ZJU-UIUC Institute

ECE 445 / ME470

SENIOR DESIGN LABORATORY

Final Report V3.0

Intelligent Pour-Over Coffee Machine

Team #15

Jie Wang
jiew5@illinois.edu

Xubin Qiu
xubinq2@illinois.edu

Jingyuan Huang
jh88@illinois.edu

Rucheng Ke
rke3@illinois.edu

Supervisor: Prof. Said Mikki

TA: Dr. Guo Hao THNG

Intelligent Coffee Team

May 28, 2024

Contents

Abstract

This report presents the implementation of an **Intelligent Pour-Over Coffee Machine**, a senior design project aimed at automating the coffee brewing process to replicate professional barista techniques. The system, controlled by a Raspberry Pi 5B, integrates 7 subsystems including *brewing, control, sensor, power, pumping, heating, and user interface*. Besides closed-loop control over the coffee brewing, the machine also involves factors such as temperature, pH, and weight sensors to ensure the coffee's quality and consistency. With fully implemented unit tests and whole-system verification, our system can produce tasty, drinkable coffee with varying settings. The project highlights the potential for further advances in the specialty coffee market. As a senior design project by PhiloCoffee Club, it shows a blend of truth-seeking innovation and learning by labor. The implementations are open-sourced here.

Keywords: pour-over coffee, automated coffee brewing, coffee machine, sensory experience, specialty coffee market, PhiloCoffee Club

PhiloCoffee Club **Official Website**

Figure 1: All rights reserved by PhiloCoffee Club ©. QR code linking to the official website.

Acknowledgement

We would like to express our heartfelt gratitude to our senior design advisor, Prof. Said Mikki, Prof. Mark Butala, Prof. Timothy Lee, and our dedicated TA, Dr. Guo-Hao Thng.

We also express our gratitude to Maxpresso@Suzhou for sponsoring us with his commercial coffee machine for our study. Thanks Mr. Cup from GIFCO@Shenzhen, Intro Coffee@Shenzhen, ZJU Coffee Club@Hangzhou, and PhiloCoffee Club@Haining for providing user interviews on our design.

Further, we want to acknowledge everyone who helped with our project: Zhenzuo Si for valuable embedded experience on ***ECE445@2023: Tea Blend Distributor***[1]; from tea to coffee, our similar senior designs are upgrading each year!

Thanks to Zicheng Ma for his brewing recipe and appliance support, and to Zhuohao Li and Jiajun Hu for moving our device before the DEMO. Thanks also to Prof. Lap Chee Tsui, Tiantong Qiao, Haoxuan Du, Jiaheng Wen, Yunhan Zhong, Zhuzi Xu, and Hanggang Zhu for testing our coffee.

MAXPRESSO®
COFFEE ROASTERY

Figure 2: Maxpresso Coffee Roastery@Suzhou, Our Off-campus Sponsor

1 Introduction

1.1 Background

Dating back to the early 20th century, the art of pour-over coffee has evolved from Melitta Bentz's simple paper filter to a globally handy craft tide [2]. Although this brewing method is highly praised for the complex flavor and creative experience, it is hard to maintain consistent quality for the public.

Modern consumers demand coffee that is not only good, but also convenient to drink [3]. This project proposes an **intelligent pour-over coffee machine** that produces coffee automatically. Our design aims to blend the hand-made art of pour-over with the precision of automation, contributing to the dynamic and expanding coffee machine market [4].

1.2 Problem and Solution Overview

1.2.1 Problem

The art of pour-over coffee brewing, famous for its complex flavor and high quality, is heavily dependent on the skills and experience of a barista. This craftsmanship leads to variability in coffee quality due to human element. Additionally, it is challenging for **common coffee enthusiasts** to replicate professional barista techniques *at home or in non-specialized settings,* particularly in areas where specialty coffee culture is less developed.

1.2.2 Solution

Imagine a coffee machine that automates the process of pouring water. It can customize each cup according to **the type of coffee bean and the desired flavor**. With the bean grounded and filter in place, the user can start the process with the press of a simple button, after which the machine dynamically adjusts its operations to create a delightful cup of coffee.

This machine should deliver sensory pleasure and the similar taste of hand-poured coffee, while saving time and effort. This machine is designed to mimic the skills of the master and conveniently deliver high quality in-place coffee. Fig 3 shows the mechanical design in SolidWorks. Using laser cutting, we constructed a acrylic plate-based transparent framework with 3D printed fabrication components. From the vision of transparent demonstration and educational openness, our system demonstrates a novel pour-over mechanism, different from traditional drip or espresso coffee machine.

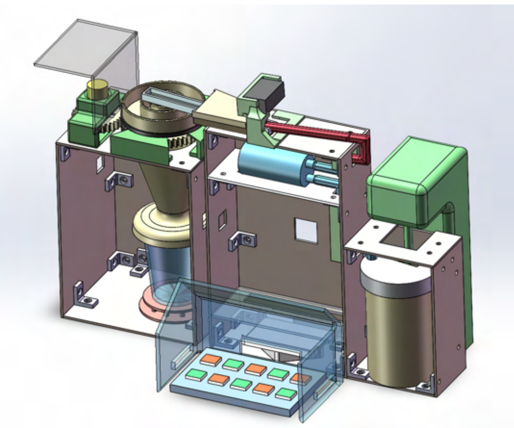

Figure 3: Physical Design in SolidWorks, Version 4.

Figure 4: Assembled Coffee Machine Ready for Demo, we prepared ground coffee in filter paper.

1.3 High-level Requirements List

- **Precision and Consistency:** The machine should replicate the scientific pour-over skill . It should control water temperature, water-bean ratio and time spent for making coffee.
- **Affordability and Accessibility:** The product should offer a more cost-effective solution than existing commercial product without sacrificing flexibility and quality.
- **Easy User Experience:** The design should be intuitive, easy to use for the specialty coffee beginner.
- **Durability and Maintenance:** The machine should be built to last, requiring minimal maintenance while operating efficiently.
- **Quality of Brew:** The coffee produced must be consistently high in quality, with taste tests confirming its superiority or equivalence to manually brewed pour-over coffee.
- **Coffee Roast Degree Difference**: The machine should be able to distinguish light roast and dark roast coffee, taking different brewing strategy on them.

2 Design

Our design has undergone three major revisions for the sake of stability and feasibility. Compared to the initial bold idea in our proposal [5], the final design document represents a much more scientifically grounded approach [6]. Considering the project period is only three months along with other coursework, the final product should be clean, approachable, and easy to build.

2.1 Design Formalization and Development

Our design was formalized through the following actions:

1. **User Study:** We conducted user studies at various coffee shops and students' club, including Maxpresso@Suzhou [7], GIF Coffee@Shenzhen [8], ZJU Coffee Club@Hangzhou [9] and PhiloCoffee Club@Haining[10] . We interviewed potential users to record their expectations and requirements for a pour-over coffee machine in both home and commercial scenarios.
2. **Literature Review:** We reviewed related patents and papers [11, 12, 13, 14]. This helped us compare similar ideas, designs, and thought processes, allowing us to develop a novel approach that enhances existing scientific coffee brewing methods. Sizhen Zuo et al.'s senior design on Tea Machine last year also helps our design a lot. We learned the optimal choice to complete the project on constricted time[1]. Our goal is to apply for a utility model patent in China upon completion of this senior design.
3. **Market Investigation:** We investigated similar product designs by watching online videos and reviewing their official websites[15, 16, 17]. Notable products that provided valuable insights include xBloom, Chemex, Bonavita, Cuisinart, and Gemilai [18, 19, 20, 19]. Sponsored by Maxpresso@Suzhou, we applied reverse engineering on the ***Gemilai CRM 4106*** [21], which enabled us to gain a deeper understanding of the design and functionality required for our machine.

After careful consideration of the project's high-level requirements and constraints, we decided on a system architecture that balances performance, scalability, and ease of implementation optimally.

2.2 System Architecture

Figure 5 shows the system block diagram of our design. Our coffee machine consists of seven subsystems, all of them are indispensable for the pour over task.

Notice that the most important contribution of our coffee machine is the brewing system. Due to time limit and food safety consideration, we simplify the sensor system, making the temperature sensor and pH sensor to be Open-Loop Controlled. While these data can still be recorded by manual detection. During running, all the system IO information is recorded in control system's logger files, providing future inspection and improvement.

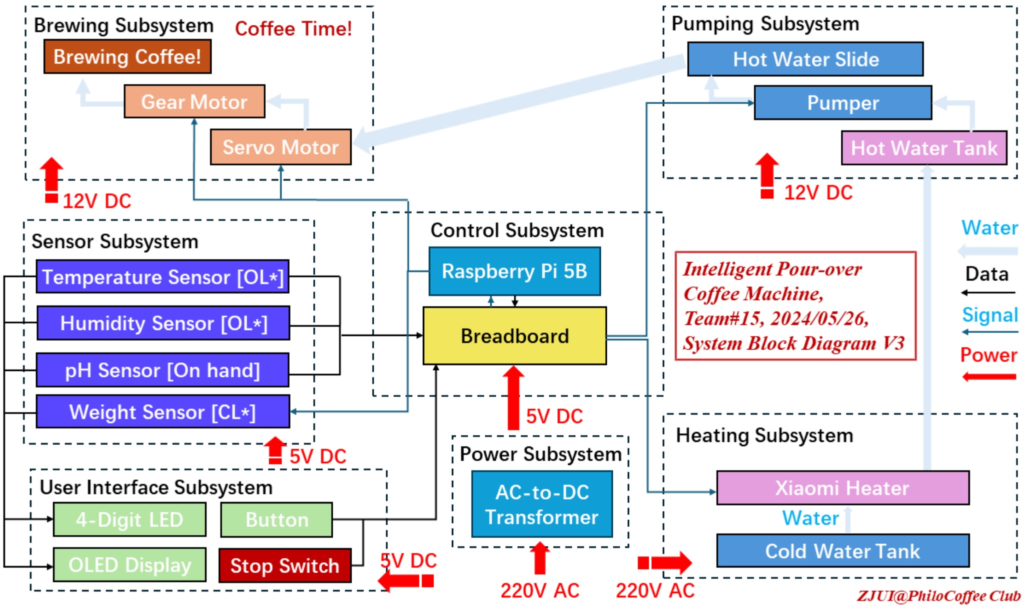

Figure 5: System architecture of our design: Power supplies electricity, control supervises, sensors provide CL feedback, heating produces hot water, pumping transports water, UI allows interaction, brewing simulates a barista (OL: Open-Loop, CL: Closed-Loop).

2.3 Mechanical Framework Design

In this project, we utilized a series of professional tools and software for mechanical design. First, we used SolidWorks 2022 Educational Edition for 3D modeling and design, which was provided by the Illinois Webstore. Through SolidWorks, we were able to create precise 3D models and conduct various simulations and analyses to ensure the feasibility and optimized performance of the design. To manufacture the actual mechanical components, we used the TOUCHFAST laser cutter from the ZJUI Innovation Lab. This fully automated CNC single-platform fiber laser cutter efficiently cut acrylic sheet materials, which formed the transparent framework of our product. Additionally, we used the Creativity Ender3 V2 3D printer to print some complex support structures, such as motor mounts and waterproof designs, facilitating further testing and improvements.

As Figure 6 illustrates, the following sections provide a detailed overview of each subsystems, highlighting their functionality within the overall design.

