ECE445 Senior Design Laboratory

Automated Tea Maker Design Document

Team #29
Kyle Humphries (kylebh2)
Tanmay Mittal (tmitta3)
Milan Patel (milanhp2)

Fall 2024

October 2nd, 2024

Introduction

Problem and Solution

Dry tea leaves are a great cost-effective alternative to buying tea bags. They offer higher quality and allow many flavors to choose from. However, storing leaves and measuring out precise amounts can become quite tedious. They must be kept within specific temperature and humidity ranges to avoid any leaf degradation over time. With multiple types of tea leaves out there, many users would benefit from being able to easily create their blends. New tea drinkers would be able to experiment and see what flavor combinations best suit their taste profile. Traditional methods of storing tea leaves include containers, tins, and ceramic jars. These provide no real-time feedback as to whether or not the conditions for that specific leaf are being met. Users may unknowingly lose out on money as the aroma and quality of the leaves diminish. Also, blending different flavors requires precise measurements to get the most out of the final result. An all-in-one storage and dispensing unit currently do not allow users to automate the monitoring and dispensing process simultaneously.

Our proposed solution is to offer users an integrated product that allows them to get the most out of their tea experience. They would be able to store dry tea leaves inside separate vertical units. Each unit will house temperature and humidity sensors to monitor the quality of each unit. This information will be relayed back to a series of LEDs attached to the device. Different colors will provide users with insight into the conditions in which the leaves are being stored. This will also be communicated to a simple app that users would be able to use. They would be able to see these conditions in the app and be notified when levels get low. IR sensors would help with this as the levels within the vertical units decrease over time and eventually go below the sensors. The app would also alert users when their blend was ready to be measured out. The second benefit would be allowing users to input different combinations and amounts of each flavor into the app. Once they are ready to dispense the blend, they place their mesh spoon in a weighing tray where the product will fall. A series of motors will measure out the correct proportion of the blend and drop it into this weighing area, directly into the spoon. Once the weight sensors under the tray notify the dispensing process to stop, users would simply be able to remove the spoon and steep their hot water. This solution will automate the entire process of storing and measuring out tea leaves. Users will be able to easily experiment with different blends and get the most out of their tea-drinking experience. The final enclosure would roughly be 6 inches by 8 inches by 8 inches.

