An-Najah National University
Faculty of Engineering & Information Technology Computer
Department of Computer Engineering

DrinkCraft

Prepared By:
Hadi Irshaid
Zahi Abu Shalbak

Supervised By:
Dr. Anas Toma & Dr. Asharf Armoush

A Graduation Project submitted to the Department of Electrical and Computer Engineering in partial fulfillment of the requirements for the degree of B.Sc. in Computer Engineering

Najah National University
June, 2025

Abstract

This project presents the design and implementation of DrinkCraft, a fully automated juice vending machine developed as a smart, self-service system capable of preparing custom fruit-based beverages on demand. The system integrates mechanical, electronic, and software components to handle the entire drink preparation process from cup dispensing to final topping. DrinkCraft features a 3D-printed cup dispenser, fruit containers with controlled stepper motors, IR sensors for positioning, and a central mixing mechanism powered by pumps and a mixer.

The hardware is primarily driven by an Arduino Mega, supported by an ESP32 module to enable wireless order input via a web interface or a local keypad and LCD. The system architecture emphasizes modular design and seamless communication between components, ensuring a reliable workflow that replicates human-level interaction in juice preparation. The user can choose drink types and topping preferences, which are processed in real-time and translated into mechanical actions. Additionally, a basic cleaning mechanism is incorporated to maintain hygiene standards.

DrinkCraft aims to bridge the gap between traditional vending machines and smart robotic kiosks by offering a cost-effective, scalable prototype that showcases automation, user interaction, and efficient hardware-software integration.

Acknowledgment

We would like to express our sincere gratitude to our supervisors, Dr. Anas Toma and Dr. Asharf Armoush, for their guidance and support throughout the course of this project. We also thank our families, colleagues, and friends for their encouragement.

Table of Contents

Chapter 1

Introduction and Motivation

Contents

1.1 Overview

The demand for smart, automated, and contactless solutions in the food and beverage industry has grown significantly. DrinkCraft is a fully automated juice vending machine that prepares customized juice orders on demand. This machine integrates mechanical, electrical, and software components to automate the entire drink preparation cycle — from dispensing a cup to delivering the final juice with optional toppings.

1.2 Problem Statement

Manual juice preparation is time-consuming and lacks consistency. Traditional methods also introduce hygiene and labor-dependency concerns. This project aims to solve these problems by developing a self-service machine that ensures speed, accuracy, and cleanliness.

1.3 Objectives

- Design and build a 3D-printed cup dispenser mechanism.
- Develop a conveyor system with IR sensor feedback.
- Implement a mixing system with fruit and water dispensing.
- Integrate topping and lid placement modules.
- Provide two ordering methods: via web interface and keypad.

- Enable remote control using ESP32 and serial communication with Arduino Mega.

1.4 Motivation

DrinkCraft demonstrates how embedded systems, IoT, and automation can be applied to a real-world scenario. The project also contributes to the growing field of intelligent vending systems, aligning with modern consumer expectations of speed, customization, and interactivity.

1.5 Methodology Overview

We adopted a modular approach to design, implementation, and testing. Each component — mechanical or electronic — was developed independently and later integrated into the complete system. The software control logic was written for the Arduino Mega, with communication facilitated through ESP32 for web-based interaction.

Chapter 2

Background and Related Work

Contents

2.1 Automated Vending Machines

Automated vending machines have evolved to serve not just snacks and beverages but also hot food, electronics, and even custom-made items. These systems typically involve an interface for order input, a delivery mechanism, and a processing unit.

2.2 Juice Dispensing Systems

Most commercial juice dispensers are limited to predefined flavors and lack dynamic ingredient mixing. Our system extends this concept by incorporating fresh fruit dispensing and real-time mixing, offering greater flexibility and freshness.

2.3 Embedded Control Systems

Microcontrollers like the Arduino family are widely used in automation due to their cost-effectiveness, flexibility, and ease of use. ESP32, with its built-in WiFi capabilities, is commonly used for IoT-based communication.

2.4 Stepper Motor Applications

NEMA17 stepper motors are standard in 3D printers and automation systems due to their precision. In our system, they are used for cup dispensing and fruit release.

2.5 IR Sensor Integration

IR sensors are used in industrial systems for object detection. In DrinkCraft, they ensure proper alignment of the cup along the conveyor and facilitate positioning for mixing and topping.