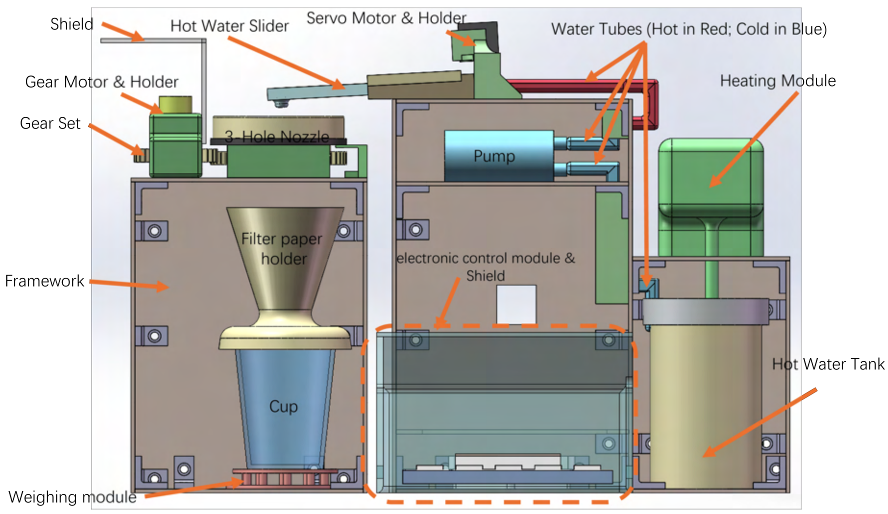

Figure 6: Systematic Decomposition with Specific Hardware Names

2.4 Brewing System

The brewing system is the main innovation in our project and also the most complex one. We met these challenges and gave our solution on how to pour over intelligently.

1. **How to Brew in Rotation with Directed Water Flow?** Our solution is a combination of a hollow ring gear, a three-hole nozzle, a hot water slider, a gear motor and a servo motor. Comparing existing products on the market, we found this structure to be most efficient and scientific to do the pour-over job. A 12V DC 370 gear motor [22] drives and controls the rotational speed and direction. Different rotation circle are switched by a 5V PWM SG5010 Servo Motor[23].

   After the hot water tank is filled with enough water, the pump transfers the water into the hot water slider. The hot water slider then delivers water to three different holes on the nozzle according to the coffee recipe, simulating different brewing movement, as illustrated in Fig 10 14 12.

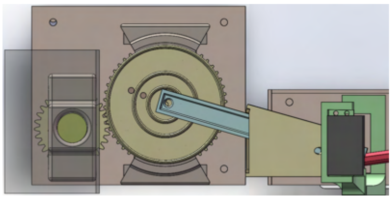

Figure 7: Pour over water at center, 30 degree

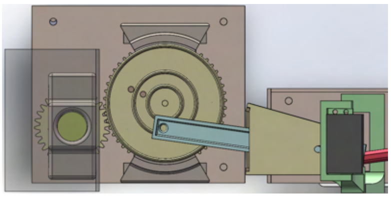

Figure 8: Pour over water at middle, 15 degree

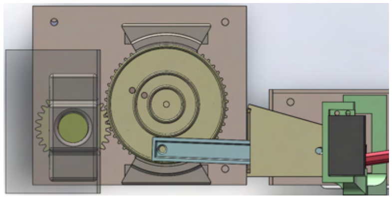

Figure 9: Pour over water at outer, 0 degree

   The servo motor realized pour-over circle decision. With pre-tested *'pwm_value-to-angle' dictionary,* It has three discrete angle settings, designed based on previous user studies.[6]. These angles ensure the hot water is delivered accurately to the three different holes of the brewing nozzle, as shown below.

Figure 10: Machine pour over water at center in practice.

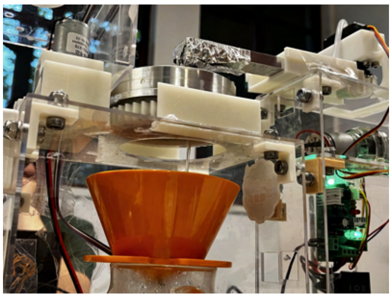

Figure 11: Machine pour over water at middle in practice.

Figure 12: Machine pour over water at outer in practice.

2. **How to Choose and Position Motors?**

   - **Motor Selection**: The system is stable only if it is operates on suitable motors. We tested 5V DC130 Toy Motor[24], 6V TT Motor[25], 12V DC 370 gear motors at 30, 60, 120 RPM separately [22] and 24V 45kg DC Gear Motor[26]. Then we found under the friction force and torque condition of our brewing system, the 12V 370 gear motor provides most appropriate torque and stable operation. This is essential for a senior design using 3D printed material to maintain consistent rotational motion under varying loads.
   - **Gear Torque Analysis:** The torque τ produced by a DC motor is given by $\tau = K_T I$, where K_T is the motor's torque constant and I is the current.

     Consider a module with a 55-teeth spur gear and a 95-teeth ring gear in figure 13. Given gear-gear friction, moment of inertia, and viscous damping, we derive the following equation for the gear motor module:

$$T_{out} = I_{g_2}\ddot{\theta}_1 + I_{g_1}\ddot{\theta}_2 + (I_d + I_w)\ddot{\theta}_2 + Z_f + T_{f_2} + b_{v_1}\dot{\theta}_1 + b_{v_2}\dot{\theta}_2 \quad (1)$$

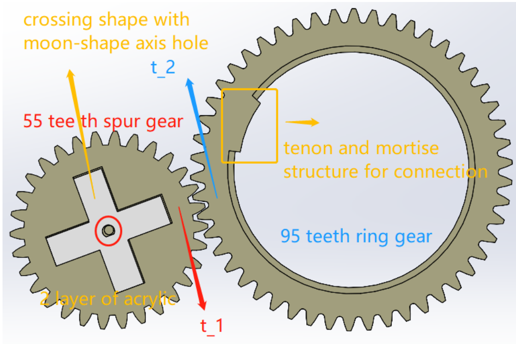

Figure 13: Gear Module with 55-teeth Spur Gear and 95-teeth Ring Gear

Figure 14: Gear Module Running In Practice, the 9kg torque force can deform the holder.

According to the right-hand rule, when the brewing nozzle rotates clockwise, it exerts a downward torque on the holder support, corresponding to an upward torque on the bracket as the motor rotates counterclockwise. Conversely, a counterclockwise rotation of the brewing nozzle exerts an upward torque on the holder support and a downward torque on the bracket.

Considering the brewing nozzle is glued to the ring holder, it is necessary to ensure the torque is directed upwards to stabilize the system. Therefore, the motor should rotate clockwise. By empirical methods, we get a helpful record for the mech design, where multiple trials indicates the optimal motor speed for our system is around 30% to 95%. The motor is controlled via the `Motor` class from the `gpiozero` library, as detailed in `hardware\motor.py`.

- **Motor Driver Selection:**By controlling the PWM current via L298N Motor driver[27], we can adjust the torque to ensure it is sufficient for rotating the brewing mechanism without stalling. This is because L298H Bridge has two Output, making it ideal for low voltage signal from Raspberry Pi 5B to the motors.

For switching different rotation circles, we selected a 5V PWM SG5010 Servo Motor [23]. The servo motor is ideal for its ability to provide precise angular positioning through pulse-width modulation (PWM). We ensure the servo motor can handle the varying torques by selecting one with a sufficient torque rating. This guarantees smooth and accurate transitions between different rotation circles.

The three-hole nozzle is connected with a ring gear whose teeth face outward. We place a sliding ring between the brewing nozzle and three C-shaped holders. This sliding ring maintains fixed contact with the holders and allows the nozzle to rotate freely, remaining in position. The ring gear is then driven by a spur gear, spinning the whole brewing head.

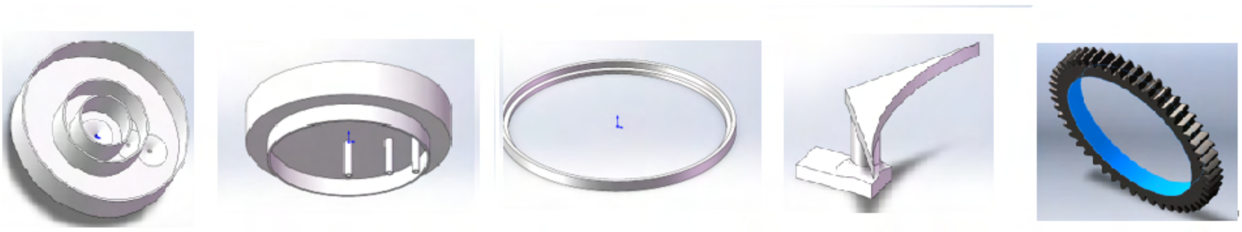

Figure 15: Brewing Nozzle (top & bottom view), Nozzle Holder, Holder Support, Spur Gear Ring

To ensure the stable rotation of the gear motor in our custom conditions, we designed a "cross-shaped" connection component to prevent the driving gear from slipping on the motor shaft, as shown in Fig 13. This non-circular, half-moon-shaped connection ensures reliable performance.

The hot water flows down the 3-degree off-angle hot water slider, enters a specific hole of the three-hole nozzle controlled by the servo motor, and then brews the coffee by passing through the coffee filter.

3. **How to Ensure Food-Grade Water Channel and Prevent Contamination by Left Water:** All components in contact with water or coffee are made from food-grade materials to prevent contamination. For instance, the hot water pipe is made from PP material, safe for hot water up to 130°C.

   The outlets of all holes, including those in the three-hole brewing nozzle and the end of the 3-degree inclined hot water slider, are non-ideal circular shapes with specially designed tips to disrupt the surface tension and prevent water from remaining.

   As shown in Figure 16, the hot water slider has a 3-degree inclination to promote natural downward water flow. The servo motor holder matches this angle to fit the slider's lean, allowing the hot water slider to deliver water into three different layers of the brewing nozzle, achieving distinct brewing radii, similar to actual pour-over techniques.

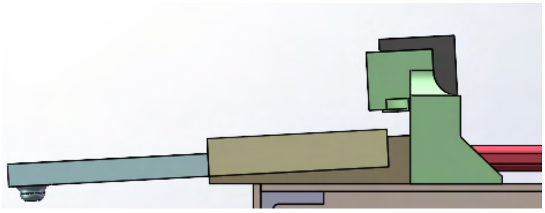

Figure 16: 3 degree inclination to self-clean left water

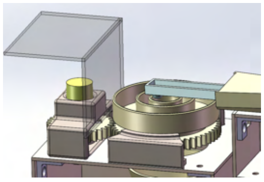

Figure 17: Gear Motor Shield for Preventing Water

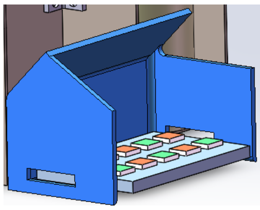

Figure 18: Shelter to prevent water spilling onto breadboard

4. **How to Prevent Water Spillage and Short-Circuiting?** Our design includes safeguards to prevent liquid spillage and overflow, which could cause electrical short circuits. We added a waterproof shelter for the gear motor (Figure 17). Properly sealed joints and controlled water flow mechanisms are also implemented to mitigate the risk of water spillage and short-circuiting, as shown in Figures 19, 16.

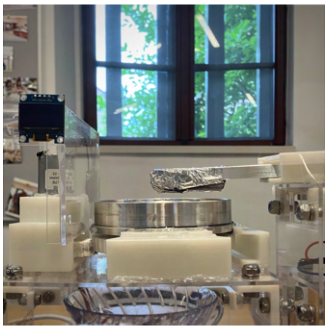

Figure 19: The hot water slider is wrapped with foil to prevent water overflow

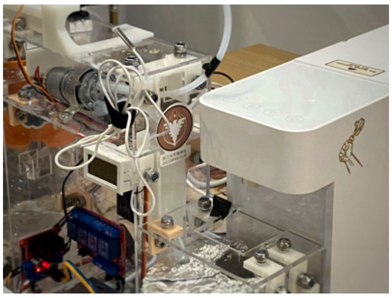

Figure 20: System design showing the arrangement for maintaining temperature and managing extra heat.