Visual Aid

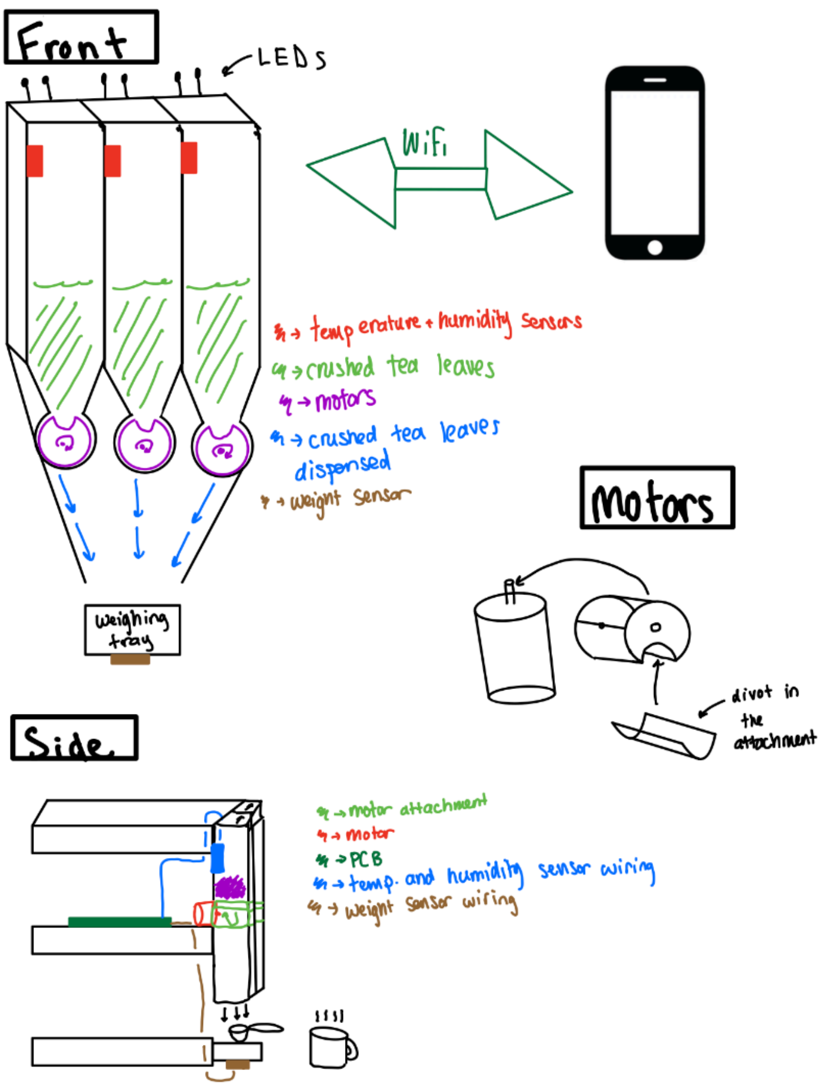


Figure 1: Rough sketch of the proposed design

The proposed solution is to have a unit that stores the tea leaves and is connected to a series of motors. The motors will require an additional attachment to allow leaves to fall into a small divot as they rotate. The backside of the unit will house our PCB, electronic components, wiring, and power source. The communication between the app and microcontroller will take place through WiFi. Finally, the blend will fall into a spoon where it will be measured in real-time. Users can then prepare their cup of tea.

High-Level Requirements List

- The dispensing component should be able to dispense precise amounts of blend combinations, as specified through the app. The result should be measured within $\pm$5 grams of the intended weight according to how much tea will be prepared.
- Temperature and humidity sensors within the vertical storage units should be able to accurately measure conditions and update users. This should convey temperature readings with an accuracy of $\pm$1% and humidity readings with an accuracy of $\pm$5% for relative humidity. These results need to be conveyed back to the app as well as LEDs attached to the storage units.
- Users must be able to utilize a simple app to trigger the dispensing process as well as check the conditions of each storage unit. Updates to and from the microcontroller must have minimal latency. Users should be able to request different amounts of dry leaves and store combinations for future use. There should also be an emergency stop option where users can halt the current dispensing process.

