An-Najah National University

Faculty of Engineering & Information Technology

Computer Engineering Department

Presented in partial fulfilment of the requirements for Bachelor degree in Computer Engineering

Graduation Project 2

REFRESHG

Students:

Amer Kobari

Tariq Marmash

Supervisor:

Dr. Hanal Abu Zant

May 25, 2025

Acknowledgement

“

We would like to start by thanking Dr. Hanal Abu Zant, our supervisor, for all his help and support throughout this project. His guidance, advice, and feedback were very important in making our work better. We are really thankful for the time and effort he gave us. We also want to thank our friends for always being there for us. Your encouragement, help, and teamwork made this project much easier and more enjoyable. We truly appreciate your patience and support. A big thank you goes to our families for their love, care, and constant support. You believed in us from the start and helped us stay strong, even when things got tough. Your love and sacrifices gave us the motivation we needed to keep going. Finally, we are grateful to everyone who helped us with this project in any way. Whether it was through advice, assistance, or just being there for us, your support means a lot, and we truly appreciate it.

”

-Amer, Tariq

Disclaimer

This report was written by Amer Kobari and Tariq Marmash at the Computer Engineering Department, Faculty of Engineering, An-Najah National University. It has not been altered or corrected, other than editorial corrections, as a result of assessment and it may contain language as well as content errors. The views expressed in it together with any outcomes and recommendations are solely those of the students. An-Najah National University accepts no responsibility or liability for the consequences of this report being used for a purpose other than the purpose for which it was commissioned.

Table of Contents

List of Figures

1 Abstract

The goal of the RefreshGo project is to create an automated mojito vending machine powered by an Arduino Mega, integrated with sensors, motors, and various electronic components that will be discussed later in the report. The machine functions like a small-scale beverage factory, where a cup is first dispensed and then moved on a conveyor belt through a series of stages, including ice dispensing, flavor mixing, and soda or energy drink addition. The user can select from multiple flavors such as Strawberry, Apple, and Raspberry Blue, each managed by independent pumps for accurate dispensing. RefreshGo supports three methods of ordering: directly through physical push buttons on the machine, remotely via a mobile application developed using Blynk and connected through an ESP32, and uniquely, through an ESP32-CAM system that captures an image of a preordered or previously prepared drink, analyzes its visual features such as color, and replicates the drink composition accordingly using onboard processing. Payment is handled using RFID technology, allowing customers to tap their card for quick and seamless transactions. The system is designed to maximize efficiency, reduce manual labor, and provide precise, customizable mojito preparation in a fully automated format. This makes it ideal for implementation in modern cafes, events, and entertainment venues. While automated drink machines already exist, RefreshGo stands out by combining ingredient customization, remote control, RFID payment, and visual drink recognition into one cohesive and intelligent solution. This project introduces a level of personalization and automation that is not currently available in existing beverage vending systems.

2 Introduction

2.1 General Background

As individuals, we often experience delays and inefficiencies when ordering beverages, especially in busy environments like cafes, events, or entertainment venues where preparing a customized mojito can be time-consuming and labor-intensive. RefreshGo addresses this challenge by introducing an automated mojito vending system. This innovative solution offers a fast, efficient, and fully automated process for preparing high-quality, customized mojitos on demand. By integrating advanced hardware with software technologies such as flavor control, RFID payment, mobile app ordering, and image-based drink replication, RefreshGo transforms traditional drink preparation into a streamlined, precise, and user-friendly experience. This project redefines how

beverages are served in high-demand settings, ensuring convenience, personalization, and consistency for every customer.

2.2 Problem

In cafes, events, and public venues, preparing customized mojitos manually often leads to delays, especially during busy periods. Customers frequently experience long wait times due to the step-by-step nature of drink preparation, which can cause errors and inconsistencies in the final product. Traditional methods also make it difficult to handle multiple customized drink orders efficiently, frustrating customers and placing pressure on staff. Our project, RefreshGo, solves these issues by automating the entire mojito-making process. From dispensing the cup to adding ice, flavors, and water, the machine offers a smooth and rapid experience for both users and operators. With multiple ordering methods—including buttons, mobile app, and image recognition—the system improves service speed and order accuracy.

2.3 Objectives

To address these challenges, RefreshGo delivers a complete automated solution for preparing and customizing mojito drinks. The system manages the full process—starting from cup dispensing, through flavor and ingredient addition, to final delivery—without the need for manual labor. Users can customize their mojito using onboard push buttons or a mobile app connected via ESP32 and Blynk. Additionally, the system features an innovative ESP32-CAM module that can replicate a drink based on a photo by analyzing its color. An RFID payment system allows for quick, contactless transactions. The primary goal of this project is to enhance beverage service efficiency, reduce human error, and increase the speed of personalized drink preparation. RefreshGo is designed to improve customer satisfaction and support high-demand environments where fast and customizable service is essential.

2.4 Scope of the Project

The RefreshGo project is designed for implementation in cafes, entertainment venues, event spaces, and other businesses seeking to modernize beverage services through automation. It appeals to customers who prioritize speed, customization, and convenience when ordering drinks. The scope involves designing and building a fully automated mojito vending machine that integrates mechanical components, RFID systems, user-friendly interfaces, mobile app control, and camera-based drink recognition. It is suitable for a broad customer base, ranging from busy individuals to groups at large public gatherings, and aims to perform reliably in environments with high customer turnover.

2.5 Importance

The development of RefreshGo represents a major innovation in the beverage service industry. By automating the mojito preparation process, the project reduces labor costs, minimizes human error, and increases customer satisfaction through faster, more precise, and personalized service. During peak hours or crowded events, it allows businesses to handle more orders without compromising quality. Its ability to replicate drinks based on images further distinguishes it from traditional vending solutions. As automation continues to shape the future of food and beverage services, RefreshGo offers a forward-looking, scalable solution that meets modern expectations of speed, customization, and convenience.

2.6 Report Organization

This report outlines the development of the RefreshGo project, including:

- An introduction to the project and its objectives.
- A discussion of the constraints and challenges faced during the development process.
- A literature review exploring existing solutions in automated drink machines.
- The methodology used to design and implement the system.
- An analysis of outcomes, data gathered, and performance metrics.
- A discussion of limitations and areas for future improvement.

Finally, the report concludes with recommendations, references, and resources that supported the project's successful development.

3 Constraints, Standards, and Earlier Course Work

3.1 Constraints

3.1.1 Cost and Budget Constraint

Managing the budget of the RefreshGo project while ensuring all essential features were included posed a major challenge. The cost of electronic components—such as the Arduino Mega, ESP32, stepper motors, pumps, and RFID modules—was high. Additionally, repeated testing with ingredients like flavors and ice led to wastage and accidental damage to components, increasing overall expenses. These financial constraints limited our ability to experiment with advanced features or improve the structural design beyond core functionality in order to stay within budget.

3.1.2 Project Structure Constraints

Designing the mechanical structure of the mojito machine was difficult due to our limited experience with mechanical layout planning. We encountered multiple issues related to the precise positioning of components such as the cup dispenser, flavor pumps, conveyor belt, and stepper motors. This led to repeated visits to the carpenter for adjustments and corrections, resulting in time loss and resource strain. Additionally, limited access to university tools and dependency on external workshops made it harder to make structural improvements quickly.

3.1.3 Time Duration Constraint

Although the RefreshGo project was scheduled across the full semester, several time-related challenges affected our ability to work consistently. The semester included multiple breaks such as New Year's and other official holidays, which significantly reduced our available working days. Additionally, occasional city closures forced us to shift to online communication, limiting hands-on progress on the hardware aspects of the project. We also had to juggle the project alongside other academic responsibilities, including final exams and deadlines from other courses. Furthermore, one of our team members began an internship in Tunis during the latter part of the semester, which required us to speed up our project discussions and decision-making. This time pressure meant we had to finalize certain stages earlier than planned, limiting the window for testing and refinement.