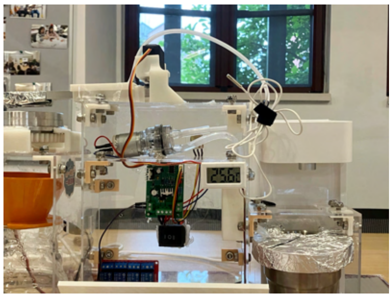

Figure 21: Water Channel is placed on top for after-use cooling

5. **How to Maintain Temperature and Manage Extra Heat?** To ensure the best temperature for brewing, we used data from previous user studies[6]. We maintain temperature via a two-layer stainless steel 304 hot water tank with an air-filled inter-layer for insulation, and a foil-wrapped lid to prevent heat evaporation (Figure 20).

   To manage extra heat, we positioned the hot water channel at the top of the machine (Figure 21). This setup avoids natural heat convection that could heat components unnecessarily. For more details, please refer to the Heating System section.

2.5 Control System

1. **Finite State Machine Design**

   Finite state machine (FSM) is a paradigm that abstracts the working mechanism of embedded systems-From laundry machines[28], cooking machines [29], robots [30], to vending machines [31], FSM is widely used as it is the simplest way to intelligently manage the programmable system. In this project, we describe our coffee machine with the FSM shown in Figure 22. It includes the states of setup, brewing, heating, and pumping, and sets the execution order according to different input commands.

   FSM Summary: The FSM describes the entire coffee brewing process. After setting the brewing method, the coffee machine prompts user to add certain volume 95 degree water into the hot water tank. Then, the pump raises the water up to the brewing system, evenly distributing hot water into the three-hole nozzle outlet.
   Depending on the chosen coffee type and style, the machine then activates a gear motor and the servo motor to perform a series of actions, such as a direct or rotating pouring water. If the weight sensor shows that the coffee concentration meets the set standard, or the expected time is up, then the brewing cycle ends. If not, the process is repeated for another cycle. A more detailed description is provided in subsection 2.3.

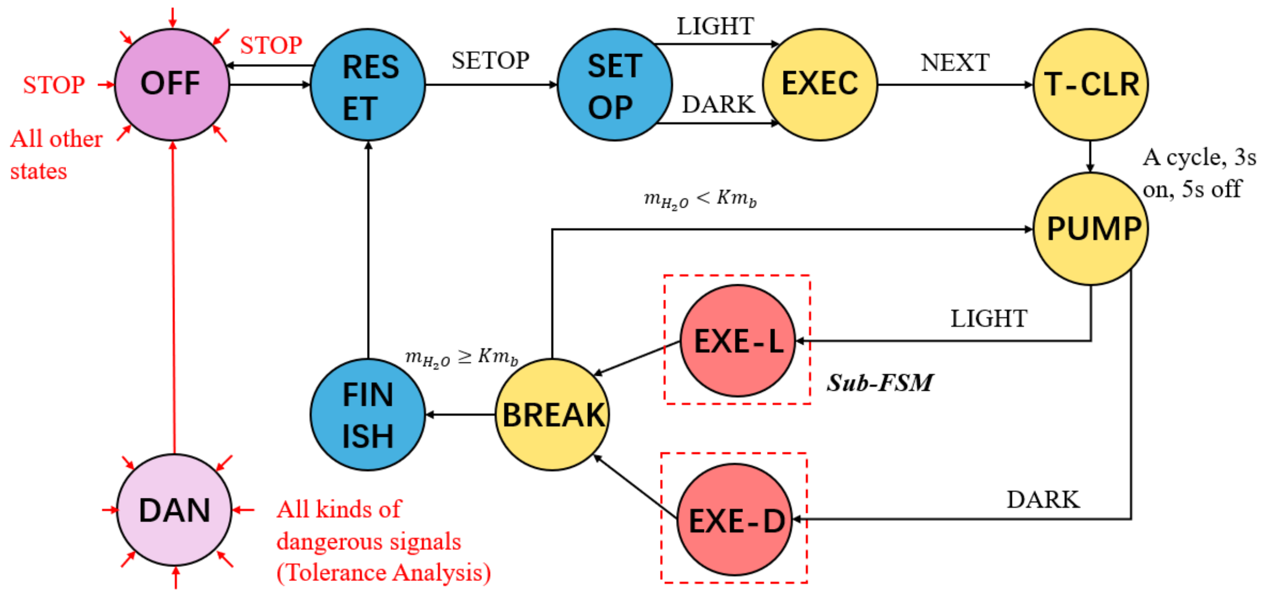

Figure 22: The finite state machine (FSM) diagram for the coffee machine. Each bubble stands for a state (what the coffee machine is doing). The arrows stand for transitions, which means the coffee machine starts to do another thing when a condition is met, or a user instruction is input. For the states' description, see Table 1. For sub-FSMs like EXE-L, EXE-D, see Figure 23 for details.

Typical Pour-Over Coffee Recipe:

Given the light-roast coffee bean, our FSM takes specific brewing procedure to make coffee. Here is a light roast recipe coming from *Zicheng Ma*, a professional pour-over coffee maker. For the details of different coffee recipe, please refer the Appendix B and 'README.md' on the Github.

(a) Prepare 250ml 95°C hot water and grind 15 grams bean with 32 fineness degree.

(b) Pre-wet the filter cup and set up the weight sensor with a tare.

(c) Start with a 20-30 gram cycle pour, ensuring all grounds are saturated.

(d) Rest the coffee grounds for 20 minutes.

(e) Continue with a slow, direct pour to 100 grams, then quickly pour around the edges for 20g.

(f) Finish the direct pour to reach a total weight of 225 grams without disturbing the flow.

(g) Wait the rest coffee liquid to drip through filter, and clean whole set with remaining hot water.

Table 1: Key Parameters for Brewing Specialty Coffee

Grind Size	Medium-fine grind
Coffee-to-Water Ratio	1:15
Water Temperature	Boiling (approximately 95°C)
Brewing Time	About 1 minute and 30 seconds
Raw Material	15 grams of fresh, light roast coffee beans, brewing into 225 grams of coffee

State Description

We use a Moore Machine with 15 distinct states for the FSM, described as follows:

OFF The coffee machine is not powered on.

RESET The coffee maker is inactive and awaiting initialization.

SET-OP Set up the operation. The machine prompts user to select flavour and mode.

EXEC Executes the brewing process.

T-CLR Timer clear. Resets the timer to zero, typically to record the duration of brewing.

PUMP Pumps water from the hot water reservoir to the three-hole nozzle.

FLOW Channels water from the hot reservoir to the brewing nozzle via the slider.

EXE-L/D Execute differnt coffee recipe. textitL/D corresponds to light or deep roast coffee modes.

BREAK Pauses operation after a brewing cycle to allow the weighing module to assess water sufficiency.

FINISH Completes the brewing process.

DAN Dangerous condition. the machine halts to ensure safety when an abnormal condition occurs.

2. **Spatial-Temporal Diagram** Figure 23 presents a space-time chart illustrating the sequence of steps in the coffee machine's operation.

   Time points t_0 to t_7 denote the progression of stages, showing how the motor and pumper work concurrently. The pumper operates intermittently, and the process halts once the water level is sufficient. Further details are provided in the Appendix: **User Journey Evaluation**.

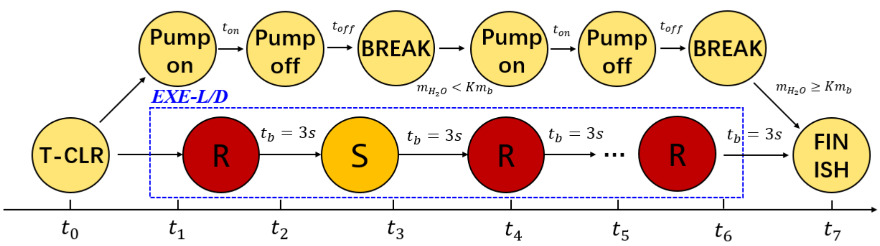

Figure 23: Spatial-temporal diagram of the brewing process, showing concurrent workflows of the pumper (top) and motor (bottom).

3. **MCU Decision: From STM32 to Raspberry Pi 5B**

   Initially, we selected Alientek's STM32 as the Microprogrammed Control Unit (MCU) due to its efficient performance in C programming and lower risk of circuit delays [32]. Please refer Appendix for our effort on STM32.

   However, the complexity of the GPIO setting and unfamiliarity with embedded system programming led us to reconsider our choice of MCU. Given our rich experience with Python, and the Raspberry Pi 5B's robust support for Python GPIO programming, we made the difficult decision in the final month to switch to the Raspberry Pi 5B as our MCU. This transition greatly accelerated our development[33].

   For further implementation, here is a list for MCU selection from our notebook on Raspberry Pi 5B versus STM32:

   Advantages

   (a) **Python and Open-source Library**: The vast libraries and community support for Python on Raspberry Pi enabled faster integration and debugging. Making it a better fit for us.

   (b) **Hardware Flexibility:** Raspberry Pi 5B offers more computational power and flexibility with built-in features like Wi-Fi, Bluetooth, and multiple USB ports, reducing the need for additional components.

   (c) **Future AI Support**: As the advancement of Large Language Models(LLMs), we are considering to include LLMs Agent in the future, which enables machine to be more customized and intelligent.

   Disadvantages:

   (a) **Power Consumption**: Raspberry Pi consumes more power. This can be resolved with an extra power card.

   (b) **Real-time Performance**: STM32 can handle real-time tasks with lower latency compared to the Raspberry Pi, which runs a full operating system. But in practice, this does not affect a lot.

   (c) **Cost**: The cost of a Raspberry Pi 5B (78usd) is higher than an STM32(26usd). This can be resolved with money.

   In conclusion, while the switch to Raspberry Pi 5B had its challenges, the overall benefits in terms of development speed and programming ease outweighed the drawbacks for our project needs.

4. **Software Engineering: Object Oriented Programming**

   Software engineering provides a reliable framework for implementing FSM. According to the FSM design, we need to develop algorithms that enable the coffee machine to transition from one state to another based on input signals. Each state performs specific tasks, which we program separately, leveraging appropriate application programming interfaces (APIs).

   Software Architecture:

   Here is the project directory structure tree to control and manage various electronic components. This structured approach, based on object-oriented programming (OOP) principles, minimizes technical debt and optimizes the system architecture.

```
smart_coffee_machine/
|- main.py               # Main control script, entry point of the program
|- fsm.py                # Small operating system, fsm for user interaction
|- coffee.py             # Coffee brewing recipe, custom programming
|                        # for light and dark roast coffee type.
|- config.py             # Configuration file, containing parameters such as
|                        # temperature thresholds and GPIO ports
|- hardware/             # Hardware control modules
|   |- __init__.py       # Initialization for importing hardware modules
|   |- sensor.py         # Sensor management module, e.g., weight sensor
|   |- pump.py           # Water pump driver module
|   |- motor.py          # Gear motor and servo driver module
|   |- servo_dict.json   # Dictionary mapping servo drive values to angles
|   |- display.py        # Driver module for OLED and 4-digit LED displays
|   |- led.py            # Driver module for 4-digit LED display
|   |- hx711_pi5.py      # Custom driver for HX711 pressure sensor for Pi 5B
|
|- utilities/            # Utility modules
```

```
|   |- __init__.py        # Initialization for importing utility modules
|   |- logger.py          # Logger utility for recording system logs
|   |- timer.py           # Timer utility for handling time-related functions
|
|- communication/          # Communication modules (reserved interface)
    |- __init__.py         # Initialization for importing communication modules
    |- server.py           # Network server module managing external APP commands.
    |- chatgpt.py          # Reserved for GPT agents for advanced HCI interactions
    |- protocol.py         # Defines data exchange formats and communication protocols.
```

This modular approach, underpinned by OOP principles, ensures a robust, maintainable, and scalable software architecture. Each component is encapsulated, promoting code reuse and reducing the risk of architectural anti-patterns.