2.6 Relevant Projects

Some similar systems include smart coffee machines and robotic bartenders. However, these often rely on liquid syrup dispensers rather than handling fresh, semi-solid ingredients like fruit.

2.7 Technology Stack Overview

- **Microcontrollers:** Arduino Mega (main control), Arduino Uno (testing), ESP32 (web communication)
- **Motors:** NEMA17 stepper motors, DC motor for lid pressing
- **Sensors:** IR sensors for cup positioning
- **Actuators:** Pumps for liquids, rotating fans for fruit dispensing

Chapter 3

System Design

Contents

3.1 Overview

DrinkCraft's architecture is divided into several subsystems, each performing a key role in the drink preparation process. The design is modular to ensure scalability and ease of maintenance.

3.2 Hardware Architecture

The hardware is built around an Arduino Mega, which controls all components including motors, pumps, and sensors. An ESP32 microcontroller handles communication with a web interface.

3.2.1 Main Components

- **Arduino Mega:** Central controller for hardware logic
- **ESP32:** Handles HTTP requests and relays orders
- **Power Supply:** Provides regulated power to motors and controllers
- **Motor Drivers:** A4988 or equivalent drivers for stepper motors

3.3 Software Architecture

The software consists of multiple modules:

- Cup dispensing
- Belt and IR control
- Ingredient dispensing
- Pump/mixer logic
- Topping and lid handling
- Serial communication parser for ESP32 commands

3.4 Control Flow

1. User submits order via keypad or website
2. ESP32 sends serial message to Arduino Mega
3. Arduino Mega runs the control routine in sequence:
   (a) Dispense cup
   (b) Align using IR sensor
   (c) Dispense ingredients into mixer
   (d) Add water, mix, and pour juice
   (e) Handle toppings and lid if requested
   (f) Clean mixer for next order

3.5 Communication Protocol

ESP32 communicates with Arduino Mega over serial (UART). Messages follow a structured format, such as:

```
CMD_ORDER:3
CMD_TOPPING:1
```

Figure 3.1: Esp MCu used in the serial web communication

Chapter 4

Component Descriptions

Contents

4.1 Cup Dispensing Mechanism

The cup dispenser uses a NEMA17 stepper motor to move a 3D-printed holder. The motor's rotational motion is converted into linear motion to release one cup at a time.