3.2 Standards

When building RefreshGo we followed a couple of standards and practices to achieve better functionality and ease of understanding the code.

3.2.1 Programming Language and IDE

The software for RefreshGo was developed using the Arduino IDE, which is based on C/C++. This platform was used to write and upload code to both the Arduino Mega and the ESP32. The Arduino controlled components such as pumps, stepper motors, and sensors, while the ESP32 handled wireless communication with the mobile app and ESP32-CAM. We also utilized Blynk libraries to simplify cloud integration and real-time communication with the app, creating a seamless end-to-end operation flow.

3.2.2 Application Development

To enable remote ordering, we used the Blynk platform for mobile application development. Blynk allowed us to create a simple and intuitive interface for users to select mojito flavors and place orders via their smartphones. This application communicates with the ESP32 module over Wi-Fi and forwards order details to the Arduino. Additionally, the system supports real-time

status updates and efficient control. Blynk's cloud-based infrastructure ensured stable and responsive communication throughout the development and testing phases.

3.3 Earlier Course Work

The development of RefreshGo was deeply supported by our previous coursework in computer and electronics engineering. Courses on microcontrollers, embedded systems, and digital electronics provided a solid foundation in using components like the Arduino Mega, stepper motors, LCDs, and sensors. Labs and projects gave us hands-on experience that proved critical when working on the hardware integration. The microcontroller lab was especially useful, as it taught us how to program and interface with real-world components. Furthermore, we participated in an IEEE Arduino training course, where we gained practical knowledge in system design and project development using Arduino and ESP32 boards. These academic experiences helped us understand the technology behind RefreshGo and equipped us to tackle complex integration challenges during the project.

4 Literature Review

Automated food and beverage systems have significantly evolved from their early beginnings, transitioning from simple manual operations to advanced, highly automated solutions that leverage modern technologies. These advancements have led to faster, more precise, and more efficient preparation processes, capable of handling complex tasks such as product customization, portion control, and real-time user interaction. This shift toward automation began accelerating in the late 20th century and continues to transform the food and beverage industry. The demand for intelligent beverage vending systems like RefreshGo reflects a broader trend toward user-centered, on-demand service experiences.

The growing interest in automated drink preparation is driven by customer expectations for speed, personalization, and convenience—especially in public settings such as cafes, events, commercial spaces, and entertainment venues. Traditional manual preparation struggles to meet these demands during peak times, leading to inefficiencies and inconsistencies. Automated systems, in contrast, enable rapid and uniform service, reducing human error and improving hygiene standards. The global food and beverage automation market has experienced consistent growth, largely due to its ability to reduce labor costs and increase operational scalability.

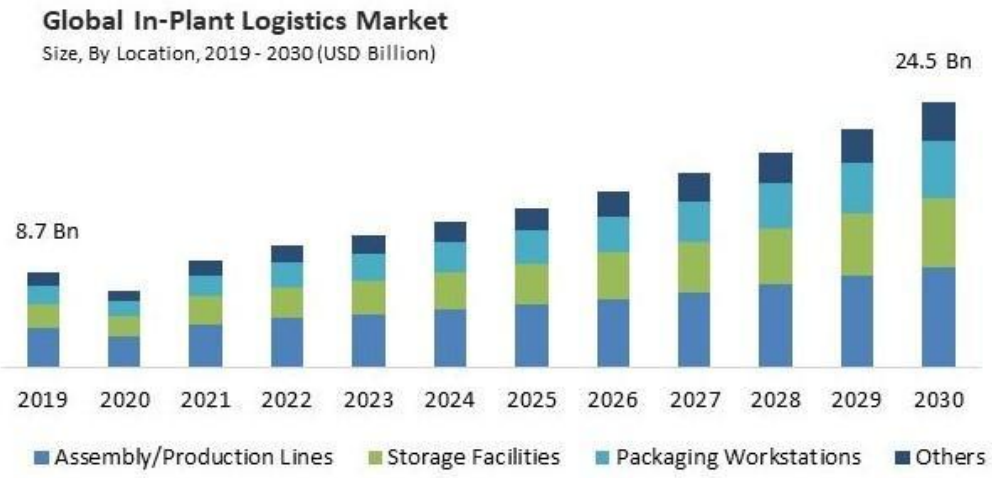

Figure 4.1 Production lines Logistics Market

In recent years, technologies such as the Internet of Things (IoT), Artificial Intelligence (AI), and smart sensors have become integral to the design of modern vending machines and robotic kiosks. These tools enable features like real-time ingredient tracking, customer interaction via mobile apps, and cashless payments through RFID or facial recognition. RefreshGo incorporates several of these technologies—using ESP32 for wireless communication, Blynk for app integration, and ESP32-CAM for visual recognition—to offer a complete drink preparation experience. The addition of image-based drink replication through color analysis further differentiates RefreshGo from standard machines, allowing users to recreate drinks based on appearance alone.

Automated drink machines now serve diverse environments, including corporate offices, restaurants, event venues, and public spaces, offering fast, scalable, and customizable service. These solutions are not only improving customer satisfaction but also reshaping how beverage services operate in today's fast-paced and digitally connected world. RefreshGo fits squarely into this evolving landscape, contributing to the advancement of compact, intelligent drink production systems designed for high efficiency and maximum personalization.

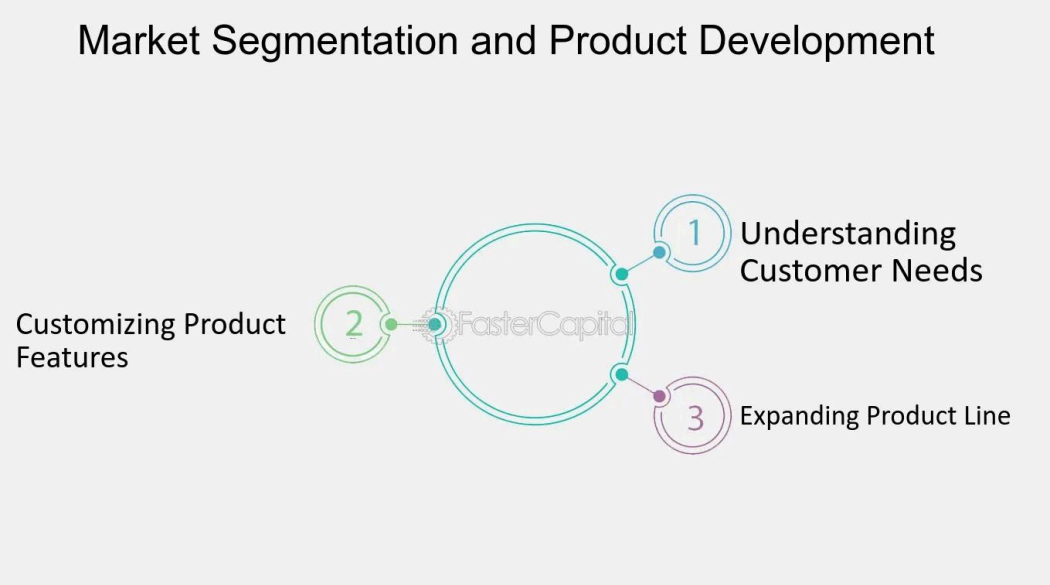

Figure 4.2 Product Lines Market Segmentation

Automated beverage systems like RefreshGo are highly adaptable and can be implemented in a variety of environments, including cafes, event venues, entertainment centers, and public spaces. These systems allow for consistent drink preparation, customizable options, and efficient service, even during periods of high demand. By automating the mojito-making process, from cup dispensing to flavor mixing and payment handling, RefreshGo ensures a smooth workflow and high-quality beverage output. However, the integration of advanced hardware components such as pumps, stepper motors, RFID modules, and camera-based recognition increases the complexity and cost of development, as well as the need for routine maintenance. Despite these challenges, there is great potential for further advancement in smart beverage systems, including improved drink personalization, sustainable operation, and integration with other digital services. These opportunities align with broader market trends in segmentation and product development, where understanding customer needs, customizing product features, and expanding product lines are key priorities (Figure 4.2).