2.6 User Interface System

The following subsection discuss around our implementation on how the control system operates with the user. Figure 24 shows the overview interaction, the OLED display here is for debugging.

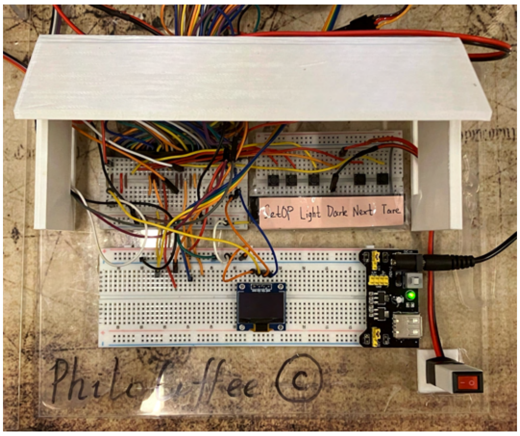

Figure 24: User Interface System, Top View Setting

Initially, the system is in the **OFF** state, which means it's not fully connected and ready to operate. To activate it, you'll need to plug in the Raspberry Pi, Xiaomi water heater, pumper, motor, and 5V stabilized voltage board.

Once fully connected, the system transitions to the **RESET** state. In this state, the OLED (Organic Light Emitting Diode) display will prompt you to press the SETOP button to select your preferred coffee roast: dark or light. Figure 25 shows the coffee machine's main interface, indicating the **RESET** state.

Upon selecting, the system enters the **SETOP** state. Here, the OLED display will ask you to press the LIGHT/DARK button to confirm your choice. Figure 26 show the coffee machine's instruction.

Next, the system moves to the **EXEC** state. In this state, the OLED display will instruct you to prepare the necessary water and coffee grounds. Once ready, you'll need to press the heater to dispense water and fill the coffee maker's hot water tank. Afterward, press NEXT to initiate the brewing process. Figure 27 and Figure 28 show the coffee machine has received the user's selection, and remind the user to add the ingredients.

The system then enters the **T-CLR** state, where the timer is set and cleared, ready to monitor the brewing process.

In the subsequent **PUMP** state, the system's software controls the pumper to transfer water from the hot water tank to the 3-hole nozzle. To prevent the pumper from dispensing too much water at once, which could lead to hot water splashes and potential injuries or damage to the electronic control system, the system is designed to pump water in cycles, with set pumping and resting times. Figure 29 shows the coffee machine's brewing interface, indicating the coffee machine is brewing the user's coffee, reflecting the mass of water.

Figure 25: Welcome to the main interface, it shows a wonderful opening.

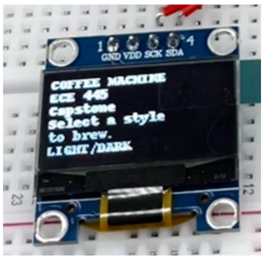

Figure 26: The OLED tells the user to select a brewing style, either light roasted or dark roasted.

Figure 27: The screen can tell the users that they selected the light-roasted coffee.

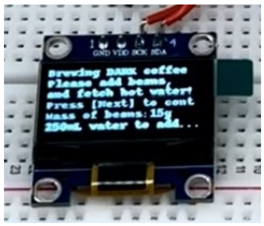

Figure 28: The screen can tell the users that they selected the dark-roasted coffee.

The system then progresses to the **EXE-L/D** state, where the OLED display indicates that the coffee is brewing. The system's software controls the movement of the servo motor and the gear motor, each with its own fixed working program. The servo motor rotates the corresponding gear rhythmically, while the gear motor rotates at a certain speed. This combination of motor movements simulates the manual process of pouring hot water over different parts of the coffee grounds to extract a variety of coffee flavors.

After the gear motor and servo motor have completed their work cycle, the system enters the **BREAK** state. Here, the weight sensor weighs the water to determine if the coffee concentration has been adequately diluted. If so, the brewing process is complete.

In the final **FINISH** state, the OLED display notifies you that the brewing process is over. You can then remove the cup from the weight sensor and enjoy your freshly brewed coffee. If you'd like to brew another pot, simply press the SETOP button to restart the process. Figure 30 shows that the machine has finished a brewing cycle.

However, if at any point you need to stop the brewing process, you can press the emergency stop button. In the **DAN** state, the OLED display will confirm that the coffee maker has ceased operation and invite you to use it again in the future. The system will then exit the program, halting all electrical components. Figure 31 shows that the machine stopped by an exception.

Figure 29: The light-roasted coffee is brewing, the pumper is working, and initially there is no water in the cup.

Figure 30: The coffee machine finishes brewing a cup of light-roasted coffee, using 34 seconds.

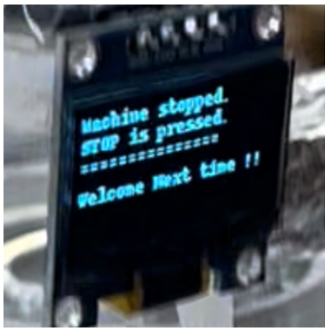

Figure 31: The machine stopped due to the emergency stop button is pressed.

Unit Tests on Raspberry Pi 5B The decision to using the Raspberry Pi 5B as the MCU significantly accelerated our project's progress. The Raspberry Pi 5B's support for Python programming played a key role in our project, as it fits well into our technology stack. The following are a series of unit test on buttons, LEDs and OLED displays.

As Fig. 32 shows, the button can transition the FSM from the RESET state to the SETUP state, allowing the user to select the coffee brewing mode (light roast or dark roast). Additionally, it assists the weight sensor during the tare operation and can function as an emergency stop button to ensure the coffee machine's safety.

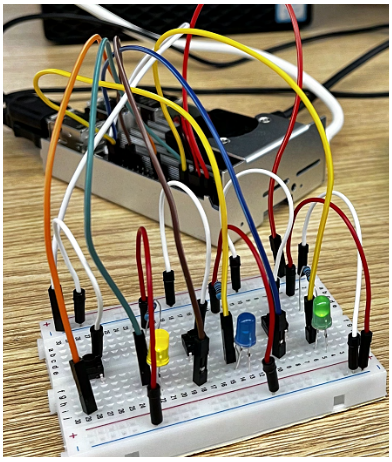

Figure 32: The button can control the LED for signal representation.

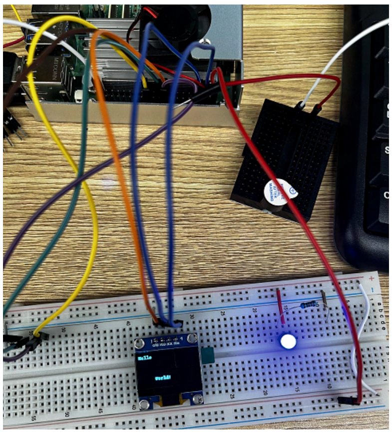

Figure 33: The OLED can display the strings by our code.

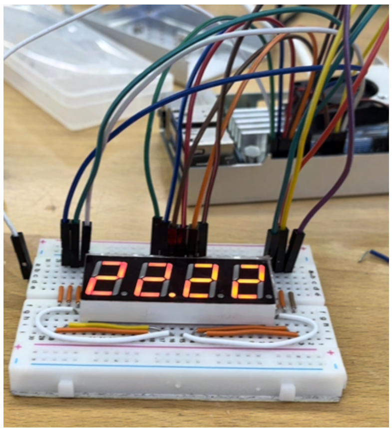

Figure 34: The 4-digit LED display for the weight sensor.

As Fig. 33 illustrates, the OLED serves as the main display for the coffee maker, guiding the user through operations and providing feedback. The OLED screen displays information according to the FSM state.

As depicted in Fig. 34, the HX711 module functions as an electronic scale, measuring the mass of water and coffee beans. It sends real-time data to the Raspberry Pi 5B via the I2C GPIO interface for closed-loop control. The 4-digit LED display accurately shows the measured mass for the user.

2.7 Heating System

Heating water is crucial in our project as it directly impacts the brewing process's efficiency and user safety. Professors Timothy Lee and Said Mikki emphasized the importance of a well-designed heating system due to inherent dangers as observed in previous projects. Initially, we tried to build our own heating subsystem using PTC heating elements with a counter-flow heat exchange scheme. However, tests revealed that our custom-built system was too slow, leading to long waiting times and a poor user experience. Additionally, sourcing heating modules safe for intimate contact with boiling water proved challenging.

1. **Initial Design with PTC Heating Elements** Our initial approach involved using Positive Temperature Coefficient (PTC) heating elements. PTC heaters are known for their self-regulating properties, which enhance safety by preventing overheating. Despite these benefits, our implementation faced significant challenges:

   - **Heating Speed:** The PTC elements were too slow to heat the water to the desired temperature, resulting in about 20 minutes for to wait, for example, heating a cup of 500 ml 25 degree Celsius water to boiling point typically takes over a quarter hour.
   - **Efficiency:** The thermal transfer efficiency of our setup was lower than expected, further exacerbating the slow heating issue.

   Given these drawbacks, we realized that our PTC-based system was not viable for providing a quick and efficient heating solution necessary for a satisfactory user experience.

2. **Integration of Commercial Hot Water Dispenser into our Heating System** To address the limitations of our initial design, we turned to a commercial product: the MSYSJ03MH Instant Hot Water Dispenser developed by Mi Home (Xiaomi Group). This device can heat water to 95°C in approximately 3 seconds, offering a much faster solution compared to our custom-built system. The key specifications of the Mi Home heater are:

   - **Rapid Heating:** Achieves 95°C in roughly 3 seconds.
   - **Power Requirements:** Operates on an external 220V power source.

- **Compact and Safe Design:** Features an integrated safety mechanism and efficient thermal insulation.

By incorporating the Mi Home Instant Hot Water Dispenser, we achieved a significant improvement in the heating performance of our coffee machine. Although this approach involves using an external commercial product, it provides the best compromise between performance, safety, and user convenience.

3. **System Integration and Safety Considerations** The integration of the Mi Home Instant Hot Water Dispenser into our heating system required careful planning to ensure seamless operation within our overall design. Key integration aspects include:

   - **Power Supply:** The heater operates on a 220V external power source, which necessitates proper insulation and safe handling to prevent electrical hazards.
   - **Control System:** The heater is controlled via our Raspberry Pi 5B-based system, ensuring precise temperature management and coordination with other subsystems.
   - **Safety Features:** The instant hot water dispenser includes built-in safety mechanisms, such as automatic shut-off and thermal protection, reducing the risk of overheating and other potential dangers.

   We also designed additional safety measures to further enhance the system's reliability:

   - **Emergency Shut-Off:** An emergency stop button is implemented to immediately cut off power in case of malfunction or unexpected issues.
   - **Thermal Insulation:** Double-layer stainless steel with an air gap ensures effective thermal insulation, minimizing heat loss and preventing accidental burns.

