Smoothie Recipe Maker

Team 49 – Avyay Koorapaty, Anay Koorapaty, Max Gendeh

ECE 445 Project Proposal – Spring 2025

TA: Jason Zhang

Feb. 13th, 2025

Table of Contents

1 Introduction **2**
1.1 Problem 2
1.2 Solution 2
1.3 Visual Aid 3
1.4 High-level Requirements 4
2 Design **4**
2.1 Physical Design 4
2.2 Block Diagram 5
2.3 Subsystem Overview 5
2.3.1 Actuation 5
2.3.2 Sensors 5
2.3.3 Control 5
2.3.4 UI 5
2.3.5 Power 6
2.4 Subsystem Requirements 6
2.4.1 Actuation 6
2.4.2 Sensors 6
2.4.3 Control 6
2.4.4 UI 6
2.4.5 Power 6
2.5 Tolerance Analysis 6
3 Ethics and Safety **7**
4 References **8**

1 Introduction

1.1 Problem

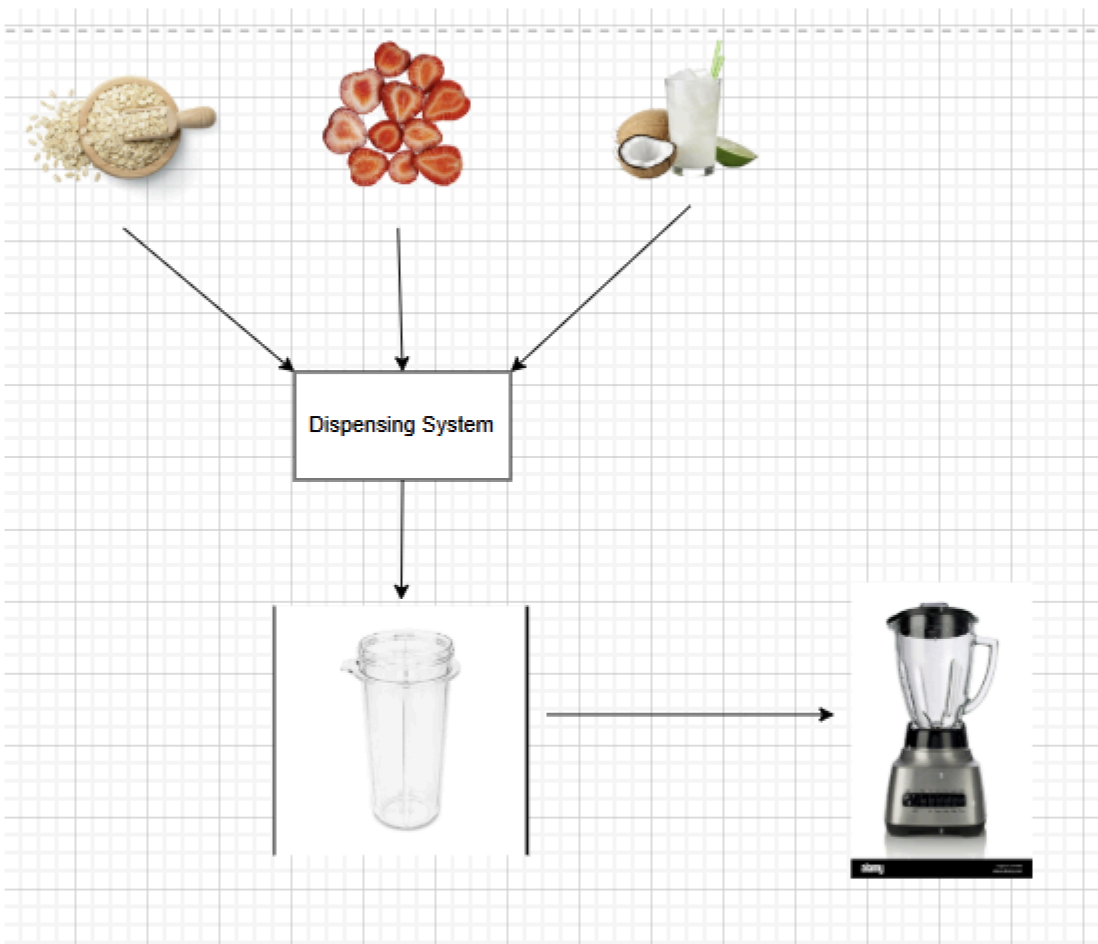
Making smoothies often requires measuring different ingredients with different measurement utilities. Liquid ingredients must be measured in mL, while solid ingredients are usually in units of cups or tablespoons. Additionally, the measurement steps are repeated for every subsequent smoothie made. To make the process more efficient, we will automate measuring and dispensing of ingredients. This will simplify creating smoothies with different recipes and reduce repetitive steps when creating more than one smoothie.

1.2 Solution

Our system will be able to make smoothies by following preset smoothie recipes or recipes the user creates. Our solution will be a three compartment structure, each compartment with one ingredient to dispense and a motor/solenoid valve for dispensing. The ingredients will be dispensed one at a time into a cup sitting on a load cell. The motors and solenoid valve are part of the Actuation Subsystem, and the load cell is part of the Sensors Subsystem. Complementing this, we will have software to control the motors incorporating the load cell measurements. This is the Control Subsystem. To be able to create different kinds of smoothies, there will be a UI for inputting recipes and selecting preset recipes. This is the UI Subsystem. These subsystems will work together to automatically dispense the appropriate amounts of ingredients into the cup for different recipes, making the ingredient measuring and placing process more efficient. The user can then empty the cup's contents into a blender.

We will use a microcontroller and the software we upload to it to control the motors and solenoid valve for proper dispensing. The microcontroller takes information from the load cell and the UI recipe to control the motors. The UI will have an LCD display and buttons to allow the user to select a preset recipe or input their own recipe.

1.3 Visual Aid

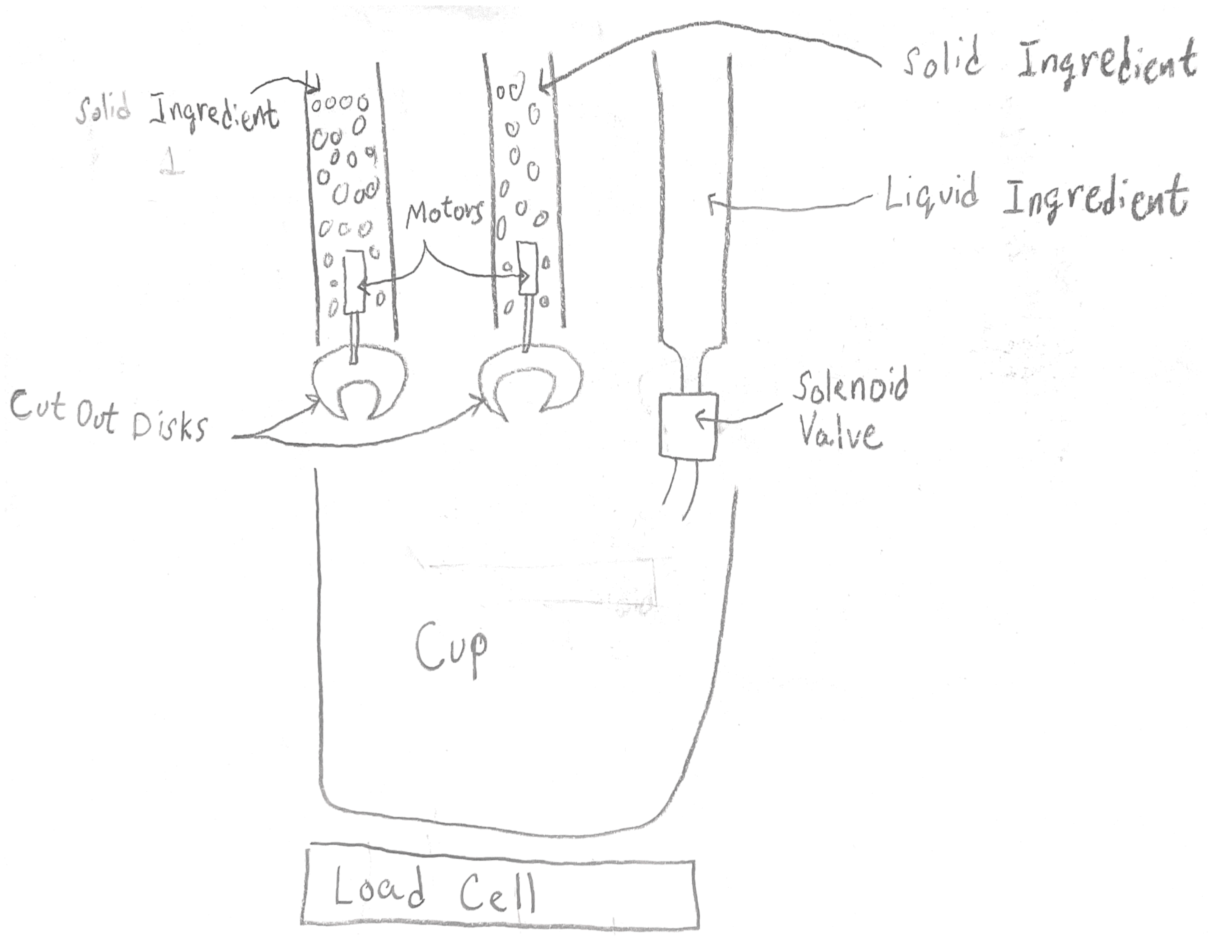


1.4 High-level Requirements

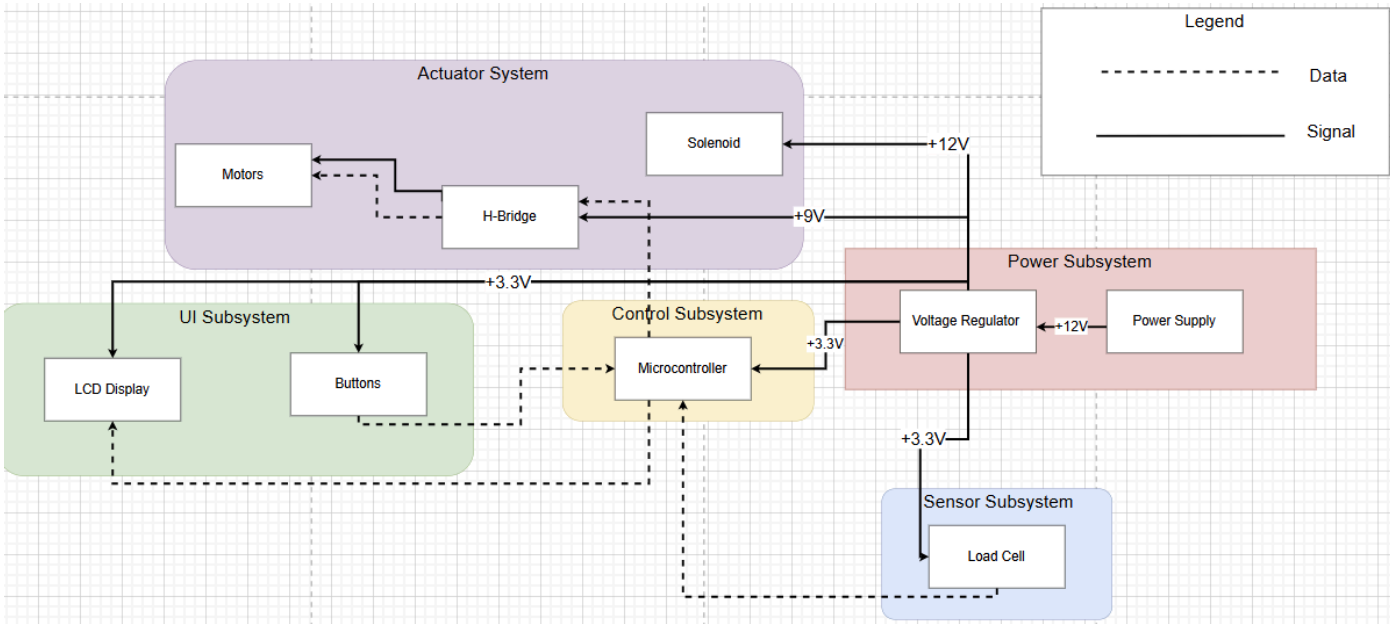
- The system takes a maximum of 3 minutes from the start of dispensing to all of the ingredients being inside the blender.
- Measurement accuracy will be to ±20% of intended ingredient amount.
- The system will dispense the correct ingredients given a preselected or user inputted recipe.

2 Design

2.1 Physical Design



2.2 Block Diagram



2.3 Subsystem Overview

2.3.1 Actuation

The Actuation Subsystem contains the stepper motors and solenoid valve that are used to dispense each ingredient into the cup. This subsystem is controlled by the Control Subsystem.

2.3.2 Sensors

The Sensors Subsystem contains the load cell, which is used to dispense the correct amount of each ingredient. This subsystem is an input to the Control Subsystem.

2.3.3 Control

The Control Subsystem contains the ESP32 microcontroller and interfaces with the UI and Actuation subsystems. The microcontroller receives inputs from the buttons and load-cell sensor and outputs signals to the H-bridge and LCD display.

2.3.4 UI

The UI Subsystem contains the buttons and LCD display. The buttons are used to select between recipes and input a new recipe. The LCD display provides visual aid to the user during this process, displaying the current recipe and ingredient amounts. The buttons are inputs to the Control Subsystem and

2.3.5 Power

The Power Subsystem contains a power supply and voltage regulator. It outputs power to all of the other subsystems.

2.4 Subsystem Requirements

2.4.1 Actuation

1. Motors must be able to rotate the disk between 0 and 180 degrees at different speeds, in order to control dispensing rate.
2. The solenoid valve must be able to control flow rate of the liquid ingredient to dispense the liquid in a controlled manner.

2.4.2 Sensors

1. Load cell must be at least precise to 10 gram units.

2.4.3 Control

1. Software should utilize load cell input to correctly send signals to motors and the solenoid valve.
2. Machine should correctly dispense the recipe inputted by users

2.4.4 UI

1. Button inputs must be correctly reflected on the LCD.
2. LCD should be able to display recipe name and each ingredient's amount in the recipe.

2.4.5 Power

1. Power supply supplies constant 12V.
2. Voltage regulator converts power supply voltage to 9V for the H-bridge and 3.3V for other components.

2.5 Tolerance Analysis

An aspect of the design that poses a threat to the functionality of our smoothie maker is the selection of our motors, particularly the torque the motor has relating to the rpm of the motor. This affects how fast we can turn the disk to stop the dispensing of ingredients. If it is too slow we will overshoot our ingredient measurements and if it is too fast we will undershoot our ingredient measurements. We will be using a stepper motor so we must calculate its rpm.

A stepper motor has a command pulse rate which moves the motor in steps and not continuously.

$$rev/second = (step/second)/(step/revolution.$$

$step/second = command\ pulse\ rate$
$step/revolution = 360/step\ angle$
$60 * (step/second)/(step/revolution) = rpm$
$C1 = motor\ circumference$
$C2 = disk\ circumference$
$C2/C1 = rotations\ of\ motor\ required$

Let us say we want to achieve a delay d = 0.2 seconds for the disk to turn and stop the flow of ingredients.

$(C2/C1)/(rpm/60) = d$
$rpm = 60C2/(C1d)$

We want to choose a stepper motor with the right command pulse rate and step angle to achieve and rpm, depending on the delay. As delay approaches 0, rpm -> infinity. We don't want to make our rpm too high, otherwise the disk could jam the machine. We have to balance the tradeoff between delay and rpm when making our choice for the stepper motor.

3 Ethics and Safety

We will address the ethical and safety issues with reference to the IEEE and ACM Codes of Ethics and relevant regulatory standards [1].

1. Ethical Guidelines

   Ethical guidelines require that we prioritize public safety and the well-being of users [1]. Because our project interacts directly with users, we want to provide full transparency of the capabilities of the machine. We aim to provide clear documentation, labeling intended use, limitations, and potential risks. An example is during an unlikely event when one of the ingredient chambers gets clogged, we'll let the user know on the LCD to try giving the machine a shake.

2. Safety Standards

   We take user safety extremely seriously. Blenders are known to be dangerous and our design ensures that our product will not interfere with the blending process. Food safety is another concern. We aim to provide that through temperature sensors that let users know for example when the temperature of milk has exceeded a safe temperature for storage. Materials in contact with ingredients will be food-grade and easy to clean to prevent contamination. Finally, with the elimination of any batteries, we minimize the risk of fires and electrical hazards.

3. Regulations

We will adhere to campus safety and lab policies throughout the development of this project. Our team has set aside time for weekly reviews and feedback on our design throughout the project's lifecycle. With this, we will be able to collect valuable insight and feedback to ensure our project stays within relevant regulations and standards.

4. Respect and Compliance

   Our team is committed to developing an environment of respect and ethical awareness. All members will treat all users fairly and hold each other accountable for any outcomes our project may have. Through this, we hope there will be efficient teamwork and good progress throughout the project lifecycle.

4 References

[1] IEEE-CS. "Code of ethics," IEEE Computer Society, https://www.computer.org/education/code-of-ethics (accessed Feb. 12, 2025).

[2] Oats-Oatmeal-PNG-Images.png (1200×1200)(Oats)

[3]diced strawberries png - Search(Diced strawberry)

[4] blender-D93N9B.jpg (849×1390)(blender)

[5] How To Determine The RPM On Stepper Motors | Sciencing