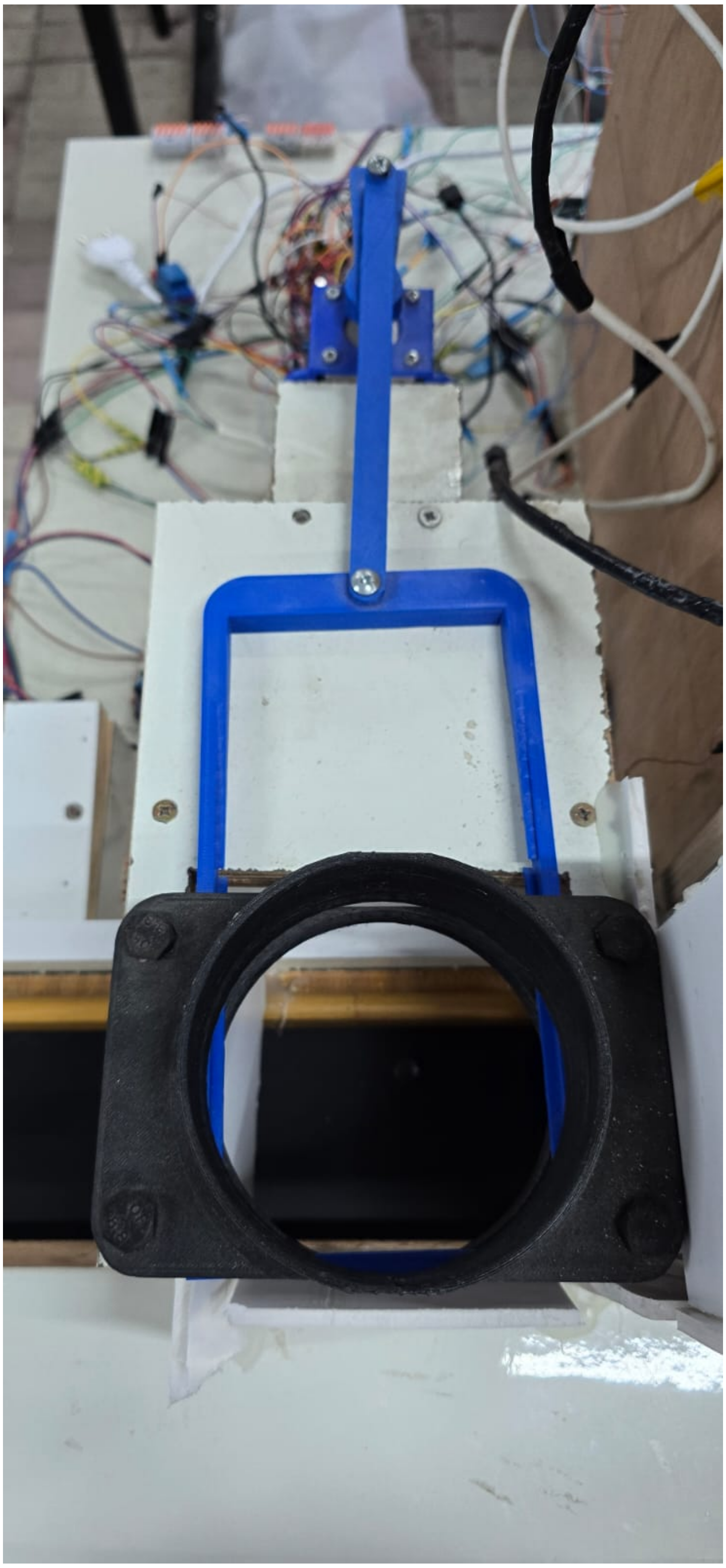

Figure 4.1: 3D Printed Cup dispensing mechanism

4.2 Conveyor and IR Sensors

A belt mechanism transports the cup. The system includes IR sensors to detect cup position:

- **Sensor 1:** Confirms cup is under mixer
- **Sensor 2:** Detects cup at topping station
- **Sensor 3:** Detects cup at final delivery

Figure 4.2: IR Sensor

4.3 Ingredient Dispensers

Each fruit container is equipped with a stepper motor and rotating fan-like structure to release solid ingredients like pineapple or watermelon.

Figure 4.3: Ingredient Dispensers driven by NEMA 17 Steppers

4.4 Mixer and Pumps

The fruit falls into a mixer. A pump introduces water to assist blending. After mixing, a second pump transfers the juice to the cup. A third pump is used during cleaning.

Figure 4.4: The Pump used in water distribution and cleaning process

Figure 4.5: The Mixer used in the project

4.5 Topping Dispenser

When toppings are requested, the cup is moved to a secondary position. A separate module dispenses toppings such as blueberries.

Figure 4.6: The Pump used in water distribution and cleaning process

4.6 Lid Sealing Unit

A lid is applied using a mechanical pad driven by a DC motor to press and secure the lid on the cup.

Figure 4.7: The Sealing Unit

4.7 Cleaning System

The cleaning phase uses water pumped into the mixer. The mixer spins for a short period, and the wastewater is collected via a third pump into a waste container.

Chapter 5

User Interaction and Interfaces

Contents

5.1 Overview

DrinkCraft provides two input methods for the user to place an order: a physical keypad and LCD screen interface, and a web-based UI served via ESP32. This dual-mode input allows flexible operation and remote access.

5.2 Keypad and LCD Input

A 4x4 keypad is used to navigate through a simple on-device menu. The LCD screen provides feedback on selected orders and current status.