   The decision to use a ready heater is a strategic compromise, which ensures both the performance and safety of our coffee machine. This commercial product's rapid heating capability and integrated safety features align well with our high-level requirements, ensuring a high-quality and user-friendly brewing experience.

2.8 Pumping System

The pumping system in our intelligent coffee machine is engineered to transport water from the hot water tank to the brewing subsystem and subsequently onto the coffee grounds. It is a critical component that ensures precise water flow, which is essential for replicating the pour-over brewing technique. In addition, the controllability of the output pumping flow rate regarding the power in a given amount is also of high significance to pump the hot water in a steady way. Below are the detailed descriptions and components of the pumping system.

1. **System Components and Design:**

   The primary components of the pumping system include a diaphragm pump, control circuitry, and tubing to direct the flow of water.

   - **Diaphragm Pump**: Selected for its ability to deliver consistent and controllable flow rates, essential for accurate brewing.
   - **Control Circuitry**: The pump is controlled using a PWM signal generated by a microcontroller (Raspberry Pi 5B), allowing precise control over the flow rate.
   - **Tubing**: Food-grade silicone hoses are used to ensure safety and maintain the quality of water of over 95 degrees Celsius at least delivered to our brewing system.

2. **Flow Rate Control:**

   The flow rate is a crucial factor in the brewing process, affecting the extraction time and the quality of the coffee. The control system uses PWM signals to adjust the pump's speed, thereby controlling the flow rate.

   - **Control Method**: The flow rate is controlled by varying the power percentage delivered to the pump. The relationship between power percentage and flow rate is non-linear and was determined through experimental calibration.

- **Polynomial Fit**: A polynomial model was developed to accurately predict the flow rate based on the power percentage, ensuring optimal brewing conditions.

3. **Experimental Setup and Results**

   To determine the optimal operating conditions for the pump, we conducted experiments varying the power percentage and measuring the resulting flow rate and time to dispense a set volume of water.

   - **Experimental Setup**: The pump's power was varied from 20% to 100%, and the corresponding flow rates and times were recorded.
   - **Results**: The flow rate increased with power percentage, following a non-linear trend. The optimal flow rate for brewing was found at approximately 50% power, aligning with industry standards for coffee extraction.

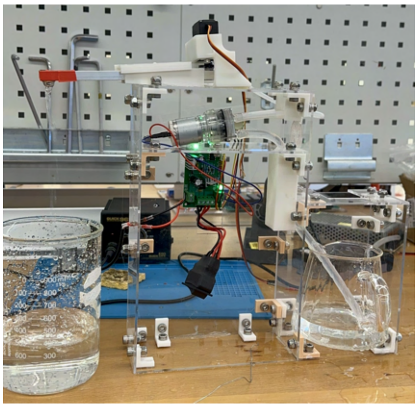

Figure 35: Initial setup of the pumping system.

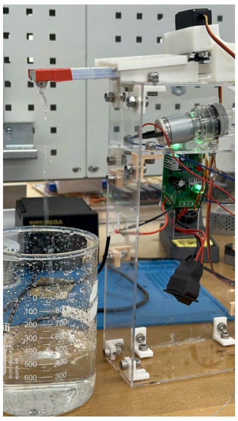

Figure 36: Pumping system during operation.

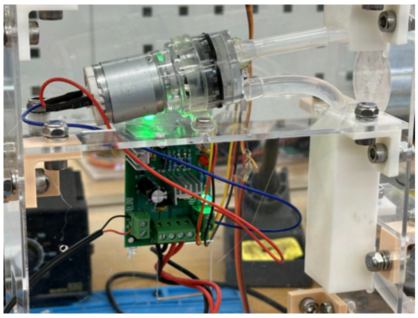

Figure 37: Close-up of the pumping mechanism.

Figure 38: Different stages of the pumping system.

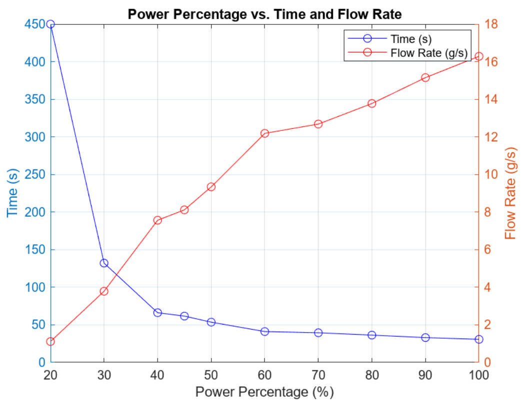

Figure 39: Power Percentage vs. Time and Flow Rate with Dual Y-axes

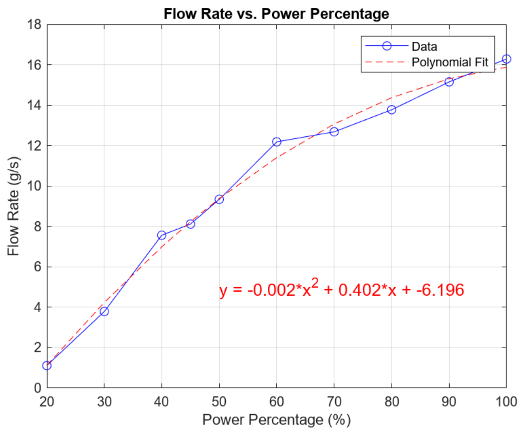

Figure 40: Polynomial Fit of Flow Rate vs. Power Percentage

The pumping system, with its precise control and optimized flow rate, ensures consistent coffee quality by delivering water at the correct rate and volume. The polynomial fit model provides a reliable method for controlling the pump, making the brewing process efficient and repeatable.

2.9 Sensor System

In order to verify the functionality of our coffee machine, real-time sensors are indispensable. Typically, temperature and weight sensor are widely implemented in real-time control system [34, 35]. Here is sensor details in our system:

1. **Weight Sensor for Closed Loop Control**

   Via a HX711 Load Cell Amplifier, our coffee machine can measure the mass of ground coffee beans and water poured into sharing pot. If the bean-water ratio is met, then the system ends the brewing process.

   As shown in figure 41, the detected mass info is displayed via 4-digit LED display. The accuracy is 0.1g, range is 5kg, and mass detect frequency at 80Hz[36]. The mass can also be reflected on OLED display at 50Hz[37]. Because there is no driver for HX711 on Raspberry Pi 5B, a relatively new MCU, we rewrote an open-source driver to measure the weight in real-time[38, 39].

2. **Temperature Sensor for Open-Loop Control**

   Initially, we designed the temperature sensor for closed-loop control. However, we didn't meet in time to add another sensor for closed-loop control in the short development time. Instead, as shown in figure 42, we used a food-grade sensor out-of-loop. It can be safely used for measuring temperatures in different parts of the coffee machine.

   The temperature sensor has a long cable that allows it to be placed at various points within the coffee machine to provide real-time temperature feedback, which is helpful in system development. The measured temperatures at different positions are summarized in the following table:

   Table 2: Average Temperatures at Different Positions in the Coffee Machine

Position	**Average Temperature (°C)**
Outlet	95
Before Pour Over	88
After Pour Over	67
Final Coffee Liquid	44

   Although the current setup uses the temperature sensor in an open-loop control system, we plan to upgrade to a better food-grade temperature sensor in the future. This will allow us to integrate the sensor into the system for closed-loop control, improving accuracy and efficiency in maintaining optimal brewing temperatures.

3. **Humidity Sensor for Daily Coffee Selection** We also employed a humidity display at the side of our machine, as shown in Figure 43. This display shows the current temperature and humidity, which helps users select coffee beans and adjust brewing parameters accordingly.

   Different humidity levels can significantly affect the coffee brewing process. High humidity can cause coffee beans to absorb moisture, leading to changes in grind size and extraction rates. Conversely, low humidity can cause beans to lose moisture, which also affects the brewing parameters. Here are some academic insights on how to adjust coffee brewing based on humidity:

   (a) **Grind Size**: In high humidity conditions, it is recommended to use a coarser grind to prevent over-extraction, as the beans may have absorbed moisture. In low humidity, a finer grind might be necessary to compensate for the drier beans.

   (b) **Water Temperature**: Higher humidity levels may necessitate a slight decrease in brewing temperature to avoid over-extraction. Conversely, in lower humidity, a higher brewing temperature might be needed to achieve the desired extraction level.

   (c) **Storage and Handling**: Proper storage of coffee beans in airtight containers can mitigate the impact of humidity fluctuations. Ensuring beans are stored in a stable environment helps maintain consistent quality and flavor profiles.

   (d) **Brewing Time**: Adjusting the brewing time is another factor to consider. High humidity might require a shorter brewing time, while low humidity may require a longer duration to ensure optimal extraction.

By incorporating real-time temperature and humidity data, our coffee machine can assist users in making informed decisions about bean selection and brewing adjustments, ultimately leading to a more consistent and high-quality coffee experience.

4. **pH Sensor for Coffee Drink Evaluation on Hand**

   We use the pH sensor for various reasons. Initially, we proposed separating the testing liquid from the drinking liquid to keep the coffee edible. At the same time, we can know how much acid is in the coffee to adjust the user's drinking experience. Finally, we attached an on-hand pH sensor and found the pH of the coffee is around 4.5, as the figure shown.

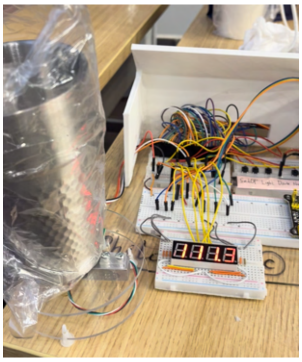

Figure 41: Weight sensor works on the 4-Digit LED display.

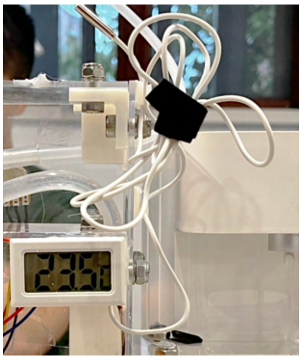

Figure 42: Temperature sensor works for open-loop feedback.

Figure 43: Humidity sensor showing 23°C and 70% RH.

Figure 44: Coffee result with pH sensor, tested in separate cup.

2.10 Power System

To supply electrical power for the whole system, we need to ensure all components receive the correct voltage and current for safe and efficient operation, as illustrated in Figure 45.

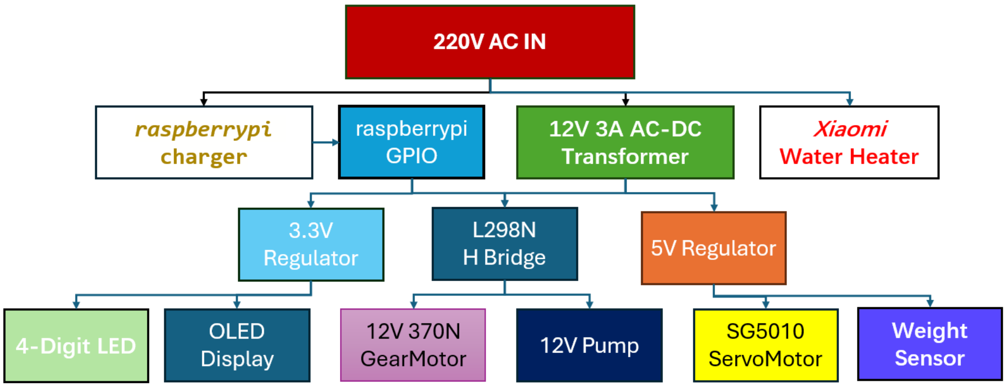

Figure 45: Power system diagram showing the distribution of power from 220V AC input to various DC actuators.