5 Methodology

After reviewing relevant literature, analyzing market trends, and evaluating user needs, it became evident that an automated mojito vending machine like RefreshGo is a timely and valuable solution. This system aligns with the increasing shift toward automation in the food and beverage industry, offering benefits such as rapid service, high customization, and minimal human intervention. RefreshGo aims to optimize the drink preparation process by delivering consistent, personalized mojitos through an intelligent, automated workflow. The project is designed to enhance operational efficiency while improving the customer experience in public and commercial environments.

This chapter will cover the system architecture, hardware components and devices, software and libraries, implementation details, final design of the project, and integration with the mobile application.

5.1 System Architecture

In this section, we are going to delve deep into how we designed the architecture of the project. We designed a product line with a length of 1 meters, width 14.5 cm. At first we designed it without the top components so we can work easily, then we added the upper parts that involve the ingredients and connected it with the belt part. And finally added the oven part to the structure.

- **This was the initial design:**

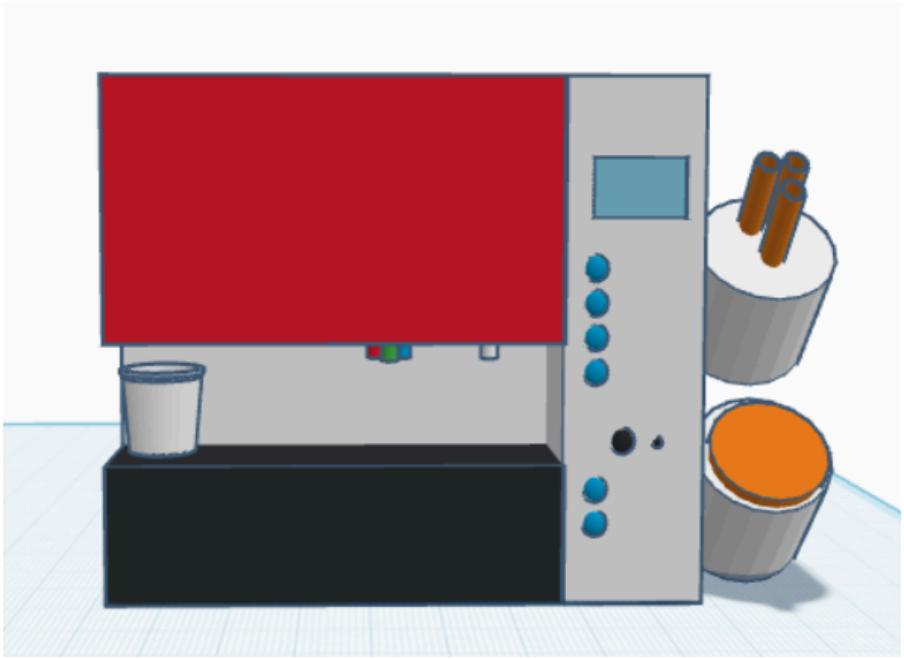

Figure 5.1 Initial Design

- **Cup Dispenser Structure:**
  We used a 3D-printed cup dispenser to automatically release one cup at the start of each order. Mounted at the beginning of the belt system, it is triggered by a stepper motor when payment is confirmed. The design fits our cup dimensions and integrates easily into the machine, ensuring smooth and hands-free operation.

Figure 5.2 Cup dispenser

Figure 5.3 Cup dispenser

- **Ice Dispenser Structure:**
  We use Cereals Dispensers as an ice dispenser which is controlled by a stepper motor that rotates to release a fixed amount of ice into the cup. It is positioned as the first station after

cup dispensing. The motor is programmed to turn for a specific number of steps to ensure accurate ice quantity. This setup ensures consistent ice delivery for every mojito..

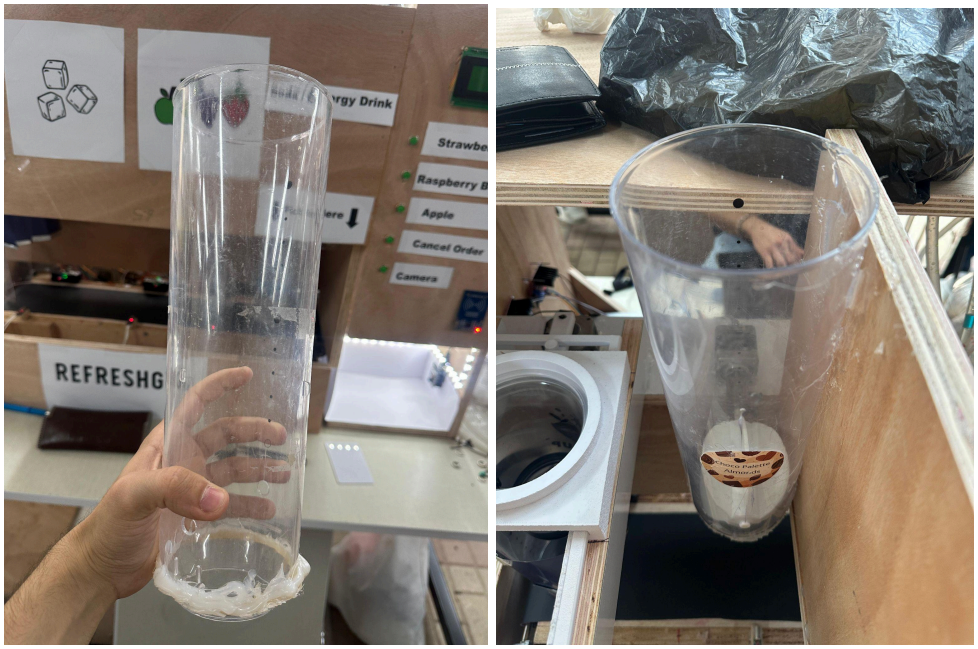

Figure 5.4 Ice Dispenser Structure

- **Flavor Dispensing System:**
  Three flavors—Strawberry, Raspberry Blue, and Apple—are each connected to individual pumps and food-grade pipes. The Arduino controls each pump through assigned pins, dispensing the selected flavors accurately into the cup. This ensures clean, precise, and consistent flavor mixing for every order..

Figure 5.5 Flavor and Base Drink Dispensing Structure

- **Input System:**
  The RefreshGo input panel is designed to be simple and user-friendly. At the top, a small LCD screen welcomes the user and gives instructions. Below it are five labeled push buttons: Strawberry, Raspberry Blue, Apple, Cancel Order, and Camera. Users can choose flavors by pressing the buttons or use the Camera button to scan a drink with the ESP32-CAM. An RFID reader handles quick, contactless payment. At the bottom, there's a well-lit camera box with a white LED strip that turns on during image-based ordering to help the camera read the drink clearly. The whole setup makes ordering easy and clear for anyone using the machine.

