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

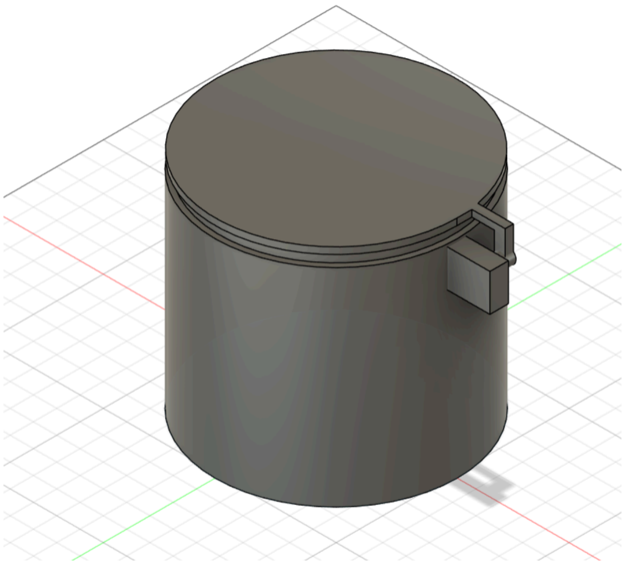
Many coffee connoisseurs care about having the perfect level of bean freshness, as it gives you the best coffee and depth of flavor! Having beans roasted 1 day ago is too fresh, but after a short time, they become too old. The issue is that when you buy fresh coffee beans from a roastery, they only give 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 CO_2 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 CO_2 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



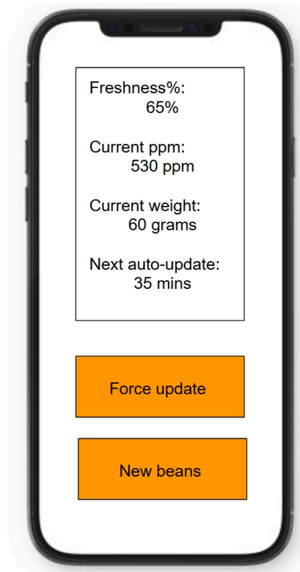


Figure 1.1(left) - visual aid of the closed container as it would appear

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

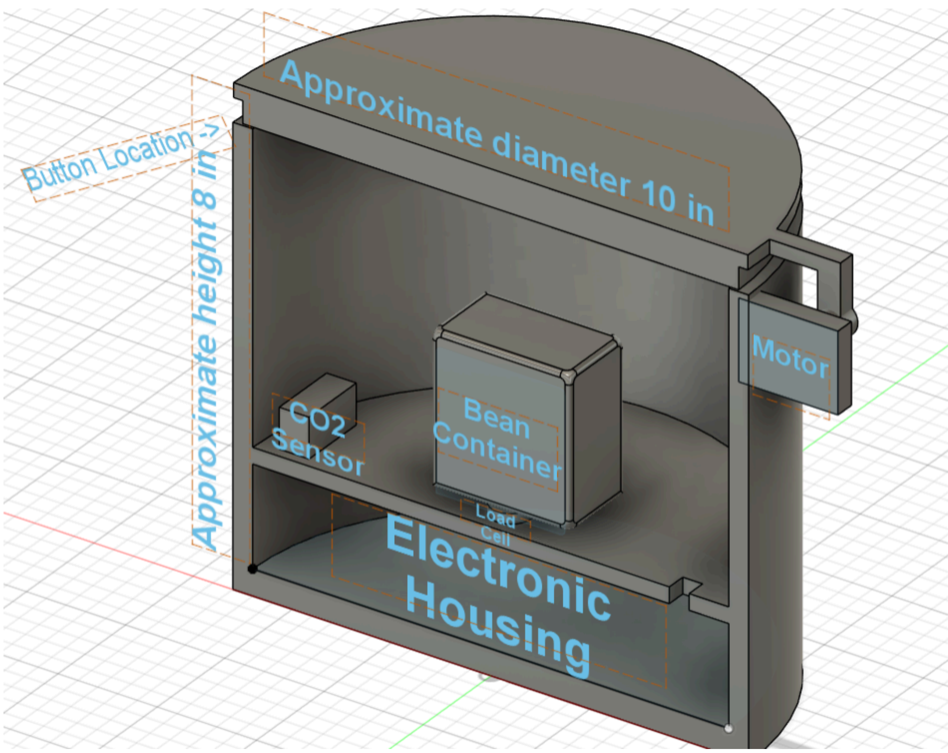


Figure 1.3 - cross-section of container with labeled components and areas along with dimensions

