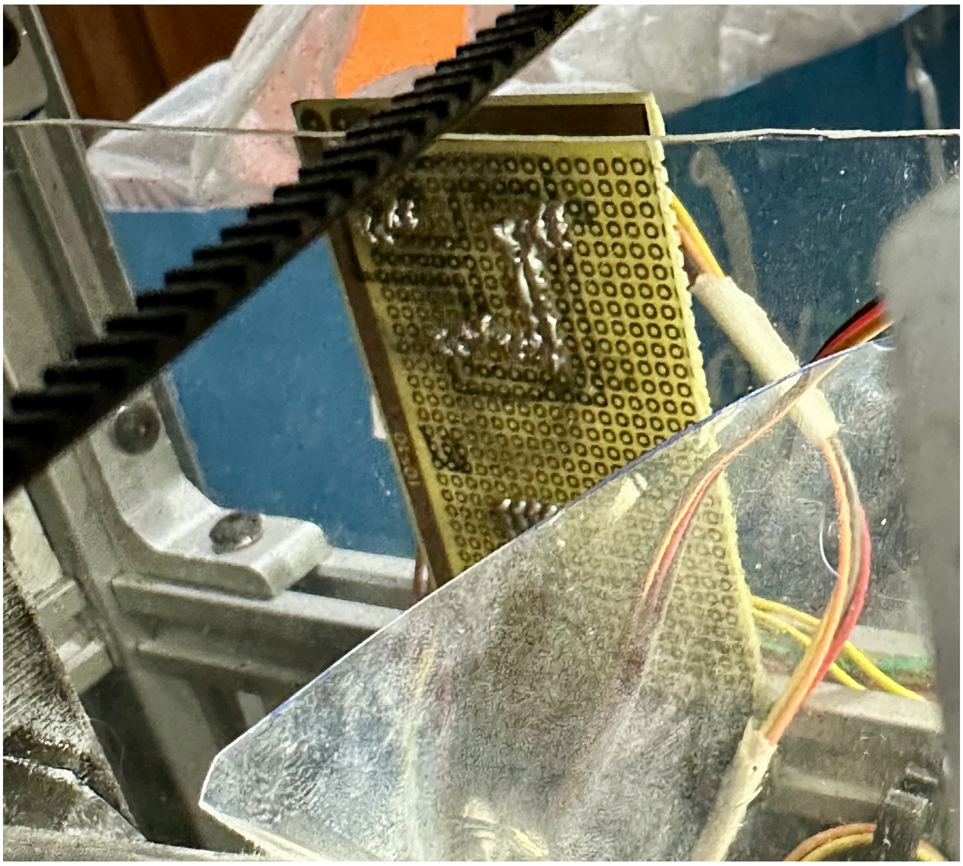

Figure 28: Photo of Load Cells Board Bottom

3.3 Software

The software took a proactive approach, always checking for new information in the database. It isn't an energy-saving method but it was a way that was previously tested by the students that had worked even with not ideal internet connections. The ESP always checks for new requests, changing its state from available to busy in the database to let anyone using the app know it isn't ready to receive a new measurement request.

3.3.1 Firmware

While planning the firmware it was found that the use of threads would make the program more decoupled, leading to more easily understandable code that could work better and be developed separately, which would mean less work that had to be done intertwined. One of the threads running on the ESP32 would only be responsible for the process, controlling the motors and getting the information from the sensors. The other thread would only be responsible for communicating with the database to know when to start the processes and to send the process information to the API.

3.3.2 Interface (APP)

The app was developed in Ionic[9]. Using Ionic, the app would be cross-platform and could be coded using tools previously used by the students, making it faster and more accessible to develop.

3.3.3 Web Server

To manage the data, a cloud-based server was created. This way the API could manage the way to access it from the APP or the firmware.

3.3.4 API

The REST API was made in Python[10] using Flask[11], a web framework for building web applications. It communicates with an SQLite database using an ORM, or Object Relational Mapper, as an abstraction layer between Python code and SQL database. This made the code database-agnostic, allowing the group to change to another database if needed. The backend provides an endpoint for the machine to update its status and to get the indicator to start a new process. It also provides endpoints to log into the App, which generates a JWT token to authenticate any other requests it makes; query the machine and process status; start a new process; and get all processes history. The API and the SQLite database were hosted in the *pythonanywhere*[12] cloud that could be accessed via an internet connection through the ESP32 and the APP.