Figure 5.6 System Input

5.2 Hardware Components

- **Arduino Mega 2560:**

  The Arduino Mega, based on the Atmega2560 microcontroller, was chosen for our project due to its extensive functionality compared to the standard Arduino board. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UART ports, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. Given the complexity of our project and the numerous hardware components involved, the Arduino Mega served as our central processor. Because it connects all the parts of the project together, and runs the code we wrote using C++ we developed. [4]

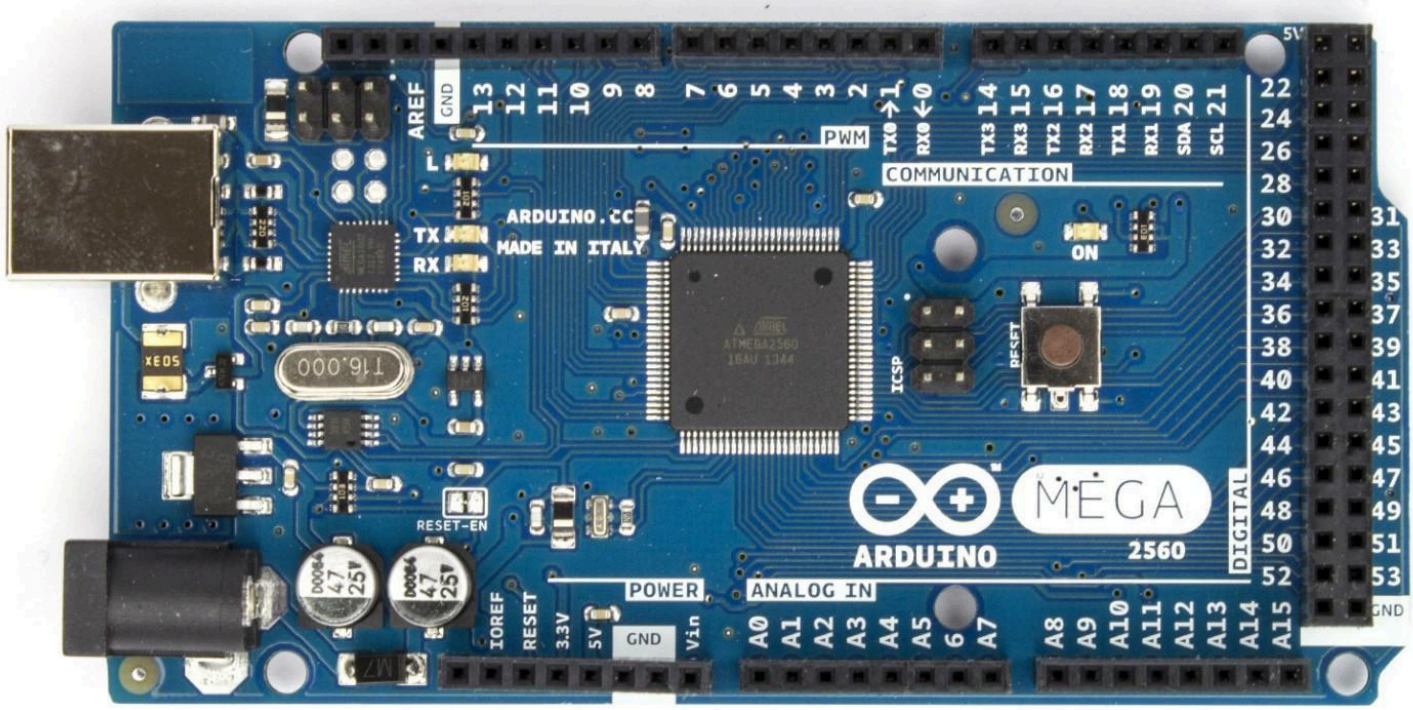

Figure 5.7 Arduino Mega

- **ESP32-CAM Module:**
  The ESP32-CAM is a low-cost development board with a built-in OV2640 camera module and microSD card slot. It supports both Wi-Fi and Bluetooth and is suitable for image processing tasks. In our project, we used the ESP32-CAM to implement image-based drink recognition. When a user presents a previously made drink in front of the camera, the ESP32-CAM captures the image, extracts the dominant RGB color values, and sends the data to the ESP32 module. The Arduino then uses these values to reproduce the same drink by activating the corresponding flavor pumps. This component plays a key role in enabling the "Visual Order" feature of RefreshGo, allowing users to re-order drinks based on appearance.

Figure 5.8 ESP-32CAM

- **Liquid Pumps:**

The pumps are responsible for dispensing the flavored syrups and water into the user's cup. Each pump is connected to a specific flavor—Strawberry, Apple, or Raspberry Blue—along with an additional pump for water. The pumps operate using 12V DC and are controlled through digital pins on the Arduino Mega. Each pump activates for a specific duration to ensure accurate proportioning of the selected flavors. These pumps are a core part of the liquid dispensing mechanism and ensure hygiene and accuracy during the preparation process.

Figure 5.9 Pumps

- **White LED Strip:**

A white LED strip was installed inside the cup placement area of the RefreshGo machine to provide focused illumination during the image-based ordering process. This LED strip turns on exclusively when the user selects the third ordering method using the ESP32-CAM. The purpose of the LED is to enhance visibility and ensure consistent lighting conditions for the camera to accurately capture the color of the drink. Without this lighting, ambient changes in the environment could affect the accuracy of RGB color detection. The strip is powered via a 12V line and controlled through a relay connected to the Arduino. This component plays a critical role in ensuring reliable image recognition and accurate drink replication.

Figure 5.10 White LED Strip

- **ESP-WROOM-32:**

The ESP-WROOM-32 module supports 2.4 GHz ~ 2.5 GHz Wi-Fi (IEEE 802.11 b/g/n) and Bluetooth/BLE. It can act as a standalone MCU or provide wireless connectivity to host MCUs. In our project, it simplified IoT by connecting to the internet and enabling communication with our Blynk-based phone application. [5]

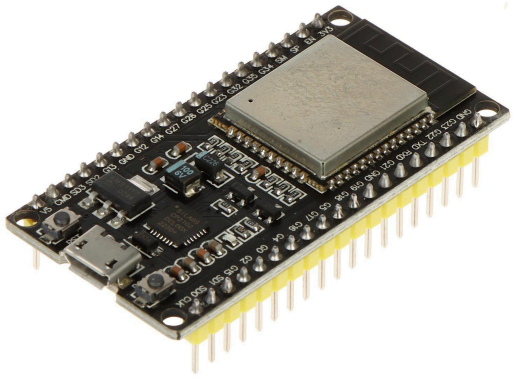

Figure 5.11 ESP-WROOM-32

- **Power Supply:**

We used a computer power supply to meet the voltage needs for our project because It can deliver the 5 volts required for various devices and the 12 volts needed for stepper motors. Furthermore, the power supply provides an adequate current output to fulfill the requirements of our project.