1.4 High-Level Requirements

- The freshness rating must be reported as a percentage, representing the CO2 remaining in the beans relative to its original state (100%).
- The weight sensor readings must accurately reflect bean withdrawal by ±2 %, and combined with CO2 sensor data, they must determine the CO2 loss per gram of beans.
- The user can select from three bean types, and press a button to open/close the outer lid for bean withdrawal.

2. Design

2.1 Block Diagram

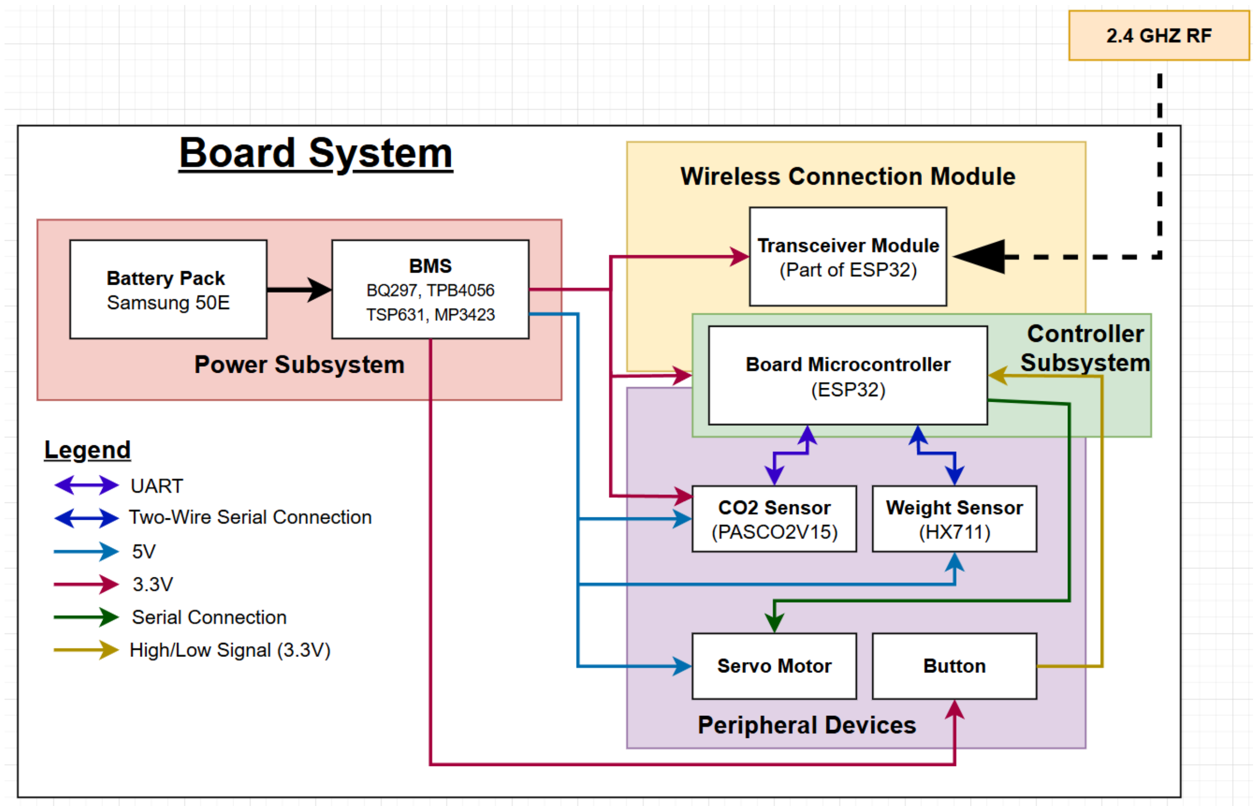


Figure 4.4 - Block Diagram of components

2.2 Subsystem Overview

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

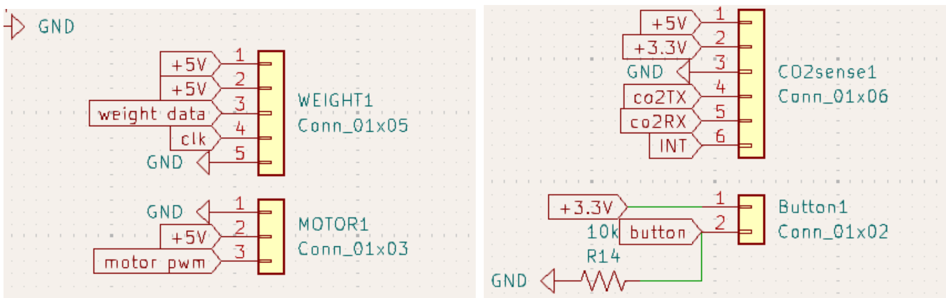


Figure 4.5 - peripheral connections on PCB

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 CO2 sensor is not overwhelmed by the concentration of CO2 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 CO2 sensor.

Parts:

CO2 sensor - PASCO2V15 (UART support)

Weight sensor - HX711 (serial support) along with load sensor

Servo Motor - HS-318 Servo Motor (PWM Input)

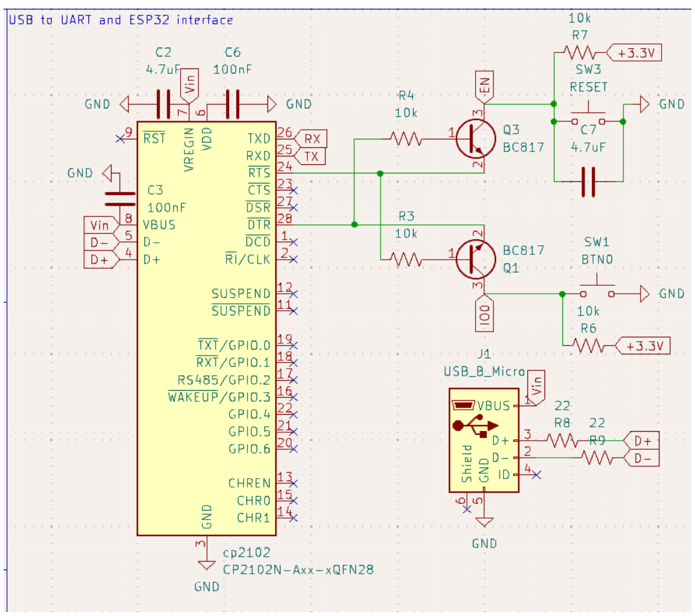
Requirements and Verification:

Requirements	**Specific component**	**Verifications**
This subsystem must send and receive reliable data to the controller subsystem.	CO2 sensor	The CO2 sensor outputs approximately 420 ppm at ambient and increases in a higher concentration of CO2 which can be done with calibrated scales and coffee beans determining the loss of CO2 and increase in ppm respectively. The decrease in weight should coincide with the increase in ppm over time and during every ppm reset the weight will be taken again. The resulting calculation from this process should show that the CO2 decreases the same amount within a 10% error margin. This process will also be done to calibrate the container in order to get an accurate freshness measurement. All values of the ppm and weight measurements will be taken over time (no beans will be removed during this process as a control) and the resulting CO2 losses

		will be compared.
	Weight sensor	This sensor must output data consistent with previously tested objects on a calibrated scale and correctly outputs data within 5%. Multiple trials comparing results will be recorded and the average error will be calculated and should be below 5% error.
	Servo Motor	The servo motor must rotate a given angle provided by the controller within 10 degrees. This motor will be run multiple times between two angle values to be sure the angle does not diverge greatly from the expected servo angles.
The motor must respond according to the button input and CO2 sensor readings.	Button and CO2 sensor	Container automatically opens before reaching 3000 ppm in the container and opens upon the user pressing the button which opens the container at the next available time in the operation cycle and closes when the button is pressed again.
The CO2 sensor must periodically record a measurement of the ppm of the CO2 in the	CO2 sensor	The CO2 sensor will be configured to update the ppm reading of the CO2 every hour or whenever the user utilizes a forced update utilizing the web server. This means that the user should be updated within approximately 1 minute from pressing the force update.

container and immediately update the server notifying the user.		