4 Results

When measurements were made on a wobbly table, the weight would vary a lot. In 10 measurements that we experimented with the amount of weight on the table while the sample was being weighted the results changed in more than 100g, as shown in Table 3. In these measurements, we could see that the mean was 145.9g with an error of +-10g. This is something that can not happen if the machine were to be used in a professional space. Since the process had a more constant weight measurement for the same sample when the table was stable, it was added to the instructions that the table must be stable and the person has to stay away from the machine to avoid these types of errors to occur. When tested with a more stable table and the user away from the table, as shown in the Table 4 When tested in the manner depicted in the User Manual the amount of errors was diminished to a mean of 291.2g with a 4.19g error.

While we saw these errors on the machine we thought it might be significant, so we tested by hand if we could measure by hand. The results of this experiment

Measurement of Weight
188g
194g
128g
138g
87g
180g
180g
105g
137g
153g

Table 3: Measurement Experiments on Unstable Table

Measurement of Weight
299g
305g
288g
280g
284g

Table 4: Measurement Experiments on Stable Table

are shown in Table 5 This experiment had a 293.2g mean and a +-5.6g error. This is near the amount of error gotten by the machine but something that isn't accounted for in this statistic is the amount of unsureness that we got by guessing if the volume was consistent every time we measured it. In this way, the manual way had much more guesswork, which led us to think that the measurements were untrustworthy.

4.1 Difficulties

This subsection highlights the difficulties in each part of the development of the system of the project.

Measurement of Weight
315g
284g
299g
280g
288g

Table 5: Measurement Experiments done by Hand

4.1.1 Mechanical

There were many difficulties on the mechanical part, the first one being that the machine had to be planned and modeled rapidly for the electronic and hardware testing, using all the planned time and more as it was a very complex project. Assembling all the components that had to be in hands and integration was hard, as the students didn't have much contact with mechanics throughout graduation classes. The need for change was also tough when it was decided that the main PCB would be best placed in the upright position in the middle of one of the side acrylics.

4.1.2 Hardware & Firmware

Electronics The main difficulty in electronics was not being able to pass the electronic circuits from the breadboard to the universal board without trouble. Another universal board utilized for the load cells was made separately because it was better to be the nearest possible from the load cells. The universal board made only from soldering had about 20 Ohms of resistance between the extremities, which made the circuit not work, so it was remade with jumpers lowering the resistance to approximately 5-6 Ohms, and then the circuit worked. Another problem the group faced was that our ESP32 serial ports connected to the USB chip burned during tests, this made it impossible to upload new firmware to it using the USB connector, and we had to buy a new ESP32 using our response plan extra budget.

Firmware The firmware test was done side by side with the electronics so there were times when an electronic part wouldn't work and there would have to be more tests to find if the error was in the code or the electronics had a wire that had come loose again. Some bugs in the ESP interface concerned the developers, where the code of the current file being worked on was shown, but it wouldn't load the changes. Sometimes this bug was fixed by turning the software off and on, other times it had to be the computer itself that had to be turned off and on again. When both of these didn't work we found that the mistake would only be "fixable" by not using that file anymore and having to copy-paste the code onto another.

4.1.3 Interface (APP)

The main difficulty during the app development was finding some bugs introduced during coding, but, other than that, there were no other problems.

4.1.4 API

Since the API was coded in a language and framework used daily by group members, there were no difficulties in this part of the project. The problems were similar to the APP development cited in the previous subsection.

4.2 Integration

Integrating the electronic components in the machine was easy since the project in Solid Works had already fitted them to the mechanics. The only issues were with the servo doors that had to be turned all the way in before being fitted to the trapdoor mechanisms so they closed correctly, and the wires that weren't that well planned on the length needed to connect them to the PCB.

The main issues of electronics integration appeared when we got the components inside the machine and tried to make it work, with several problems of wires disconnecting making the tests less than optimal. In the end, the integration worked and the wires were either tapped not to move or hot glued making the tests more stable.

The backend integration with the ESP32 had a few errors, but one that took some time to debug was the variable that started the process. It was not initialized with zero and it was picking trash values from memory, this made the process start sometimes by itself when the machine was turned on. After finding and fixing this, the rest of the integration went well. The backend integration to the APP had some errors in the logic of the process that were easily fixed.

4.3 Budget

The initial budget was done by searching for prices we found online or if it was hard to ship, like the aluminum and acrylics, which were quoted with suppliers that members have previously worked with. The planned spents are shown in Table 6.

The table would have been a little different since we used the ESP32 that our members already had, but we burned one so we had to buy another to substitute. The amount spent for the PCB would have been less if there had been enough time from the end of the tests of the electronic schematic but in the end, with all things bought it was about the same at 70 reais.

4.4 Schedule

When the Schedule was being done there wasn't as much of a clear idea of what would have to be done near the end. The tasks to begin the project were very clear, as they were to study, buy, and test the components we planned to use. The next parts of integration and integration tests for the system were more blurred since there wasn't as much study already done to know what specific tasks would have to be done to make a component functional as intended on the project. This lack of knowledge kept the tasks that were near the end of the project more abstract and with a time expectancy not as accurate.

MATERIALS	AMOUNT	UNIT COST (R$)	TOTAL COST (R$)
ESP32	1	33,33	33,33
INFRARED SENSOR	1	9,05	9,05
LOAD CELL 1KG	2	6,95	13,90
LIMIT SWITCH	4	3,70	14,80
STEPPER MOTOR NEMA 17	1	38,00	38,00
STEPPER MOTOR DRIVER	1	4,99	4,99
SERVO MOTOR	2	11,05	22,10
VIBRATION MOTOR	1	23,35	23,35
POTENTIOMETER	4	0,50	2,00
HX711	1	4,19	4,19
COOLER 8CM x 8CM 12V	2	14,49	28,98
SUPPLIES FOR PCD DEVELOPMENT	1	120,00	120,00
LINEAR BEARING 8mm	2	10,50	21,00
SUPPORT FOR LINEAR AXIS SHF 8mm	4	10,25	41,00
LINEAR GUIDE 8mm	1	15,18	15,18
PULLEY GT2 20	1	7,50	7,50
FLAT TENSION PULLEY GT2	1	18,62	18,62
BELT GT2	1	29,00	29,00
BELT CLAMP GT2	1	3,00	3,00
DAME STRUCTURAL PROFILE	2	37,00	74,00
ALUMINUM			100,00
STRUCTURAL PROFILE	5	30,00	150,00
STRUCTURAL CORNER	40	2,50	100,00
POWER SUPPLY 12V 20A	1	38,95	38,95
GYROSCOPE MPU-6050	1	16,11	16,11
ACRYLIC PLATE	6	25,00	150,00
3D FILAMENT	1	25,00	25,00
Mancal	4	5,00	20,00
Seal	4	5,00	20,00
Screw		6,00	6,00
Angle		7,00	7,00
TOTAL			1157,05

Table 6: Budget

4.5 Hours Worked

The number of hours worked was not as easy to track because if not noted when it started and right as it ended people would forget how much time was spent and there would be times when approximations would be done. The amount of hours worked was not computed correctly as many hours were forgotten to be added to our calculator. In the end, these forgotten hours were added it would make our work hours as high as the expected one. (Figure 29)

