Smart Snack Dispenser (SSD)

ECE 445: Senior Design Laboratory

Project Proposal

Team No.23

Eric Nieto Gonzalez || Elinor Simmons || Adam Kramer

(enieto5@illinois.edu)||(elinors2@illinois.edu)||(adamlk2@illinois.edu)

TA: Surya Vasanth

Professor: Yang Zhao

February 12th, 2025

Table of Contents

1. Introduction

1.1 Problem

One common problem many people face is difficulty in controlling snack portions, which can lead to overeating and unhealthy eating habits. Mindless snacking, especially when working, studying, or watching TV, often results in consuming more than intended. Similarly, there seem to be no machines handling this issue within the current market, leaving individuals to rely on willpower alone or resort to ineffective portioning methods such as manually separating snacks into smaller bags. Without a structured approach, people often struggle to regulate their intake, leading to issues such as weight gain, unhealthy eating patterns, and difficulty in maintaining a balanced diet.

1.2 Solution

The smart snack dispenser addresses this issue by allowing users to set portion sizes and control snack intake. By offering a structured approach to snacking, it helps users develop healthier eating habits, prevent overindulgence, and manage calorie intake more effectively. This solution is particularly beneficial for individuals trying to maintain a balanced diet and/or are tracking their food intake. The machine will offer a specific set of 3 snacks. This includes M&M's, Almonds, and Goldfish . The machine will also plug into a wall outlet and serve as a home appliance.