2.2.2 Subsystem 2 - Controller



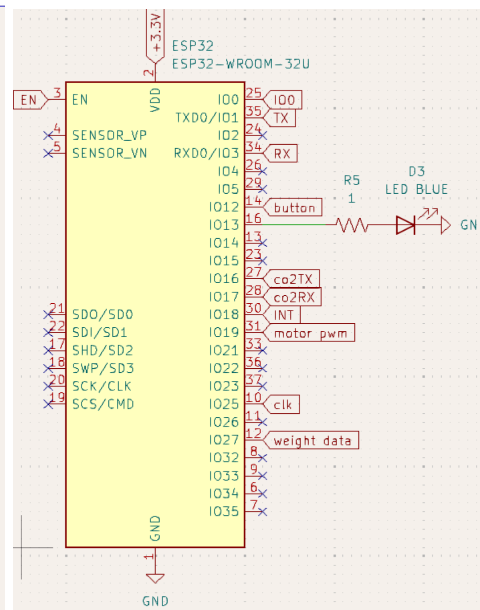


Figure 4.6 (left) - STM32 USB-UART connection and charging port
Figure 4.7 (right) - ESP32 all connections

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 and Verification:

Requirements	**Verifications**
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.	ESP32 has a single and is able to host servers that the user may interact with. Which should be reachable as long as both the container and phone are within

	an internet range. Which should have the ability to communicate with the ESP32 with minimal delay at most 5 seconds communication delay.
The ESP32 must be easily programmable to quickly debug	There should be no other steps beyond simply plugging in the usb micro b connection to the board and uploading a program to reprogram the ESP32.
The controller should be interrupted by the user pressing the outer button or interacting on the app to update or putting in new beans.	Regardless of what current process the ESP32 is doing it should put the user first and accommodate the users commands with minimal delay. This should never result in a mixup in commands or a crash in the software onboard.

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 CO2 stored in the beans. Through the mobile interface, the freshness of the beans will be reported as the approximate percentage of CO2 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 CO2 calculations.

Parts:

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

Requirements and Verification:

Requirements	**Verifications**
Reliable Wi-Fi connection to have constant connection to the controller subsystem in order to send and receive data.	Verify that the controller subsystem maintains a continuous and stable Wi-Fi connection by monitoring connection status over a prolonged period and under varying conditions (e.g., distance, interference). This can include automated tests that repeatedly ping the controller and log any disconnects or latency spikes.
A server should be running on the microcontroller to receive requests from the mobile interface.	Confirm that a server is running on the microcontroller by sending test requests from the mobile interface and verifying that valid responses are received within an acceptable response time. Log and

	analyze the responses to ensure all endpoints behave as expected.
Sensor readings and a freshness report should be displayed and user inputs should be available on the mobile interface.	Validate that sensor readings and the freshness report are correctly displayed on the mobile interface by comparing live data against known calibration values and verifying that the display updates. Also, test user inputs (such as selecting bean types or triggering the lid) to confirm that changes are immediately reflected on the interface and that commands are properly executed by the controller.

