Faculty of Information Technology

IN1901 Microcontroller Based Application Development Project

Juice Maker Machine

IT group 54

Index Number	Name
224109V	Kovintharajan K.
224249A	Vethishan R.
224116N	Lingaraj U.
224252C	Thajeevan V.
224206P	Vithushan K.

Table of Contents

1. Introduction

The Juice Maker Machine is a revolutionary solution for event planners, designed to address the challenge of catering to diverse juice preferences among large groups of guests. This innovative machine allows each individual to customize their juice, selecting their preferred flavor and sweetness level in real-time. Whether at weddings, conferences, or parties, guests can personalize their drinks to match their tastes, ensuring a satisfying experience for everyone. The machine is intuitive and efficient, streamlining beverage service and reducing the stress of managing multiple preferences at once.

With the Juice Maker Machine, event hosts can now offer a truly tailored experience, enhancing guest satisfaction without compromising on convenience. The machine's advanced technology allows for quick and precise customization, delivering the perfect juice for every palate. By meeting the unique needs of each attendee, this device ensures that everyone leaves happy, making it an essential addition to any event planning toolkit.

2. Literature Survey

The demand for personalized food and beverage experiences has been on the rise, driven by a growing consumer desire for customized products that suit individual preferences. In the beverage industry, this trend has been especially pronounced as consumers seek personalized drinks that cater to their taste, health needs, and dietary preferences. Studies have shown that providing options for customization in beverages significantly increases customer satisfaction and loyalty, particularly in environments like restaurants, cafes, and events. As event planning increasingly focuses on enhancing guest experiences, personalized drink solutions, like juice customization, are becoming integral to successful event management.

Several technologies have emerged to address beverage customization. For example, Coca-Cola's *Freestyle* machine allows users to mix and match flavors, offering 100 combinations at the touch of a button. This trend has been mirrored in coffee machines with similar self-service options. However, few innovations have focused on juice customization at events, which require not only flexibility in flavor options but also adjustments in sweetness levels and nutritional considerations. The existing literature highlights a gap in the development of advanced juice-dispensing systems that cater to real-time, on-demand customization at large-scale events. The Juice Maker Machine aims to fill this gap, offering a solution that aligns with current consumer trends and technological advancements in personalized food and beverage services.

3. Problem in Brief

In event planning, one of the major challenges is catering to the diverse preferences of guests, particularly when it comes to beverages like juices. Traditional approaches, such as offering pre-made or limited juice options, often fall short in meeting individual tastes. Guests may have varying preferences for sweetness levels, flavors, or even the types of fruits they enjoy, leading to dissatisfaction for those whose needs aren't met. As events grow in size and complexity, the inability to provide personalized drinks can hinder the overall guest experience, creating a gap between what is offered and what attendees actually want.

The logistical difficulty of customizing juices on-demand for large groups exacerbates this problem. Event planners must either prepare a wide range of options, which is often impractical, or settle for a one-size-fits-all approach that may not satisfy everyone. This creates a need for a solution that can offer real-time customization, allowing each guest to adjust their drink to their exact liking without causing delays or operational inefficiencies. Finding a way to streamline the process while delivering personalized beverages has become essential for ensuring both guest satisfaction and event success.

4. Solution in Brief

The Juice Maker Machine offers a practical and innovative solution to the challenge of meeting diverse juice preferences at large events. This advanced machine enables guests to customize their drinks in real-time, choosing from a variety of flavors and adjusting sweetness levels to suit their personal tastes. With its user-friendly interface, it simplifies the process of beverage service, allowing each guest to create their ideal drink quickly and easily. By eliminating the need for pre-made options, it enhances guest satisfaction without overburdening event staff or causing delays.

For event planners, the Juice Maker Machine streamlines the management of beverages, reducing the logistical complexities of serving a wide range of individual preferences. The machine's ability to provide personalized drinks on-demand ensures that every guest is catered to, enhancing their overall experience. Its efficiency and scalability make it an ideal solution for events of any size, from intimate gatherings to large-scale functions, ensuring a seamless and enjoyable experience for all attendees while reducing the operational strain on event organizers.

5. Aim and Objectives

5.1 Aim:

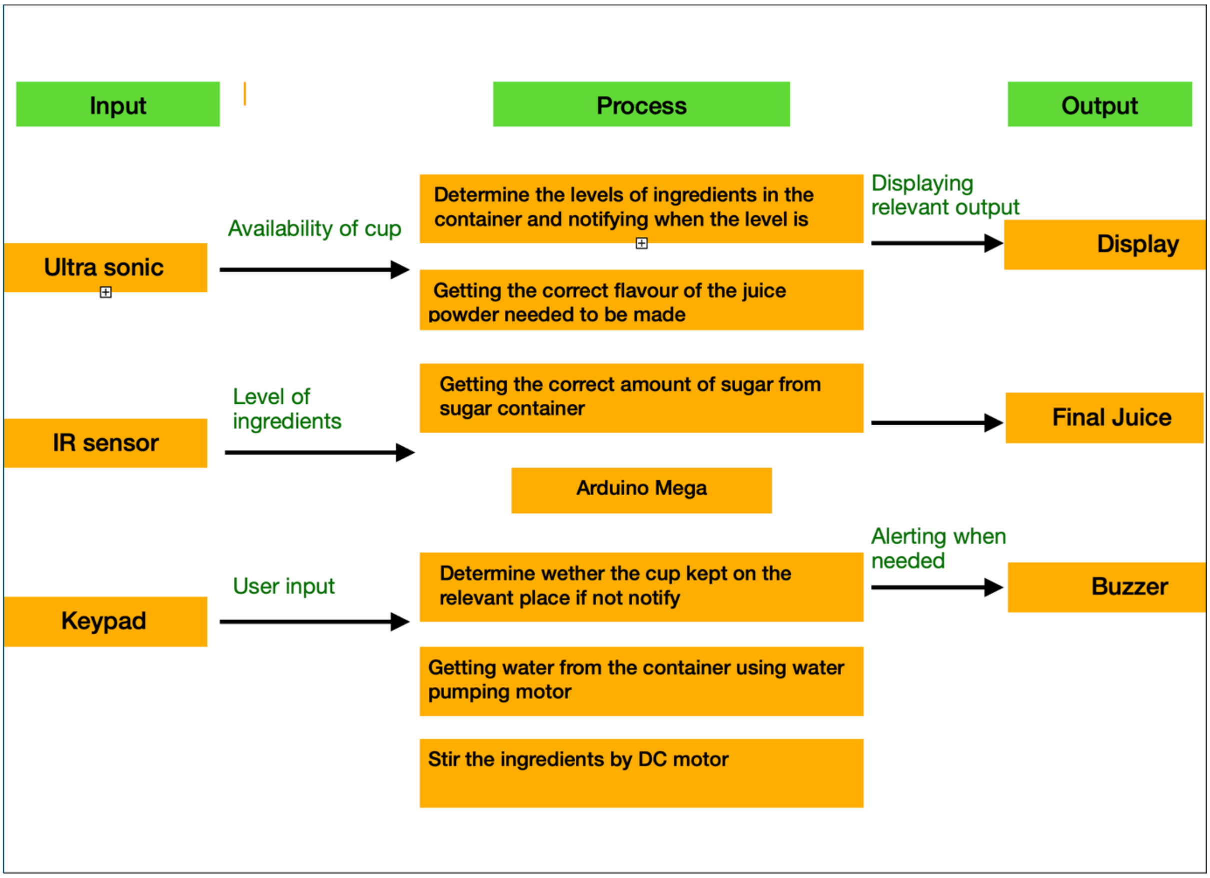
The main purpose of this Juice Maker Machine is to explore the development, functionality, and impact of the beverage customization at events. It seeks to demonstrate how the machine enhances guest satisfaction by offering personalized juice options, and how it streamlines beverage service for event planners.