1.3 Visual Aid

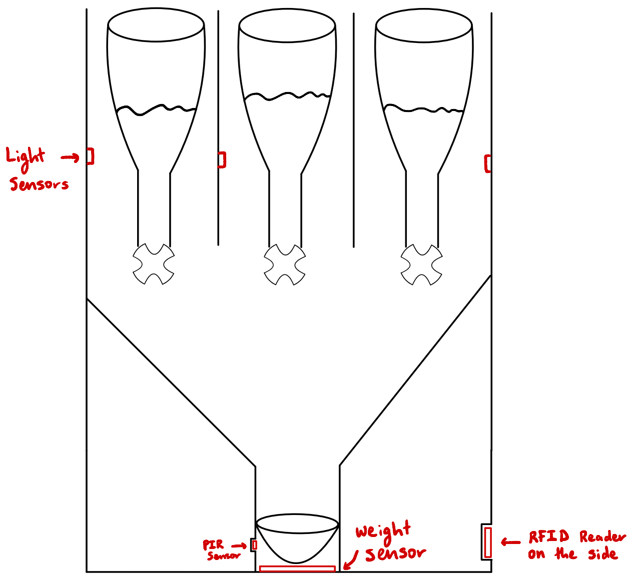


Figure 1: Visual Aid Internal View

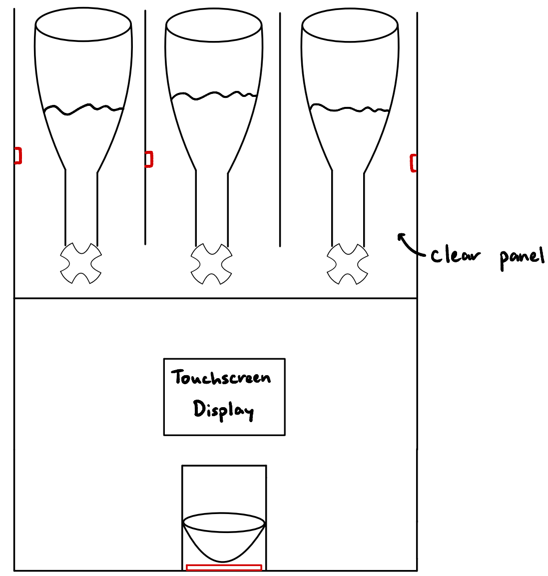


Figure 2: Visual Aid Front View

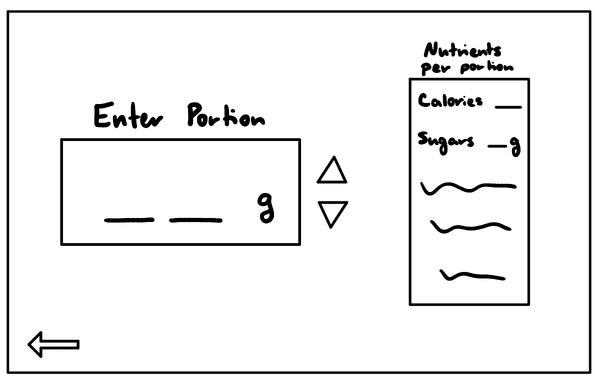


Figure 3: Example of Portion Selection Screen

The dispenser will have overall dimensions of 24 [in.] x 16 [in.] x 22 [in.] (L x H x W), housing all internal components efficiently. Each snack storage compartment will measure 7 [in.] x 8 [in.] x 10 [in.], providing a total storage capacity of 560 cubic inches (equivalent to 320 oz). Additionally, the bowl input will have dimensions of 5 [in.] x 5 [in.] x 5[in.], ensuring a suitable space for snack collection. This sizing ensures efficient snack organization while keeping the dispenser compact and suitable for various settings.

1.4 High Level Requirements

For this project to be considered successful, specific parameters and measurable goals must be met. These goals define the necessary performance benchmarks and ensure objective evaluation of the outcomes:

- Accuracy: For our worst-case scenario (minimum amount dispensed), the machine must dispense the correct portion within a 15% tolerance. The machine must also correctly keep track of the nutrients that are in each portion.
- Speed: The machine must dispense the snack within 4 seconds or less after the snack has been selected. It also must register user input immediately as it is selected on the touchscreen.
- Usability: The UI must be smooth and organized. It should be easy for the user to dispense a snack or find their personalized data.