2.2.4 Subsystem 4 - Power System

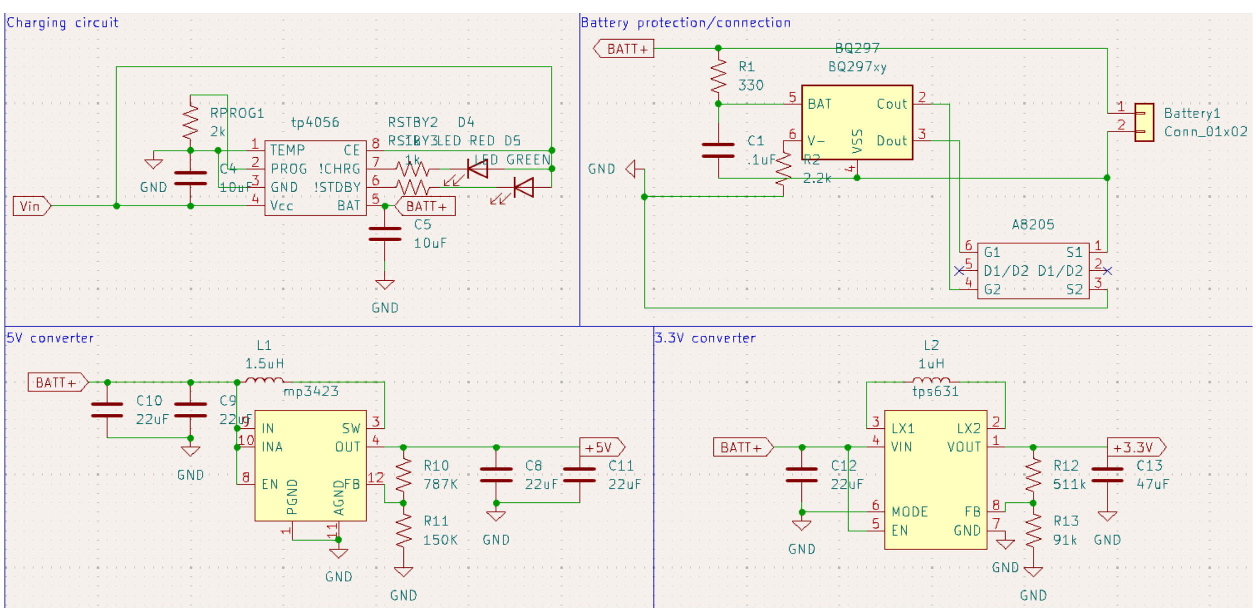


Figure 4.8 - Power system modules including 5V, 3.3V converter and BMS systems

Functionality: For power, we have decided to use a Samsung 50E battery, which is 5000mAh and supplies 3.6-4.2V at a maximum of 9.8A. This battery is also rechargeable via USB using our 3PEAK linear charging circuit (using the tp4056 component). The battery will also be connected to a protection circuit using TI's BQ297 single cell battery protector. From the battery, we will connect it to 2 high efficiency DC-DC converters (both above 90% efficiency with our circuits). The first DC-DC converter is the TI TPS631, which converts our battery input into a 3.3V output with a maximum output current of 1.5A. Our second DC-DC converter is the Monolith Power MP3423 step-up converter, which allows us to output 5V at 3.1A maximum. These two DC-DC converters will then supply the power to all the other components in our circuit, as they all run off of 3.3V or 5V. In our outputs section, you can see the breakdown of which components run at which voltage and what is our maximum expected current that they will draw.

Interaction:

Inputs:

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

Outputs:

3.3V Circuit:

- ESP32 at 500mA max
- PASCO2V15 CO2 Sensor at 10mA max
- Push button with negligible current

5V Circuit:

- Motor at 1A max
- PASCO2V15 CO2 Sensor at 300mA max
- SEN-13329 and SEN-13879 weight sensor module at 5mA max

Part:

Lith-ion 3.7V rechargeable Battery

Battery Protection: Texas Instruments BQ29700DSER

Battery Linear Charger: 3PEAK TPB4056B2X-ES1R

3.3V 1.5A Buck Converter: Texas Instruments TPS631000

5V 3.1A Buck Converter: Monolith Power MP3423

Requirements and Verification:

Requirements	**Verifications**
Supplies continuous power to all subsystems at rated voltages without dropping current required during operations when the motor is activated. Is able to recharge and continue functioning without a jump in voltage or current from the recharge port.	The power supply for the 3.3 V should never diverge more than 0.2 V even if the motor is running at peak power or if the user plugs in the device. The 5V power supply should never diverge more than 0.4 V under the same conditions. This allows for enough leeway for all devices to operate at their rated voltages. This will be tested by monitoring the voltage of the power lines when turning on the motor or plugging in the devices as these are the times with the greatest disturbance in the system.
Protects the battery from overcharge and protects from surge in power demand.	To test that the battery is safe from overcharge, the current to the battery will be monitored while charging to be sure that the on board linear charging IC is correctly operating. To test battery protection a similar process will be done

	when a large demand is suddenly pulled from the battery and the battery protection module correctly eases the battery into the new power demand.

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:

$$420(ppm) \,/\, 10\text{^}6 \;*\; 9.051(g) \;=\; 0.0038(g)$$

total amount of CO2 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 CO2 sensor range.

3. Cost and Schedule

3.1 Cost Analysis

The following table lays out all material and parts that will be required to construct this project and includes the estimated labor cost for those working on the project.