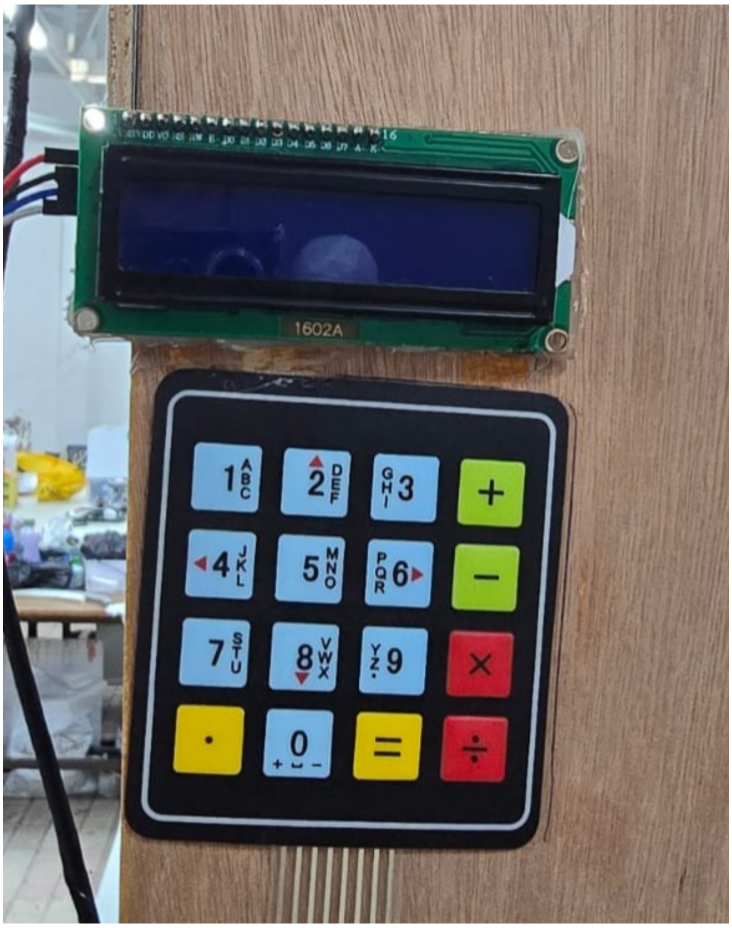

Figure 5.1: Keypad & LCD local inputs

- User can select drink type (e.g., pineapple, watermelon, mix)
- User may optionally choose toppings
- Order is confirmed and sent to Arduino Mega

5.3 Web Interface

The ESP32 hosts a lightweight HTTP server, exposing a menu to the user through Wi-Fi.

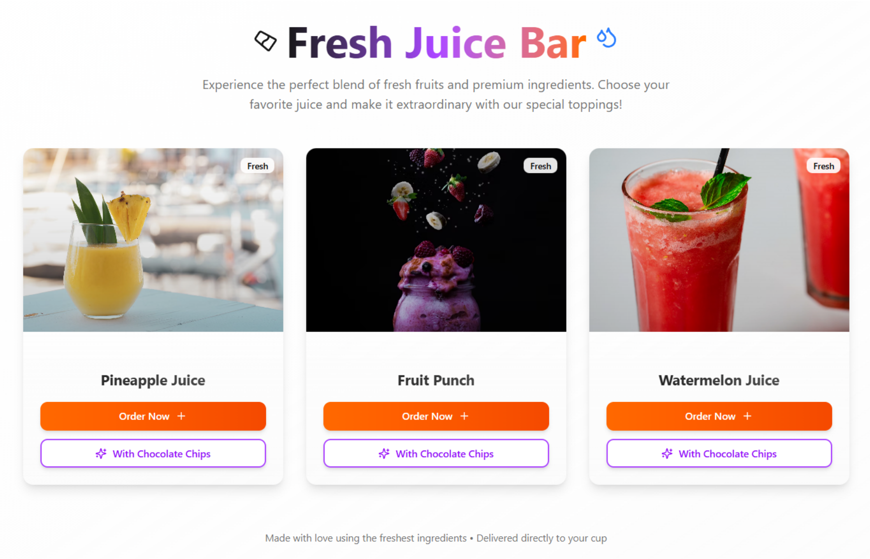

Figure 5.2: The website view

- Orders are selected via a form or buttons
- Once submitted, the ESP32 sends a serial command to Arduino Mega

5.4 Advantages of Dual Input

- **Accessibility:** Web interface allows ordering from a distance
- **Redundancy:** Keypad acts as fallback in case Wi-Fi is unavailable
- **User Preference:** Supports both tech-savvy users and walk-up customers

Chapter 6

Implementation and Integration

Contents

6.1 Mechanical Assembly

All structural components were 3D-printed using PLA filament. The parts include:

- Cup holder and pusher
- Fruit dispenser fan units
- Mixer mount and fluid tubes

6.2 Electronics and Wiring

The system is powered via a 12V power supply. Stepper motors are driven through A4988 drivers. Sensors and LCD are connected to digital I/O pins on the Arduino Mega.