5.2 Objectives:

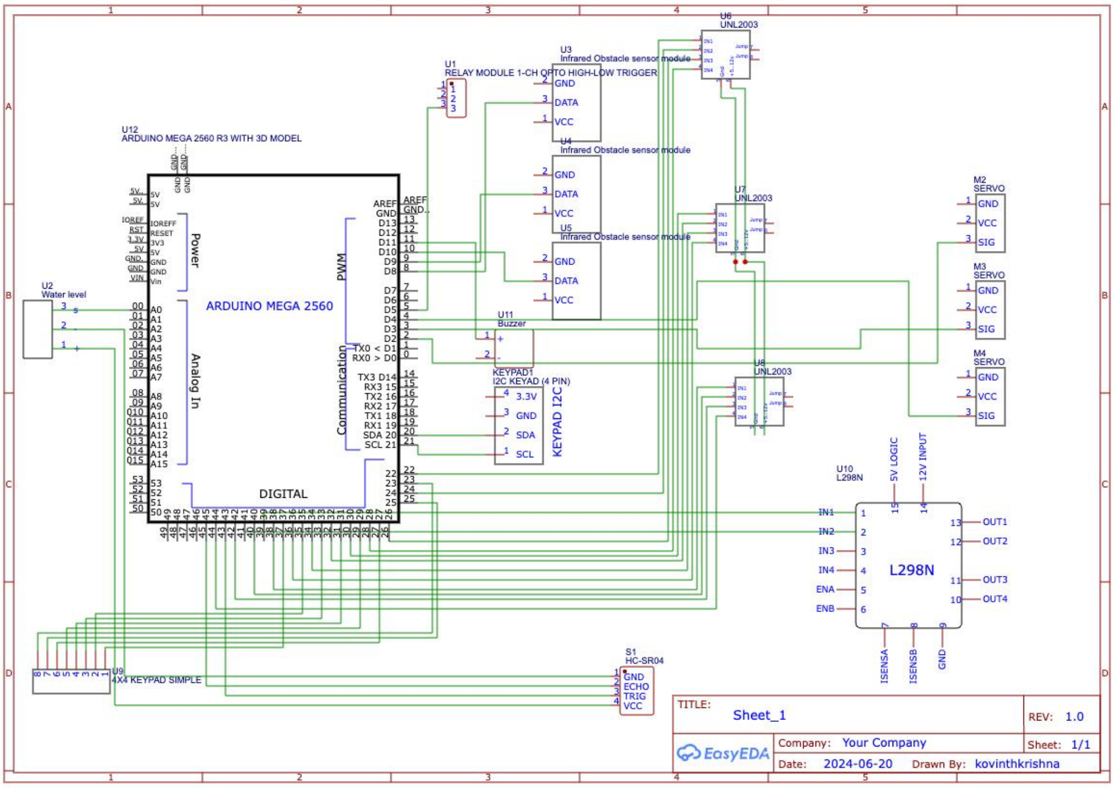
- Identify the challenges in traditional event beverage services with a focus on satisfying diverse guest preferences for juice flavor and sweetness.
- Present the design and technological features of the Juice Maker Machine that enable real-time, personalized juice customization.
- Evaluate the benefits of the machine in terms of improving guest experience, operational efficiency, and overall event satisfaction.
- Analyze the potential impact of the machine on event planning and management, including its scalability for different types and sizes of events.
- Provide recommendations for the integration of the Juice Maker Machine in various event scenarios to maximize its effectiveness.

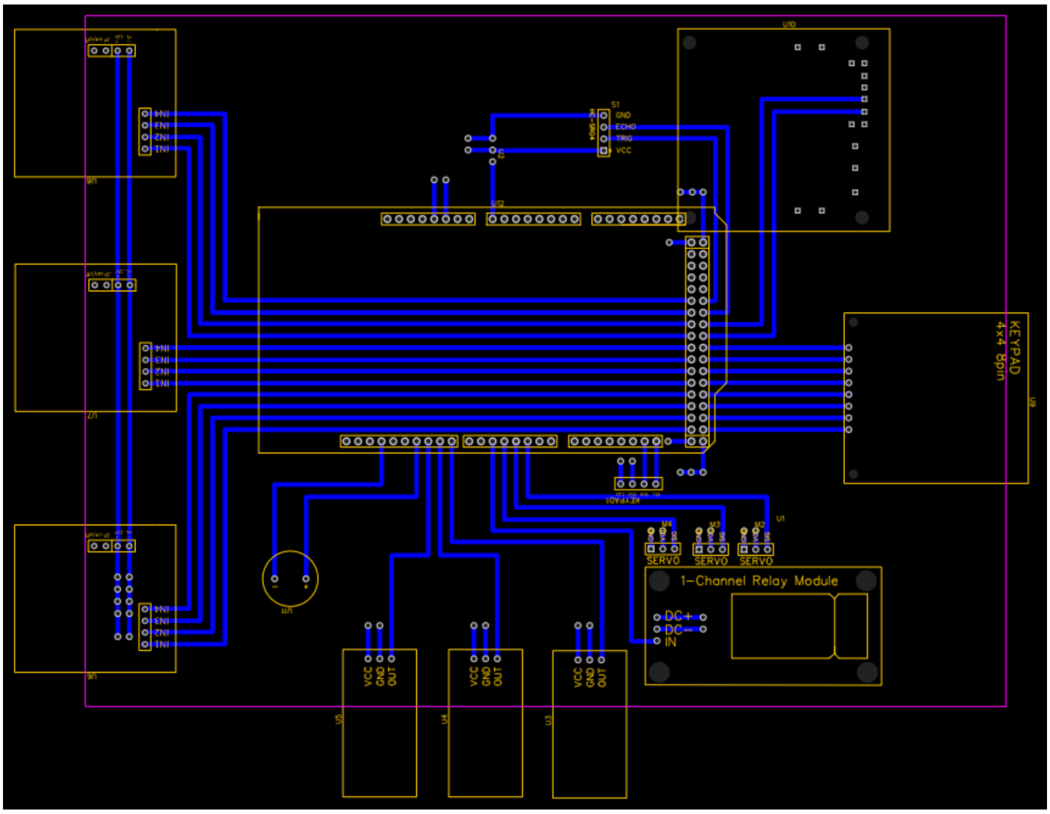
6. Analysis and Design

6.1 System Block Diagram

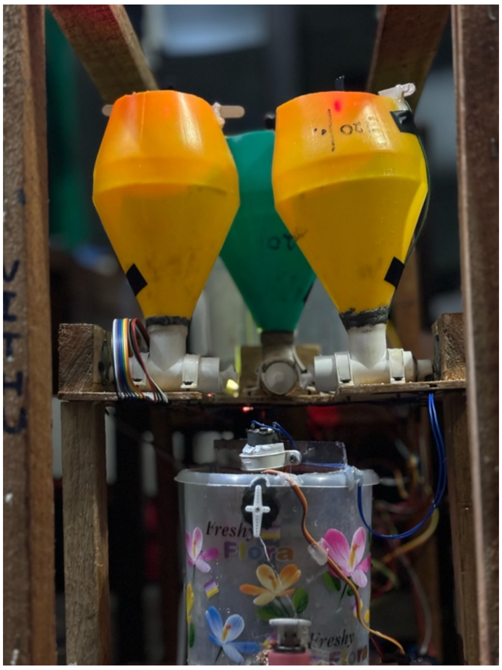


6.2 PCB Design





6.3 Implementation



7. Further Works

As we look ahead to the future of our Juice Maker Machine, we are excited about the endless possibilities for improvement and expansion. Here is a list of potential enhancements and developments we plan to explore in the coming months and years:

- **Enhanced User Interface**: Improving the user interface, both physical (if there's a touchscreen or buttons) and digital (if there's a smartphone app), can make the system more user-friendly and attractive.

- **Enhanced Fruit Variety and Ingredient Expansion**: Future upgrades could include a broader range of ingredients beyond traditional fruits, such as adding vegetables, herbs, or superfoods like chia seeds and ginger.

- **Integration of Nutritional and Dietary Preferences**: Future iterations of the machine could include options for guests to customize their juices based on dietary needs or nutritional goals. For example, users could select low-calorie, or vitamin-enriched options.

- **Mobile App Integration and Remote Ordering**: Developing a mobile app that allows guests to customize and order their juice remotely before arriving at the machine would reduce wait times and enhance convenience, especially at large events.

- **Sustainability and Eco-Friendly Options**: Incorporating eco-friendly elements such as biodegradable cups, energy-efficient operations could enhance the machine's environmental impact.

- **Customizable Temperature Control**: Adding a feature to allow users to adjust the temperature of their juice (chilled, room temperature, or slightly warm) would provide an extra level of customization.

- **Energy Efficiency**: Optimize power consumption to reduce energy costs and environmental impact.

- **Modular Design**: Design the system in a modular way to allow for easy upgrades and maintenance.

- **Customizable Cup Sizes**: Add the option to select different cup sizes, and ensure the system adjusts ingredient quantities accordingly.

- **User Profiles**: Implement user profiles, allowing individuals to save their preferences and easily reorder their favourite drinks.

- **Safety Features**: Consider additional safety mechanisms, such as emergency shut-offs, to ensure the system operates safely at all times.

8. Individual Contributions

8.1 Member – Kovintharajan K. – 224249A

Responsibilities:

- **LCD I2C Integration:**

I studied and documented the operation of the LCD with an I2C Module and established the electrical connections between the LCD and the Arduino Mega 2560, ensuring proper wiring of the SDA and SCL pins for smooth communication. I then programmed the Arduino Mega to control the display, allowing it to show relevant information such as the available juice flavours, sugar levels, and the current states during juice preparation. I also ensured proper calibration of the display's contrast and backlight to provide clear and readable output for the user interface of the juice maker.