Part Costs and Description						
Part Number	**Manufacturer Part Number**	**Description**	**Quantity**	**Unit Price**	**Extended Price USD**	**links**
1568-1436-ND	SEN-13879	LOAD CELL AMP HX711	1	$10.95	$10.95	link
336-5886-1-ND	CP2102N-A02-GQFN20R	IC USB TO UART BRIDGE QFN20	1	$5.59	$5.59	link
296-43985-1-ND	BQ29700DSER	IC BATT PROT LI-ION 1CELL 6WSON	1	$0.52	$0.52	link
4518-8205ACT-ND	8205A	MOSFET 2N-CH 20V 5A SOT23-6	1	$0.37	$0.37	link
5503-TPB4056B2X-ES1RCT-ND	TPB4056B2X-ES1R	LINEAR BATTERY CHARGER 1 CELL 8-	1	$1.05	$1.05	link
1589-1642-1-ND	MP3423GG-Z	IC REG BOOST ADJ 9A 14QFN	1	$3.46	$3.46	link
296-TPS631000DRLRCT-ND	TPS631000DRLR	DC DC CONVERTER	1	$1.42	$1.42	link
1568-1900-ND	SEN-14729	LOAD CELL 5KG STRAIGHT BAR TAL22	1	$13.12	$13.12	link

445-8657-1-ND	MLZ2012M1R0HT000	FIXED IND 1UH 800MA 100 MOHM SMD	1	$0.10	$0.10	link
445-6757-1-ND	MLZ2012N1R5LT000	FIXED IND 1.5UH 900MA 100MOHM SM	1	$0.10	$0.10	link
BH-18650-PC-ND	BH-18650-PC	BATTERY HOLDER 18650 PC PIN	1	$3.12	$3.12	link
1568-1488-ND	PRT-12895	BATTERY LITH-ION 3.7V 2.6AH 1865	1	$6.62	$6.62	link
PASCO2V15AUMA1	Infineon Technologies	CO2 SENSOR IN PPM	1	$20.92	$20.92	link
Values include: 1, 22, 330, 1k, 2k, 2.2k, 10k, 91k, 150k, 511k, 787k	--	Various resistors of different values	17	$0.10	$1.70	--
Values include .1uF, 4.7uF, 100nF, 10uF, 22uF, 47uF	--	Various capacitors of different values	19	$0.10	$1.90	--
Colors include: Green, Red, Blue	--	different colored LEDs	3	$0.14	$0.42	--
BC817-25LT1G	onsemi	TRANS NPN 45V 0.5A SOT23-3	2	$0.15	$0.30	link
10118194-0001LF	Amphenol ICC (FCI)	CONN RCPT USB2.0 MICRO B SMD R/A	1	$0.41	$0.41	link
HS-318	Hi-Tec	25 TOOTH SERVO MOTOR	1	$18.78	$18.78	link
TS02-66-50-BK-260-LCR-D	Same Sky (Formerly CUI Devices)	PUSHBUTTON THT	2	$0.10	$0.20	link
--	OVERTURE	1.75MM PLA 3D PRINTER FILAMENT	1	$24.00	$24.00	link
Total part cost:					$115.05	

Labor Cost

Number of People	**Average hourly salary**	**Estimated individual hours worked**	**--**
3	$45	200	$27,000 --
GRAND TOTAL COST:			**$27,115.05**

Note: average salary estimation was calculated using an average ECE salary post-graduation in illinois. The Estimated individual hours were approximated based on the amount of hours previously worked and the expected amount of hours going forward to meet project goals.

3.2 Schedule

Week of	Task	Assignments
March 10	Complete setup for a breadboard demo using dev board. Part should arrive and can begin sorting to begin soldering on PCB arrival. Begin CAD for container design.	breadboard demo
March 17	Enjoy Spring Break	spring break
March 24	All parts arrived, began soldering and board testing. Print CAD design for container and test mounting. Iterate as needed.	
March 31	Iterate PCB design as needed and order new boards if required.	third round PCBs / individual reports
April 7	Continue testing and debugging to correct or better all components.	fourth round PCBs
April 14	Begin preparing for the demo and	team contract assessment

	assembling what is to be presented. Begin final presentation and papers to outline project and all components.	
April 21	Present mock demo and make any final corrections as needed for final demo.	mock demo
April 28	Present in final demo Start coordinating presentation	final demo/mock presentation
May 5	Complete final presentation and submit papers.	final presentation / papers / checkout

4. Ethics and Safety

4.1 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 CO2 is retained. The initial CO2 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 CO2 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.

4.2 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.

5. 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.