Design

Block Diagram

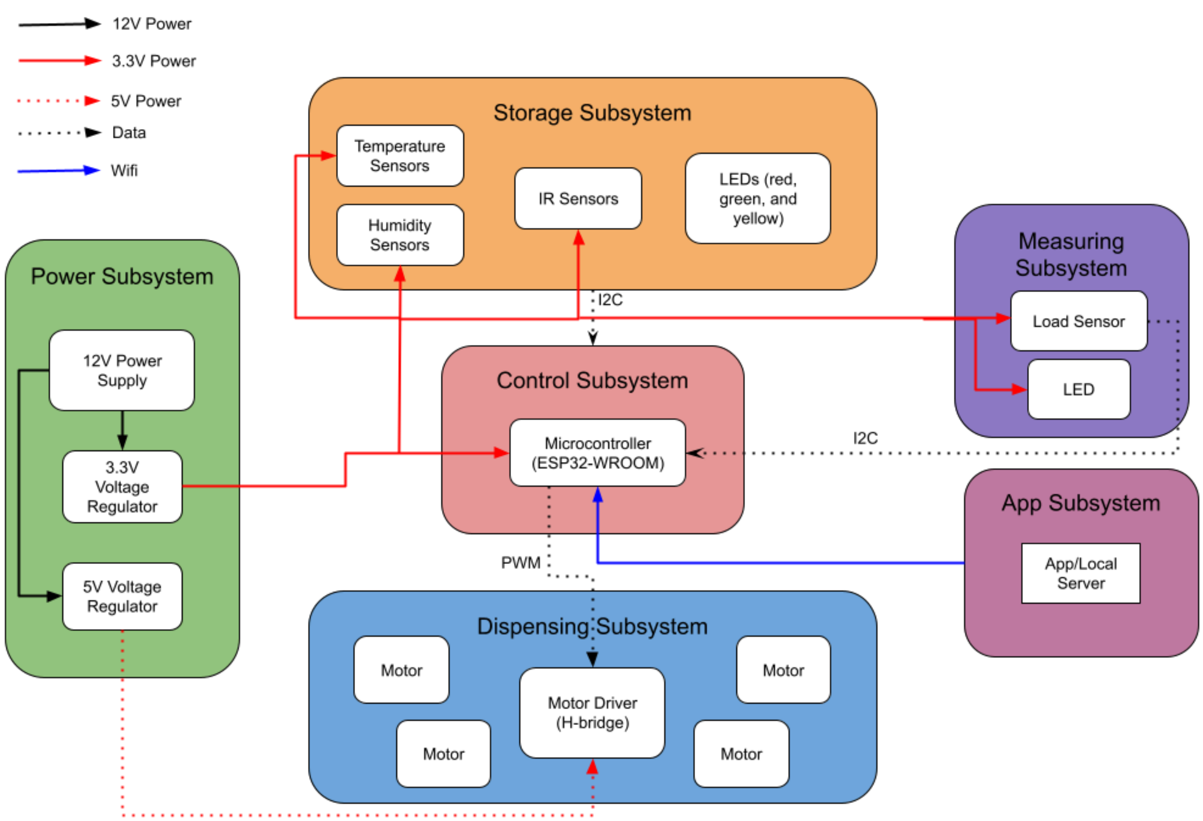


Figure 2: Proposed block diagram

Power Subsystem

The power subsystem will be responsible for taking a centralized 12V source and converting it to necessary voltages for other subsystems in the design. Specifically, lower voltages of 5V and 3.3V are required. Buck converters will be used to convert power drawn from a wall source to the desired voltages for different subsystems. Both of these voltages will be the result of voltage regulator circuits that hold the voltage and current steady. The 5V line will need to help power the motor driver within the dispensing subsystem. A series of stepper motors will help dispense precise amounts of tea leaves. The 3.3V line will power the rest of the subsystems. The storage subsystem will contain a series of sensors that require a fairly small amount of power. The load sensor contained under the tray that sits within the measuring subsystem will also require 3.3V. Most importantly, our microcontroller (ESP32 WROOM), is the final component that will need to be powered by this line. This will be responsible for performing all operations and directing different subsystems.

Requirement	Verification
Must be able to supply 5±0.5V to the dispenser subsystem.	1. Assemble the dispensing unit and hook up the power supply (wall connection) 2. Use a multimeter to measure the voltage across the buck converter connected to the dispensing components 3. Ensure the voltage readings are correct by checking that they are within the range of 5±0.5V
Must be able to supply 3.3±0.5V to the sensors, LEDs, and microcontroller.	1. Assemble the dispensing unit and hook up the power supply (wall connection) 2. Use a multimeter to measure the voltage across the buck converter connected to all sensors, LEDs, and the microcontroller 3. Ensure voltage readings are within 3.3±0.5V for each sensor, LED, and microcontroller