Figure 5.12 Power Supply

NEMA 17 42HS02 Stepper Motor:

We used 6 Nema 17 stepper motors, two of them are used for the ketchup and mayonnaise sauce syringes, and the other four motors used for topping ingredients (tomatoes, lettuce, onions and cheese). [6]

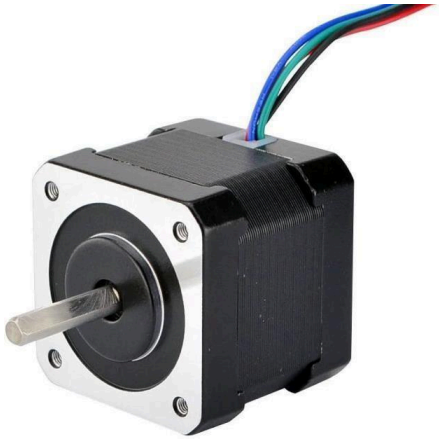

Figure 5.13 Nema 17 Stepper Motor

- **L298N motor driver:**

We used 6 H-bridges to drive the stepper motors, one for each motor. The dual H-bridges of the L298N module are capable of driving one of the stepper engine's electromagnet coils. The stepper motor's shaft can rotate precisely in small steps, either forward or backward, by activating its coils in a specific sequence. We connect the A+, A-, B+ and B- wires from the stepper motor with the L298N Driver. [7]

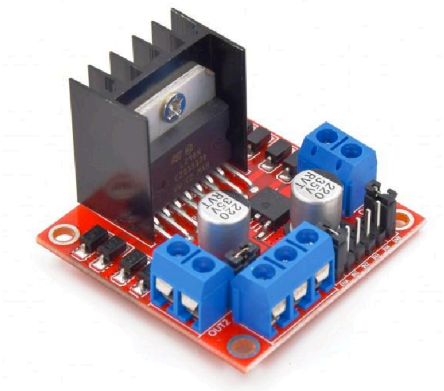

Figure 5.14 L298N motor driver

- **J-5718HB2401 Stepper motor:**

Nema 23 stepper motor operates by energizing its coils in a precise sequence, causing the rotor to rotate in fixed steps (typically 1.8° per step). The direction and speed of rotation are controlled by the input pulse signals from a stepper motor driver. We used this motor to control the rotation of the cup dispenser and the ice dispenser, integrating it with a YS-DIV268N driver.

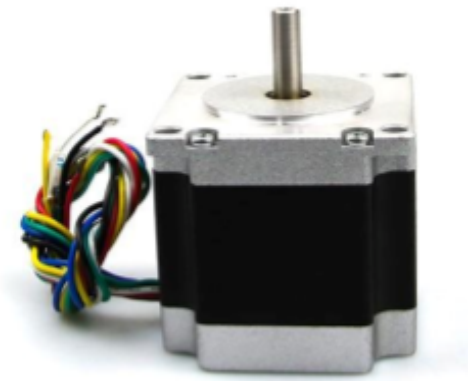

Figure 5.15 J-5718HB2401 Stepper motor

- **Jabnow Push Button Module:**

  A set of 5 digital push buttons used for local flavor selection on the machine. Each button corresponds to a specific option, allowing users to easily customize their drink by pressing the desired choices.

Figure 5.16 JABNOW PB

- **RFID Reader:**

  Radio Frequency Identification (RFID) is a wireless technology that consists of two main components: tags and readers. The reader uses one or more antennas to transmit radio waves and receive responses from the RFID tag. In our project, we used RFID to process customer payments by reading their card when purchasing.

Admin’s tag: Customer’s card:

Figure 5.17 RFID Reader

- **LCD 16x4:**

  The LCD 16x4 is a 20 characters by 4 lines liquid crystal display module. It’s used for displaying output like text and simple graphics. We used it to display text such as asking the user what flavors they wants in their order are and other instructions such as scanning the card request for the customer and printing the progress of his order that will be shown on the screen as a percentage.

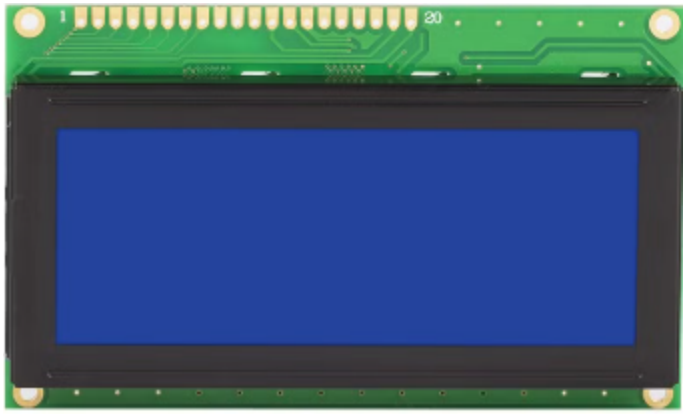

Figure 5.18 LCD 16x4

- **LDR sensor:**

A light-dependent resistor (LDR) sensor module is a tool used to measure variations in light intensity. When the amount of light striking the LDR's surface grows, its resistance lowers, and when the amount of light drops, it increases.The voltage across the LDR lowers as the light intensity rises because the LDR's resistance drops. The digital output changes states based on whether the light intensity falls or rises, depending on how the module is configured. We used this in our project to determine whether an object was reached in a particular location on several stations. We used 8 LDRs, one on each.

Figure 5.19 LDR Sensor

- **Laser:**

The laser module produces a coherent, concentrated light beam that is frequently utilized for precise targeting or sensing applications. It consists of a driver circuit for power control, a laser diode, and frequent optics for beam shaping or focusing.
We created tripwires, measured distances, and detected objects using the laser module in our project. We have constructed systems that are able to recognize when the beam is broken and initiate the intended action by merging the laser module with a light sensor.

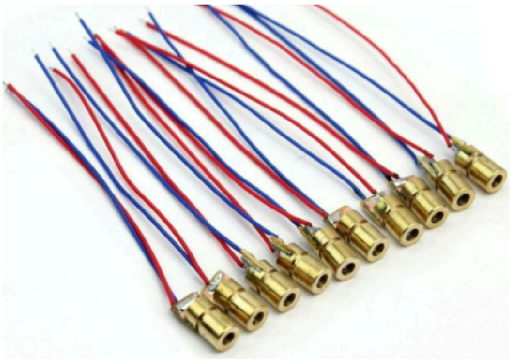

Figure 5.20 Laser

- **Ultrasonic Sensor:**

  An electronic component that uses ultrasonic sound waves (through air) to measure the distance of the target object and the reflected sound is converted into an electrical signal. We used it to track the amount of components left in the container containing the ingredients, so we thought of an ultrasonic sensor to detect the distance. The administrator receives this information, allowing them to restock the supplies as needed.

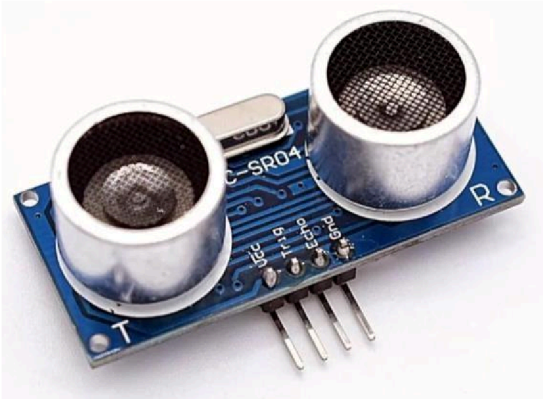

Figure 5.21 Ultrasonic Sensor

- **DC motor with gearbox:**

The DC motor was powered by a 12V power supply and controlled by a relay to regulate its direction. Its role was to drive the movement of the belt that required smooth and consistent rotation, making it an essential part of our machine system.

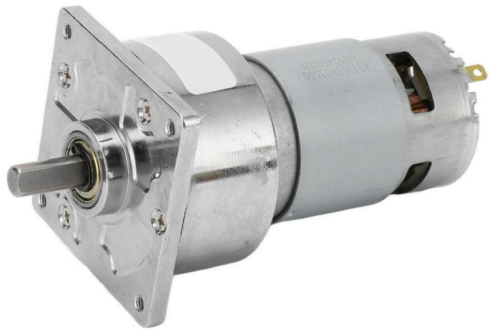

Figure 5.22 DC motor with gearbox

- **Two Channel Relay Module**

We used it to control the two heaters during the meat cooking stage, ensuring precise control over the cooking duration to achieve consistent and optimal results.