- **Ultrasonic Sensor Integration:**

For the ultrasonic sensor, I designed the electrical connections and programmed the sensor to work with the Arduino Mega 2560. This included configuring the sensor’s trigger and echo pins and integrating the sensor readings into the overall system. I implemented code to detect whether a cup was placed correctly for filling. The sensor readings were used to ensure accurate detection, which was essential for initiating the juice-making process.

- **Studying Components and Programming:**

In our project, I played a key role in studying the various components related to both the LCD and the ultrasonic sensor. I chose the LCD with I2C for its simplicity and reduced wiring, and the ultrasonic sensor for its accuracy in detecting objects. After thoroughly studying the technical specifications and operational details, I successfully programmed the LCD to display the selected juice flavour, sugar level, and system status, improving the overall user interface of the juice maker.

- **Code Integration:**

I integrated the control logic for the LCD and ultrasonic sensor with the system's core functionalities. This involved ensuring that the sensor readings from the ultrasonic sensor were processed correctly and displayed in real-time on the LCD module. I managed the communication between the components, ensuring that user inputs, such as flavour selection and sugar level, were synchronized with the cup detection data from the ultrasonic sensor, allowing the juice-making process to proceed smoothly.

Contribution:

I contributed significantly to the development and completion of the juice maker project, particularly in integrating the LCD I2C module and the ultrasonic sensor. By studying the specifications of these components, I designed the electrical connections and programmed the Arduino Mega to interface effectively with both modules. I implemented logic that allowed users to select and view juice flavours and sugar levels on the LCD, while also ensuring the cup was correctly positioned before filling. Additionally, I played a key role in assembling the overall system, ensuring all components were securely mounted and functioned seamlessly together.

LCD with I2C Module

The LCD I2C Module is a compact display unit that simplifies communication using the I2C protocol, requiring only two communication lines (SDA and SCL) to connect with a microcontroller like Arduino. It is typically used to display text or numerical values, making it an excellent choice for projects where visual feedback is needed.

Specifications:

- **Display:** 16x2 LCD
- **Interface:** I2C (SDA, SCL)
- **Operating Voltage:** 5V
- **I2C Address:** 0x27 (Default, configurable)
- **Backlight Control:** Built-in
- **Adjustable Contrast:** Through a potentiometer on the module
- **Dimensions:** 80mm x 36mm x 12mm

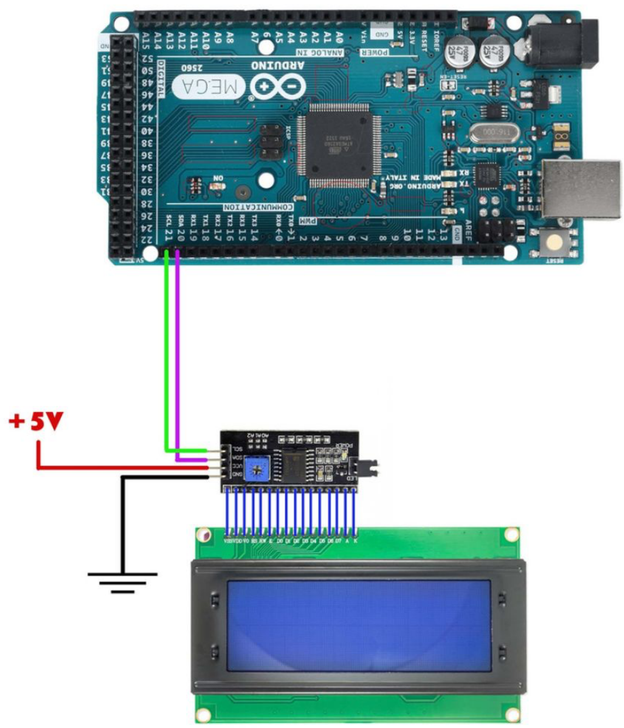
Wire Connections:

- **SDA (Data Line):** Connects to the microcontroller's SDA pin.

- **SCL (Clock Line):** Connects to the microcontroller's SCL pin.
- **VCC:** 5V power supply.
- **GND:** Ground.

How It Works:

The I2C protocol allows communication between the microcontroller (like the Arduino Mega 2560) and the LCD using only two pins. The I2C module attached to the back of the LCD controls the sending and receiving of data, allowing the display of messages or sensor readings. A typical use case is showing sensor data such as distance measured by an ultrasonic sensor or status updates of the system.



Ultrasonic Sensor (HC-SR04)

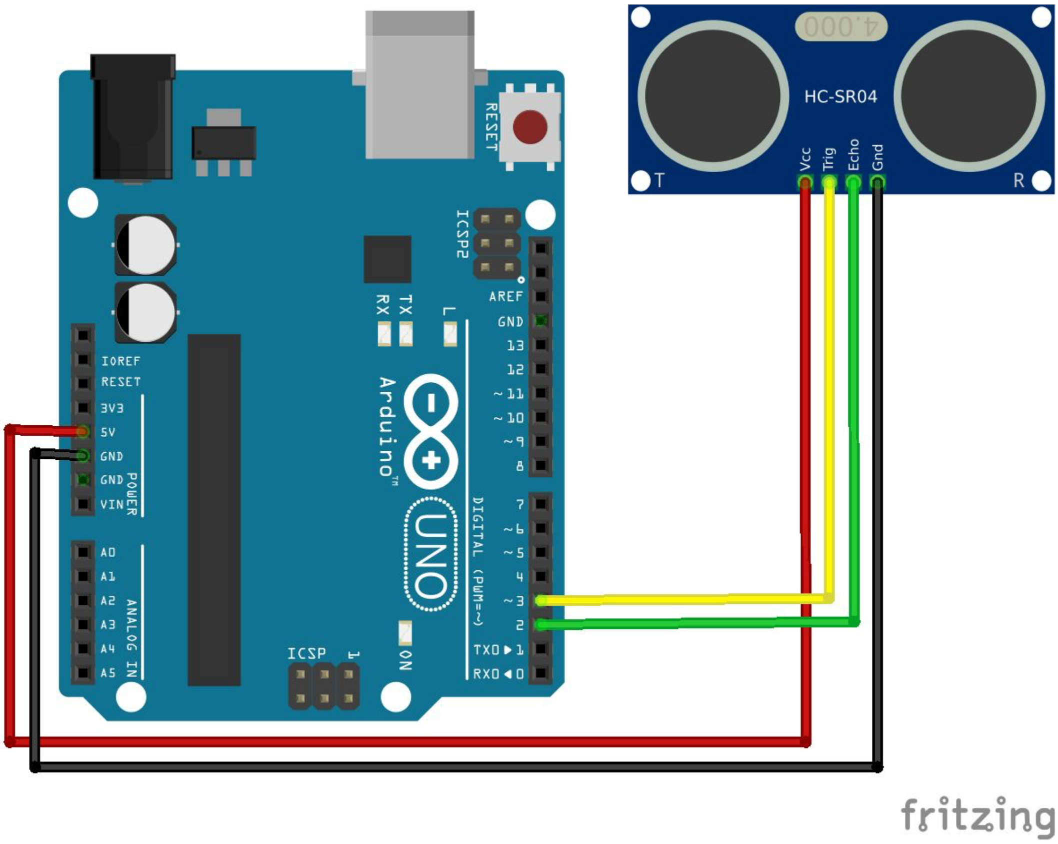
The Ultrasonic Sensor is a simple, easy-to-use distance sensor that measures distance using ultrasonic waves. It's widely used in robotics and automation systems to detect obstacles or measure distances.

Specifications:

- **Operating Voltage:** 5V
- **Operating Current:** <2 mA
- **Measuring Range:** 2 cm to 400 cm
- **Accuracy:** ±3 mm
- **Operating Frequency:** 40 kHz
- **Connection Pins:**
    - **VCC:** 5V Power supply
    - **GND:** Ground
    - **TRIG:** Trigger pin to initiate distance measurement
    - **ECHO:** Echo pin to read the signal

How It Works:

The sensor works by sending out an ultrasonic pulse from the TRIG pin and waiting for the pulse to reflect back after hitting an object. The ECHO pin then receives this reflected pulse, and the time taken for the pulse to return is used to calculate the distance. The microcontroller calculates this by using the speed of sound in air (roughly 343 meters per second). The sensor is ideal for distance measurements and obstacle avoidance.



8.2 Member – Vethishan R. – 224249A

Responsibilities:

- **Studying water-level sensor and programming it to Arduino Mega**

I studied and documented about a water-level sensor for integration with an Arduino Mega in the juice maker machine include selecting an appropriate sensor based on the machine's needs, such as liquid type and tank size, and ensuring compatibility with the Arduino Mega. This involves proper wiring, sensor calibration, and writing code to accurately read and process water-level data. The code helped to trigger actions based on predefined thresholds, such as stopping the machine when water level is low. Additional tasks include testing and debugging the system to ensure real-time monitoring, integrating the sensor with other components like pumps and valves and issuing alerts.

