Pour-over-and-over

Rio Pacheco, Corrado Govea, Elijah Knupp

Department of Electrical and Computer Engineering, Carnegie Mellon University

Abstract—**A system capable of automatically brewing a barista-level cup of coffee utilizing the pour-over brewing technique. Typically, this process is difficult to learn and largely inaccessible to those with fine motor control deficiencies. Our machine will bridge this gap by requiring minimal input and effort from the user. It will have, at minimum, 5 different brewing presets (with customizable variables) and will be able to brew up to 300mL of coffee. The machine will have a precise and repeatable brewing process, with all variables being within 10% of what the user specified.**

Index Terms—**Automation, Coffee, Gantry, Robot**

I. INTRODUCTION

THIS world of coffee brewing is diverse and complex. There is a plethora of different brewing methods, each with their own advantages and disadvantages. However, there is a consistent pattern that emerges when one begins to explore these different options - quality and convenience do *not* typically go together. For the coffee connoisseur, they will focus on the quality of their coffee and will spend the time and effort required to make a premium cup of coffee. A popular choice for these individuals is the pour-over coffee brewing method, and for good reason. As the name suggests, this method requires an individual to pour boiling water over a bed of coffee grounds in a specific, time-intensive manner. They will repeat this process several times to brew a single cup of coffee. Their labor is rewarded with coffee that has full-bodied, highly specific, and differentiable flavors. Another advantage of this brewing method is the high level of customization it offers. Under normal circumstances, when the individual is using a more convenient and rigid coffee brewing process, their options to customize their cup of coffee are usually limited to choosing a different coffee bean. However, with the pour-over coffee method, users may change all the variables associated with the brewing process. This includes adjusting the flow rate, pour pattern, water temperature, and bed agitation. For the casual coffee drinker, who may just want a cup of coffee at the push of a button, this highly variable and detail-oriented approach to coffee brewing may sound like a nightmare. Additionally, this brewing method requires precise physical movements (with boiling water, nonetheless), which can cause accessibility issues for those with fine motor control deficiencies. This often forces those individuals to choose a more convenient coffee brewing experience.

We aim to bridge this gap between a convenient and quality cup of coffee through our machine: the Pour-Over-and-Over automatic coffee machine. After pouring in room-temperature water into the machine, pouring coffee grounds into the filter, and selecting the brewing option they desire, the user will not have to perform any further actions (other than removing the cup of coffee after the brewing process is complete!). Our machine will heat and pour the now-boiling water over the coffee in the exact manner the user specified. For the user looking for a convenient experience, they can simply choose one of the 5 presets that will be programmed into the machine. For the coffee connoisseur, they may edit these presets or create an entirely new preset that will then be saved to the machine for later and repeated use. Our machine will prioritize precise repeatability, with us aiming for the capability of brewing coffee that is within 10% of quantifiable metrics of a cup of coffee brewed using identical variables. As the taste of coffee is highly subjective, this allows the user to determine what a good cup of coffee is to them, with our machine being able to reproduce that same cup of coffee using highly controlled quantitative variables and goals.

II. USE-CASE REQUIREMENTS

A. *Capability*:

To ensure we can make a full cup of coffee we want the device to hold 300ml of water. This will allow us to ensure that we have enough water to brew most if not all types of single cup recipes and ratios. Most pour-over recipes do not exceed 300ml of water, and when they do it ends up being a 2-cup recipe rather than 1-cup.

B. *Parameter Accuracy*:

To produce coffee to the specifications of a recipe the user desires we must ensure that individual parameters are accurate. These parameters are water temperature, poured water amount, and flow rate. It should be able to heat water up to a desired temperature, with a ∓5°F margin from the set temperature. The amount of water poured over the coffee grounds should be ∓10g from the desired amount. To ensure we can control coffee bed agitation we want the flow rate to be controllable, from 0g/s to 12g/s at most.

C. *User Experience*:

Given our goal of making this an accessible experience for people who are not familiar with specialty coffee or pour-over in general, we want to ensure that there is a plethora of options to choose from at the start. This means that the device should have a minimum of 5 preloaded presets to brew coffee with. Specifically, this means 5 different recipes, all with unique pour patterns, timings, agitation, and end results.

D. Repeatability:

To ensure that users can trust that a recipe will produce the same cup of coffee every time (given the same coffee beans), we need to ensure that the amount of coffee extracted is consistent between cups. We will measure the TDS of the resulting cup and ensure that there is at most a ∓0.5% difference in TDS between cups of coffee.

