ECE 445
Spring 2025
SENIOR DESIGN LABORATORY
PROJECT PROPOSAL

CO2ffee: Coffee Bean Freshness Tracker

Team: 4

Abrar Murtaza (abraram2@illinois.edu)

Nathan Colunga (colunga4@illinois.edu)

Joshua Meier (joshua51@illinois.edu)

TA: Surya Vasanth

Professor: Arne Fliflet

February 13, 2025

1. Introduction

1.1 Problem

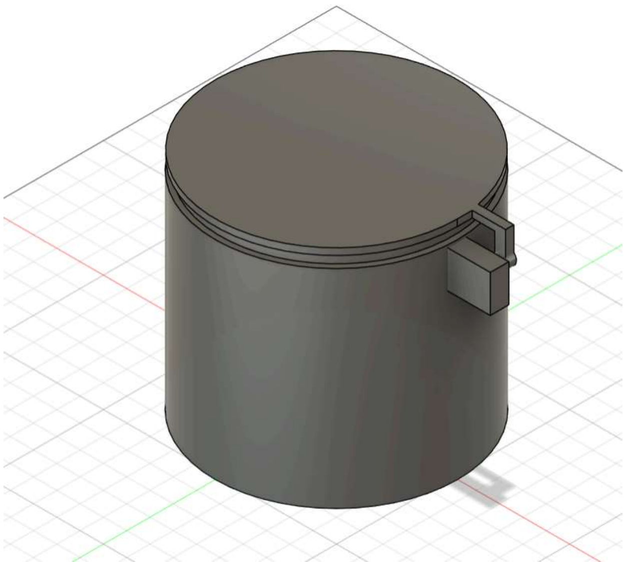
Many coffee connoisseurs care about having the perfect level of bean freshness, as it gives you the best coffee and depth of flavor! Having beans roasted 1 day ago is too fresh, but after a short time, they become too old. The issue is that when you buy fresh coffee beans from a roastery, they only give you an estimated date for when you should use them (typically within a month). Because of this, those who are picky with their coffee quality don't actually know exactly when their beans are actually considered fresh. If their coffee beans are too new, it makes the coffee taste overly acidic [2]. If the beans are old, they risk losing the same punch of flavor that the main notes of the coffee provides, leading to stale coffee [2]. If coffee enjoyers do not know when their beans are at the perfect level of freshness, they end up getting a suboptimal extraction and an imperfect cup of coffee [1][3]. In order to solve this issue, we are making a custom coffee container which detects how fresh beans are. This way, users can know when their beans are at the perfect freshness level so that they can get an optimal extraction to make the perfect cup of coffee!

1.2 Solution

For our design, we plan on creating a container designed to track the amount of CO2 remaining in the coffee beans, as this correlates directly with their freshness [1]. This is based on the weight of beans that were added, as well as the detected concentration of CO2 that builds up in the container over time. Our system consists of an inner and outer container. The inner container holds the beans and preserves them for as long as possible, utilizing an airtight seal combined with a degassing valve. The combination of the seal and valve assures that no oxygen enters the container with the beans, while allowing for CO_2 to be released from the inner container to allow for consistent pressure within that container. The outer container includes all the electronic components, including the weight sensor to measure the weight of the beans (calculations will account for the inner container), the CO_2 sensor, the motor for opening the outer container, and all the other components necessary for our design (which will be hid in a separate section of the outer container).

For operation, when the user wants to add a newly roasted bag of beans to the device, they will press a button to open the outer container, manually open the inner container, and add the beans to the inner container. They will utilize a mobile interface to indicate that new beans were entered. In addition to this, they will add what roast type the beans are (light, medium, or dark). Once both containers are closed (manually for the inner container and using the button for the outer container), the container will then detect the weight of the beans that are added and the immediate initial concentration of CO_2 in the container (which should be ~420 ppm at atm). While the container is closed, CO_2 will be released from the inner container into the outer container. Every hour, the container will update the current CO_2 concentration of the outer container and mass of the beans. Using that new information, the system will then calculate the amount of CO_2 that was released per gram of coffee beans. The system will compare this rate of release to the initial rate of release for the beans, along with the type of beans that were added to the container, and calculate the percentage of CO_2 remaining in the beans. It is important to note that we will be assuming that 100% of the CO_2 is still contained in the beans upon being placed in the container. We will be assuming this because the expected use case is that beans will be added almost immediately after being bought, making the time since the roasting extremely short and the escaped CO_2 negligible.