2. Design

2.1 Block Diagram

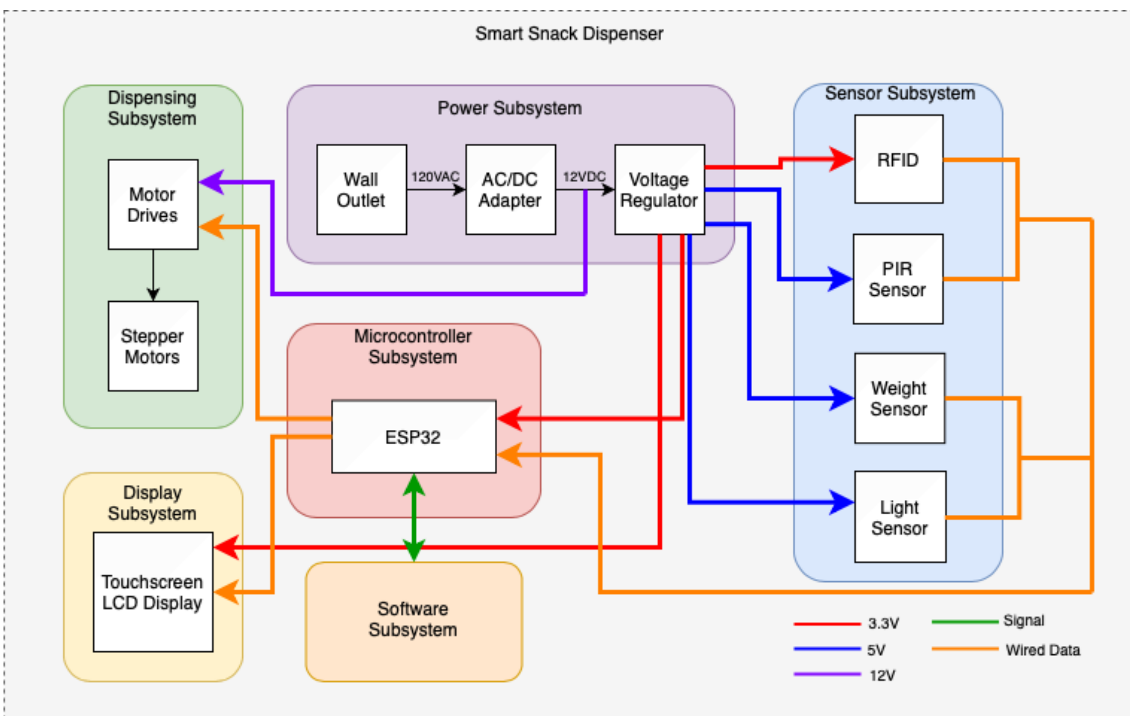


Figure 4:Block Diagram Representing Subsystems

2.2 Subsystem Overview

2.2.1 Dispensing Subsystem

The dispensing subsystem is responsible for accurately and efficiently delivering snacks to the user while preventing jams or over-dispensing. The dispensing subsystem will be driven by three stepper motors, one for each snack, which are driven by motor drives controlled by the microcontroller. Each stepper motor will have an attached rotary fin dispensing wheel which rotates so that the snack falls between the fins in the wheel and then gets dumped into the weighing area. The wheel's fins should be spaced out large enough to allow ideally 2 pieces for M&M's and almonds to be dispensed at a time, and 4 pieces for goldfish. The minimum and maximum amount the user can dispense is 15 grams and 100 grams. The snacks should be stored in separate hoppers which each feed into a small opening that their respective wheels can grab from.

2.2.2 Microcontroller Subsystem (ESP32)

The ESP32 microcontroller serves as the central hub for processing inputs from the sensor subsystem and controlling the outputs of the snack dispenser. It is responsible for receiving data from various sensors and making real-time decisions based on this information. For example, when an RFID scan is detected, the ESP32 verifies the user's profile and retrieves their personalized snack settings to be shown in the LCD display. If the PIR sensor confirms that a tray is present, the microcontroller signals the motor subsystem to dispense the selected snack. The weight sensor ensures accurate portion control by sending data to the ESP32, which adjusts the motor speed accordingly. Additionally, the light sensor will alert the microcontroller to send a low stock notification to the display. The ESP32 also manages the software subsystem, including the lockout system and snack recommendations, ensuring seamless operation and an intuitive user experience. By coordinating all these processes efficiently, the ESP32 enables the smart snack dispenser to function reliably and meet user needs effectively.

2.2.3 Sensor Subsystem (RFID, PIR Sensor, Weight Sensor, Light sensor)

The sensor subsystem features four different sensors: the RFID to allow each user to have their own profile, the PIR sensor to make sure a bowl is present before dispensing, the weight sensor to measure the desired portion, and the light sensor to monitor stock level. The sensors in the smart snack dispenser must accurately receive data from the environment and relay it to the microcontroller subsystem, which will determine the appropriate responses for snack dispensing and user interaction. These sensors are crucial for the microcontroller to process input data effectively, but they are even more critical for ensuring the overall functionality of the machine. If the RFID sensor fails, the system may not recognize individual users, preventing access to personalized snack settings. If the PIR sensor does not detect a tray, snacks could be dispensed improperly and end up wasted. A malfunctioning weight sensor could lead to incorrect portion sizes, undermining the machine's ability to regulate snack intake. Lastly, if the light sensor fails to monitor snack levels correctly, it will prevent the user from getting their desired amount if they do not have enough on hand to refill. Ensuring that these sensors operate correctly is essential for maintaining a smooth, efficient, and reliable snack dispensing experience.

2.2.4 Touchscreen LCD Display Subsystem

The touchscreen LCD display serves as the primary interface for users to interact with the smart snack dispenser, providing real-time updates and controls. It is directly linked to the ESP32 microcontroller, which processes inputs from various sensors and delivers relevant information to the display. When a user scans their RFID tag, the LCD retrieves and displays the user's personal settings and nutrition tracking when necessary. The PIR sensor ensures a tray is present before dispensing, and if not detected, the screen alerts the user with a prompt. As snacks are dispensed, the weight sensor provides live portion measurements, which are displayed on the screen to confirm accuracy. Once the snack level goes below the light sensor, a notification will be shown on the LCD to alert the user to refill soon. Additionally, the LCD plays a key role in the lockout system and recommendation feature, informing users if they have reached their snack limit and suggesting alternative options. By integrating all these functionalities, the touchscreen LCD enhances usability, making the snack dispenser both intuitive and efficient for users managing their dietary habits.