1. **Power Distribution:**

   The power system begins with a 220V AC input, which is then converted to 12V DC, then fed into the L298N H-Bridge for actuators' control[27]. To meet the voltage requirements of different components, the 12V DC is further regulated down to 5V and 3.3V using voltage regulators. And *Xiaomi water heater* is charged separately by another standalone 220V charger due to safety consideration. The whole diagram can view figure 47

2. **Signal Transmission:**

   Signal transmission is managed via the Raspberry Pi GPIO, which communicates through PWM and I2C signals. The GPIO signals are carefully connected to avoid noise and interference, ensuring reliable connection within the coffee machine.

3. **Current Considerations:**

   Given the different power needs of our system, it is important to manage the total current flow. Test result shows the total load can reach over 2.1 A current. However, The Raspberry Pi power supply was designed with a maximum current of 3mA per GPIO pin. If you load each pin with 16mA, the total current is 272mA. The 3.3V supply will collapse under that level of load[40]. If we supply all the motors from only Raspberrypi, then the MCU may burn out easily. We chose a 12V 3A AC-DC transformer to provide sufficient current capacity for the devices. Each regulator's output current is carefully calculated to ensure that the total system current remains within safe limits. We need resisters in series to prevent overheating and potential damage to the circuit.

4. **Importance of Common Ground (GND):**

   As we used a separate 5V regulator to supply enough current for the breadboard, thus a common ground connected to Raspberry is very important for signal transmission. We need to minimize voltage differences between interconnected components, reducing the risk of ground loops and signal noise.

5. **Careful Wire Connection** Connecting the wire physically is an elaborate skill. As there are numbers on the breadboard, my idea is that we connect the GPIO pins on the Raspberry Pi 5 to an empty breadboard at first, as shown in Figure 46. There are two reasons. On the one hand, the order of GPIO distribution on the Raspberry Pi 5 is messy. On the other hand, it would be more convenient if we connected the pins with physical components with clear GPIO numbers.

6. **Testing and Verification**: Perform thorough testing with a multimeter to verify correct voltages and ensure no short circuits before powering up the system.

By carefully managing out power, our coffee machine is functionally reliable, following the safety code for product.

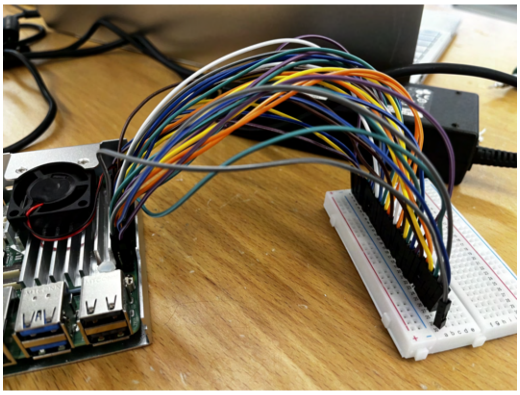

Figure 46: The innovative wire connection design is a standout feature, enabling effortless identification of each wire's corresponding GPIO pin. This meticulous organization enhances both functionality and aesthetics.

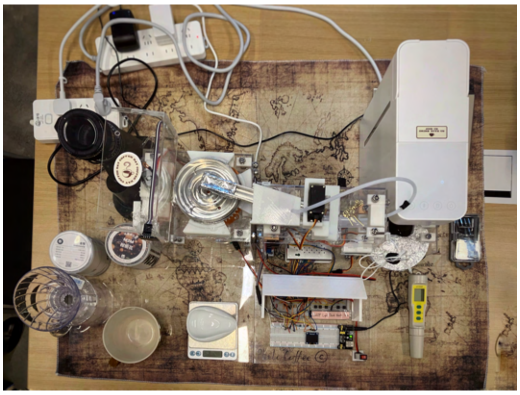

Figure 47: Machine Top View with All Power Setting.

3 Cost Analysis

3.1 Material Cost

Here is a comprehensive breakdown of the expenses for prototyping our intelligent coffee machine. This analysis covers three primary cost categories: material costs, labor costs, and the cost of coffee beans. Material costs include all the components required for the construction and functionality of the machine. Labor costs account for the time and effort invested by our team in research, design, programming, assembly, and testing. Additionally, the cost of coffee beans is considered to provide a complete financial overview.

To ensure accurate and transparent financial planning, our team uses *Notion*, a collaborative software tool, to record notebook and all costs associated with the project[41]. It helps us manage our budget effectively and ensures that all team members can see latest cost information.

Item	**Price (in USD)**
Mechanical Components	
Steering Engine	1.5
Water Tank	3.0
Non-ferrous Metal Rolled Materials, Aluminium Sheet	10.5
Customized Metal Rings, Gears, and Guiding Gutter	105.0
3D Printed Parts	11.0
Silicone Hose Flexible Water Pipe (Food Grade)	7.0
Electronic Components	
pH Meter	7.5
Electronic Scale	13.5
STM32 Developer Board	8.0
Raspberry Pi Kit	68.0
RTS-2316x1 Electronic Temperature Sensor	1.5
DC Motor	4.0
DC 12V Pump	14.0
Microcontroller Motor	4.0
Breadboard Wires, Goggles	12.5
Other Small Electronic Parts and Components	11.0
Additional Equipment	
MSYSJ03MH Instant Hot Water Dispenser	28.0
Network and Connectivity	
Network Interface and Adapters	9.0
Total Material Cost (excluding Coffee Bean)	**319.0**
Coffee Bean Cost (for reference)	**41.0**

Table 3: Material Cost Analysis for Our Project

Table 3 records the estimated material costs for our design. We have sourced various components essential to emulate professional barista techniques, including heating elements for temperature control, DC motors for mechanical movements, and silicone hoses for fluid transfers. The cost estimates are based on current market prices in USD, adhering to our budget provided by ZJUI's Innovation Lab.

3.2 Labor and Development Costs

Table 4 summarized the labor and manufacturing cost to date included in our design. Our labor costs are computed based on the cumulative hours spent by team members in designing, programming, and assembling the coffee machine. This cost is factored in to provide a realistic estimate of the project's non-material expenses.

Table 4: Human Labor Cost Table

Description	Hours	Rate (in USD)	Total (in USD)
Product Research	20	20	400
Design and Programming	70	20	1400
Assembly and Testing	70	20	1400
Documentation and Reporting	40	20	800
Metal Processing	5	40	200
Total Labor Cost	**205h**	**-**	**4200**

Table 3 summarizes the labor and manufacturing costs to date. Our labor costs are computed based on the cumulative hours spent by team members in designing, programming, and assembling the coffee machine. This cost provides a realistic estimate of the project's non-material expenses.

3.3 Overall Project Cost

Table 5 shows the overall project cost, including material costs, labor costs, and coffee beans cost.

Table 5: Overall Cost Table

Cost Type	Estimated Cost (in USD)
Material Costs	319.0
Labor Costs	4200.0
Coffee Bean Cost	41.0
Grand Total	**4560**

3.4 Cost Optimization

While our current cost estimates are based on prototype quantities, we are actively seeking bulk pricing options for future scalability. For example, the cost for metal three-hole nozzle and slide can be greatly reduced from $103.56 to even $5 per set. We pay the price for food-grade material, which contains high custom fee for few products. And the usage of RaspberryPi increases the cost for MCU by 200%, in large-scale production, we can utilize cheap STM32 board for system control. Further reducing the cost for single product. Additionally, we are exploring assistance from the school and coffee shops for possible sponsorship of the expenses. We hope our senior design can be exhibited in the demonstration room of ZJUI after our graduation.

4 Conclusions

4.1 Accomplishment

We implemented 80% of our previous design document, given only three months for the coursework duration. Finally, our accomplishments correspond to the high-level requirements are listed as follows:

- **Precision and Consistency:** At the end of the project, our elec-mech design can achieve following features:
  1. The piezoelectric-based weighing subsystem is able to weigh the coffee beans within a 2-gram accuracy.
  2. The heater is able to generate hot water at the desired temperature of 95 degrees Celsius at the outlet, with an on/off delay time of less than 0.5 seconds.
  3. The pumping subsystem is able to pump water at a programmed flow rate ranging from 1.1 grams per second to 16.5 grams per second. The fitted polynomial describing the relation between the pump power percentage P and the water flow rate Q (in grams per second) is $Q = -0.002P^2+0.402P-6.196$.
  4. The servo motor in the brewing system successfully rotates the hot water slider to three fixed angles, delivering hot water into three different holes.
  5. The gear motor that drives the brewing nozzle is controlled to have a precise rotation speed ranging from 0.101 rotations per second to 9.642 rotations per second. These two bounds are measured by conducting a working test.
- **Affordability and Accessibility:** The machine should be be a cost-effective solution without compromising flexibility or quality. To ensure food grade safety of the pipes that are able to hold hot water of temperature up to 95 degrees Celsius, we choose to use pipes made from PP (poly-propylene) that can sustain hot water up to 130 degree Celsius. We also use the Tin foil for the barbecue to wrap the end of the hot water slider and the hot water tank to avoid the spurtter of hot water doing operation. These food grade material is affordable and widely accessible on market, the overall material cost on our final model of coffee machine is 200 USD . The housing of our coffee machine is made by acrylic board, which costs 15 USD. The most part of our expenditure lies in the manufacturing the metal parts of the machine (three-hole nozzle, nozzle ring, and hot water slider) which costs 120 USD , and the Raspberry Pi beard as the controller of our machine which costs 70 USD. By carefully selecting affordable materials and open-source software, we can reduce more costs while maintaining high performance.
- **Easy User Experience:** Our coffee machine is carefully designed to provide users with an interesting and effortless coffee-making experience. By simply pressing a button and adding water and beans, users can choose their favourite brewing styles, and see the transparent pour-over process, which would be educational and demonstrative for potential product investors.
- **Durability and Maintenance:** The FSM supports machine to brew coffee arbitrary times. And breadboard wiring makes it easy for quick trouble shoot. By estimation, the durability of our coffee machine is able to operate robustly for over 20 hours before the first maintenance. After that, the first maintenance consists the acrylic board connection refastening, Raspberry connection check, wearing parts replacement including tin alloy and PP pipes.
- **Quality of Brew:** Our pour-over coffee machine has been tested by over five coffee tasters on our campus, who praised the coffee's temperature and acidity. They appreciated the flexible brewing operation, as well as the programmable settings for hot water flow rate, brewing nozzle rotation control, and overall delicacy. Users noted that the bitter taste could be reduced by selecting coffee beans with different roast degrees, further mentioned that the temperature sensor does not measure temperature at the brewing nozzle's exit. In all, our machine successfully produces similar favor to human barista.
- **Coffee Roast Degree Difference**: Our high flexible mechanical design can brew the coffee as professional barista with most necessary movement. The programmable APIs make it easy to adjust for different roast degrees, ensuring custom flavor. We used two types of beans from M2M Coffee: the dark-roasted Golden Mandheling and the light-roasted Yirgacheffe [42], both of which performed excellently. The dark roast shows chocolate and nutty flavors, while the light roast is similar to jasmine and citrus. Experiments have proven that our coffee extraction is effective at different roast degree.

Here is our prototype machine for user study and demonstration:

Figure 48: Our Coffee Machine, settled in Guantong Residential College.

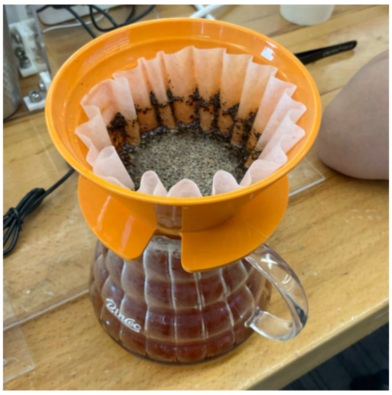

Figure 49: The coffee grounds after brewing are very good in outlook.