- **Designing the stirring mechanism using DC motor using L289N driver**

I selected 12V DC motor and L298N driver , ensuring the motor provides enough torque and speed to stir the juice effectively without overheating or malfunctioning. The stirring blade designed efficiently and well-coupled to the motor shaft for smooth operation, with proper alignment to prevented wobbling. The motor and stirring mechanism integrated seamlessly into the juice mixer, considering space constraints, safety features, and ease of maintenance. Additionally, the motor controled through the L298N driver for adjustable speed and direction using PWM (Pulse Width Modulation) for optimal stirring and decrease the time of disolving. The control system programmed to operate the motor at specific times during the juicing process, and power management optimized to ensure efficient and reliable operation. Lastly, thorough tested and calibrated of the stirring system to ensure consistent performance, durability, and easy maintenance.

- **Designing the structure of spiral dispenser (Vis sans fin)**

Designing the spiral dispenser (*vis sans fin*) for the juice maker machine involves several critical responsibilities. The first is selecting appropriate materials, ensuring they are durable, corrosion-resistant, and food-grade compliant to maintain hygiene and withstand wear. The geometric design optimized for smooth operation, with the correct pitch, diameter, and length to handle the flow of sugar and flavour powder efficiently without clogging. Structural integrity is essential to ensure the spiral can endure the pressure from various flavours while maximizing juice extraction and minimizing waste. It integrated seamlessly with other machine components, such as the motor and juicing chamber, while considering ease of cleaning and maintenance through modular design. I used 3D printer to create this structure.

- **Code Integration**

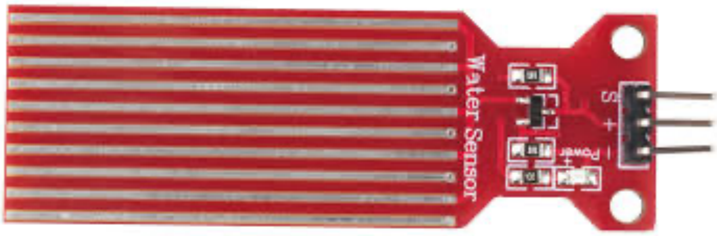
I integrated the control logic for the Water Level Sensor and DC motor with L298N driver with the system's core functionalities. This involved ensuring that the sensor readings from the water level sensor were processed correctly and displayed in real-time on the LCD module. I controlled the speed of the motor for decrease wobbling and spilling of the juice while mixing. I used L298N to make the motor to spin in both directions it decreased the time to get disolve.

Contribution:

I contributed significantly to the development and completion of our project, particularly in integrating the Water level sensor and the DC motor with L298N. By studying the specifications of these components, I designed the electrical connections and programmed the Arduino Mega to interface effectively with both modules. I implemented logic that allowed sensor readings to be displayed clearly on the LCD screen. In addition, I worked on constructing the overall system, ensuring that all components were securely mounted and functioned harmoniously.

Water Level Sensor

A water level sensor is a device used to detect and measure the level of water or other liquids in a container, tank, or system. It provides real-time data about the liquid's height, which can be used to monitor and control processes where liquid levels are critical, such as in industrial, domestic, and environmental applications.

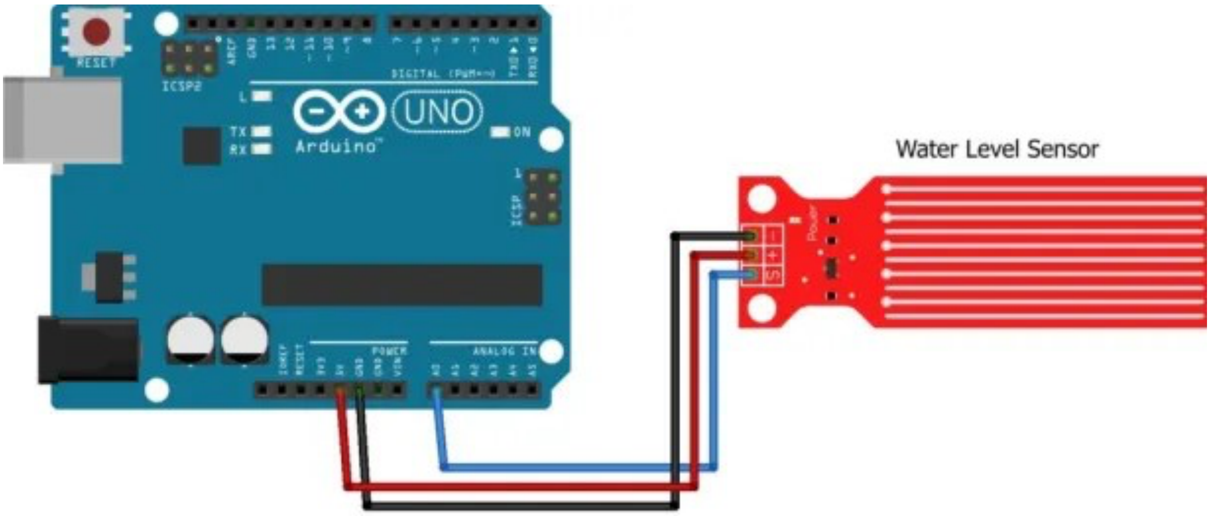


Specifications:

- **Operating Voltage:** 3.3V – 5V
- **Measuring Range**: 2cm – 10m
- **Accuracy:** ±1% to ±3% of full scale
- **Operating Tempreature:** -10°C to 80°C
- **Connection pins:**
    - **VCC:** 5V Power supply
    - **GND:** Ground
    - **SIG**: Analog signal output

How it works:

A typical water level sensor detects the presence and amount of water by sensing electrical resistance changes or using mechanical, ultrasonic, or pressure-based methods. For example, in simple resistive-type water level sensors, the sensor consists of a series of exposed metal traces (copper or stainless steel) on the sensor board that sense the water level based on conductivity. When the water makes contact with these traces, the sensor detects the change in resistance and converts this into an electrical signal. The higher the water level, the more of these conductive traces get covered by water, allowing the sensor to measure how much liquid is present.


Water Level Sensor

12V Brushed DC motor with L298N

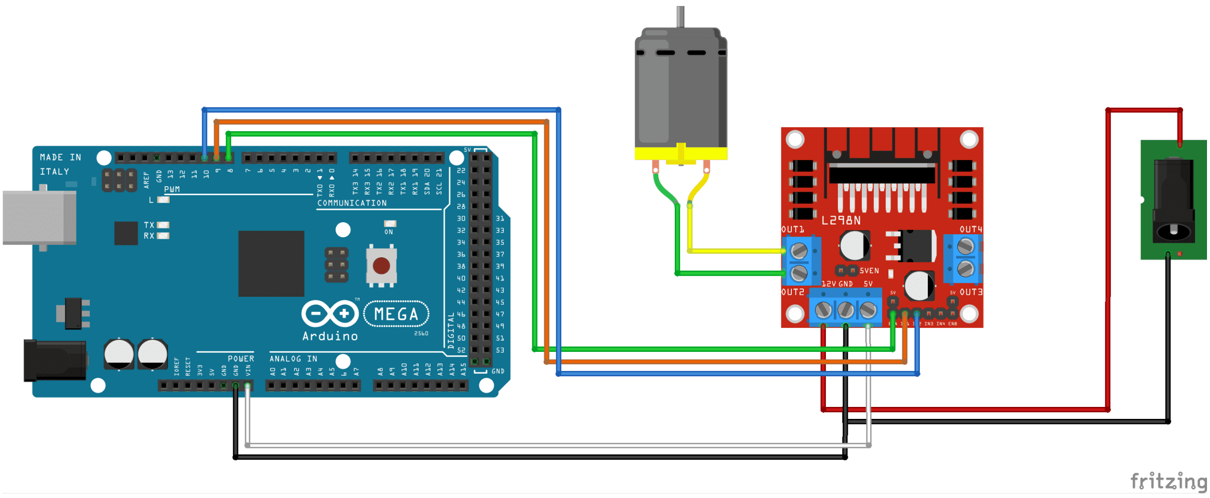
A 12V brushed DC motor is a simple, widely used type of motor that operates on direct current (DC) and uses mechanical brushes to manage the current flow to the motor windings, causing rotation. It is commonly found in various applications due to its cost-effectiveness, ease of control, and availability

Specifications:

- **Operating Voltage:** 12V
- **Operating Current**: 0.5 – several amps (Depending on motor size)
- **Speed:** 1000 to 10000 RPM
- **Connection Pins:**
    - **Positive Terminal (+) :** 12V Power supply
    - **Negative Terminal (-) :** Ground

How it works:

A 12V brushed DC motor works by converting electrical energy into rotational mechanical energy through the interaction of magnetic fields. When a 12V power supply is connected, current flows through the motor's rotor windings, creating a magnetic field that interacts with the stationary magnetic field of the stator, typically made of permanent magnets. This interaction generates torque, causing the rotor to spin. The motor uses a commutator and brushes to reverse the direction of current through the rotor windings at the right time, ensuring continuous rotation in the same direction. By swapping the power supply polarity, the motor's direction can be reversed. The speed of the motor is controlled by varying the applied voltage or using Pulse Width Modulation (PWM), which adjusts the power delivered to the motor. The simplicity and ease of control make the 12V brushed DC motor widely used in various applications, from household appliances to automotive systems.