6.3 Software Modules

The control program was written in C++ using the Arduino IDE.

- State machine logic for task sequencing
- ISR routines for sensor detection
- Serial handler to parse incoming ESP32 messages

6.4 Integration Workflow

Each module was first tested individually. Integration steps:

1. Cup dispenser tested with limit switches
2. Mixer tested with timed pump input
3. Serial commands validated over USB and ESP
4. Final system integrated and debugged using serial logs

6.5 Safety Considerations

- Emergency stop button (planned for future)
- Overcurrent protection using fuse for power supply
- Sensor calibration to avoid cup misalignment

Chapter 7

Testing and Results

Contents

7.1 Testing Methodology

The system was tested both as individual modules and as an integrated system. Each component underwent stress and reliability testing.

7.2 Cup Dispensing

- Repeated for 50+ cycles
- No jams or multi-cup releases observed

7.3 Conveyor and IR Sensors

- Cup detected reliably at all three IR checkpoints
- False positives filtered by timing logic

7.4 Fruit Dispensers and Pumps

- Stepper rotations adjusted to ensure 1 serving per request
- Pumps activated via delay-based timers (accuracy 90%)

7.5 Mixer and Cleaning

- Mixing duration tuned for optimal consistency
- Cleaning sequence tested after each operation

7.6 Interface and Communication

- Web orders processed under 1 second delay
- Keypad inputs stable, with LCD reflecting order status
- Serial communication logged and verified

7.7 Performance Summary

Component	**Test Cycles**	**Success Rate**
Cup Dispenser	50	100%
Mixer	30	97%
Web Ordering	25	100%
Topping Dispenser	20	95%
Cleaning System	30	93%

Table 7.1: Component Testing Summary

Chapter 8

Challenges and Limitations

Contents

8.1 Hardware Integration

Integrating multiple motors and sensors on a single Arduino Mega required careful pin management and power regulation. Current spikes from multiple motors running simultaneously were mitigated using capacitors and current-limiting resistors.

8.2 Sensor Calibration

IR sensors are sensitive to ambient light and angle of reflection. We had to fine-tune detection thresholds and implement basic filtering logic in code.

8.3 Pump Timing and Liquid Flow

Controlling pump timing accurately was difficult due to minor differences in tubing length and motor efficiency. Variations affected water and juice dispensing consistency.

8.4 Mixer Residue

Even after the cleaning cycle, some fruit residue was occasionally left behind. Mechanical cleaning or more advanced drainage would be needed in future versions.

8.5 Time limitations

Although the project was intended to be developed within a dedicated workspace designed for such hardware-based systems, delays in preparing the workspace meant it was not accessible for the first two months. This significantly impacted our timeline, as we lost valuable development time during this critical early phase, Specially for a big project like ours.

8.6 Mechanical Precision

3D-printed components occasionally suffered from tolerance issues, especially in high-load areas like the cup dispenser and fruit holders.

8.7 Development Constraints

- Limited access to professional fabrication tools
- Budget limitations on sensor and pump quality
- Short development timeline restricted feature scope

Chapter 9

Conclusion and Future Work

Contents

9.1 Conclusion

DrinkCraft is a successful implementation of an automated juice vending system that combines mechanical, electrical, and software engineering. The final prototype can accept user input, dispense fruit and water, mix juice, add optional toppings, and seal the cup — all without human intervention.

The project demonstrates how real-time systems and embedded control can be applied in practical scenarios. We effectively managed a range of components, from stepper motors to web communication, to build a cohesive and functional product.

9.2 Key Achievements

- Modular and scalable system design
- Integration of web and hardware interfaces
- Real-time automation using microcontrollers

9.3 Future Work

- Add a touchscreen for better UI/UX
- Improve liquid measurement using flow sensors
- Introduce temperature control for ingredients
- Implement cleaning alerts and maintenance logs

- Expand topping variety with interchangeable modules

9.4 Final Thoughts

This project was not only an academic success but also a testament to the power of cross-disciplinary teamwork. It bridges the fields of automation, IoT, and embedded design, and opens the door for future innovation in smart vending systems.