2.2.5 Software Subsystem

The software subsystem contains all of our internal features. This includes having two modes. The main mode will include the lockout feature and snack recommendation system. It will also keep track of the nutrition in each dispensed portion and the total daily intake. There will also be a casual mode that will not have the lockout system activated and will allow the user to dispense any snack they want without tracking the nutrients. The lockout system will be used to help with portion control. The user will set a daily limit for a nutrient of their choice (calories, sugar, etc.) and once that limit is hit, the machine will prevent the user from dispensing anymore snacks that day. If the user is close to their limit and tries to dispense a portion that will go over their limit, the machine will ask the user to select a smaller portion or recommend a different snack. The machine will also offer recommendations for portion size in order to help with moderation. This subsystem will also contain our database that holds all of the nutrition information for each snack.

2.2.6 Power Subsystem (AC/DC adapter and voltage regulators)

The power subsystem is required to power all of our electrical components, such as the stepper motors and microcontroller. This will require an AC/DC adapter that will convert the 120V from the wall outlet to 12V. We will have several voltage regulators to adjust the voltage to the needed level for our microcontroller, stepper motors, and sensors.

2.3 Subsystem Requirements

2.3.1 Dispensing Subsystem

- The stepper motors should rotate with enough speed and torque to balance the speed and accuracy of the dispensing process.
- The feeder should be sized to dispense a given snack at a consistent rate while monitoring the weight on the scale.
- The snacks should be stored in hoppers which funnel each snack into a small hole where they may be dispensed by the rotating feeder underneath the hopper.

2.3.2 Microcontroller Subsystem

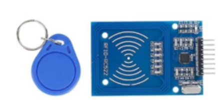
- The *ESP32 microcontroller* should activate the appropriate stepper motor when a user selects a snack, driving the rack and pinion system to push the snack stock over a hill for controlled dispensing.
- The microcontroller should verify that the PIR sensor detects a tray before releasing a snack, preventing unintended spills.
- Upon scanning an RFID tag, the microcontroller should retrieve the user's profile and display their snack preferences, nutritional goals, and history on the touchscreen LCD display.
- The microcontroller should adjust the speed of the motors accordingly as the machine gets closer to the desired amount.
- The microcontroller should trigger the motors when signaled by the light sensor and stop once the stock has reached the desired position.
- The microcontroller should output a notification to refill once the rack has reached the end of the stock container.
- The microcontroller should enforce dietary restrictions by activating the lockout system if a user reaches their daily snack limit, preventing further dispensing and suggesting alternatives through the recommendation system.

2.3.3 Sensor Subsystem

- The *HX711 load cell amplifier* with a *500g strain gauge weight sensor (SEN-14728)* should measure the dispensed snack's weight, ensuring accuracy within ±5% tolerance of the serving size.
- The *RCWL-0516 PIR sensor* should detect whether a tray is present before dispensing a snack and alert the microcontroller if not.

- The *TEPT5700 ambient light sensor* should monitor the snack levels and alert the microcontroller when the detected light level increases beyond a preset threshold.

- The *MFRC522 RFID reader* should scan user tags and relay individual snack preferences, calorie limits, and nutritional goals to the microcontroller for personalized snack dispensing.



2.3.4 Touchscreen LCD Display Subsystem

- The *4.0-inch TFT LCD Touchscreen Display (MSP4022)* allows users to select snacks, specify portion sizes, and view nutritional information, offering an interactive and intuitive experience.

- Displays daily snack intake, compares it to set goals, and provides alerts when the user approaches or exceeds their calorie limit.
- Users can toggle between Casual Mode for unrestricted snacking and Main Mode for tracking calorie intake and managing nutrition.
- Notifies users when snacks are low or when maintenance is needed, ensuring smooth operation and user awareness.

2.3.5 Software Subsystem

- The lockout system activates only in the main mode and once the calorie limit is reached.
- The recommendation system provides a snack that accurately represents the user's response to the question.
- The software subsystem keeps track of the correct amount of nutrients in each portion and keeps a running total for the day.

2.3.6 Power Subsystem

- The voltage regulators should provide the correct voltage for the respective component.
- The system must be able to provide the necessary current for each component, while maintain the correct voltage.