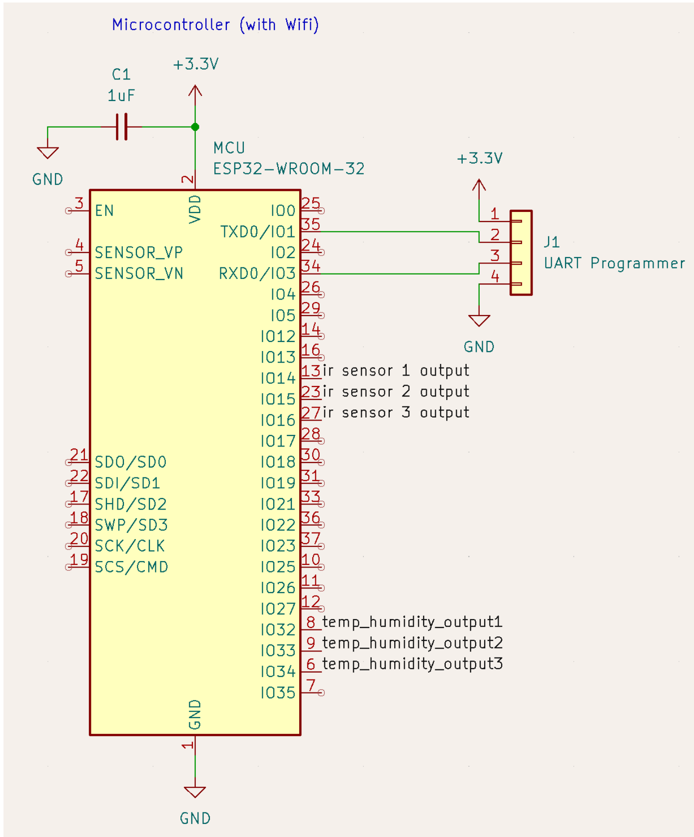
Control Subsystem

The control subsystem will include our microcontroller, the ESP32 WROOM. This specific MCU supports WiFi/Bluetooth functionality which is crucial for the user interface of our design. The MCU will help monitor storage conditions, trigger and verify the measuring process, and control motors accordingly. Real-time feedback will be relayed to a local app/website where users can begin the dispensing and choose blends. They can also check the temperature, humidity, and stock levels of the leaves being stored. The motor driver requires PWM/GPIO signals from the control subsystem as well. Finally, LEDs located on the measuring and storage subsystems will need to be lit up whenever conditions fall outside a predetermined threshold and when the dispensing process is complete. Sensor readings will need to be converted from analog to digital signals before feeding into the microcontroller.

Requirement	Verification
Must be able to receive signals from temperature, humidity, and IR sensors and convert from analog to digital signals.	1. Test each sensor individually to verify that signals can be received continuously. 2. Use commercial products like

	thermometers and humidity meter sensors for measurements to gather readings to compare sensor readings. 3. Compare the readings from commercial sensors and our sensors to verify that they match up. 4. Perform similar testing for the sensors when on the PCB and operating in normal customer conditions and compare the readings from our established baseline.
Must be able to send PWM signals to the motor driver that will control various stepper motors in the dispensing unit.	1. Individually test each motor driver by sending PWM outputs from the microcontroller. Use an oscilloscope to verify that the packets sent over follow the correct protocol. 2. Check if the motor drivers are receiving the signals and verify that the signal looks the same as the one sent by the microcontroller.
Must be able to obtain readings from load sensors efficiently and stop sending signals to the motor controller to stop the dispensing process. Readings should be produced with an accuracy of $\pm$0.5g from the intended amount requested.	1. Test the load sensor individually to ensure that the sensor isn't defective by uploading a simple sketch to take readings periodically and display the output. 2. Compare the load sensors output readings to conventional scale readings for the same material. 3. Request various tea amounts from the device and check with the scale if the correct amount of tea was dispensed. 4. Check the PWM signals sent to the motors and see if they receive the shut-off signal in time to ensure the correct amount of tea is dispensed.

The following is a rough schematic of the microcontroller and USB adapter used to program it. It will provide support for various sensor connections, including temperature, humidity, and break beam. The ESP32-WROOM also has Wifi capabilities. Included below are the proposed GPIO pins that will be used to receive signals from the sensors. These pins will convert them from analog to digital so the microcontroller can decide how to direct the readings.



Schematic 1: Control subsystem

App Subsystem

The app subsystem will host a local server that can be accessed through WiFi. Users will be able to monitor and dispense different blends. They will be presented with an interface that shows the temperature, humidity, and amount in each tea leaf storage unit. This will be able to communicate, over WiFi, with the microcontroller. The ESP32 WROOM is capable of hosting its own simple server or connecting to an existing network. We will need to develop our own protocol (most likely leveraging HTTP GET/SET) to run through a series of different states. The frontend can be implemented using Javascript/CSS while the backend can utilize C++ or Python. We will consider various languages and see which interfaces best with the MCU. The app will also need to be able to efficiently convey different instructions to the microcontroller. Finally, users can expect alerts when levels within the storage containers fall below a threshold. Also, they will be notified when the blend is done being dispensed and ready to steep.