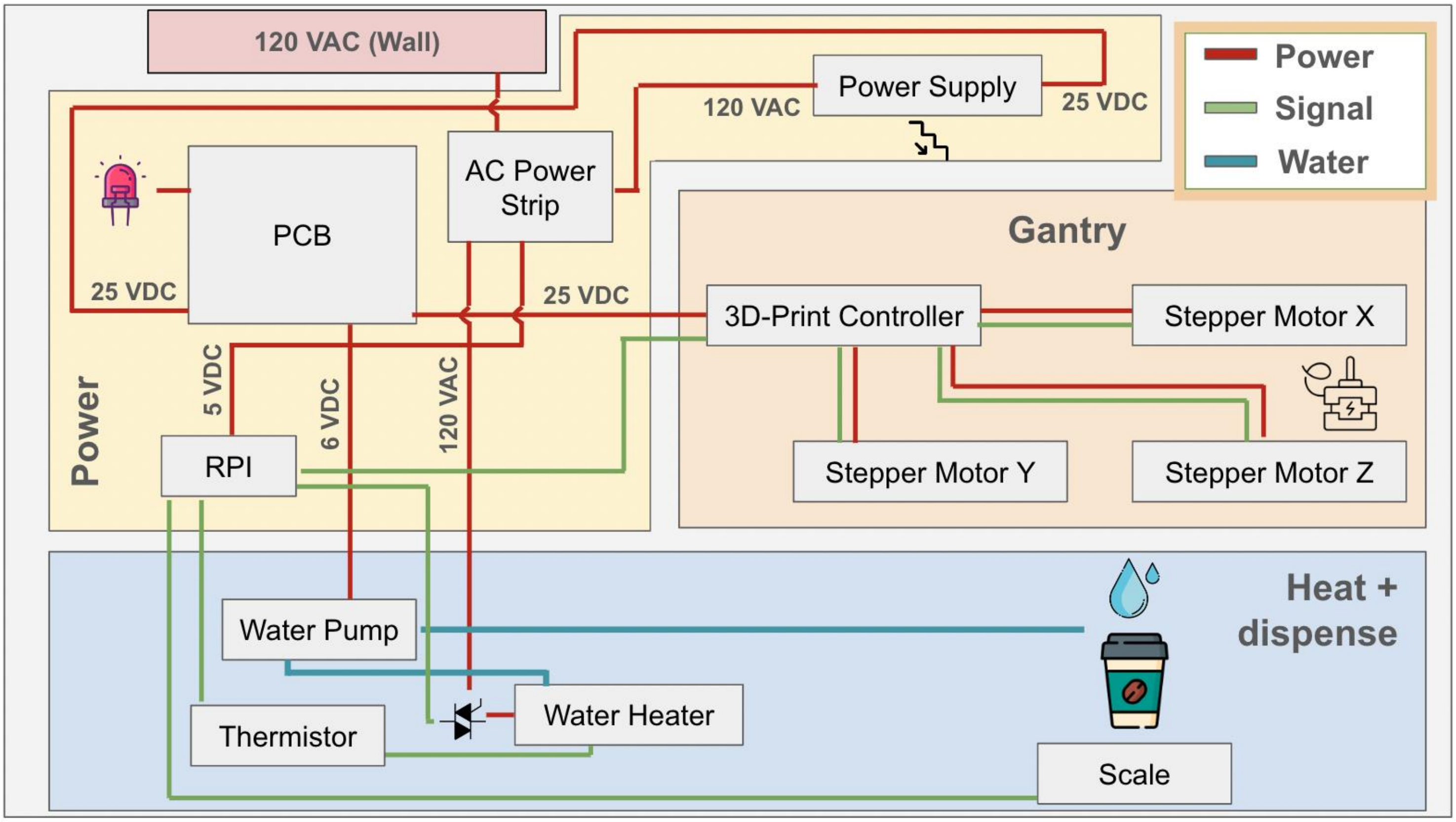


Fig. 1. Overall system diagram

III. ARCHITECTURE AND/OR PRINCIPLE OF OPERATION

Our system is made up of a 3-axis gantry system, a water distribution system, analog and digital microcontrollers, and a custom PCB for power distribution.