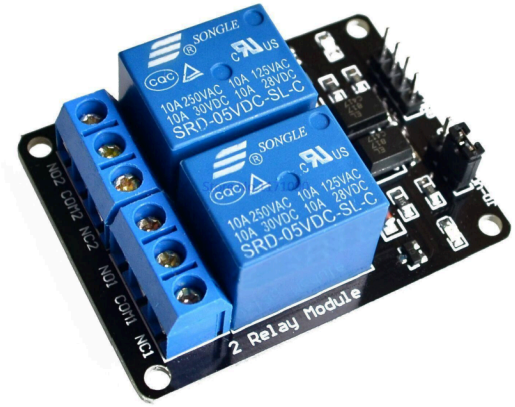

Figure 5.23 Two Channel Relay Module

- **Wires:**

We used jumper wires and standard wires for the connections between components, male-to-male, female-to-female, and male-to-female.

Figure 5.24 Jumper Wires

5.3 Software and Libraries

5.3.1 Libraries used in Arduino code

- **SPI.h**: A built-in library that provides functions for using the Serial Peripheral Interface (SPI) protocol. Which connects between microcontrollers and peripheral devices such as sensors, displays, and RFID modules.

- **Servo.h:** A library that provides an easy way to control servo motors using Arduino.

- **MFRC522.h:** A library is used to interface with MFRC522 RFID reader modules, enabling Arduino projects to read and write data to RFID tags and cards. It simplifies communication with the module over SPI, providing functions for authentication, data reading, and writing.

- **LiquidCrystal_I2C.h:** An Arduino library that interacts with LCD displays that use the I2C communication protocol. I2C communication makes connecting an LCD display to the Arduino board much easier with less pins.

- **Wire.h:** A library used in Arduino programming to facilitate communication between devices over the I²C (Inter-Integrated Circuit) protocol. It allows multiple devices, like sensors, displays, and microcontrollers, to communicate using just two wires: SDA (data line) and SCL (clock line).

- **Stepper.h:** A library in Arduino used to control stepper motors. Stepper motors are precise motors that divide a full rotation into multiple steps, allowing you to control their position, speed, and direction accurately.

5.3.2 Libraries used in ESP32 code

- **Wifi.h:** A library specifically designed for ESP32 and ESP8266 boards, enabling them to connect to Wi-Fi networks. It provides easy-to-use functions to handle Wi-Fi connections, manage access points, and work in both client and server modes.

- **BlynkSimpleEsp32.h:** A necessary component for developing IoT applications with the ESP32 microcontroller and the Blynk platform.

5.4 Implementation

5.4.1 System Design

We used the Arduino Mega as the main controller in our mojito vending machine, RefreshGo, due to its multiple serial ports and large number of digital/analog pins, making it ideal for managing the various motors, sensors, pumps, and modules efficiently. When the system is powered on, both the Arduino Mega and the ESP32 start simultaneously. The ESP32 connects to the internet and acts as a bridge between the mobile application and the Arduino, receiving order data such as flavor selections or image-based preferences and forwarding them to the main controller. The mobile app, developed using Blynk, allows users to remotely customize their mojito by selecting flavors or placing repeat orders. Admin users are also notified through the app when any tank or ingredient needs refilling.

Before the preparation process begins, the LCD turns on and displays a welcome message, followed by instructions guiding the user through the mojito customization process. If the user is ordering directly from the machine, they can use push buttons to select their preferred flavors such as Strawberry, Raspberry Blue, or Apple. Once the customization is complete, the LCD prompts the user to scan their RFID card. The RFID module verifies the balance stored on the card, and if the payment is successful, the preparation process begins. This ensures that orders are only processed after valid payment, preventing unauthorized use of the system.

The Arduino Mega also controls a relay that manages the movement of the conveyor belt. Light Dependent Resistors (LDRs) are placed at different stations along the belt to detect the presence of the cup. When an LDR detects a cup, the Arduino stops the belt via the relay to perform a specific action, such as adding ice, injecting flavor, or dispensing water. Once the step is completed, the belt resumes movement toward the next stage, ensuring precise alignment and synchronization between actions.

The production process is divided into stages. Initially, a cup is dispensed. At the first stage, a stepper motor adds ice to the cup. The cup then moves to the flavor selection stage, where dedicated pumps dispense flavors based on the user's choices—Strawberry on pin 27, Raspberry Blue on pin 25, and Apple on pin 29. The final stage adds water using a pump on pin 23. These actions are automatically triggered by the Arduino based on the flavor selections received either through the buttons or from the ESP32. The system ensures that flavors are not only dispensed accurately but also in proportion based on the order type.

A unique feature of RefreshGo is the use of an ESP32-CAM module, which allows users to reorder a previous drink by simply presenting a cup with the desired color. The camera captures an image,

processes the RGB values, and sends the analysis to the ESP32. The Arduino then calculates the flavor ratios needed to recreate that drink, activating the corresponding pumps accordingly. This provides a visually interactive and intelligent ordering method.

Throughout the entire process, the LCD displays a percentage progress update after each stage is completed. For instance, after ice is added, it shows 25%; after flavor selection, 50%; after water, 75%; and finally, upon completion, it displays 100% with the message, "Your mojito is ready. Enjoy your drink!" This keeps the user informed of the preparation stages and adds to the overall experience.

The Arduino Mega's multiple serial ports enable it to handle the RFID module, LCD, ESP32, and motor control commands simultaneously without interference. This smooth integration between hardware and software ensures that all components work in harmony, providing a seamless, efficient, and fully automated mojito-making experience. With both local and remote ordering options, the RefreshGo system offers a smart, interactive solution that enhances customer convenience while also giving administrators easy control and oversight through the mobile app.

5.4.2 Machine Process of Work

At the beginning, the RefreshGo machine is powered off, and all ingredients flavor syrups, water, and ice are fully stocked and ready for dispensing. A stack of cups is loaded into the dispenser, and the internal system is idle, waiting for a new order. Once the power supply is activated, the Arduino Mega, ESP32, and LCD screen initialize. The LCD displays a welcome message: "Welcome to RefreshGo! Press any button to start." This message invites customers to begin their order by pressing one of the physical buttons located below the LCD.

Customers can place their mojito orders through two different methods, offering both local and remote flexibility:

- Order from machine:

  Customers at the machine start by pressing any of the push buttons. Once initiated, the LCD allows the user to select their desired flavors directly. The screen shows a list of available flavors such as Strawberry, Raspberry Blue, and Apple. Each button is assigned to a flavor, and the customer presses the buttons corresponding to the flavors they want in their drink. The selections are displayed on the LCD in real-time to confirm the order.

After choosing up to three flavors, the LCD instructs the customer to scan their RFID card for payment. Once the card is scanned and validated, the system begins the preparation process. A cup is dispensed, followed by ice using a stepper motor. Then the selected flavors are dispensed through their dedicated pumps, and finally, water is added to complete the mojito. The cup moves along the conveyor belt, stopping at each stage as LDR sensors detect its position to ensure accurate execution. The LCD updates the customer on each step, displaying progress in percentages. At the end, the screen reads: “Your mojito is ready. Enjoy!”.

- Order from mobile application:

For added convenience, customers can place their orders using the RefreshGo mobile app, built using Blynk and connected through ESP32. The app offers preset drink options and a “Custom Drink” mode, allowing users to toggle desired flavors with ON/OFF switches. Once the user taps “Place My Order,” the request is sent to the ESP32, which forwards the data to the Arduino Mega.

The LCD on the machine then displays: “Remote order received. Please scan your card.” After the RFID card is detected and validated, the system executes the order automatically, following the same step-by-step process as the manual mode. Additionally, the app notifies administrators when any flavor tank is running low, ensuring timely maintenance and refill alerts.

- Visual Order via ESP32-CAM

The most innovative feature of RefreshGo is its image-based ordering system using the ESP32-CAM module. In this mode, a user presents a previously prepared or favorite drink in front of the onboard camera. The ESP32-CAM captures an image of the drink and processes the dominant RGB color values. These values are translated into estimated flavor ratios and sent to the Arduino Mega.