Requirement	Verification
Must be able to efficiently send and receive instructions from the microcontroller with	1. Test that the device is communicating well with the app. Turn on the device

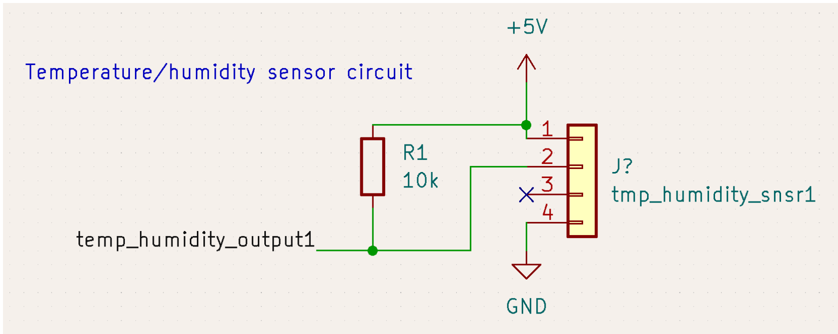
minimal latency of less than 1 second.	and change the amount of tea leaves in the storage units and see if the app receives the updated information. 2. See if the app can now send signals to the device. Input the type of blend that you want and spectate if the device accurately receives the data and does the command. 3. Test the amount of time it takes to receive the updates. Run the tea making process and wait until tea is finished. When done the user should receive a notification so start a timer once the tea is finished and see how long it takes for the information to be relayed to the app.

Storage Subsystem

The storage subsystem will house the actual leaves in separate vertical storage units. We will work alongside the machine shop to create an enclosure that can easily fit in an everyday kitchen. The units will need temperature, humidity, and IR sensors placed within them to help monitor conditions. The leaves will sit atop a series of motors with customized divots in them to scoop small amounts. The top of the storage subsystem will be covered in lids that can be removed. The backside of this enclosure will leave room for our PCB and electrical components. LEDs on top will light up different colors depending on the conditions and whether or not they fall outside the healthy range. Overall, this subsystem will work closely with the microcontroller to notify users of important information regarding their stored tea leaves. IR sensors will be pointed into the units and trigger once the tea leaf level falls below it.

Requirement	Verification
Must be able to send accurate signals from temperature, humidity, and IR to the microcontroller according to their respective data sheets. Temperature and humidity readings are expected to be 72-77F and 30%-50% for optimal storage	1. Test that the sensors are reading the correct levels. Take readings inside the compartment before testing by measuring with a thermometer and commercial humidity sensor. 2. Individually test the humidity temperature sensors compared to the values just listed.
Must be able to alert users, through an LED, when readings fall outside a specified threshold.	1. Test to make sure the LED lights work when storage is low. 2. Empty the storage compartment and check if the LED light has lightened up to the correct color.

Sensor connections will be fed directly into the PCB from where the sensors reside in the storage unit. They will require 3.3V of power and will output analog signals. The microcontroller will need to convert these to digital. The circuits may require pull-up resistors, but this depends on the exact parts and wether or not they come assembled in breakout boards already. Below is an example of a sensor component. It includes a pull-up resistor as well.



Schematic 2: Humidity sensor

Dispensing Subsystem

The dispensing subsystem will contain a series of stepper motors and be located below the storage area. The leaves will sit on top of these motors that will have custom divots in them to scoop ingredients. The size of these divots will depend on how granular the leaves end up being. A motor controller will need to process PWM signals and output the correct voltage/current to each motor according to what the microcontroller tells it. This will most likely be some sort of H-bridge circuit. The precision of our product depends heavily on whether or not these motors can reliably rotate and not get jammed up. They will eventually stop rotating as the dispensing process yields the correct weight of the blend requested. As the motors scoop a small amount of each leaf, they will fall through a funnel and drop into the measuring subsystem. This component will rely on gravity to make sure that the full amount of the measured leaves make it through. Motors will turn one by one to avoid any current complications.

Requirement	Verification
Must be able to receive signals from the motor controller and rotate stepper motors a full 360 degrees while also allowing control over varying step sizes.	1. Test that the motors can receive signals by sending the motor a PWM signal to run and seeing if the motors starts working 2. Send the signal to dump the tea leaves and ensure that the motor rotates a full 360 degrees 3. Input 3 different amounts of tea leaves to the steeping system and ensure that all the varying loads can be dumped.