As per the hardware architecture plan, it can be divided into several subsystems: the water's heating and distribution system, the gantry and frame, and the scale. For the gantry and frame, the machine will use a re-purposed Ender 3 3-D Printer. All other components will either be integrated into the 3-D printer or mounted to its frame. The water will be heated in an electric kettle (which will also serve as the water reservoir) mounted to the frame of the 3-D printer. It will have a temperature sensor mounted to its lid to monitor and report the exact temperature of the water to the system. The water will be distributed using a food-grade, high-temperature water pump. The pump will be mounted to the 3-D printer's extruder head, replacing the fan that was previously mounted there. Concerning the scale, it will be integrated onto the hotbed of the 3-D printer. It should be noted that the hotbed will be disabled, so that it does not overheat and damage the scale. The scale will be used to provide information regarding the amount of coffee grounds in the filter and the total amount of coffee dispersed by the machine.

As per the software architecture plan (see Fig. 2) the Arduino will be running C++ code to interpret the analog signals from the weight scale. This will be sent over a USB connection to the Raspberry Pi, which will be running our Django web application. The Django web application will be written in Python, HTML, CSS, and JS. A JS script will take in the serial signal of the Arduino output and format the values to be sent to the web app backend code. The backend will handle interfacing with the database to retrieve and save brew profiles, serving the data to the front end, and sending printer movement details to Octoprint. Octoprint will then convert these movement details into Gcode for the mechanical system to interpret and act on. The front end will be displayed via HDMI cable and interacted with via mouse and keyboard plugged into the RPI.

As per the power distribution plan, we take in 120VAC from the wall outlet into an off-the-shelf power strip, which will then power 3 things: the 25VDC power supply, the 120VAC kettle, and the 5VDC raspberry pi. The 25VDC will then connect to the power management board, which will step that down to 6VDC using an LDO circuit to then power the water pump. The 25V power supply will also connect directly to the gantry to power the 3-axis system through the 3D printer controlled. Lastly, we will route 120VAC from the power strip to the electric kettle. To control the AC current going into the kettle, we will use a Triac circuit which will allow us to then regulate the flow rate coming from the pump.