8.3 Member - Thajeevan V. (224252C)

Responsibilities:

- Designing juice distributing and drainage mechanism using servo motor
- Studying about keypad module and programming it to Arduino Mega
- Designing and building the structure of the machine

In our hardware project, I played a pivotal role in various aspects of the project, demonstrating a diverse skill set and contributing significantly to its success. I conducted extensive research and analysis to identify the most suitable components for our project.

- ## SG90 Servo Motor Integration:

I studied and documented the specifications and operation of the **SG90 servo motor** and designed the electrical connections between the servo motor and the other components, ensuring both safety and functionality. I then programmed the **Arduino Mega 2560** to control the servo motor, allowing for precise control of the juice dispensing and drainage mechanisms. I implemented control logic and calibration routines to ensure accurate and reliable motor operation, preventing any unintended movements and potential mechanical issues. Utilizing the knowledge I gained, I successfully integrated the servo motor into the overall system, facilitating effective automation and control of the juice maker machine.

- ## Keypad Module Integration with Arduino Mega 2560:

I have been assigned to study the keypad module and designed the hardware connections between the keypad module and the Arduino Mega 2560, considering pin assignments and voltage requirements. Then I programmed the Arduino Mega 2560 to interface with the keypad module, allowing user input through the keypad, and integrated the keypad functionality into the system, enabling user input for various control and configuration purposes.

- ## Studying Components and Programming:

I was primarily responsible for the juice distribution and drainage system of our juice maker machine. This involved a detailed study of various components to ensure optimal performance. I focused on selecting the appropriate servo motor for controlling the juice dispensing mechanism. The servo motor was chosen for its precision and reliability in managing the distribution and drainage of juice.

I also researched and implemented the keypad module for user input, programming it to interface effectively with the Arduino Mega. This allowed users to interact with the machine and select different juice options seamlessly. My work involved understanding the keypad's functionality and integrating it into the system to enhance user experience.

Furthermore, I designed and constructed the physical structure of the juice maker machine. This included assembling the components in a way that ensured durability and ease of use, while also ensuring that the mechanical parts operated harmoniously with the electronic controls.

- **Code Integration:**

In addition to coding individual components, I played a crucial role in integrating all the code elements for the juice maker machine. This involved ensuring that data from the keypad module was accurately processed by the Arduino Mega and translated into commands for the servo motor responsible for juice dispensing. I coordinated the control logic to ensure that the servo motor operated in sync with user inputs and the machine's functions. My integration work was essential for creating a cohesive system where all components worked together efficiently and reliably.

- **Building the structure of the machine**

I also contributed to the physical aspect of our project by implementing the structure of the machine itself. This involved designing and assembling the housing or enclosure for all the components, ensuring they were securely mounted and protected from environmental factors.

Contribution:

I am pleased to contribute to the successful completion of our "Juice Maker Machine" project. As part of the team, I was responsible for designing the juice distributing and drainage mechanism using a servo motor, as well as studying and programming the keypad module for the Arduino Mega.

I began by researching various servo motor systems, selecting the SG90 servo motor for its suitability in controlling the juice and water dispensing process. I then designed and implemented the distribution mechanism, ensuring smooth and efficient operation. To complement this, I studied the integration of the keypad module, allowing users to control the machine's functions. I successfully programmed it in Arduino IDE and tested it for functionality.

Finally, I worked on designing and constructing the structure of the machine, ensuring that all components were securely mounted and accessible. This hands-on experience in both hardware and software provided a great opportunity to bring the project to life.

Servo Motor (SG90) :

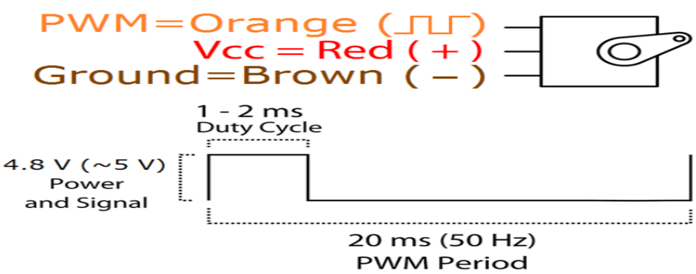
SG90 Servo Motor is a tiny and lightweight motor with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but smaller. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places.

Specification

- 3 pole ferrite, all nylon gear
- Top ball bearing
- Operating Voltage: 4.8 - 6V
- Operating speed: 0.12sec/60 degree
- Output torque: 1.6kg/cm 4.8V
- Dimension: 21.5 x 11.8 x 22.7mm
- Weight: 9g

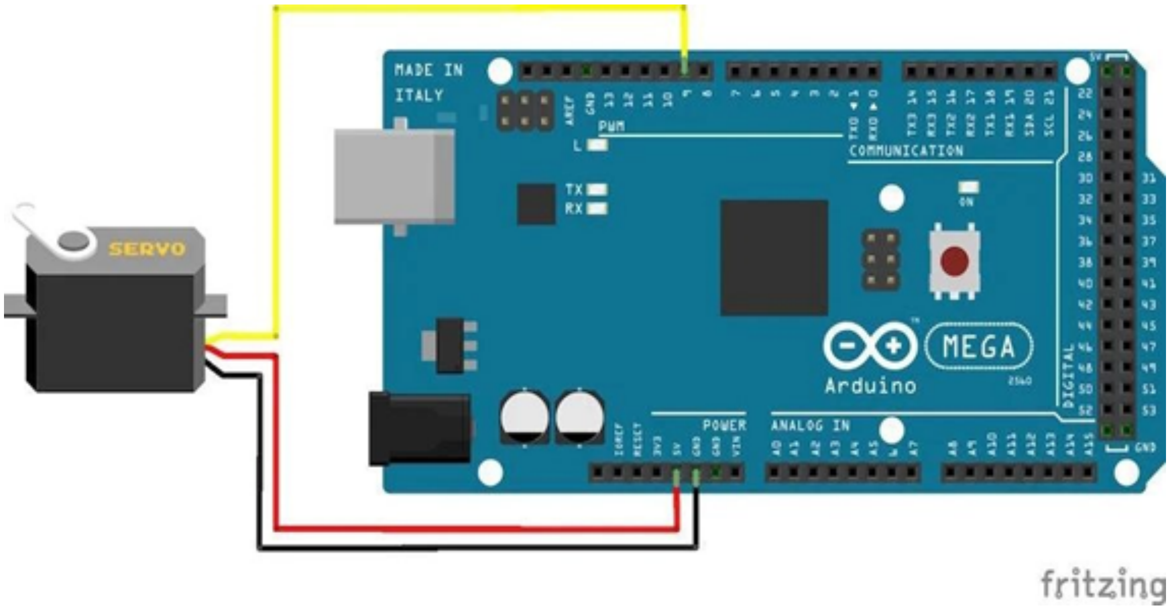
Wire Connection

- RED – Positive
- Brown – Negative
- Orange – Signal



Position "0" (1.5 ms pulse) is middle, "90" (~2 ms pulse) is all the way to the left. ms pulse) is middle, "90" (~2 ms pulse) is all the way to the right, " the left.

- **Servo Motor (SG90) Circuit diagram**



4x4 Matrix Membrane Keypad (#27899) Module:

This 16-button keypad provides a useful human interface component for microcontroller projects. Convenient adhesive backing provides a simple way to mount the keypad in a variety of applications.

Features

- Ultra-thin design
- Adhesive backing
- Excellent price/performance ratio
- Easy interface to any microcontroller
- Example programs provided for the BASIC Stamp 2 and Propeller P8X32A microcontrollers

Key Specifications

- Maximum Rating: 24 VDC, 30 mA
- Interface: 8-pin access to 4x4 matrix
- Operating temperature: 32 to 122 °F (0 to 50°C)
- Dimensions: Keypad, 2.7 x 3.0 in (6.9 x 7.6 cm)
- Cable: 0.78 x 3.5 in (2.0 x 8.8 cm)

How it Works

Matrix keypads use a combination of four rows and four columns to provide button states to the host device, typically a microcontroller. Underneath each key is a pushbutton, with one end connected to one row, and the other end connected to one column. These connections are shown in Figure 1.