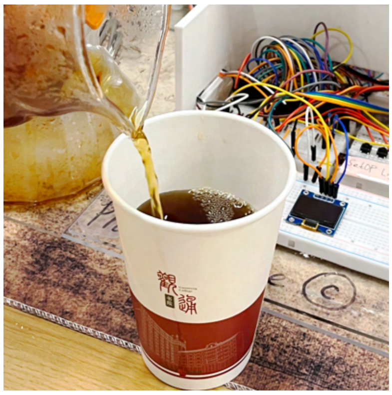

Figure 50: A cup of coffee produced by our machine, sweat and sour, like orange.

4.2 Uncertainties

During the difficult development, we made many sad decisions, and these engineering practice are also valuable lessons. Here are some of them:

- **Eliminating States in the FSM:** Due to the limitation of the sensor system, the states that detect the water height and temperature cannot be implemented.
- **Cancel a Brewing Style:** As the working period isn't long, we canceled a brewing style called "HOT" for cleaning the coffee machine.
- **Separated Heating System from the Close Loop:** As the heating system is too dangerous, we finally decided to prepare hot water directly in our senior design rather than building a water heater.
- **Short Circuit During Demo:** The charger of Raspberry Pi 5B was shorted during the demo. This is caused by an unreliable power strip at West Hall, out of the innovation lab. Fortunately the MCU is protected, functioning correctly. Take care and resettle your device in advance!
- **Loosed Mechanical Structure:** Although we've implemented code for the servo motor to spin at a certain angle, the motor usually slipped, which blames to incorrectly applying hot melt glue to adhere the three-hole nozzle onto the ring gear in our test version. Nevertheless, we shifted to physical lock (Using the concavity and convexity of the nozzle and the ring gear) of these two components, which shows robust and confident connection of the final version of our coffee machine model.
- **Switching MCU in a short time:** As stated before, though C fits well with embedded system, it's quite difficult to implement it in a short time. Using Raspberry Pi 5 instead can simplify the code implementation.
- **Overwhelmed OLED display:** If too many signals interrupt the OLED, it will shut down and cause a compilation error, stopping the entire coffee machine.
- **Subsystem Synchronization:** As the working period is too short, we are still far from getting the pumping system and brewing system synchronized, utilizing the synchronization knowledge learned from ECE 391, computer system engineering.
- **Unstable weight sensor:** Although the weight sensor can reflect the water mass, sometimes the mass is not precise.

We hope these cases can help further students' development! Feel free to contact via PhiloCoffee's Github Organization for further information.

4.3 Ethics and Safety Consideration

Aligning with IEEE and ACM ethical standards[43], we adhere to safety regulations throughout the whole project.

4.3.1 Ethics Factors

1. **Assess Impact on Society:** This project democratizes high-quality coffee, reducing the need for extensive training to enjoy well-brewed coffee. It aims to enhance life quality while minimizing waste and negative impacts.

2. **Fairness and Non-discrimination:** The machine is designed to be accessible and usable by all individuals, ensuring no biases based on personal characteristics.

3. **Honesty and Transparency:** The development process is conducted transparently, with clear communication of the machine's capabilities, limitations, safety features, maintenance needs, and potential risks.

4.3.2 Safety Factors

1. **Burn Prevention:** The machine includes safety features like an emergency button to control temperature and prevent accidental spills[44]. Components are insulated, and clear warning labels and user guides are provided. Temperature regulation mechanisms prevent overheating.

2. **Food Grade Transmission:** The machine is designed for easy cleaning to avoid pathogen breeding, using materials like aluminum alloy for food safety[45]. Rounded internal edges aid in cleaning, and hygiene protocols ensure water quality and minimize contamination risks.

3. **Mechanical and Pressure Safety:** The machine adheres to *Pressure Systems Safety Regulations (PSSR)* with regular inspections and safety checks. We test and maintain safe level pressure for the pump, release remaining pressure to prevent over-pressure scenario [46].

4. **Electrical Safety:** Electrical components comply with safety standards to prevent electric shock, short-circuiting, or fire. Features include proper grounding, insulation, overcurrent protection, waterproofing, fail-safe mechanisms, and redundant circuits. Compliance with *International Electrotechnical Commission (IEC) standards* is ensured, with regular safety assessments.

In Innovation Lab @ D221, our team follows safety and ethical requirements, protecting ourselves and future machine users.

4.4 Future Work

In the future, we aim to further enhance the capabilities and efficiency of the Intelligent Pour-Over Coffee Machine. We have several key areas targeted for development:

- **Implement a complete, closed-loop hot water system.** Currently, the method to prompt the user to heat water is primitive; we plan to automate this process to streamline operations and enhance user experience.

- **Integrate Large Language Models (LLMs) as agents to enable more intelligent analysis of brewing experiences and techniques.** This integration aims to leverage advanced AI to refine and optimize the brewing process based on accumulated data and learning algorithms.

- **Develop a more comprehensive power system by integrating the entire system into a single PCB.** This will facilitate easier wiring and enhance safety by minimizing exposed components and connections.

These advancements will focus on improving not only the technical aspects of the machine but also the practical usability and safety features, pushing the boundaries of automated coffee brewing technology.

4.5 Summary

As Prof. Timothy Lee said during demo, "this cup of coffee contains our sweat and tears". Before we got involved in this senior design, we didn't have much of a concept of hardware and software integration, even though we had taken a lot of hardware and software-related courses during our undergraduate studies. After we four worked the coffee machine out together, we felt a sense of accomplishment when letting friends to taste the coffee our machine made.

Figure 51: Professors happily drinking our coffee during demo

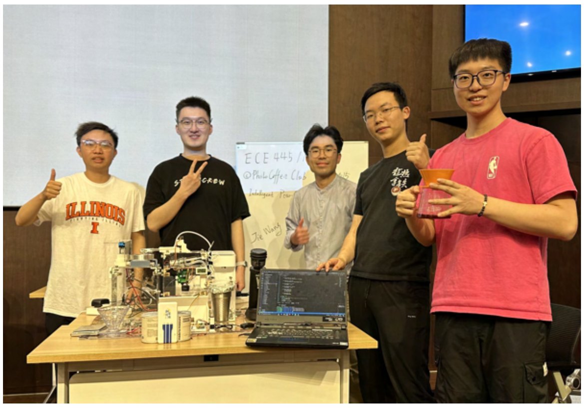

Figure 52: PhiloCoffee Club members constructed this senior design together.

Figure 53: Our Team Signature on the Machine.

Figure 54: Prof. Mikki and TA Dr. Thng's Signature.

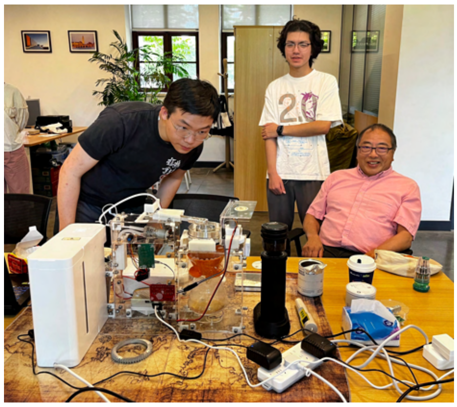

Figure 55: We invited Prof. Lap Chee Tsui to try our gadget. He encouraged our effort on the innovation.

All in all, it is an interesting senior design with both innovation and commercial potential. That's a wonderful creation from ***PhiloCoffee Club.*** Thanks everyone for helping us during our development. We believe our hard work deliver a good end for our graduation at ***ZJU-UIUC Institute***!

References

[1] Zhenzuo Si et al. *Tea Blend Distributor*. Senior Design Laboratory Final Report Team 14. Sponsor: Prof.Timothy Lee, TA: Xinyi Xu. University of Illinois at Urbana-Champaign, May 2023.

[2] CoffeeGeek. *Pour Over History and Development*. http://www.coffeegeek.com.

[3] Expert Market Research. *Coffee Machines Market Trends, Share Growth 2024-2032*. http://www.expertmarketresearch.com.

[4] N. Stanek, M. Zarebska, Ł. Biłos, et al. "Influence of coffee brewing methods on the chromatographic and spectroscopic profiles, antioxidant and sensory properties". In: *Scientific Reports* 11 (2021), p. 21377. DOI: 10.1038/s41598-021-01001-2. URL: https://doi.org/10.1038/s41598-021-01001-2.

[5] Jie Wang et al. *Proposal: Intelligent Pour-over Coffee Machine based on Pre-trained Imitation Learning V2*. Senior Design Laboratory Project Proposal Team 15. ZJU-UIUC Institute, Mar. 2024.

[6] Jie Wang et al. *Design Documen: Intelligent Pour-over Coffee Machine V2.1*. Senior Design Laboratory Design Document Team 15. TA: Dr. Guo Hao THNG. ZJU-UIUC Institute, Apr. 2024.

[7] Maxpresso. *Maxpresso's Bilibili Video Channel*. https://space.bilibili.com/1895786708. Coffee shop owner at Suzhou. 2023.

[8] GIFCO. *Official WeChat Account of GIFCO*. WeChat ID: GIFCOFEEE-CO. Previously known as GIF Coffee. Offers interesting food and coffee in Shenzhen. 2024.

[9] . *ZJU Coffee Club WeChat Official Account*. WeChat Official Account: ZJUCOFFEE. Affiliated with Zhejiang University. 2024.

[10] PhiloCoffee Club. *PhiloCoffee: Exploring the Intersection of Coffee and Philosophy*. https://philocoffee.github.io/. 2024.

[11] Thierry Jean Robert Fabozzi and Stephane Hentzel. "Coffee Maker". US20210228018. Justia Patents Search. July 2021.

[12] Louis Dakis. "Method of Using a Multi-Layer Brewing Cup". US8808778. Justia Patents Search. Aug. 2014.

[13] Jenn Chen. "Intellectual Property In Coffee: Imitation Is No Longer Flattering". In: *Sprudge Coffee* (Oct. 2018). URL: https://sprudge.com/intellectual-property-in-coffee-134342.html.

[14] Norton Rose Fulbright LLP. "Distinguishing Trademark and Patent Protection and the Nespresso Coffee Capsules Court Battle". In: *Financial Institutions Legal Snapshot* (May 2024). URL: https://www.financialinstitutionslegalsnapshot.com/2024/05/distinguishing-trademark-and-patent-protection-and-the-nespresso-coffee-capsules-court-battle/.

[15] Brew Smartly. *8 Best Automatic Pour Over Coffee Makers of 2024 - Reviews and Buying Guide*. https://www.brewsmartly.com. 2024.

[16] Coffee Affection. *7 Best Automatic Pour Over Coffee Machines in 2024 – Top Picks & Reviews*. https://www.coffeeaffection.com. 2024.

[17] James Hoffmann. *James Hoffmann YouTube Channel*. https://www.youtube.com/user/jimseven. 2024.

[18] The Spruce Eats. *The 10 Best Pour Over Coffee Makers of 2024, Tested & Reviewed*. https://www.thespruceeats.com. 2024.

[19] Batch Coffee. *7 Best Pour Over Coffee Makers 2024 (Plus 1 To Avoid)*. https://www.batchcoffee.co.uk. 2024.

[20] Chemex. *Best Pour Over Coffee Makers 2024*. https://www.chemexcoffeemaker.com. 2024.

[21] Gemilai. *Gemilai CRM 4106 Pour-Over Coffee Machine*. http://gemilai.chujub2b.com/. 2023.