1.3 Visual Aid



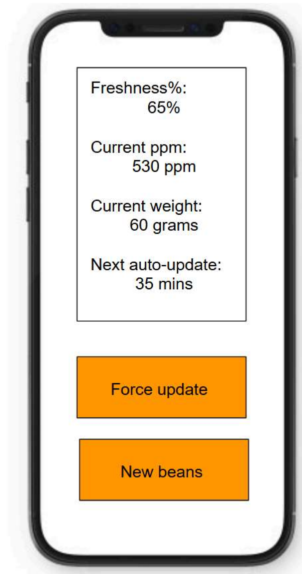


Figure 1.1(left) - visual aid of the closed container as it would appear

Figure 1.2 (right) - visual aid of phone interface to interact with electronics

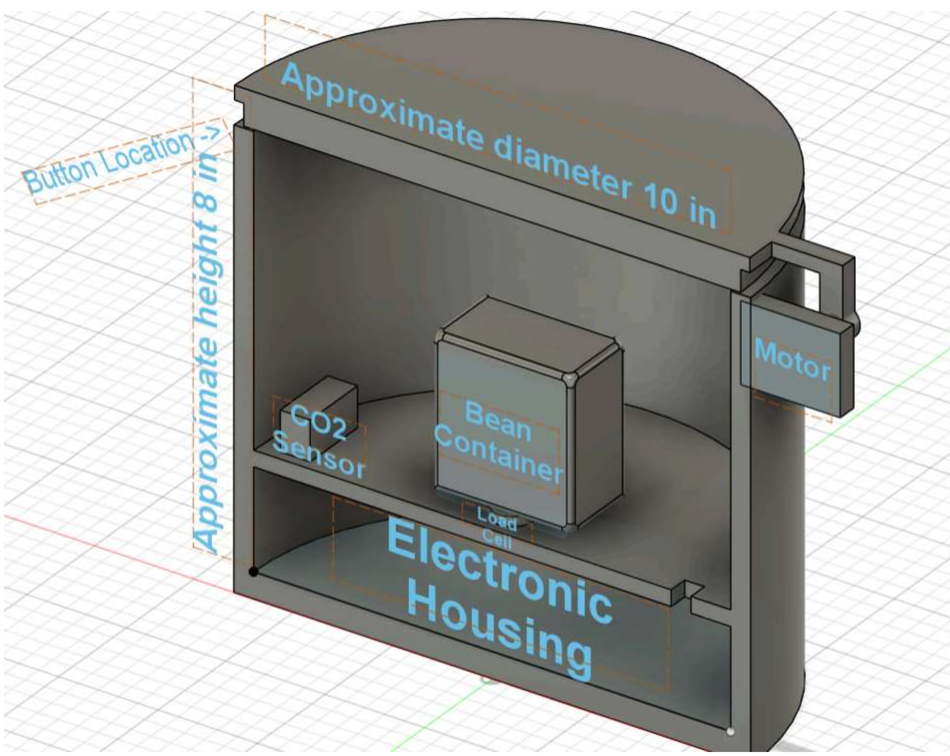


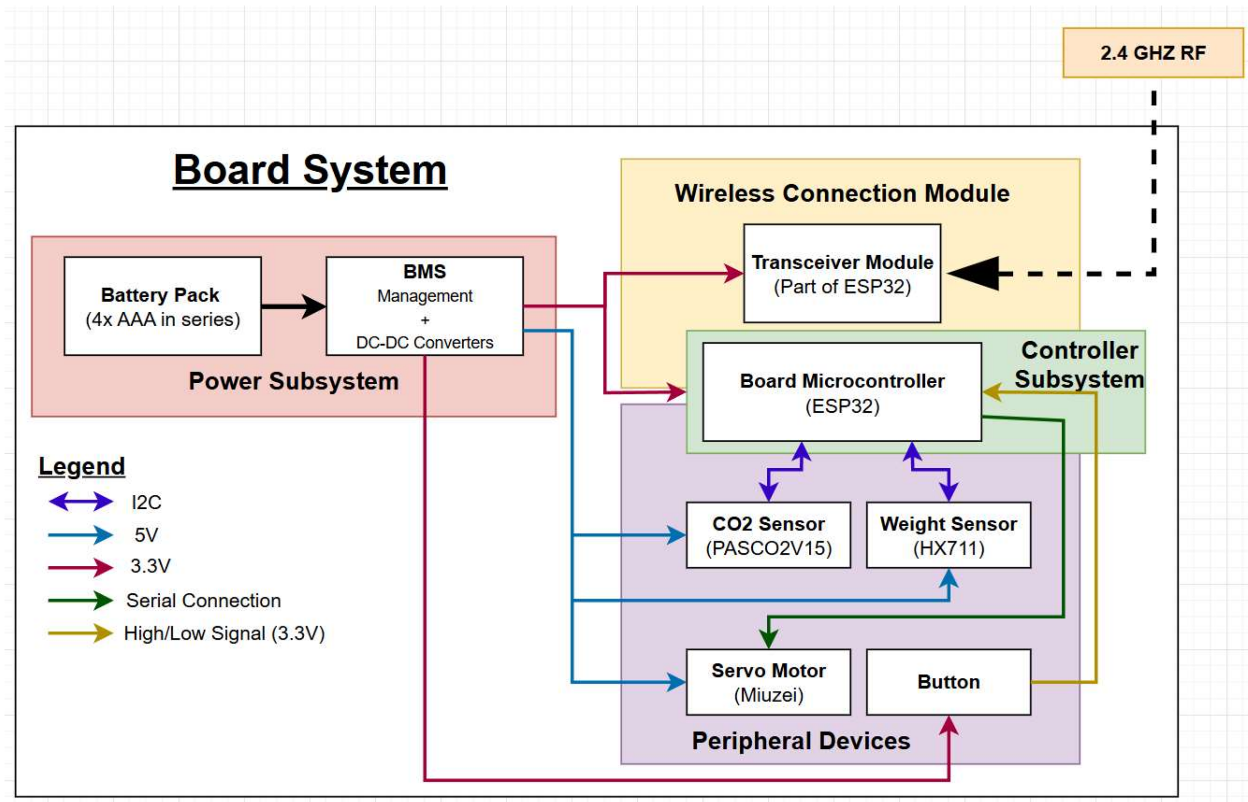
Figure 1.3 - cross-section of container with labeled components and areas along with dimensions

1.4 High-Level Requirements

- The user can select from 3 different bean types: light, medium, or dark roast.
- The user can press a button to automatically open or close the lid for bean withdrawal.
- Weight sensor reading after bean withdrawal should be reflected with a ±2% accuracy.
- The lid automatically opens for 5 seconds to aerate when CO_2 levels reach 3,000 ppm.
- Sensor measurements (CO2 and weight) should not be taken during aeration, only immediately before the lid is opened, after the lid is closed, during user-initiated updates, or every 1 hour by default. These readings are used to update the beans' freshness rating (percentage of CO2 remaining per gram of beans).

2. Design

2.1 Block Diagram



2.2 Subsystem Overview

2.2.1 Subsystem 1 - Peripheral Devices (i.e. Sensors, Inputs, and Motors)

Functionality: There will be three main sensors integrated into this subsystem. The weight sensor that determines the amount of beans in the container by mass, which will be used to calculate the carbon dioxide released per gram. The carbon dioxide sensor, which determines the ppm of the carbon dioxide in the closed container, which is also used in the CO2 per gram calculation. There will also be a button to open and close the container as well as a motor that will be used to rotate the lid on a hinge to open and close it. The motor will also be engaged every time the concentration builds up too much so that the CO_2 sensor is not overwhelmed by the concentration of CO_2 within the container. The weight sensor would most likely be made up of a few load cells and connected to the microcontroller via an HX711.