Must be able to stop when notified by the microcontroller that the dispensing process has been completed. There should be minimal latency with 0 extra rotations.	1. Connect the dispensing components (motors) to the output of the buck converter used to step down voltage from wall to 5V 2. Send the signal within microcontroller code to the motor to run and then send the signal for the motor to stop. 3. View whether the motor stops spinning and that there are 0 extra rotations

Measuring Subsystem

The measuring subsystem will be the final destination for the tea leaves once they are dispensed. Users will be able to place a spoon here that they will use to steep their tea. The spoon will need to be calibrated since the tray that it sits on will be on top of a load sensor. We are considering having a button on this subsystem that users can press to "zero" out of the scale. The load sensor will need to accurately and quickly relay updated measurements to the microcontroller. This is what will lead to the most precise measurements. Once the desired amount has been reached, there will be an LED that lights up here notifying users that their blend is ready for a cup of hot water.

Requirement	Verification
Must be able to accurately make readings using a load sensor from 0-500 grams with 5% accuracy.	1. Include code on the microcontroller to output readings from the load sensor 2. Add different amounts of dry tea leaves, ranging from 0-500 grams, to the load cell and note values output from the microcontroller 3. Manually weigh the same amount on a smaller, cheap weighing scale 4. Compare the two values and ensure that readings from the microcontroller are within $\pm$5% of the expected value
Must be able to register fine changes in measurements and allow users to calibrate a steeping spoon to "zero" out the scale.	1. Include code on the microcontroller to output readings from the load sensor and provide the option to “zero” out the scale 2. Manually add different small amounts of dry tea leaves to a stepping spoon that will rest on the load cell and check debugging output from microcontroller 3. Remove dry tea leaves from the steeping spoon and keep the spoon

	on the load sensor 4. Verify microcontroller readings are non-zero and press the button near the measuring subsystem to “zero” out the scale 5. Check that the new readings from the microcontroller are now zero

Tolerance Analysis

A main component of our design is accurately measuring the desired weights of different tea blends. One requirement of our control subsystem is to have this accuracy within 0.5 grams of the intended amount the user requests. We plan on working with the machine shop to create custom attachments for the motors in the dispensing subsystem. These attachments will go on the ends of the motor and appear as divots in the shape of a half circle. The radius and length of this divot will depend on the amount of a single tea leaf we want a single rotation of a motor to dispense.

The measurements will depend on the density of the dry tea leaves we use. Since this will vary depending on the type of leaf, we decided to have the tolerance be quite high initially. As we progress with the design, we may work at refining this. Overall, the goal is to have a single rotation measure out 0.5 grams of that tea leaf. The leaves will be a bit more crushed to make it easier to scoop. The following calculations represent how we plan on shaping the motor attachment to attain this goal. We will assume that the average density of the tea leaf we will work with will be between 0.2 g/cm^3 and 0.4 g/cm^3.

Density of dry tea leaves = 0.2 g/cm^3 to 0.4 g/cm^3

Volume upper bound (smaller density):
Volume = Mass / Density = 0.5 g / 0.2 g/cm^3 = 2.5 cm^3

Volume lower bound (larger density):
Volume = Mass / Density = 0.5 g / 0.4 g/cm^3 = 1.25 cm^3

The cross-section of the motor attachment's divot will be in the shape of a half-circle with radius r and length L. The following shows the volume of dry tea leaves the divot can hold with these dimensions:

Volume = Half-circle cross-section area × Length
Half-circle area = $(1/2)\ \pi\ r^2$

So, the volume of the divot becomes:
Volume = $(1/2)\ \pi\ r^2 \times L$

The range for the product of r^2 and L is as follows:

$$2.5 = (1/2)\,\pi\, r^2 \times L$$
$$r^2 \times L = (2.5 \times 2) / \pi \approx 1.59$$

$$1.25 = (1/2)\,\pi\, r^2 \times L$$
$$r^2 \times L = (1.25 \times 2) / \pi \approx 0.795$$

Depending on what we choose for the radius (r) and length of the divot (L), we can get close to a single rotation measuring out 0.5 grams of dry tea leaves. The range for the product shown above will be from 0.795 to 1.59 since the range of tea leaf densities is from 0.2 g/cm^3 to 0.4 g/cm^3.