2.4 Tolerance Analysis

For our Smart Snack Dispenser, we primarily care about the error involved in dispensing an accurate weight of a snack to the user. This can be broken down into two primary concerns: the tolerance of the sensors themselves, and the tolerance of our system's ability to provide as close to the ideal number of pieces of snack in our dispensing wheel each time. The load cell that we've chosen has a 0.05% error, so for 500g that is a 0.25g error. We are defining our worst-case scenario to be at the smallest amount we can dispense (15g) and to include the maximum error from the load cell. This is because errors are amplified at smaller amounts. For M&M's and almonds we would like the machine to ideally dispense 2 pieces at a time. We aren't too worried about the machine dispensing less than two pieces at a time because it can always turn more to dispense more. However, we are concerned about the machine dispensing more than two at a time. We think it's reasonable to assume that the dispenser will be designed well enough so that only a maximum of one extra piece could fit. We are basing this assumption on the fact that we will design the dispensing wheel and repeatedly test it to make the necessary changes needed to get the quantity that we desire. This would mean that it's possible for the machine to dispense three pieces instead of two. One M&M weighs about 0.91 grams. In the scenario that the machine has dispensed 15 M&M's (this would be about 13.65 grams) and it decides to dispense one more time to get closer to the ideal amount, there is a chance three pieces could come out during this time. We can repeat this same scenario, but with the 11 almonds (this would be about 13.2 grams with one almond weighing about 1.2 grams). For goldfish, we are going to dispense 4 pieces at a time, so in the scenario that 24 pieces have been dispense (about 13.104 grams at 0.546 grams per piece), we would be worried about the machine dispensing 5 more pieces instead of 4.

$$M\&M's: \left|\frac{\left((18)(0.91)+0.25\right)-15}{15}\right| * 100 = 10.87\%$$

$$Almonds: \left|\frac{\left((14)(1.2)+0.25\right)-15}{15}\right| * 100 = 13.67\%$$

$$Goldfish: \left|\frac{\left((29)(0.546)+0.25\right)-15}{15}\right| * 100 = 7.23\%$$

We can see that when comparing each scenario, almonds are the worst-case scenario. Since the error is 13.67%, we believe a tolerance of 15% is fair for our worst-case scenario. We believe this would only improve for larger amounts. This tolerance also accounts for the largest load cell error, which we believe would be unlikely for such a small amount. We also used the average weights per piece on the higher side to over design our tolerance.

M&M’s (~0.875 g each)

Max storage suggestion: 62 oz or about 1757.67 g (big container of m&m’s from Costco)

- (min amount dispensable) 15 g of m&m’s is ideally 17-18 m&m pieces (14.875-15.75 g)
    - About ½ serving and 75 kcal
- (max amount dispensable) 100 g of m&m’s is ideally 114-115 m&m pieces (99.75-100.625 g)
    - About 3.5 servings or 500 kcal

Almonds (~1.15 g each)

Max storage suggestion: 3 lbs or about 1360.78 g (big container of almonds from Costco)

- (min amount dispensable) 15 g of almonds is ideally 13-14 almonds (14.95-16.1 g)
    - About ½ serving, 85 kcal
- (max dispensable amount) 100 g of almonds is ideally 86-87 almonds (98.9-100.05 g)
    - About 3.3 servings, 566.67 kcal

Goldfish (~0.545 g each)

Max storage suggestion: 66 oz or about 1871.07 g (big container of goldfish from Costco)

- (min amount dispensable) 15 g of goldfish is ideally 27-28 goldfish (14.73-15.27 g)
    - About ½ serving, 70 kcal
- (max amount dispensable) 100 g of goldfish is ideally 183-184 goldfish (99.81-100.36 g)
    - About 3.3 servings, 466.67 kcal

3. Ethics & Safety

3.1 Ethics

The smart snack dispenser raises ethical concerns related to user privacy, data security, and autonomy. Since the system tracks individual snacking habits and nutritional data through RFID, it is essential to comply with ethical standards outlined by organizations such as the *IEEE Code of* Ethics and the *ACM Code of Ethics and Professional Conduct*. *IEEE's Principle 1* emphasizes the need to prioritize the well-being and privacy of individuals, ensuring that users' personal information remains secure and is not shared without explicit consent. Strong encryption, secure authentication, and strict access control measures must be implemented to prevent unauthorized access to user data.