Once the camera-based analysis is complete, the LCD prompts the user to scan their RFID card. After successful payment, the system begins replicating the drink by dispensing the detected flavor proportions. This method offers a fast and intuitive way to reorder a favorite mojito without needing to manually select ingredients.

With these three ordering methods—manual via push buttons, remote via mobile app, and smart replication via image recognition—RefreshGo provides a flexible and engaging user experience. Regardless of how the order is placed, the preparation process is fully automated,

contactless, and guided by a synchronized interaction between sensors, motors, and intelligent controls.

5.4.3 Flow Chart

The flow chart for ordering:

Figure 5.25 Flow Chart

5.5 Final Project

This picture shows the final project “Burger Station”, a smart production line which is used for preparing a customized burger to allow customers to have an easy and effective experience while ordering burgers at any time and any place, whatever the circumstances. It shows all the components we explained earlier in the report. Mainly, the oven with its components, the production line stations with the belt and the containers that carry all the ingredients.

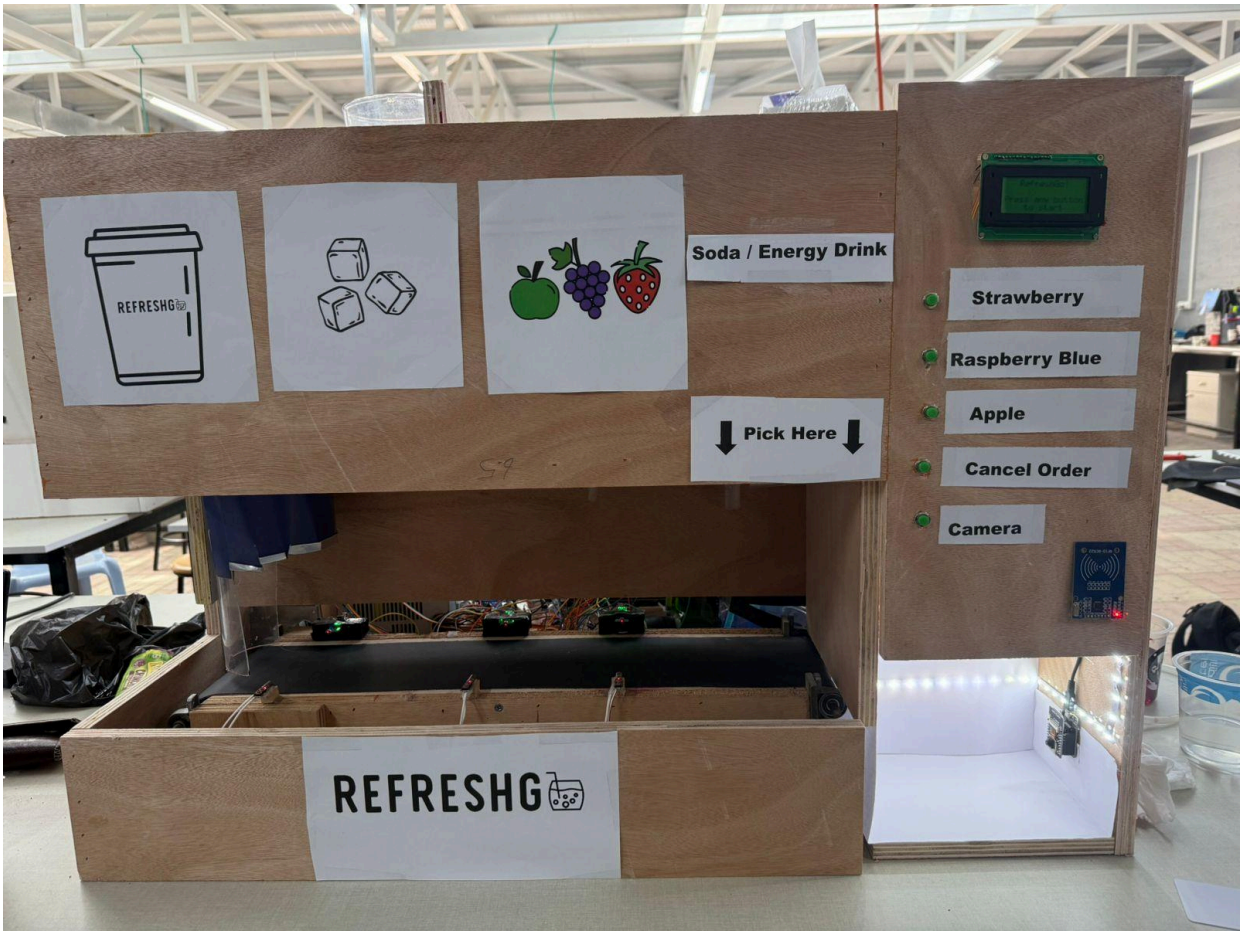

Figure 5.26 Final Project

5.6 Mobile Application

We developed a mobile application on Blynk with many features. The user can make its order and customize it by choosing the ingredients he wants.