Interaction:

Inputs:

1. Communication to the controller subsystem to take clock information for most sensors and serial input to the motor.
2. The power system supplies the subsystem with 5V power, as most of the components have this voltage rating

Outputs:

1. Communication to the controller subsystem in the form of data using I2C and for both weight and CO_2 sensor.

Parts:

CO_2 sensor - PASCO2V15 (I2C support)

Weight sensor - HX711 (I2C support)

Servo Motor - Miuzei Waterproof Servo Motor (Serial Input)

Requirements:

1. This subsystem must send and receive reliable data to the controller subsystem.
2. This subsystem must have a reliable power supply able to simultaneously power each individual unit.
3. The motor must respond according to the button input and CO_2 sensor readings.

2.2.2 Subsystem 2 - Controller

Functionality: This subsystem consists of most likely a microcontroller with wifi compatibility to connect with all peripheral devices and must support I2C as many peripherals rely on this communication. The microcontroller will read data from the CO2 and weight sensors to perform calculations on the percentage of CO2 remaining in the coffee beans. Based on this measurement, data will be output to the mobile interface via a wireless connection module, reporting as the freshness report of the coffee beans. The push button input and motor outputs are also connected, so on each press, the microcontroller can send data to the motor to begin opening or closing the lid, alternating on each press.

Interaction: This subsystem will interact with all other subsystems, as it requires power and is the mode of communication through which all peripherals will be connected.

Inputs:

1. Communicates with peripheral subsystems to take in data from each sensor.
2. Takes 3.3V from the power subsystem, as this is what the ESP32 operates under for optimal performance.
3. Connects via Wi-Fi to the network to receive data from the user in regard to new bean status and bean type information.

Outputs:

1. Communicates with peripheral subsystems to maintain clock cycle and operate motor rotation.
2. Connects via Wi-Fi to the network to send data to the user in regard to bean freshness and specific status of data measurements.

Parts:

Microcontroller - ESP32-WROOM-32E-H4 (Wi-Fi compatible/low power)

Requirements:

1. This subsystem will need a constant supply of 3.3V from the power supply to operate optimally.
2. The esp32 Wi-Fi module must have a reliable Wi-Fi signal in order to update the user of the coffee status and receive the needed information for calibration.
3. Reliable information must be supplied to the controller subsystem from the peripherals for calculations to be accurate.

2.2.3 Subsystem 3 - Wireless Connection Module and Mobile Interface

Functionality: The wireless connection module is built in the ESP32 microcontroller and this will allow data communication between the mobile interface and microcontroller. On this mobile interface, there will be an input to indicate which type of coffee beans have been entered into the container. Based on this input, our system will be able to estimate the initial amount of CO_2 stored in the beans. Through the mobile interface, the freshness of the beans will be reported as the approximate percentage of CO_2 still remaining in the beans.

Interaction:

Inputs:

1. This subsystem will only interact with the controller subsystems Wi-Fi component to retrieve freshness updates and data measurements for the user.

Outputs:

1. This subsystem will send data via Wi-Fi to the controller subsystem in order to update the system on the type of bean and when a new package of beans was entered into the system to restart the CO_2 calculations.

Parts:

Wi-Fi connectivity component - ESP32 with Wi-Fi compatibility

Requirements:

1. Reliable Wi-Fi connection to have constant connection to the controller subsystem in order to send and receive data.

2.2.4 Subsystem 4 - Power System

Functionality: For power, we plan to utilize four rechargeable batteries along with a power controller board. The power controller board will manage the power being taken in from a USB port to recharge the batteries and also manage the output power to all sensors and microcontroller. We will most likely utilize triple rechargeable batteries and design our own power board for this subsystem.

Interaction:

Inputs:

1. Will not take inputs from any subsystem, but will have a USB-C port for recharging purposes.

Outputs:

1. Will connect to all other subsystems to supply power to each component at rated voltages, which includes at least 3.3 volts for the ESP32 and 5V for most peripherals.

Part:

power control IC - possible choice is TI's BQ25176J

Requirements:

1. Supplies contiguous power to all subsystems at rated voltages without dropping current required during operations when the motor is activated.
2. Is able to recharge and continue functioning without a jump in voltage or current from the recharge port.