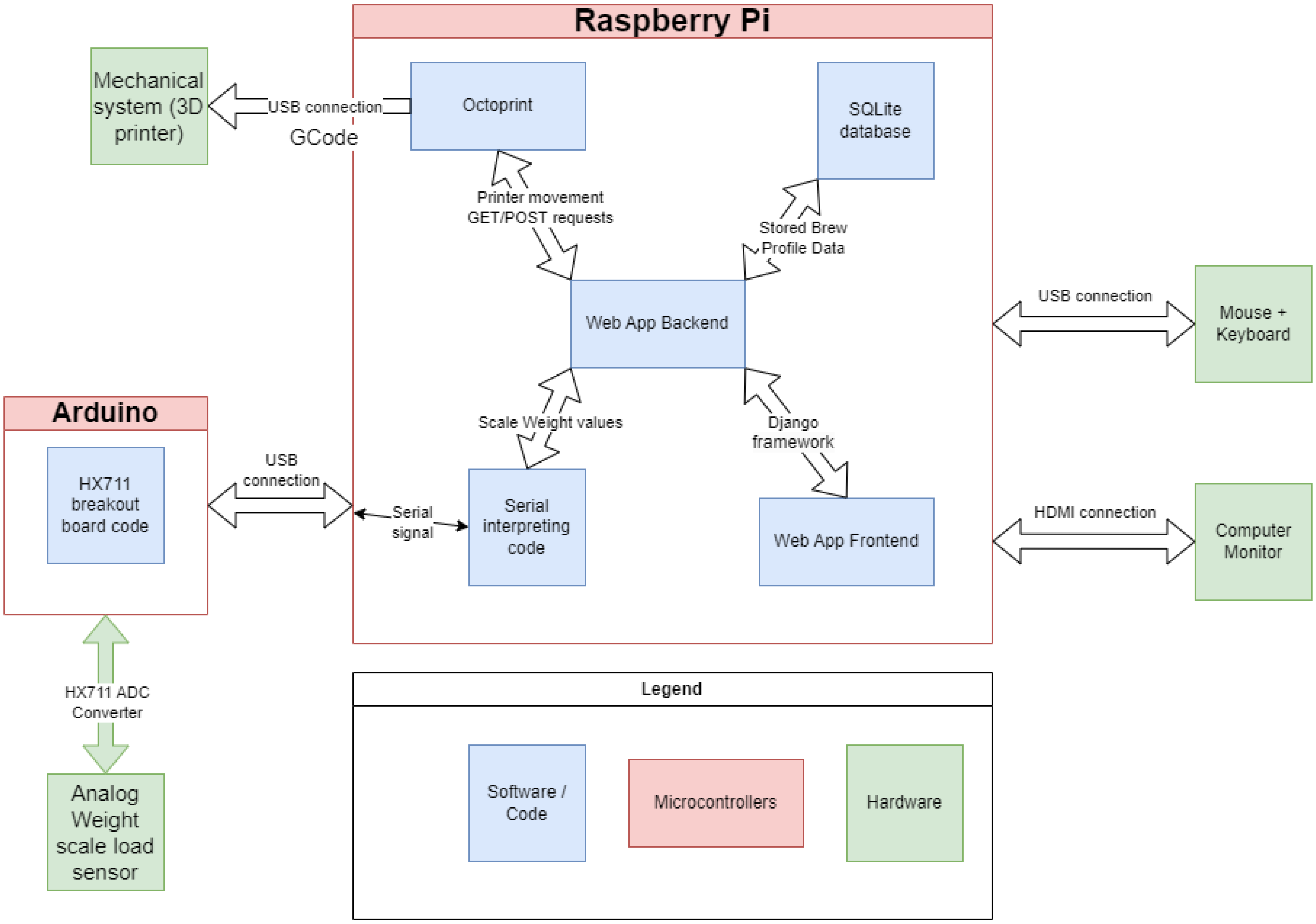


Fig. 2. Software diagram

IV. Design Requirements

A. *Water Tank Capacity of 300mL*

We require that our system can brew a full cup of coffee in order to be feasible for our market audience. In order to fulfil this requirement, the water tank, which in our case is the boiler, must be able to hold enough water to brew a full cup of coffee at once. One cup is equivalent to 236.6mL. However, some of the water will be absorbed by the coffee grounds – specifically, "about 0.5 ounce of fresh water is lost per cup of coffee" [1] which is equivalent to roughly 15mL. To have enough margin, our kettle is required to hold 300mL of water, which gives us a 19% safety margin.

B. *PID Temperature Control of +/-5 F*

Part of our promise to our users is to be able to pour repeatable cups of coffee, and this can only be achieved with reliable parameter selection. For water temperature control, we have established that our PID heating system must have a +/-5F accuracy from the set temperature throughout the pour.

C. *Total Water Flow margin +/- 10g*

An important parameter in pour-over coffee is the ratio of coffee grounds to water poured – this is because higher concentration of coffee leads to a richer end product, while a lower ratio leads to a lighter brew. To ensure the accuracy of our machine, we have set a design requirement of +/- 10g of water from the total set amount to be poured over the grounds – that is, total water poured before the grounds absorb any water.

D. Water Flow Rate Control Range

On the same line of keeping parameters consistent across pours, another very important one is water flow rate. This is because higher flow rate causes more turbulence during the infusion process. To ensure the turbulence is kept consistent, we are controlling the water flow through our pump at a range of 0g/s to 12g/s.

E. Default-Loaded Presets

One of the biggest goals of our project is to make pour-over coffee more accessible, such that it is suitable for experts and beginners alike. We figure that the best way to do this is to have default, pre-loaded presets with carefully curated parameters such that users can simply power on the machine, select a preset, and begin pouring. To give enough flexibility to beginner users, we chose to have a minimum of 5 presets loaded into the machine.

F. Total Dissolved Solids Control

Measuring the quality of coffee, our end result, can be very subjective. Parameters like “flavor” and “bitterness” are very hard to quantify, especially for non-experts in coffee. For that reason, we have chosen to evaluate our end result by measuring the total dissolved solids (TDS) in the final product. Our requirement is for +/- 5% error in TDS throughout the pours.

G. Accurate Patterns

The highlight of pour-over coffee – and our project – is having users create their own patterns for water pouring. For this reason, we want our pouring patterns to be repeatable. Thus, we have chosen our design requirement to be for the gantry to be able to reproduce the same pattern 5 consecutive times without going off-path by more than 500mm. This will ensure pouring patterns are consistent.

V. DESIGN TRADE STUDIES

A. Espresso Machine Parts vs Hardware Hacking

Initially we had considered using off the shelf parts used for espresso machines to be our water heating and distribution system. Our train of thought was that these parts have been industry tested and would likely be more than enough for our purposes. However, it was found that these parts would not align with our use case requirements. Specifically, the boiler we were considering could only hold a maximum of 100ml of water. This would not be nearly enough to heat the amount of water we needed to brew one full cup of coffee. The pump, which was designed to work in tandem with the boiler, would also need to be scrapped. This led us to look into hardware hacking existing solutions instead since heating water safely and effectively to boiling is not as easy as it seems. We ended up choosing to take an off the shelf water bottle heater and modify it with a custom PID loop and temperature sensors. This would allow us to heat it to a desired temperature, rather than only boiling. Also, since the bottle was already insulated it would mean that we did not need to worry about heat or water leakage onto nearby electronics.

B. Custom 2-axis Gantry vs. 3D printer

During our ideation phase we looked into creating a simple 2-axis system that would allow us to control the direction of water flow. One of the pros of this design is that we would be able to spec it to the exact size that we needed, thus making the system much more efficient. However, when we looked into pricing for all of the individual components it ended up costing us a similar amount of money to an off the shelf 3D printer system, but with much less functionality than a 3D printer. We ultimately decided that our goal was to make good coffee, not design a 2-axis system from scratch. Thus, we ended up choosing the 3D printer as it better suited our needs and also gave us another axis of freedom to play with as opposed to the 2-axis system we initially thought of.

VI. SYSTEM IMPLEMENTATION

This section will provide a more in-depth breakdown of how the Pour-over-and-over machine is going to be implemented. It will be broken down into three main sections – software, hardware, and electronics – each with a detailed breakdown of their respective subsystems.

A. Software

Arduino

The Arduino will be running code for the HX711 breakout board, pulled from a publicly available GitHub repository [2]. This code is written in C++ and will allow the Arduino to process signals from the scale and send weight values to the RPI.

Raspberry Pi

The RPI will be running two separate pieces of software. Octoprint will be running in the background to handle the movement of the 3-axis system. In the foreground will be our web application which the user will interact with to choose their brew profiles and set parameters.

The web application, running on the Django framework, will allow users to interact with the machine. Through this web application, they will be able to set parameters such as water temperature and pour pattern, as well as change and create existing brew profiles. The data for these profiles will be stored on an SQLite database. The web application will also have a portion of its codebase dedicated to interpreting the signals received from the Arduino to allow us to read scale weight measurements while brewing coffee.

Octoprint is an open-source interface for connecting a computer to a 3D printer system. This will allow us to keep many of the existing systems of the 3D printer rather than modifying them to fit our needs. Specifically, our web application will send coordinates to Octoprint which will then convert these to GCode, a language readable by the 3D printer system. The GCode will then be sent over a USB connection and interpreted by the 3D printer.

B. *Hardware*

Gantry and frame

Rather than crafting a gantry from scratch, our team has opted to repurpose a 3-D printer for our gantry system. Specifically, we decided on the Creality Ender 3 (Figure 3). This has several advantages. Firstly, due to the popularity and availability of 3-D printers, this was a cost-effective solution (Under $200). This model comes mostly assembled as well, with the motors, pulleys, and gears pre-mounted to the frame. This avoids many integration and construction related issues. Additionally, the 3-D printer's gantry was built to perform motions and actions that we needed our gantry to perform as well. By simply removing the nozzle and tubing from the 3-D printer's gantry, we can quickly integrate the tubing and pumps necessary for our pour-over machine. We will remove the fan assembly and nozzle from the 3-D printer's extruder head, and replace it with a custom, 3-D printed frame on which the water pump will be mounted onto. As noted previously, the Ender 3 will serve as the base on which all the other components will be mounted on. To attach these components, including the water heater, PCB, scale, and Rpi, custom mounts will be 3-D printed.

Fig. 3. Creality Ender 3 3-D printer: *https://www.creality.com/products/ender-3-3d-printer*

Water Heating and Distribution

The water will be stored and heated in a 300mL electric kettle attached to the frame of the 3-D printer. It will be mounted in such a way that the kettle will be easily accessible for the user, so that they can easily detach and fill the kettle. The lid of the kettle will be modified so that a temperature probe can be mounted to it. This will allow the system to constantly read and evaluate the temperature of the water as the kettle heats it. This will allow the user to select an exact temperature (within a reasonable range) for which they wish their water to be dispensed at. The dispensing will be accomplished through a food-safe water pump attached to the extruder head of the 3-D printer through a custom 3-D printed mount. It will replace the fan system that is normally mounted there. The tubing will basically replace the filament tubing (it will be larger, so larger zip ties will be needed to attach it to the wire bundle). The original filament nozzle will be removed and replaced by the water tubing. As of now, there are no plans to add a nozzle to the end of the tubing, as it assumed that the tubing is small enough to dispense the water in a precise manner. However, if this is not possible, a 3-D printed nozzle will be created and attached to the end of the tubing.