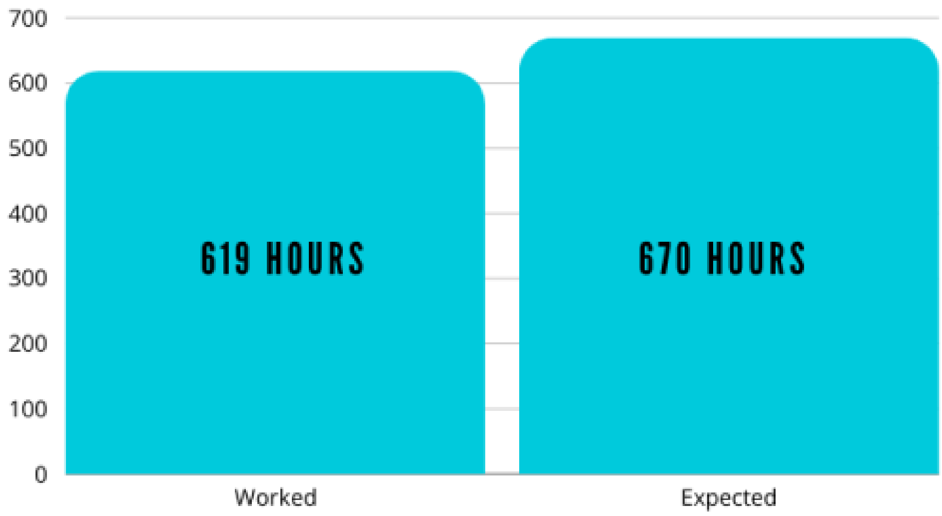

Figure 29: Chart of Worked x Expected Hours

5 Conclusion

The project was successful in its objectives because the planning that took a huge part of the project was well done. The team had the goal to meet every requirement, and they were met. The planning part was important, as we thought about how it would be easier to execute and test the project before finalizing each requirement.

Though the class wasn't easy and the amount needed to work on it every week was exhausting, seeing all projects meet their goals week by week was very inspiring, and to know that we were all on similar boats of the same crew.

The amount of uncertainty following the subjects of a strike had the students unsure of the progress and the allotted time confused, since some weeks there was enough time to do everything calmly and others there would have to be time allocated to pick up all things back from other subjects.

While thinking about future improvements to the system created, the team teams first thought was a better way to level the sample, since this was the most complicated part of the project and the one that could most easily malfunction.

Another way to get the volume correctly would be to not level the sample into a container but get cameras to get the volume without leveling the sample. Another thing to improve in the machine would be a system to bag the grains used for the measurement so that the client can easily keep the sample and store what was used to measure for future inspections.

References

[1] J. D. COÊLHO, "Trigo: ProduÇÃo e mercados." https://www.bnb.gov.br/s482-dspace/bitstream/123456789/636/3/2021_CDS_151.pdf.

[2] M. Manley, M. Engelbrecht, P. Williams, and M. Kidd, "Assessment of variance in the measurement of hectolitre mass of wheat, using equipment from different grain producing and exporting countries." https://www.sciencedirect.com/science/article/pii/S153751100900066X.

[3] Laboratuvar, "Hectoliter weight determination." https://www.laboratuvar.com/en/gida-analizleri/fiziksel-analizler/hektolitre-agirligi-tayini.

[4] A. Jain, "How to use hectolitre brass chamber container slit knife digital balance for 1kg seeds abron hindi." https://www.youtube.com/watch?v=PgmMNritAqI.

[5] U. S. D. of Agriculture, "Operation of a bulk weighing scale." https://www.ams.usda.gov/resources/operation-bulk-weighing-scale.

[6] M. M. K. Lauro Okuyama, Henrique Caramori Paulo, "New microchondrometer to measure hectoliter weight in small samples of wheat." https://www.researchgate.net/figure/Hectoliter-weight-standard-deviation-and-t-test-significance-of-60-wheat-grain-samples_tbl2_341052701.

[7] T. H. S. Team, "The hectoliter scale proof of requirements." https://parallel-scaffold-ec7.notion.site/PROOF-OF-REQUIREMENTS-4ba2a63c5d1b4f44bce2a2bfa774230c.

[8] T. H. S. Team, "The hectoliter scale blog." https://parallel-scaffold-ec7.notion.site/Hectoliter-scale-e9c8e874d7f0444c88b8114017271cce.

[9] Ionic, "Ionic framework." https://ionicframework.com/.

[10] P. S. Foundation, "Python programming language." https://www.python.org/.

[11] Pallets, "Flask." https://flask.palletsprojects.com/en/3.0.x/.

[12] Anaconda, “pythonanywhere by anaconda.” https://www.pythonanywhere.com/.