2.3 Tolerance Analysis

One of the main concerns initially when tackling this design is being assured that coffee beans release enough CO2 into the air to be tracked by a sensor. Using the data in a degassing bean research paper [3] we know that during the first day approximately 2 mg of CO2 is degassed by the coffee beans per gram of beans for a fast-dark roasted coffee bean. We may assume that 300 g of coffee beans were placed into the coffee bean container within the vessel, as this is a reasonable approximation given the amount of space we have granted the container within the vessel. This means that 0.6 grams of CO2 was released into the air around the coffee beans. The total volume of our vessel using the values in the Visual Aid is approximately 10 liters, however this includes many of the components as well as the beans themselves so to make it more accurate it will have to be measured when the actual vessel is built and the dimensions of all the components is taken into account. For this calculation though, we will estimate 7 liters is the amount available for free air. From here we have all we need to calculate the theoretical ppm after 1 day of fresh beans in the container.

ppm of CO2 of typical air = 420 ppm

Air density at sea level = 1.293 g/L

First, calculate the amount of grams of air in the container:

$$7(L)\ *\ 1.293(g/L)\ =\ 9.051(g)$$

Then to calculate the original amount of CO2 in the air:

$$420(ppm) \ / \ 10^6 \ * \ 9.051(g) \ = \ 0.0038(g)$$

total amount of CO_2 in air after ~24 hours

$$0.6(g) \ + \ 0.0038(g) \ = \ 0.6038(g)$$

Finally, converting this value back to a ppm value that we would actually be measuring:

$$0.6038(g)/(9.051(g) \ + \ 0.6) \ * \ 10^6 \ = \ 62,569 \ ppm$$

This is such a high number that we would have to do nearly 20 aeration cycles throughout the first day (which will be triggered upon reaching near 3000 ppm) as to not overload the CO_2 sensor range.

3. Ethics and Safety

3.1 Honesty and Transparency in System Expectations

The coffee bean freshness detector operates under the assumption that when beans are initially placed in the container, 100% of their CO_2 is retained. The initial CO_2 content is then estimated based on the selected bean type. Based on these assumptions, the reported freshness report is an approximate percentage calculation of how much CO_2 remains in the beans. In accordance with IEEE Code of Ethics I.5, to ensure honesty and transparency, we will clearly inform users of these limitations and intended use of the product.

3.2 Power System Safety

Since our system relies on rechargeable batteries and a power controller board, it is essential to prevent overheating, electrical hazards, and potential fire risks. According to the ACM Code of Ethics 1.2, we must mitigate risk and avoid harm, ensuring that our power system operates safely and reliably under all conditions. To meet these standards, when the batteries are not in use they will be held in a designated battery box that will have no contact with possible conductors. They will also be secured in their housing with screws and covers to not be loose among other electrical components.

3.3 Safe and Sanitary Container for Beans

Since our product is designed to store coffee beans, we must adhere to the IEEE Code of Ethics I.1, which emphasizes prioritizing public safety and health. To meet this standard, all components that come into contact with the coffee beans will be made from food-safe materials, ensuring it does not react with coffee oils or beans. Meanwhile, the outer container will likely be made from PLA, which remains stable and does not react with CO2 in any way.

4. References

[1] Grant, TJ. "A Guide to Measuring the CO2 Levels of Roasted Coffee." *MTPak Coffee*, 13 May 2021, https://mtpak.coffee/2021/05/measuring-carbon-dioxide-co2-levels-roasted-coffee/.

[2] Raper, Adam. "Is Your Coffee Too Fresh? – Clive Coffee." *Clive Coffee*, Clive Coffee, 2 Apr. 2018, https://clivecoffee.com/blogs/learn/is-your-coffee-too-fresh?srsltid=AfmBOoqs96cwngkEma8nq-4lCJSyu2RNiSOkk2UFrxMS0Z1cUGBRQTfU.

[3] Smrke, Samo, et al. "Time-Resolved Gravimetric Method To Assess Degassing of Roasted Coffee." *ACS Publications*, 1 Oct. 2017, https://pubs.acs.org/doi/10.1021/acs.jafc.7b03310.