[22] NFPshop. *25mm Metal Gear Motor Model NFP-JGA25-370*. 2024. URL: https://nfpshop.com/product/25mm-metal-gear-motor-model-nfp-jga25-370-en.

[23] TowerPro. *SG-5010*. 2024. URL: https://www.towerpro.com.tw/product/sg5010-4/.

[24] Farnell. *DS18B20 5V Toy Motor*. https://www.farnell.com/datasheets/1863913.pdf. 2015.

[25] Adafruit. *TT Motor Datasheet*. https://www.verical.com/datasheet/adafruit-brushless-dc-motors-3777-5912007.pdf. 2015.

[26] Phidgets. *DCM4002 24V/45Kg-cm/182RPM/18:1 DC Gear Motor*. https://www.phidgets.com/?prodid=1094. 2024.

[27] STMicroelectronics. *L298 Dual Full-Bridge Driver*. Accessed: 2024-05-27. 1997. URL: https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf.

[28] Chen Xizhen et al. "Design of automatic washing machine based on Verilog HDL language". In: *Proceedings of 2011 International Conference on Electronics and Optoelectronics*. Vol. 3. 2011, pp. V3-38-V3–40. DOI: 10.1109/ICEOE.2011.6013294.

[29] Giuliano Donzellini et al. "The Finite State Machine as System Controller". In: *Introduction to Digital Systems Design* (2019), pp. 363–464.

[30] Maksym Figat, Cezary Zieliński, and René Hexel. "FSM based specification of robot control system activities". In: *2017 11th International Workshop on Robot Motion and Control (RoMoCo)*. IEEE. 2017, pp. 193–198.

[31] Ana Monga and Balwinder Singh. "Finite state machine based vending machine controller with auto-billing features". In: *arXiv preprint arXiv:1205.3642* (2012).

[32] Alientek. *Alientek Official Website*. https://www.alientek.com/. Accessed: 2024-05-27. 2024.

[33] Raspberry Pi Foundation. *Raspberry Pi Documentation*. https://www.raspberrypi.com/documentation/. Accessed: 2024-05-27. 2024.

[34] Hugh Calkins et al. "Temperature monitoring during radiofrequency catheter ablation procedures using closed loop control. Atakr Multicenter Investigators Group." In: *Circulation* 90.3 (1994), pp. 1279–1286.

[35] MJ Long et al. "Novel online temperature control system with closed feedback loop for steel continuous casting". In: *Ironmaking & Steelmaking* 38.8 (2011), pp. 620–629.

[36] *Analog/Digital (A/D) Converter Chips for Electronic Scales*. URL: http://www.aviaic.com/images/ueditor/1566282174439483.pdf.

[37] TELESKY. *0.96 inch 4-pin OLED Display IIC Interface White / MK006614*. https://telesky.yuque.com/bdys8w/01/ywc0kvo0kh51sokw. Based on SOLOMON SYSTECH SSD1306 technical data. 2024.

[38] tatobari. *HX711 Python Library for Raspberry Pi*. https://github.com/tatobari/hx711py?tab=readme-ov-file. 2024.

[39] underdoeg. *Python port for RaspberryPI of the HX711 Board*. https://gist.github.com/underdoeg/98a38b54f889fce2b237. 2023.

[40] Raspberry Pi Foundation. *Raspberry Pi Documentation: Why can't I set all my pads to the maximum current?* https://www.raspberrypi.com/documentation/computers/raspberry-pi.html#why-cant-i-set-all-my-pads-to-the-maximum-current. 2024.

[41] Inc. Notion Labs. *Notion: Your Connected Workspace for Wiki, Docs & Projects*. https://www.notion.so/. Accessed: 2024-05-27. 2024.

[42] M2MCoffee. *M2MCoffee WeChat Official Account*. WeChat Official Account. Accessed: 2024-05-27. 2024.

[43] *ACM Code of Ethics and Professional Conduct*. https://www.acm.org/code-of-ethics.

[44] Lawrence J. Buckfire. *Coffee Burn Injury Lawsuits*. http://www.natlawreview.com/article/coffee-burn-injury-lawsuits. 2022.

[45] Sarah Victoria Walker et al. "Bug in a mug: are hospital coffee machines transmitting pathogens?" In: *BMJ* 383 (2023). DOI: 10.1136/bmj.p2564.

[46] Espresso Service. *Coffee Machine Safety*. http://www.espressoservice.co.uk/resources/coffee-machine-safety/.

Appendices

A. User Study and Reverse Engineering

Please refer our Design Document for user study and reverse engineering.

B. *'coffee.py'*: Brewing Recipe We Took

Here is the programmable coffee recipe code for our intelligent pour-over coffee machine. The code includes the initialization of the coffee machine, execution of different coffee brewing recipe. With this separate recipe file, we can easily adjust the brewing recipe according to the new kinds of coffee bean. There is still much space for more intelligent brewing decision-making, which could be implemented via LLMs and Knowledge Graph, to be done in the future.

```
# ./coffee.py: Programmable Coffee Recipe Book

from hardware.motor import GearMotor, ServoMotor
from hardware.pump import PumpControl
from util import logger
from time import sleep
import sys
import warnings
from gpiozero.exc import PWMSoftwareFallback

# Suppress specific warning
warnings.filterwarnings("ignore", category=PWMSoftwareFallback)

# Define GPIO pins for motors and pump
SM_PIN = 18
GM_PIN_F = 19
GM_PIN_B = 20
PP_PIN = 13

class Coffee:
    def __init__(self, ws=None, wstare=None, display=None, stop=None, mass_bean=None,
    ↪ mass_water=None, mode_list=None, mode=None, timer=None, K=None, next=None):
        self.gear_motor = GearMotor(GM_PIN_F, GM_PIN_B)
        self.servo_motor = ServoMotor(SM_PIN)
        self.pump = PumpControl(PP_PIN)
        self.ws = ws
        self.wstare = wstare
        self.display = display
        self.stop = stop
        self.mass_bean = mass_bean
        self.mass_water = mass_water
        self.mode_list = mode_list
        self.mode = mode
        self.timer = timer
        self.K = K
        self.next = next

    def safe_sleep(self, duration, interval=1):
        """
        Safely sleep for a given duration while checking the stop button and water weight.
        """
        while duration > 0:
            if self.stop.is_pressed:
```

```python
                self.display.display_text(["Machine stopped.", "STOP is pressed.", "", "Bye, welcome to drink Next time !!"])
                sys.exit()
            if self.weight_water():
                return True
            sleep_time = min(duration, interval)
            sleep(sleep_time)
            duration -= sleep_time
        return False

    def weight_water(self):
        """
        Measure the mass of water and display the current state.
        """
        self.mass_water = max(0, self.ws.weight(5))
        self.display.display_text([f"Brewing {self.mode_list[self.mode]} coffee", "Pumping water...", f" Time: {int(self.timer.elapsed())}", f"Mass of water: {int(self.mass_water)}g"])
        if self.mass_water >= self.K * self.mass_bean or self.next.is_pressed:
            return True
        return False

    def light_roast(self):
        """
        Execute the light roast coffee brewing process.
        """
        # Initial phase: Prepare the setup
        self.servo_motor.move_to_angle(0)
        self.gear_motor.set_speed(90)
        self.gear_motor.go()
        self.pump.go()
        if self.weight_water() or self.safe_sleep(5):
            return True
        self.pump.stop()

        # Second phase: Intermediate brewing stage
        self.servo_motor.move_to_angle(30)
        self.pump.go()
        if self.weight_water() or self.safe_sleep(5):
            return True
        self.pump.stop()

        # Third phase: Extended brewing period
        if self.safe_sleep(20):
            return True
        self.servo_motor.move_to_angle(0)
        self.pump.go()
        if self.weight_water() or self.safe_sleep(5):
            return True
        self.pump.stop()

        # Final phase: Finishing up
        self.servo_motor.move_to_angle(30)
        self.pump.go()
        if self.safe_sleep(3):
            return True
```

```
        self.servo_motor.move_to_angle(0)
        if self.safe_sleep(3) or self.weight_water() or self.safe_sleep(5):
            return True
        self.pump.stop()
        self.gear_motor.stop()
        return False

    def dark_roast(self):
        """
        Execute the dark roast coffee brewing process.
        """
        # Initial phase: Prepare the setup
        self.servo_motor.move_to_angle(0)
        self.gear_motor.set_speed(-90)
        self.gear_motor.go()
        self.pump.go()
        if self.weight_water() or self.safe_sleep(5):
            return True
        self.pump.stop()

        # Second phase: Intermediate brewing stage
        self.servo_motor.move_to_angle(30)
        self.pump.go()
        if self.weight_water() or self.safe_sleep(5):
            return True
        self.pump.stop()

        # Third phase: Extended brewing period
        if self.safe_sleep(20):
            return True
        self.servo_motor.move_to_angle(0)
        self.pump.go()
        if self.weight_water() or self.safe_sleep(5):
            return True
        self.pump.stop()

        # Final phase: Finishing up
        self.servo_motor.move_to_angle(30)
        self.pump.go()
        if self.weight_water() or self.safe_sleep(5):
            return True
        self.pump.stop()
        self.gear_motor.stop()
        return False

    def hot_water(self):
        """
        Dispense hot water.
        """
        self.servo_motor.move_to_angle(30)
        sleep(2)
        self.pump.go()
        sleep(15)
        self.pump.stop()
        self.servo_motor.stop()

```

```python
    def hot_rinse(self):
        """
        Perform a hot rinse to clean the system.
        """
        print("Start hot rinse")
        self.servo_motor.move_to_angle(0)
        sleep(2)
        self.pump.go()
        self.gear_motor.set_speed(-100)
        self.gear_motor.go()
        sleep(5)
        self.gear_motor.set_speed(100)
        self.gear_motor.go()
        sleep(5)
        self.servo_motor.stop()
        self.pump.stop()
        self.gear_motor.stop()

if __name__ == "__main__":
    logger.setup_logger("Test coffee")
    test_coffee = Coffee()
    test_coffee.hot_rinse()
```

C. Our Try with the STM32 as MCU

Figures 56, 57, and 58 illustrate our success in achieving signal transmission (high and low levels indicated by bright and dark LEDs), time counting, and OLED display control with the STM32.

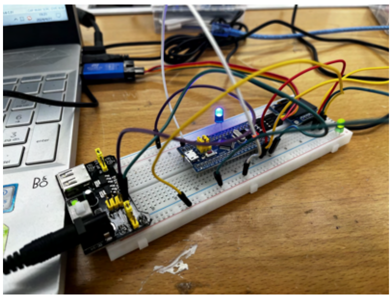

Figure 56: The STM32 successfully represents the signal on the breadboard, indicated by the lit LED.

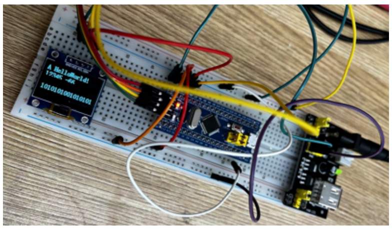

Figure 57: The STM32-controlled OLED display successfully shows the programmed strings.

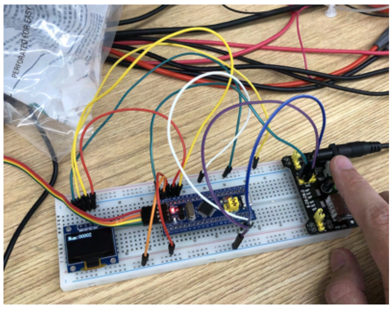

Figure 58: The STM32-controlled OLED display successfully shows the programmed time.