Let's assume we take the radius of the divot to be 1 cm. The calculation for the length of the divot will be the following for a density of 0.2 g/cm^3:

$$1^2 \times L = 1.59 \rightarrow L \approx 1.59 \text{ cm}$$

And the following for a density of 0.4 g/cm^3:

$$1^2 \times L = 0.795 \rightarrow L \approx 0.795 \text{ cm}$$

This means we can take a length between 0.795 cm and 1.59 cm. A tolerance of 0.5 grams is required because not all of the divot will actually be filled with dry tea leaves. Since they are solid, there will be gaps in the scoop which can lead to the rotation giving out less than 0.5 grams. Moving forward, we will assume that roughly between 60% and 80% of the divots can be filled with dry tea leaves. This means that a single rotation can potentially scoop out 0.3 grams to 0.4 grams. For multiple rotations, this can amount to a large amount of error. We will need to consider this by either making the divot larger than it needs to fill 0.5 grams or crushing up the leaves to make them finer. A proposed solution is to make the divot fit closer to 1 gram so that it is less of an issue when only 60% to 80% ends up being occupied.

A potential issue we may face is having to account for the time that the dispensed tea leaves are in the air before landing on the load sensor. We have decided, through testing, that we will determine an upper limit on the average time that this takes. Then, we will be able to implement this through software and have a timeout while we wait for the load sensor reading to update. It will be a delay that allows the dispensed tea leaves to fall through the design and into the cup/mug on top of the weight sensor. As for when leaves get stuck, we will have a method that allows users to abort the process and use the last know value from the load sensor and pick up the dispensing process from there.

Cost and Schedule

Cost Analysis

Quantity	Part	Part Number	Manufacturer	Total Cost ($)	Link
2	Microcontroller	ESP-32-WROOM	Espressif Systems	$5.36	Link
1	USB 2.0 to TTL Module Serial Converter Downloader	CP2102	HiLetgo	$7.39	Link
4	Temperature Sensor	296-LM61BIZ/LFT3 CT-ND - Cut Tape (CT)	Texas Instruments	$5.36	Link
4	IR Break Beam Sensor	2167-ND	Adafruit Industries LLC	$11.80	Link
3	Humidity Sensor	DHT11	WWZMDiB	$5.99	Link
1	Load Cell	TAL220	Sparkfun	$9.50	Link
3	Step Down Converter	DFR1015	DFRobot	$5.30	Link
6	5V Stepper Motor	290-028	ECE Supply Center	$43.32	Link
3	Stepper Motor Driver	1528-4489-ND	Adafruit Industries LLC	$9.00	Link
N/A	Circuit components	Resistors, capacitors, inductors, LEDs	ECE Supply Center	$10.00	N/A
				Total = $113.02	

Category	Hours	Cost
Tanmay(labor)	145	$52.65*145*2.5= $19,085.63
Milan(labor)	145	$52.65*145*2.5= $19,085.63

Kyle(labor)	145	$52.65*145*2.5= $19,085.63
Machine shop service	5	$100*5 = $500
Parts	-	$113.02
		Total = $57,869.90

Our team includes two computer engineers and one electrical engineer. The average salary, according to Grainger College of Engineering website, is $109,510 [3]. This is roughly $52.65 per hour. We expect the total spend on the project 145 hours on this project. This brings the total cost for 3 team members to $52.65 (hourly wage) * 2.5 (overhead factor) * 145 (total hours spent) * 3 (total team members) = $57,256.88. We will also be utilizing the machine shop where we expect the average hourly salary to be around $25-$30 per hour. So assuming about two people work on our project plus all the materials required, the overhead of running the machine shop, and a bit of profit we found that the labor cost plus material cost for a machine shop is around $100 an hour.