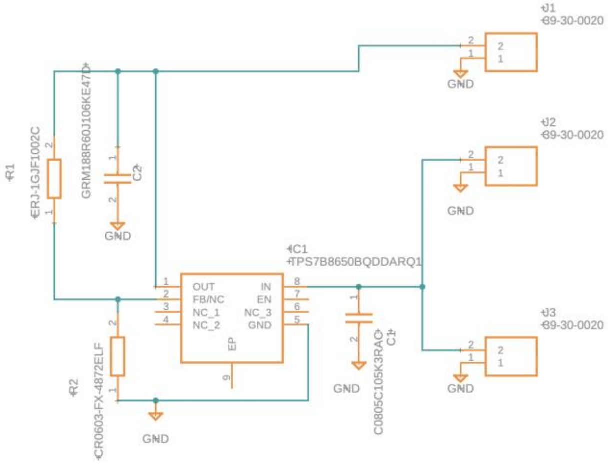
Scale

The scale will be utilized to allow for precise measurement of the ground beans in the filter and the total amount of water dispensed from the machine after the brewing process is complete. It will be mounted on the 3-D printer's hotplate. As mentioned previously, the heating function of the hotplate will be disabled to prevent damage to the scale. The scale, as ordered, is not integrable into our system. To do so, we will need to add some components so that the Rpi can take readings from the scale. Firstly, the analog signal from the scale will need to be amplified and converted to a digital signal. This will be done through the HX711 AD/C converter and amplifier. This signal will be routed to the Arduino R3. Finally, through a USB connection, this data will be sent to the Rpi, where the data can now be utilized by the system [3].

C. *Electronics*

This section of the project will be heavily focused on power electronics design, with the goal of having the cleanest possible packaging, from the wall outlet, all the way to each of our machine's components. To achieve this, a central Power Management Board will be designed to do all power conversions needed in the gantry. Our gantry has a 25V DC power supply which we will use as a power source for most electronics.

The first main component of the Power Management Board will be a LDO step-down circuit design to step the 25VDC from the gantry's supply down to 6VDC to power the water pump. We are using a TPS7B86-Q1 LDO (Figure 4), which is automotive grade, and it is rated for 500mA, which is higher than the pump which has a 300mA input rating.



1. Fig. 4. Circuit schematic for LDO 25VDC -> 6VDC Step-Down in Power Management Board

Lastly, we need to solve for controlling the AC current delivered to the kettle based on the PID control system. This will be done using a triac circuit designed to regulate the current delivered to the kettle's heating element based on the thermocouple's readings and the PID control algorithm.

VII. TEST, VERIFICATION AND VALIDATION

A. *Tests for Water Tank Capacity*

This will be a simple test, but there are some important considerations in its design. We will measure 300mL of water with an external measuring cup, and we will pour the contents into the kettle. Then, after confirming the liquids can be held by the kettle, we need to confirm that no water escaped the kettle due to the bubbles. We will turn the kettle up to the max expected temperature (100C) and let the kettle boil the water for 3 minutes. Then, we will turn the kettle off, and remove it from the surface. We will then place a sheet of paper over the surface where the kettle was and confirm that the piece of paper does not have any wet spots.

B. *Tests for PID Temperature Control*

To conduct the experiment and test our PID loop system, we will insert a thermocouple into the kettle with water. We will set a desired temperature in the system, and let the water reach the desired temperature. After the machine confirms the water has reached steady state and goes into "ready mode", we will start measuring the water temperature for 4 minutes (which is longer than the expected pour time of 3:30 minutes). We will then confirm that the temperature doesn't go more than 5F off of the set temperature. We will repeat this test at 90C, 95C, and 100C. This test will confirm that our design requirement has been met by directly measuring the water temperature in the kettle. Moreover, it will translate into giving users the confidence that their parameter selection is accurate, and that they will get the precise amount of extraction, which is highly dependent on water temperature, out of their pour.

C. *Tests for Total Water Flow Margin*

To test total water flow, we will select a desired pour volume and let the machine pour into an empty cup (since this requirement refers to water amount before it's absorbed by grounds). We will then measure the total mass poured with a kitchen scale and validate the desired margins. We will repeat this test with 180g, 200g, and 220g of water, which are around the expected uses for the machine. We know from our studies that our users value the richness of the coffee, and this is what motivated us to have a requirement for total water flow. This test will help us confirm the consistency of our pouring volume, and therefore, the reliability of parameter selection for the user.

D. *Tests for Water Flow Rate Control Range*

This will be a similar test to that of water volume, but it has an added time component. We will select a constant flow rate and turn the pump on for 5 seconds. We will then measure the water poured with a scale. The total water poured divided by 5s will give us the flow rate. We will then compare this rate to our design specification to confirm its validity. In the end, this will help us confirm that our parameter selection for flow rate is accurate, and thus give the users the confidence that they will reliably get the desired extraction rate out of their grounds, since this is so affected by the turbulence controlled by water flow rate.