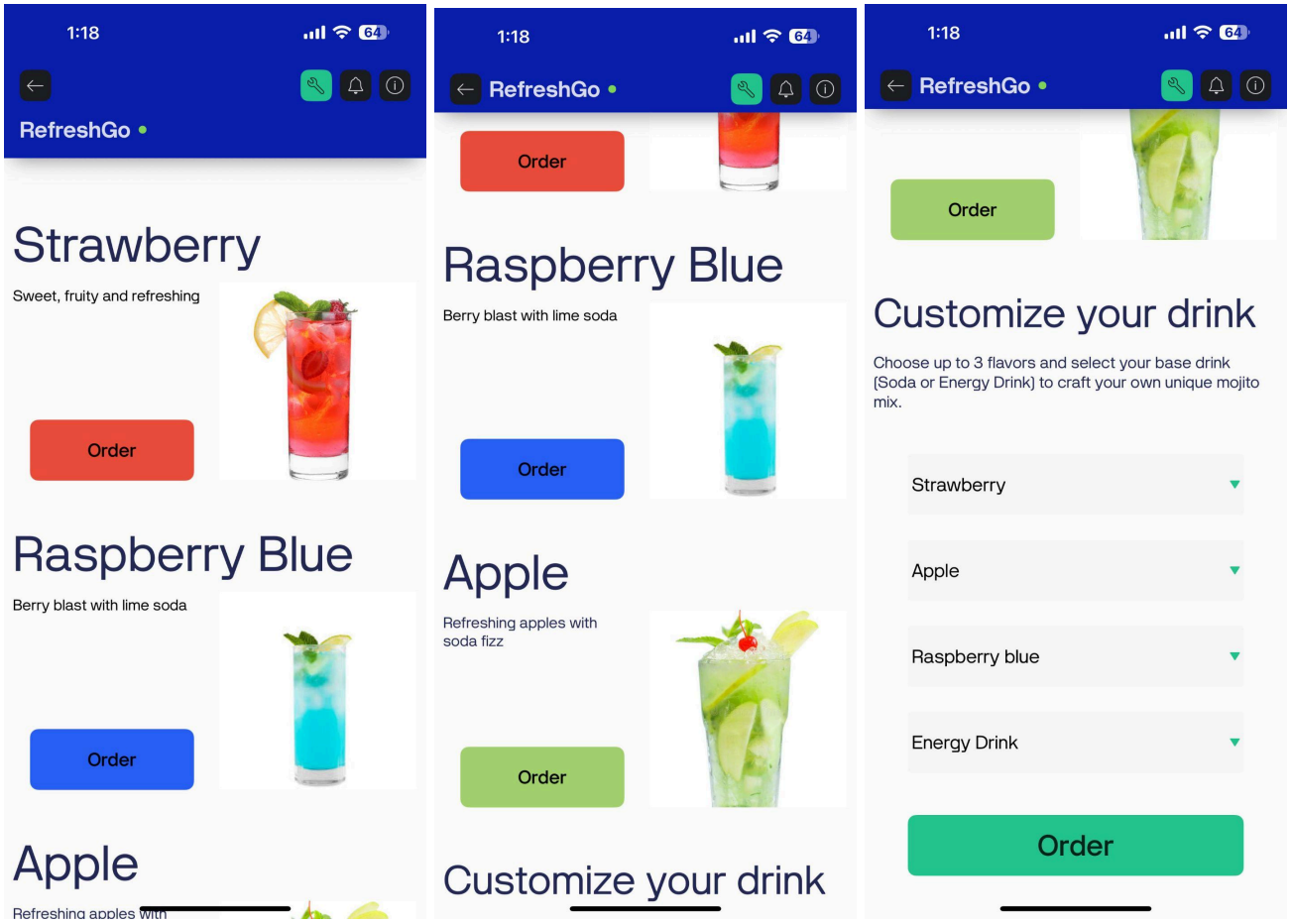

Figure 5.27 Mobile Application Dashboard

6 Results and Discussion

The RefreshGo mojito vending machine was developed to offer a fast, automated, and customizable beverage-making experience. By combining smart hardware and embedded systems, the project delivers an efficient solution that minimizes human effort while enhancing accuracy, personalization, and convenience for users. The use of automation technologies ensures that each mojito is prepared consistently and according to the customer's selected preferences.

6.1 Results

In the end, we successfully designed and built a fully automated mojito vending system capable of preparing personalized drinks with minimal human intervention. The machine integrates multiple components to work in harmony: a stepper motor-based belt system moves the cup through various stages, while LDR sensors ensure the correct timing and positioning at each stage. Dedicated pumps dispense chosen flavors—Strawberry, Apple, or Raspberry Blue—with precision, and an additional two pumps dispense soda and energy drink. The cup is first dispensed through an automated system, and the entire process is tracked to ensure a smooth workflow.

Users can place orders in three distinct ways: using onboard push buttons, remotely via a Blynk-connected mobile application, or through the innovative ESP32-CAM system, which takes a picture of a previously made drink and replicates it by analyzing its color profile. The integration with RFID provides secure, cashless payment, while the LCD guides the user through the process and displays current progress, making the experience more interactive and user-friendly. Overall, the RefreshGo machine performs efficiently, delivering customized mojitos in a reliable and engaging way.

6.2 Discussion

The development of RefreshGo demonstrates how automation can revolutionize beverage service by reducing preparation time, enhancing accuracy, and allowing for high levels of customization. By using a conveyor system and LDR sensors, the machine ensures the drink progresses correctly through each stage—from cup dispensing to flavor mixing and final delivery. The mobile app allows users to order remotely, while RFID provides fast and secure payment. The ESP32-CAM system introduces a novel feature by enabling drinks to be reproduced from visual references, adding an extra layer of customization and innovation.

Even with the successful outcome, we encountered several challenges. One major difficulty was learning to integrate multiple systems—Arduino, ESP32, pumps, motors, RFID, and the camera—into a single, cohesive machine. Even with prior knowledge, it required additional research and repeated testing to ensure all components worked together properly. Another challenge was sourcing the components, which were sometimes expensive or unavailable locally, causing delays in assembly. Time management was also a concern, as the semester included holidays and overlapping coursework, which limited the time we could spend on development.

The project also opened ideas for future improvements, such as adding more flavor options, enhancing the mobile interface, or incorporating a touchscreen for in-person ordering. We could even display advertisements or nutrition information on the LCD during preparation. RefreshGo's modular design allows for easy upgrades and scalability.

In conclusion, RefreshGo offers a practical and reliable automated solution for preparing mojitos quickly and accurately based on user input. By combining customization, automation, and ease of use, it enhances the user experience while streamlining the drink preparation process in busy or modern environments.

7 Conclusion and Recommendation

7.1 Conclusion

In conclusion, our RefreshGo mojito vending machine successfully demonstrates an innovative approach to automating the preparation of customized beverages. Designed for use in high-traffic environments such as Universities, cafés, festivals, and entertainment venues, the system utilizes advanced automation components including stepper motors, pumps, LDR sensors, relays, an Arduino Mega, and an ESP32 to manage the entire drink preparation process from cup dispensing to final delivery. The integration of RFID for cashless payments, a mobile app for remote ordering, and an ESP32-CAM for image-based drink replication makes RefreshGo a highly interactive and intelligent system. The LCD display provides clear guidance and progress updates, improving the user experience. The system's modular design allows it to be easily modified and expanded, making it scalable for future demands and adaptable to include new features. RefreshGo not only addresses the need for faster, more reliable beverage service but also meets modern expectations of customization, automation, and user convenience. It lays a strong foundation for future developments in smart drink preparation technologies.

7.2 Recommendation

Based on our current results, there are several enhancements that could further improve the RefreshGo system. Expanding the variety of flavor choices or allowing users to mix multiple flavors more freely would increase personalization and customer satisfaction. Adding eco-friendly components and optimizing power consumption would make the system more energy-efficient and sustainable in the long term. To reach a broader audience, the RefreshGo system could be deployed in busy public spaces such as malls, airports, festivals, and university campuses. Additionally, marketing efforts should highlight its unique features—remote ordering, contactless payment, and drink replication from images—to attract both users and potential business partners. These enhancements would significantly increase the system's appeal, performance, and market viability.

7.3 Future Work

Several opportunities exist for future development of RefreshGo:

- **Expanded Flavor Range:** Introduce more flavors and allow multi-flavor mixing to boost customization.
- **Touchscreen Interface:** Replace or complement the LCD with a touchscreen for a more intuitive user interface.
- **Dynamic Pricing System:** Integrate pricing options based on flavor combinations or quantity ordered.
- **Cooling Unit for Ingredients:** Add a cooling system to maintain freshness of water and ice supplies.
- **Data Logging and Analytics:** Use cloud storage and dashboards to monitor usage patterns and machine health for better decision-making.
- **Camera System Improvement:** Upgrade the ESP32-CAM to a higher-resolution camera for better image clarity and faster processing. This will improve the accuracy of drink replication by enabling more advanced image analysis techniques like color correction and edge detection.

8 References

[1] Arduino IDE Documentation: https://docs.arduino.cc/.

[2] Blynk Documentation: https://docs.blynk.io/en.

[3] Global In-Plant Logistics Market: https://www.kbvresearch.com/in-plant-logistics-market/.

[4] Arduino Mega Datasheet: https://docs.arduino.cc/resources/datasheets/A000067-datasheet.pdf.

[5] ESP-WROOM-32 Datasheet:
https://www.mouser.com/datasheet/2/891/esp-wroom-32_datasheet_en-1223836.pdf?srsltid=AfmBOopbnjBTg-2dl2OJsrYaeRa_13jKsEu3HXNUDMdVgVJ1wSIQlaF_

[6] Stepper Motor Datasheet:
https://pages.pbclinear.com/rs/909-BFY-775/images/Data-Sheet-Stepper-Motor-Support.pdf

[7] L298N Motor Driver Datasheet:
https://www.handsontec.com/dataspecs/L298N%20Motor%20Driver.pdf

[8] HY-DIV268N-5A Datasheet:
https://www.sigmaelectronica.net/wp-content/uploads/2018/11/div268n-5a-datasheet.pdf.

[9] Servo Motor Datasheet:
http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf