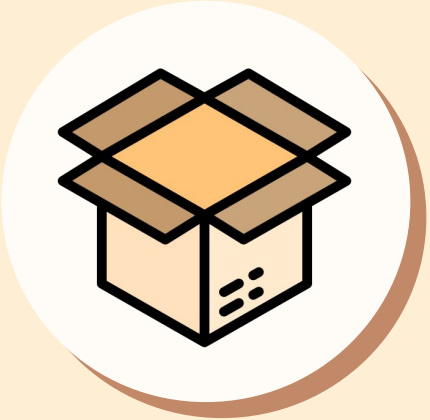
Pour Over-And-Over

Final Presentation

Team A1 | Rio Pacheco, Elijah Knupp, Corrado Govea

Solution Approach

Use Case: Programmable pour-over coffee machine



Open Source

Open loop philosophy to make product **customizable**

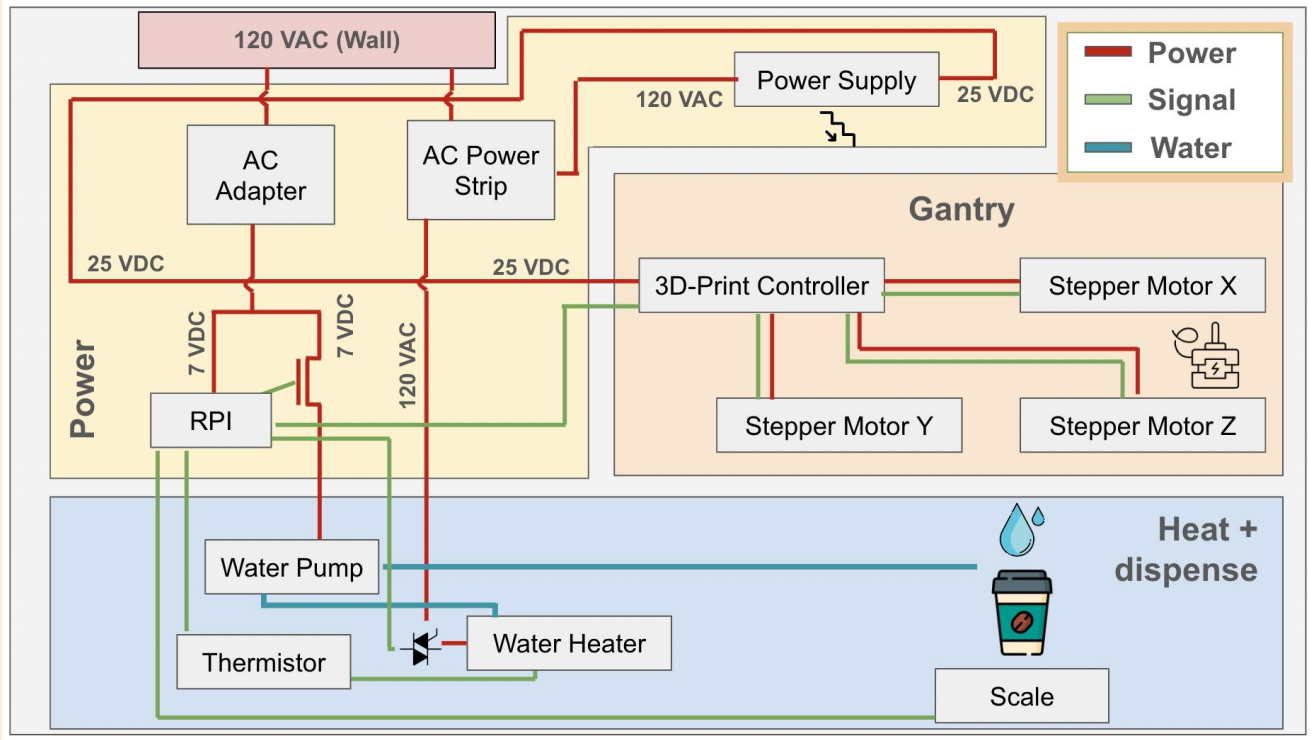
Easy to Use

“Plug-and-play” before modifications to make product accessible

Safe

Food-grade materials, touch-safe surfaces to ensure **public health**

System Specification



Key Takeaways:

- Added power MOSFET
- Removed LDO circuit from board.
- Added 7V supply for pump + MOSFET.

Implementation Plan | Changes

Removed octopi as communication medium

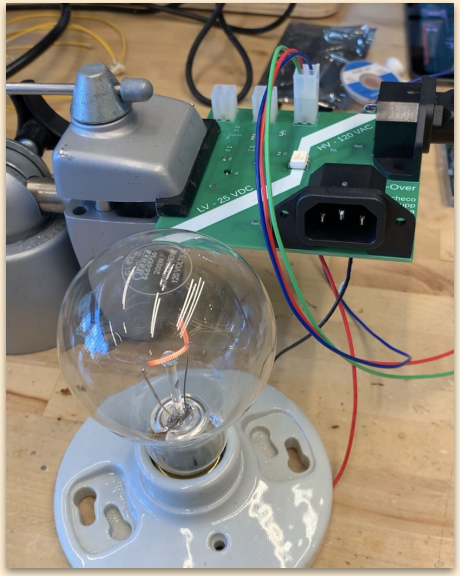
- Sending gcode over serial using **pyserial** proved to be easier and more effective

Stronger Pump

- Located on top of gantry.
- Allowed us to add a flow rate restrictor on the gantry.
- Added **power MOSFET**

Digital scale data only used for weighing coffee

- Collecting data over serial was found to be too unstable to rely on for pouring water



Solution Requirements

Goals: precise, controllable, and repeatable

	Use-Case Requirements	Design Requirements
Heat	Heat water up to **desired temperature**	**∓5℉ margin** from set temperature
	Hold enough water to allow to **brew one cup** of coffee	**300ml water tank capacity**
Dispense	Pour a **precise amount of water** over the coffee grounds	**∓5 - 10 grams**
	Have a consistent, highly **controlled flow rate** throughout the entire pour	**0 g/s - 12 g/s**
UI	Provide an intuitive, straightforward, and **accessible** experience	>= 5 **preloaded presets**
Result	Deliver a **repeatable**, highly **consistent** cup of coffee between pours	**∓0.5% TDS***

*TDS: Total dissolved solids

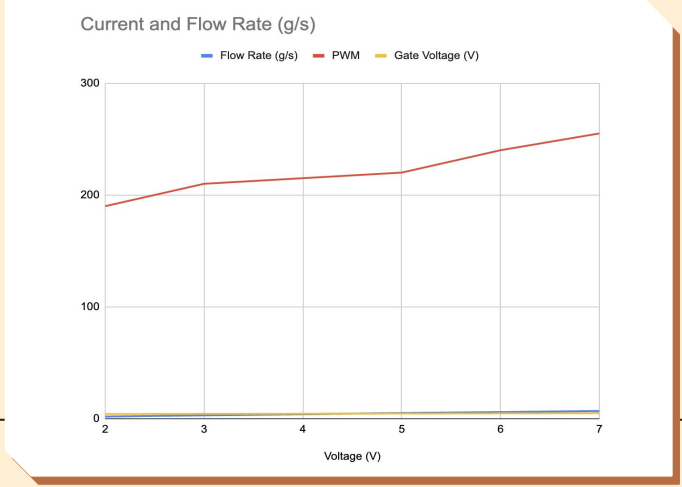
Flow Rate

Test result: Pass / Fail

Goal: 2 - 7 g/s (Updated from original design 0-12 g/s)

Method: Digitally controlled **PWM signal** from the Arduino to the gate of a MOSFET that, when “on", completes the pumps circuit

Voltage Applied to MOSFET (V)	Measured **Flow Rate**
7 (MAX)	255
6	240
5	220
3	210
2 (MIN)	190



Heating

Test result: **Pass** / Fail

Goal: 180 - 212℉ range, with ± 5℉ accuracy

Method: PID loop to send an “on/off” signal to the arduino, which switches the **water heater on and off** until the **desired temperature is reached**.

Temperature **Target** (℉)	Temperature **Min** (℉)	Temperature **Max** (℉)
180	177	183
195	190	200
205	203	207
212	212	212

User Experience

Test result: Pass / Fail

Goal: Average rating of **4 or above** (on a scale from 1 to 5) on every question asked based on their experience using the app

Method: Have **10 users** take a survey after using the web application.

Question Asked	Average Result
How useful were the features (tare, profile creation, sorting)?	**4.8**
Was it easy to navigate?	**5**
How satisfied are you with the functionality?	**4.3**
Do you see yourself using this at home?	**4**

Key Takeaways:

- Major **UI-related** metrics exceeded.
- Focus on making product **appealing**.

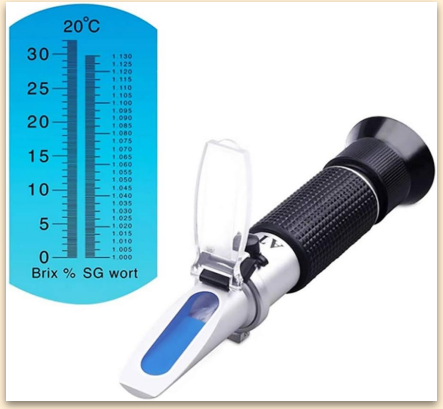
Coffee TDS

Test result: Pending/ Fail

Goal: The TDS of coffee brewed by the machine with the **same profile** should be within 0.5%

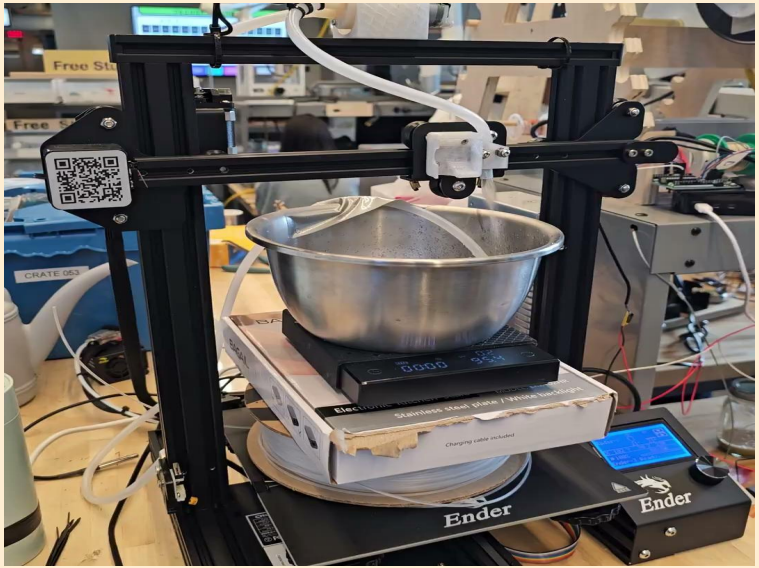
Method: Brew 3 cups of coffee using the same profile and compare TDS % between brews

Blocker: We are currently waiting for the coffee beans to arrive so we can record test data.



Testing: Next Steps

Current progress!



Fine tuning

- Further refine PID loop values to reach target temperatures more accurately

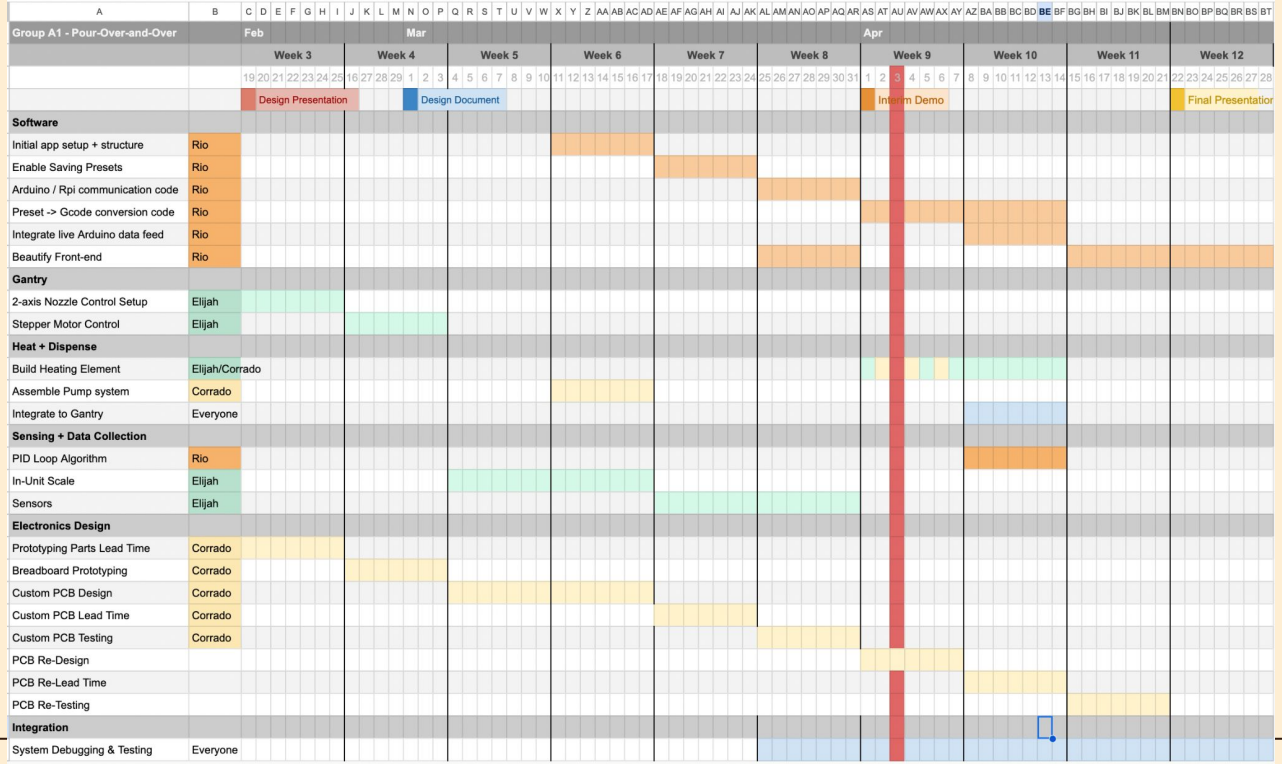
Brewing effectiveness

- Ensure that all coffee grounds are saturated throughout the brewing process

Stability

- Stress test the software interface for any further uncaught bugs

Project Management



Next Steps Distribution:

- **Calibration**: Rio
- **Integration**: Elijah
- **Corrado**: Testing

Project Management - Lessons Learned

- **Test, Test, Test… Early!!**
  - We quickly found out how difficult it was to test a system as a whole
    - Breaking the system into its subsystems was essential for effective testing

- **Read the datasheets**
  - *Like, really read the datasheets*
  - *This is how we solved our pump circuit (after 12+ hours of troubleshooting…)*

- **Order multiples of everything**