E. *Tests for Default-Loaded Presets*

To test the accessibility and ease-of-use of our device, we will conduct a test where we have 5 random people try to use the machine for the first time. They will be tasked with selecting one preset of choice from the list and start a brew. To quantify the results: once the UI is finished, we will try to select the presets ourselves and come up with a reasonable amount of time to find a preset and start the machine. We will then benchmark the external users against that target time. The test will pass if all 5 users can start a pour within the set amount of time. This will confirm that our product is accessible and easy to use, and has seamless user experience, as per our use case requirement.

F. *Tests for Total Dissolved Solids Control*

This test will consist of using a refractometer to test for total dissolved solids (TDS) in our final product. We will create 5 samples of coffee with identical parameters and then test for TDS with our refractometer and compare the results against our design requirement of +/- 5% error. This test will be crucial in evaluating our final product – Flavor of coffee can be really hard to objectively evaluate, but by measuring the total dissolved solids, we will be able to confirm that our use-case requirement of making repeatable pours has been met.

G. *Tests for Accurate Patterns*

To test that our gantry can produce accurate pouring patterns, we will conduct a simple test. We will attach a pen to the pouring head of the machine, and then we will have it produce the same pattern 5 consecutive times on the same sheet of paper. Then, we will visually evaluate the patterns produced with a ruler, to make sure that no drawing is deviated by more than 500mm from the original path. To This simple –yet very obvious– parameter of being able to produce accurate patterns is something we need to ensure in our design. After all, this is what differentiates pour-over coffee from many other brewing alternatives. By confirming that the patterns are repeatable, we are closing the set of variables that can greatly impact the final product and ensure that our design produces consistent cups of coffee for users to enjoy.

PROJECT MANAGEMENT

A. *Schedule*

The schedule for our project can be found in Table II in the appendix. It includes the major project milestones, like project deliverables, presentations, and reports. It also includes any buffer time that we expect to need throughout the project, and some buffer in the end for any unforeseen circumstances.

B. *Team Member Responsibilities*

Overall for our project, the task division is as follows: Rio will be in charge of software development, which includes creating a user interface for creating and selecting presets, enabling saving presets into the machine's storage, and communication between the RPI to the gantry. Elijah will be responsible for creating the gantry, both in physical form and in algorithm creation. This includes nozzle movement, dispensation, and heating, as well as all necessary sensors like weight and temperature. Lastly, Corrado will be working on custom electronics, which will be heavily focused on power electronics design – this will include a custom power management PCB, and any sensor data collection and processing.

C. *Bill of Materials and Budget*

Refer to Table I in the appendix of this report to locate the bill of materials and budget specifics. Currently, there is over $100 of the budget remaining.

D. *Risk Mitigation Plans*

One risk we have not tested yet is the amount of variability we have in the flow rate of the pump we are using. We have done some basic testing which showed us that flow rate does vary with varying current, but no concrete numbers were collected. In the event of the pump being unable to achieve our desired flow rate we can employ the use of two pumps in tandem. This will allow us to have even more control of the flow rate as well as double our max flow rate. To achieve this we will design two separate pump mounts for the gantry, making it so that adding another pump will be easy.

We are also accounting for a large amount of risk by ordering food-safe filament to make or replace parts as we see fit.

Another way we managed risk was by ordering multiple components that we considered to be fragile or that may fail easily. Additionally, we did not fully spend our budget, allowing for additional parts to be ordered if need be. We also ordered most of the components through Amazon Prime, giving us a quick turnaround time if future parts are needed in a timely manner.

VIII. RELATED WORK

Some other related work to our idea is the Xbloom and the Poursteady. The Xbloom looks to serve a group of people who already use pod-based machines such as Keurig or Nespresso but allow them to experience and experiment with specialty coffee and pour-over. The issue with this is there is no specificity in choosing brew parameters, as you can only change them based on an arbitrary visual scale. The expense of the machine as well as the pods proves to be unsustainable in the long term. On top of this, the added material waste of buying pods as well as the packaging for them proves to be much larger than just making pour-over coffee the traditional way. Poursteady looks to automate pour-over coffee for a cafe setting. This machine costs a whopping $11,000 at its cheapest, making it completely impossible to buy for an average consumer. While it can be useful for cafes it doesn't allow for customization of pour parameters at all. There is only one button on the machine which is to start the brew process.

IX. SUMMARY

Our open-source pour-over coffee machine aims to bring specialty coffee to any user's home. By automating the entire process, we are opening the door to consistent, high-quality pour-over coffee to experts, enthusiasts, and beginners alike. Our key success metric is repeatability, which seems to be our biggest differentiator from traditional techniques of pour-over – but this comes at the cost of challenging design criteria: having our machine be precise in reproducing all the parameters a user selects. Our team aims to overcome these challenges and produce a cup of pour-over coffee that our users can enjoy, over-and-over again.

GLOSSARY OF ACRONYMS

BOM - Bill of Materials
MQTT - Message Queuing Telemetry Transport
OBD - On-Board Diagnostics
PCB – Printed Circuit Board
RPi - Raspberry Pi
TDS - Total Dissolved Solids

REFERENCES

[1] KitchenAid “COFFEE MAKER MAKING LESS COFFEE THAN SELECTED”, 02 Mar 2024. https://producthelp.kitchenaid.com/@api/deki/pages/2512/pdf/Coffee%2bMaker%2bMaking%2bLess%2bCoffee%2bthan%2bSelected.pdf

[2] BOGDE on Github “An Arduino library to interface the Avia Semiconductor HX711 24-Bit Analog-to-Digital Converter (ADC) for reading load cells / weight scales.” Web, 17 Mar 2022 https://github.com/bogde/HX711

[3] Scalafiotti, Edo. “Build a Scientific Scale and Process Its Live Data Using Arduino, RaspBerryPi and Socket.io.” Medium, 15 July 2018, medium.com/@edoardo849/build-a-scientific-scale-and-process-its-live-data-using-arduino-raspberrypi-and-socket-io-13a5671011d9.

Appendix

TABLE I. BILL OF MATERIALS

Item Name	**Purpose**	**Manufacturer**	**Quantity**	**Cost @**	**Total**
Raspberry Pi	Control	Raspberry Pi Foundation	1	Free	$0
3-D printer	Gantry	Creality	1	$168	$168
Water heater	Water heating	DREAMOSA	1	$20	$20
Water pump	Water transport	Lightobject	2	$14	$28
10mm Silicon tubing	Water transport	Quickun	1	$10	$10
12mm Silicon tubing	Water transport	ANPTGHT	1	$17	$17
Temperature probe	Sensing	BOJACK	2	$9	$18
Refractometer	Verification	Xindacheng	1	$20	$20
HTPLA filament	Printing custom parts	Protopasta	1	$30	$30
Custom PCB	Power management	*Currently unknown*	1	$70 (est.)	$70
AC power strip	Power management	4 LEAF	1	$15	$15
Scale	Weight measurement	BAGAIL	1	$15	$15
AD/C Converter	Weight measurement	Arduino	2	$18	$36
Arduino R3	Weight measurement	ELEGOO	1	$17	$17
				Grand Total	**$464.00**

TABLE II. TEAM SCHEDULE

Group A1 - Pour-Over-and-Over		Feb	Mar					Apr		
		Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11
		19 20 21 22 23 24 25	26 27 28 29 1 2 3	4 5 6 7 8 9 10	11 12 13 14 15 16 17	18 19 20 21 22 23 24	25 26 27 28 29 30 31	1 2 3 4 5 6 7	8 9 10 11 12 13 14	15 16 17 18 19 20 21
		Design Presentation	Design Document					Interim Demo		
Software										
Preset Sequence Backend	Rio	🟧								
Communication Protocol	Rio		🟧							
Enable Saving Presets	Rio			🟧						
Front-End Pattern Creator	Rio				🟧					
Integrate Front End + Back End	Rio					🟧				
Gantry										
2-axis Nozzle Control Setup	Elijah	🟩								
Stepper Motor Control	Elijah		🟩							
Heat + Dispense										
Build Heating Element	Corrado	🟨								
Assemble Pump system	Corrado				🟨					
Integrate to Gantry	Everyone					🟦				
Sensing + Data Collection										
PID Loop Algorithm	Rio				🟧					
In-Unit Scale	Elijah			🟩						
Sensors	Elijah				🟩					
Electronics Design										
Prototyping Parts Lead Time	Corrado	🟨								
Breadboard Prototyping	Corrado		🟨							
Custom PCB Design	Corrado			🟨						
Custom PCB Lead Time	Corrado				🟨					
Custom PCB Testing	Corrado					🟨				
Integration										
System Debugging & Testing	Everyone						🟦	🟦		