Schedule

Week	Task	Team Member
9/30	Meet with machine shop to discuss next steps	All
	Look into which sensors will be used	Milan
	Start researching the power supply subsystem	Kyle
	Finish visual aid for the machine shop	Tanmay
10/7	Design review with professor and spectate another peer’s design	All
	Work on PCB design	Milan
	Assist with PCB and finalize everything needed by machine shop	Kyle
	Work on app development	Tanmay
10/14	**First-round PCB orders due,** Finish PCB and send in the design to get it ordered.	All
	Assist with finishing the app	Milan

	Track down the parts needed and start testing	Kyle
	Continue working on app	Tanmay
10/21	**Second-round PCB orders due,** Receive the PCB that was ordered last time and begin testing	All
	Assist with the circuit and electronic components	Milan
	Start testing the PCB with other electronic components	Kyle
	Finish app or get it near completion	Tanmay
10/28	**Third-round PCB orders due,** If needed make fixes to PCB and send in the new PCB design. If not, start wiring up everything else surrounding the PCB.	All
	Help with testing the parts and making sure the circuit design works as expected	Milan
	Make sure all parts have arrived and continue testing parts	Kyle
	Move working parts from circuit to PCB	Tanmay
11/4	**Fourth-round PCB orders due,** Wire up all the parts that have been proven to work to PCB and find solutions to the parts that are not working.	All
	Focus on the electrical components and getting them working with PCB	Milan
	Replace non working parts and help with moving working parts to the PCB setup	Kyle
	Start testing how the app interacts with the physical components. Check how well is app communicating with PCB	Tanmay
11/11	**Fifth-round PCB orders due,** Begin integrating PCB and parts with the mechanical subsystem.	All
	Look for weak points in the design and ensure that each subsystem is communicating well.	Milan
	Make sure all components are mounted and the enclosure is working as intended.	Kyle
	Ensure all the high level requirements are being met and document all of the technical information	Tanmay
11/18	Prepare for mock demo	Everyone
	Carryout mock demo	Everyone

	Start final presentation	Everyone
11/25	Fall break (catch up on work)	Everyone
12/2	Prepare for final demo	Everyone
	Carryout final demo	Everyone
	Carryout mock presentation	Everyone
	Continue working on final presentation	Everyone
12/9	Prepare for final presentation	Everyone
	Present final presentation	Everyone
	Return lab equipment	Everyone
	Turn in lab notebooks	Milan

Ethics and Safety

To ensure there aren't any safety concerns in the design of the Automatic Tea Maker we have identified a few scenarios where the device can potentially cause minor harm. One of the potential safety concerns may be that the motors in the enclosure might not be fully enclosed and can potentially come in contact with the user. To prevent this scenario we will make sure all the moving parts of the system are as inaccessible for the user as possible and make sure no motor is exposed. Next, we will make sure no component in the system gets dangerously hot by managing voltage, motor speed, dissipating the heat or having warning alerts to ensure safety. We will make sure all of the datasheets are met for our components in the design by utilizing step down converters and resistors to keep the devices at safe operating levels.

Overall we will make sure our design abides with all the IEEE code of ethics to ensure that we help improve the lives of our users without harming anyone else in the process[1]. We will ensure that the design is safe for use in various environments and run focus groups to see how the customer is using our product. We will also provide our contact information on the machine so that customers can give us their feedback on the product so that we can keep on improving our design. Along with our contact information we will make sure to provide a set of basic instructions and a list of specifications that must be met to be able to safely operate the device.

Citations

[1] IEEE - IEEE Code of Ethics, www.ieee.org/about/corporate/governance/p7-8.html. Accessed 20 Sept. 2024.

[2] “How to Store Loose Leaf Tea.” Té Company, https://tecompanytea.com/blogs/tea-atelier/how-to-store-loose-leaf-tea?srsltid=AfmBOoq-2firwLshDDqmf0w8pGEdSwiXIFITPQ8P8fLyXJDL195DvOly. Accessed 19 Sept. 2024.

[3] Grainger. “Salary and Hiring Data Portal.” *Illinois.edu*, 2024, ecs.grainger.illinois.edu/salary-data/data-portal. Accessed 2 Oct. 2024.

[4] ESP32-s3-wroom Datasheet https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf. Accessed 4 Nov. 2024.

[5] Load sensor (5kg) Datasheet https://www.adafruit.com/product/4541?gad_source=1&gclid=Cj0KCQiA_qG5BhDTARIsAA0UHSKZQQS8BUprBMv5XUbLj2EMTkmEkiE1b9DGn8CUZXc8_IUBGpFBje4aAo0YEALw_wcB. Accessed 4 Nov. 2024.

[6] ESP32 Web Server Example https://randomnerdtutorials.com/esp32-web-server-arduino-ide/. Accessed 4 Nov. 2024.