Additionally, the dispenser's lockout system, which restricts snack intake based on preset calorie limits, introduces ethical concerns regarding user autonomy. In alignment with *ACM's General Ethical Principles*, particularly *Principle 1.4 (Respect Privacy)* and *Principle 1.6 (Respect User Autonomy)*, the system should promote healthier habits without imposing excessive restrictions. Users must retain the ability to override or customize their limits, ensuring that the feature is supportive rather than coercive.

Furthermore, *ACM's Principle 2.9 (Ensure Fairness and Non-Discrimination)* emphasizes the responsibility of technology designers to create systems that are fair and accessible to all users. The smart snack dispenser should be designed to prevent unintentional bias, ensuring that all individuals, regardless of their dietary restrictions, cultural preferences, or physical abilities, can use the system effectively. This includes offering a variety of snack options that cater to different nutritional needs, designing an intuitive interface that accommodates diverse users, and ensuring that the system does not unintentionally disadvantage any particular group. By integrating fairness into the design, the dispenser can serve as an inclusive and equitable tool for promoting healthier snacking habits.

3.2 Safety

Safety is a critical consideration in the design of the smart snack dispenser, particularly regarding electrical, mechanical, and food hygiene risks. The *IEEE Code of Ethics Principle 1* states that engineers must "hold paramount the safety, health, and welfare of the public," which directly applies to preventing hazards associated with electrical and mechanical components.

- Electrical Safety: Since the dispenser plugs into a wall outlet, it must comply with industry standards for insulation, grounding, and circuit protection to prevent electrical shocks, short circuits, and overheating. Overcurrent protection and temperature monitoring should be included to mitigate fire hazards.
- Mechanical Safety: The motorized dispensing system should be designed to minimize risks of mechanical failure that could lead to snack jams or unintended dispensing. This is particularly important in preventing choking hazards, especially for small children. Safety mechanisms such as emergency stop features, or child-lock modes can help mitigate these risks.
- Food Safety: The design must ensure that snacks remain uncontaminated. In accordance with *ACM's Principle 1.2 (Avoid Harm),* the dispenser should minimize food exposure to external contaminants. The use of food-grade materials, airtight containers, and easily cleanable surfaces will help maintain hygiene. Users should also receive clear maintenance and cleaning guidelines to prevent bacterial buildup or cross-contamination.

By adhering to IEEE and ACM ethical and safety standards, the smart snack dispenser can be designed as a secure, user-friendly, and responsible solution that prioritizes privacy, accessibility, and well-being.

4. References

[1] *Arduino.cc*, 2025. https://docs.arduino.cc/retired/getting-started-guides/TFT/, Accessed Feb. 13, 2025.

[2] Dejan, “Arduino Touch Screen Tutorial | TFT LCD,” *How To Mechatronics*, Dec. 07, 2015. https://howtomechatronics.com/tutorials/arduino/arduino-tft-lcd-touch-screen-tutorial/, Accessed Feb. 11, 2025.

[3] Dejan, “DIY Vending Machine - Arduino based Mechatronics Project,” *HowToMechatronics*, Nov. 13, 2017. https://howtomechatronics.com/projects/diy-vending-machine-arduino-based-mechatronics-project/, Accessed Feb. 13, 2025.

[4] Espressif, “ESP32 Series Datasheet Including,” 2024. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf, Accessed Feb. 13, 2025.

[5] “IEEE Code of Ethics,” IEEE, Sept. 19, 2024, https://www.ieee.org/about/corporate/governance/p7-8.html, Accessed Feb. 13 2024.

[6] Instructables, “Build Custom ESP32 Boards From Scratch! | the Complete Guide to Designing Your Own ESP32-S3 and C3 | Full Tut…,” *Instructables*, Feb. 24, 2024. https://www.instructables.com/Build-Custom-ESP32-Boards-From-Scratch-the-Complet/, Accessed Feb. 11, 2025.

[7] Mayo Clinic Staff, “Counting calories: Get back to weight-loss basics,” *Mayo Clinic*, 2018. https://www.mayoclinic.org/healthy-lifestyle/weight-loss/in-depth/calories/art-20048065, Accessed Feb. 13, 2025.

[8] “Smart Snacks Product Calculator,” *Healthiergeneration.org*, 2016, https://foodplanner.healthiergeneration.org/calculator/, Accessed Feb. 13, 2025.