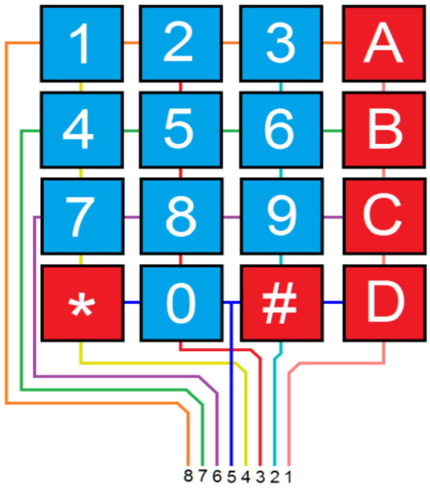


Figure 1: Matrix Keypad Connections

In order for the microcontroller to determine which button is pressed, it first needs to pull each of the four columns (pins 1-4) either low or high one at a time, and then poll the states of the four rows (pins 5-8). Depending on the states of the columns, the microcontroller can tell which button is pressed. For example, say your program pulls all four columns low and then pulls the first row high. It then reads the input states of each column, and reads pin 1 high. This means that a contact has been made between column 4 and row 1, so button ‘A’ has been pressed.

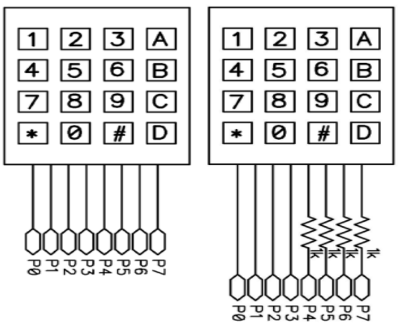
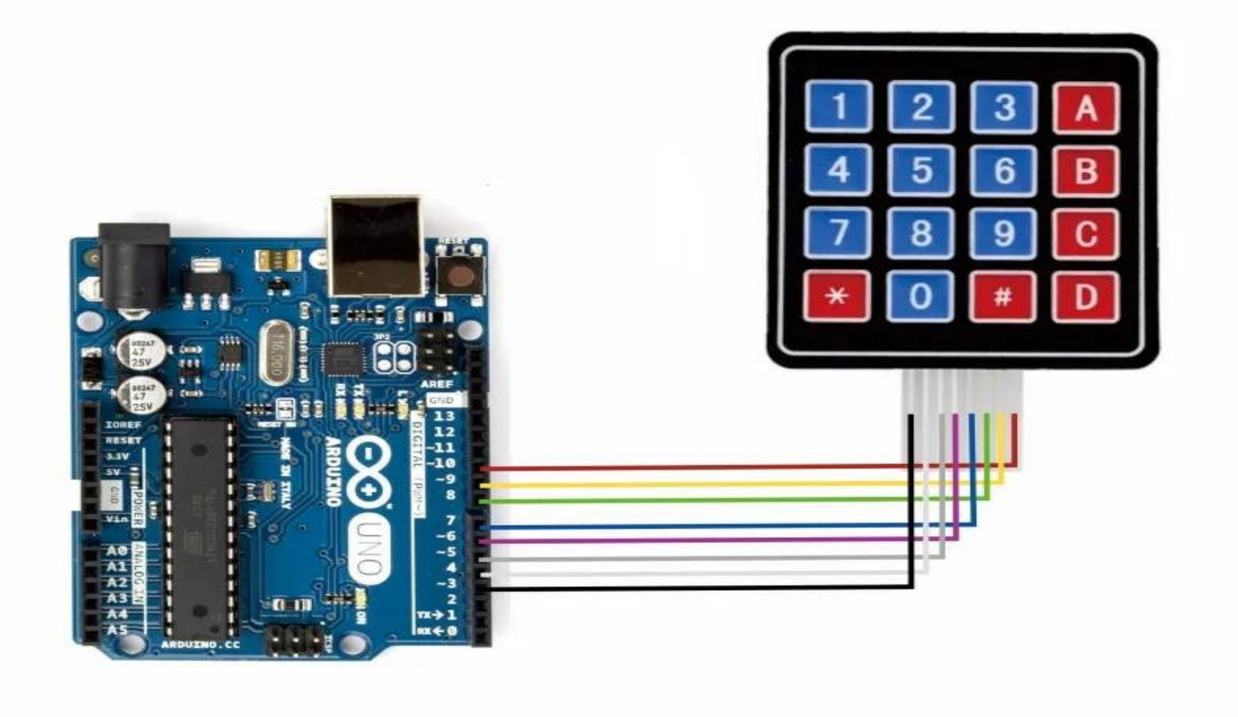


Figure 2 For use with the BASIC Stamp example program listed below.
Figure 3 For use with the Propeller P8X32A example program listed below.

Keypad module circuit diagram



8.4 Member – Vithushan K. (224206P)

Responsibilities:

- **Stepper Motor with ULN2003 Driver Integration:**

I studied and documented the operation of the 28BYJ-48 stepper motor with the ULN2003 driver, and established the electrical connections between the motor, driver, and the Arduino. This involved wiring the control pins (IN1, IN2, IN3, IN4) from the ULN2003 driver to the Arduino and ensuring a proper power supply for smooth motor operation. I programmed the Arduino to control the stepper motor's movement, allowing precise control over rotation angles and speed. This integration allowed the stepper motor to perform various functions, such as rotating dials or driving mechanical components in automation projects.

- **Water Pump 120L/H 3-6VDC Integration:**

For the water pump, I designed the electrical connections and configured the control system using a relay or transistor to manage the pump's on/off operations. I connected the pump to the Arduino through a relay module, ensuring that the Arduino could switch the pump on and off safely without drawing excessive current. I programmed the control logic to automate the water flow based on input from sensors or timed operations, enabling precise control over the pump in applications such as automated irrigation or liquid transfer systems.

- **Studying Components and Programming:**

I played a crucial role in researching the stepper motor and water pump specifications, choosing the appropriate driver (ULN2003) for the stepper motor and determining the best control mechanism (relay or transistor) for the water pump. After reviewing the technical data, I developed the necessary code to interface both components with the Arduino, ensuring smooth and efficient operation. For the stepper motor, I implemented precise control over rotation, and for the water pump, I automated its operation based on external triggers such as moisture levels or timing.

- **Code Integration:**

I integrated the control logic for both the stepper motor and the water pump into the overall system. This involved ensuring that the stepper motor's rotational control could be triggered based on user input or system conditions, and the water pump could be activated or deactivated based on sensor data. I synchronized the operation of both components with other parts of the system, ensuring that the stepper motor could move in response to system changes while the water pump operated efficiently in response to environmental triggers.

Contribution:

I made significant contributions to the successful development and operation of the system by integrating the stepper motor and water pump into the overall project. I was responsible for designing the electrical connections, programming the control logic, and ensuring smooth communication between the components and the Arduino. By carefully studying the specifications and implementing efficient control strategies, I ensured that both the stepper motor and water pump operated effectively, providing precise mechanical movement and reliable water flow in various applications. Additionally, I played a key role in assembling and testing the system, ensuring that all components functioned harmoniously to meet the project's objectives.

Stepper Motor with ULN2003 Driver

The Stepper Motor with ULN2003 Driver is a popular motor control system used in projects requiring precise control of motor position and speed. The stepper motor moves in discrete steps, allowing for fine control over its rotation, while the ULN2003 driver simplifies the interface between the motor and microcontrollers like Arduino. The driver can control the flow of current through the motor coils, enabling the step-by-step movement of the motor, making it ideal for applications like robotics, 3D printers, and CNC machines.

Specifications

Stepper Motor:

- **Type:** 28BYJ-48 Stepper Motor
- **Rated Voltage:** 5V DC
- **Step Angle:** 5.625°/64 (full step)
- **Power Consumption:** 240 mA (typical)
- **Gear Ratio:** 1/64 (for higher precision)
- **Number of Phases:** 4
- **Operating Speed:** 15-20 RPM (approx.)

- **Torque:** 34 N·mm (at 5V DC)
- **Coil Resistance:** 50Ω (per phase)

ULN2003 Driver Module:

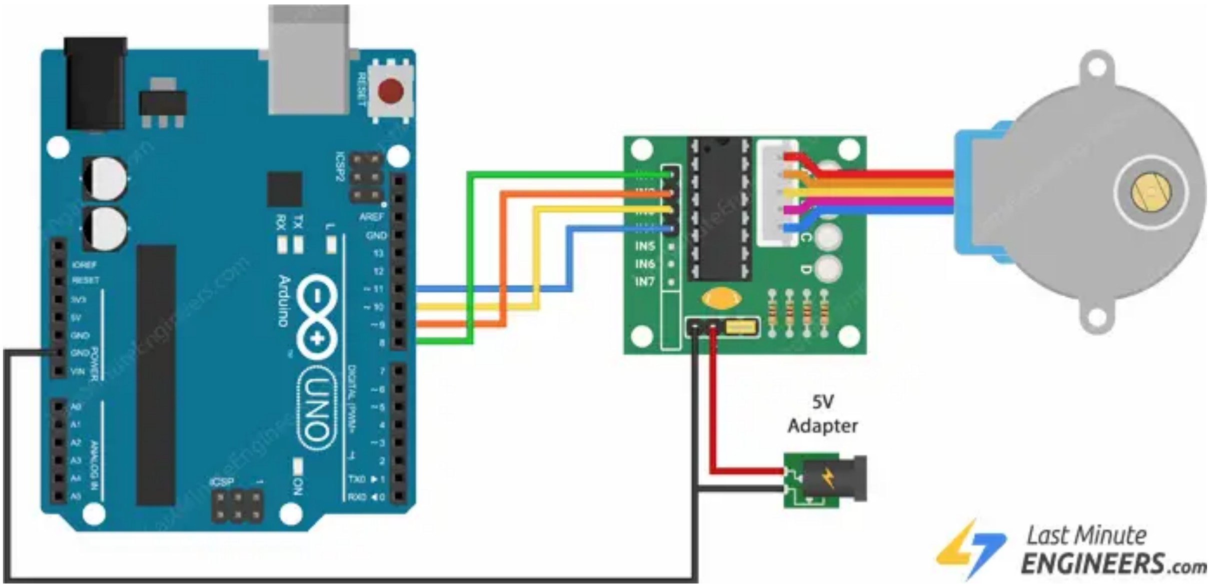
- **Input Voltage:** 5V DC
- **Number of Channels:** 7 Darlington transistor arrays
- **Maximum Output Current (per channel):** 500mA
- **Pin Interface:** 4 input pins for stepper control
- **Connector:** 4-pin motor connector for stepper motor

Wire Connections:

- **ULN2003 IN1 :** Arduino Pin 8
- **ULN2003 IN2 :** Arduino Pin 9
- **ULN2003 IN3 :** Arduino Pin 10
- **ULN2003 IN4 :** Arduino Pin 11
- **ULN2003 GND :** Arduino GND
- **ULN2003 VCC :** Arduino 5V
- **Stepper Motor :** ULN2003 Motor Connector (4-pin header)

How It Works:

The ULN2003 driver allows the Arduino to control the 28BYJ-48 stepper motor with ease. The ULN2003 translates the low-current control signals from the Arduino into high-current signals necessary to drive the stepper motor's coils. The motor works by energizing each coil in a specific sequence, causing the motor to rotate in discrete steps. The number of steps controls the motor's position, while the speed of stepping determines the motor's rotation speed. This precise control is commonly used in robotics, CNC machines, and other automation tasks where accuracy is critical.


5V
Adapter
Last Minute
ENGINEERS.com

Water Pump 120L/H 3-6V DC

The 120L/H 3-6V DC Water Pump is a compact, low-power pump used to move water in small projects, such as hydroponics, aquariums, or irrigation systems. It operates at a voltage range of 3-6V, making it compatible with common power supplies like batteries or microcontroller setups. With a flow rate of 120 liters per hour, it is effective for small-scale water circulation, cooling systems, or liquid transport applications. Its simple design makes it easy to integrate into DIY electronics projects where water movement is needed.

Specifications:

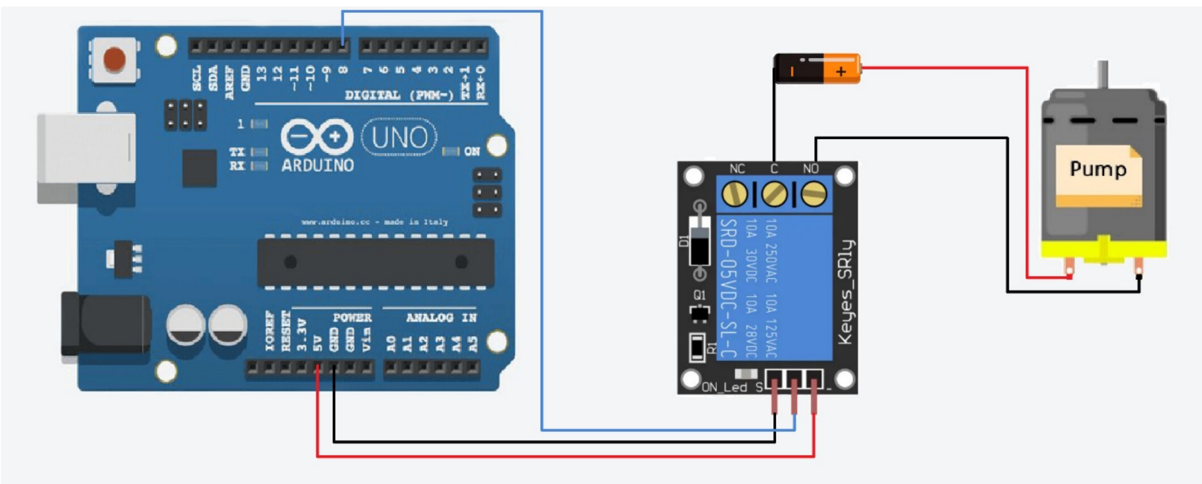
- **Rated Voltage:** 3-6V DC
- **Power Consumption:** 0.4-1.5W
- **Maximum Flow Rate:** 120 liters per hour (L/H)
- **Maximum Lift Height:** 40-110 cm (depends on voltage)
- **Operating Current:** 100-200mA
- **Noise Level:** < 35 dB
- **Water Inlet/Outlet Diameter:** 8mm (approx.)
- **Cable Length:** 1 meter (approx.)
- **Dimensions:** Typically around 45mm x 24mm x 30mm (varies by model)
- **Operating Temperature:** 0°C to 50°C

Wire Connections:

- **Water Pump Positive (Red) :** Relay NO (Normally Open) Pin
- **Water Pump Negative (Black) :** Arduino GND
- **Relay VCC :** Arduino 5V
- **Relay GND :** Arduino GND
- **Relay IN :** Arduino Pin 7

How It Works:

The water pump operates by using a small DC motor to rotate an impeller, which draws water in through an inlet and forces it out through an outlet, creating a flow of water. The pump is powered by a 3-6V DC power supply, and its operation is typically controlled by a relay or transistor circuit connected to an Arduino. The Arduino sends a signal to activate the pump, allowing it to circulate water as needed. This is useful in projects like automatic irrigation systems, water fountains, or aquariums, where water flow needs to be controlled automatically based on certain conditions (e.g., soil moisture, timing).



8.5 Member – Lingaraj U. (224116N)

Responsibilities:

- **IR Sensor Integration:**

I examined and documented the functionality of the IR sensor and established the electrical connections between the IR sensor and the Arduino Mega 2560. I then programmed the Arduino Mega to manage the IR sensor, enabling it to display relevant data, including available juice flavours and sugar levels. Should the sensor detect insufficient quantities of flavour or sugar, it will automatically identify this and relay the information to the Arduino Mega.

- **Buzzer Integration:**

For the buzzer, I designed the electrical connections and programmed it to operate with the Arduino Mega 2560. I developed code to activate the buzzer in cases where insufficient ingredients are detected in the container. Additionally, the buzzer alerts when the machine is ready to dispense juice but the cup is not detected by the ultrasonic sensor. The buzzer is also integrated to respond to keyboard inputs.

- **Studying Components and soldering:**

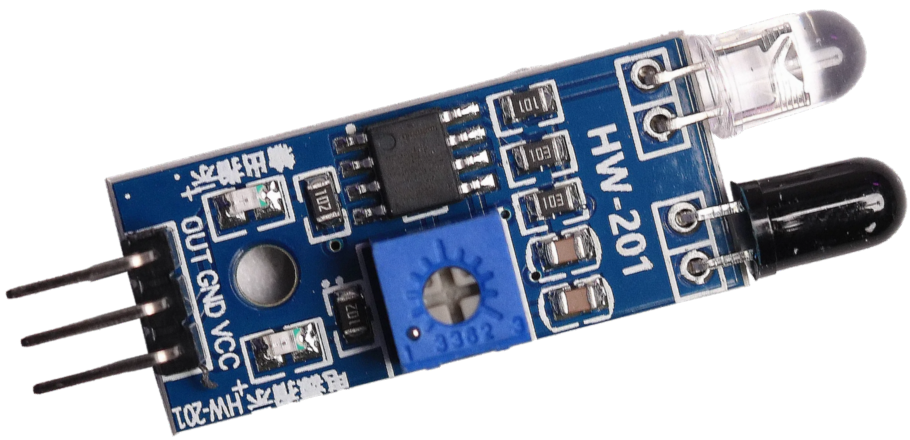
In our project, I was instrumental in studying the various components related to the IR sensor and buzzer. I selected the IR sensor to measure sugar and flavour levels, while the buzzer was utilized to signal errors. After a comprehensive review of technical specifications and operational details, I successfully programmed the IR sensor to monitor the selected juice flavours and sugar levels. Additionally, I effectively implemented the buzzer for corresponding operations, thereby enhancing the overall user interface of the juice maker. and I soldering all other necessary components in the dot board.

Contribution:

I made significant contributions to the development and completion of the juice maker project, with a focus on dot board soldering, the buzzer, and the IR sensor. By examining the specifications of these components, I designed the electrical connections and programmed the Arduino Mega to interface effectively with both modules. I implemented logic to indicate errors using the buzzer and to measure flavor and sugar levels through the IR sensor. Additionally, I played a major role in assembling the overall system, ensuring that all components were securely mounted and functioned seamlessly together.

IR sensor

An IR (Infrared) sensor detects infrared radiation, typically emitted by objects as heat. It consists of an emitter, which sends out IR light, and a detector, which senses reflections or changes in IR light. IR sensors are commonly used in remote controls, motion detection, and



temperature measurement devices.

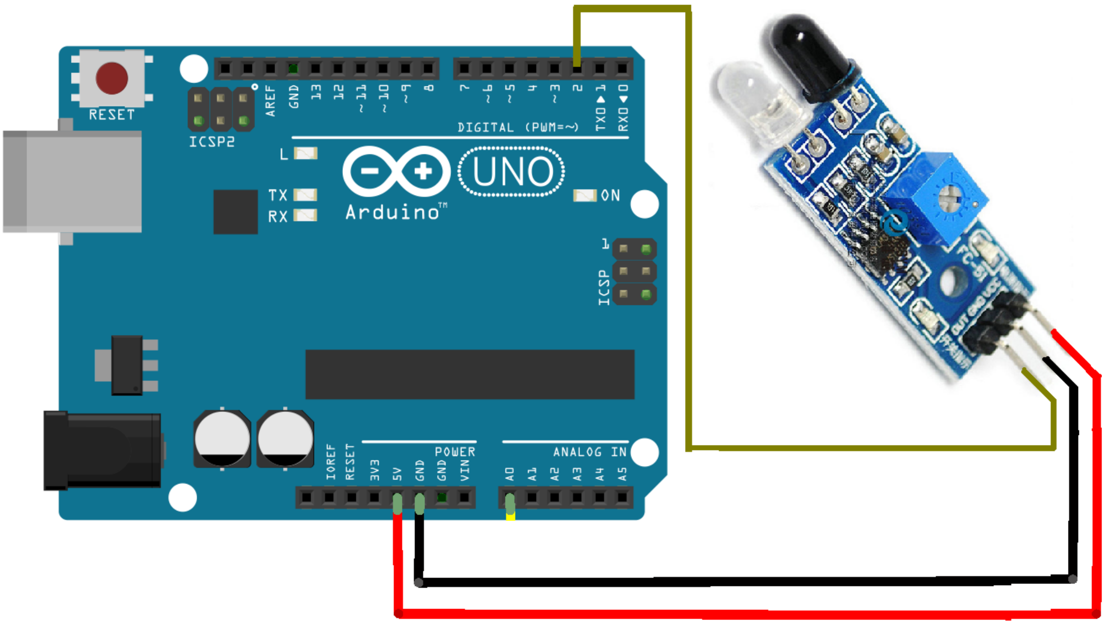
Specifications:

- **Operating Voltage:** 3.3V to 5V
- **Operating Current:** 10mA to 50mA
- **Measuring Range:** 2 cm to 30 cm
- **Accuracy:** ±1 cm to ±5 cm
- **Operating temperature: ±0.5°C to ±3°C**
- **Operating Frequency:** 30 kHz to 40 kHz
- **Connection Pins:**
  - **VCC**: Power supply pin, typically connected to 3.3V or 5V.
  - **GND**: Ground pin for completing the circuit.

- **OUT**: Output pin that provides the sensor's reading, either as an Analog signal or Digital signal depending on the sensor type.

How It Works:

In your juice-making machine, the IR sensor detects the flavour and sugar levels in powder form by emitting infrared light. The powders affect the reflection or absorption of the IR light based on their density and composition. The microcontroller interprets the sensor's output to determine the amount of flavour and sugar powder in the container for accurate measurements.



Buzzer

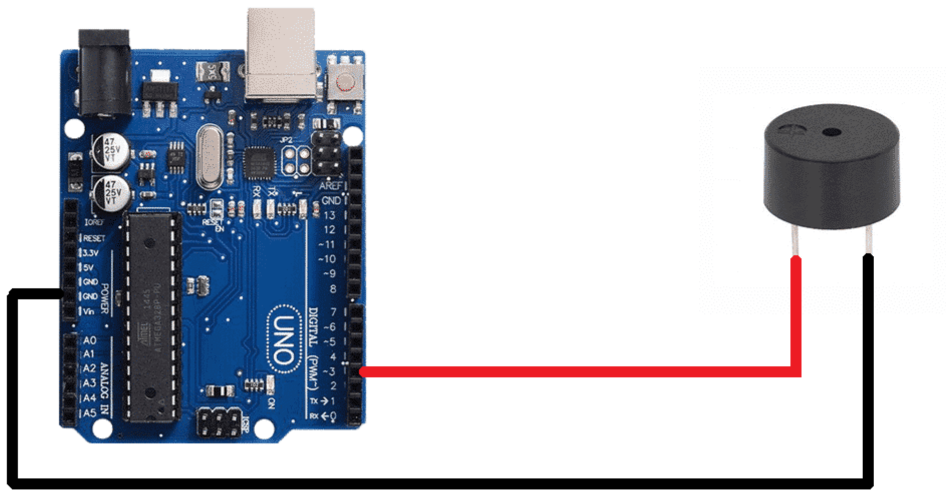
A buzzer is an electronic component that produces sound when electrical current passes through it. It consists of a piezoelectric element or an electromagnetic coil that generates vibrations. Buzzers are used for alerts, notifications, and alarms in various devices, including appliances, electronics, and security systems.

Specifications:

- **Operating Voltage:** 3V- 5V
- **Operating Current:** ≤32 mA
- **Sound Output:** 10cm with 2048Hz Square Wave: ≥85 dB
- **Frequency Range:** 50 ~ 14,000 Hz
- **Resonant Frequency:** 2048 Hz
- **Operating Temperature:** -20 to 60°C
- **Connection Pins:**
    - **VCC (+)**: The positive power supply pin, connected to the voltage source.
    - **GND (−)**: The ground pin, connected to the circuit's ground.

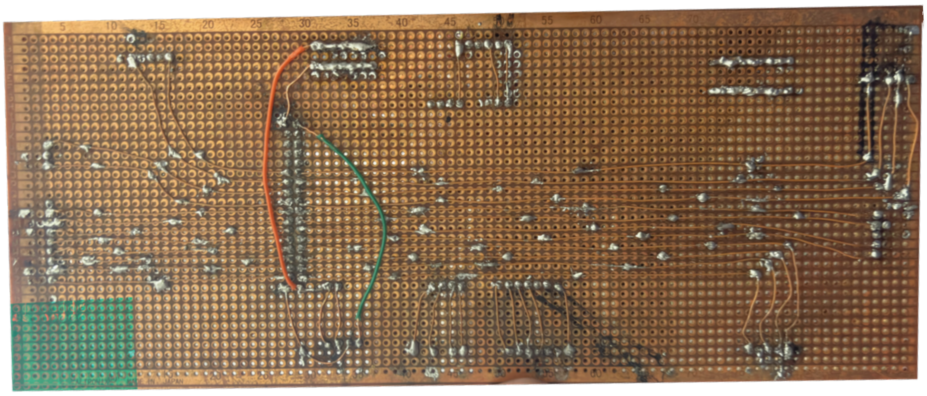
How It Works:

The buzzer in the juice-making machine serves as an alert mechanism to indicate operational errors. If the IR sensor detects insufficient amounts of flavour or sugar, the buzzer is activated. Additionally, if the ultrasonic sensor fails to identify a cup, it sends a signal to the microcontroller, which then triggers the buzzer. Furthermore, the buzzer also activates when user input is received via the keyboard.



Dot Board Soldering

A dot board is used for prototyping electronic circuits. It allows components to be inserted and connected without soldering, facilitating easy modifications and experimentation. Ideal for designing and testing circuits before finalizing with a printed circuit board (PCB), it simplifies circuit assembly and troubleshooting



9. References:

[1]“Circuit Design App for Makers- circuito.io” www.circuito.io/. Accessed 13 June. 2024.

[2]“Contact Us – Scion Electronics” scionelectronics.com/contact-us/. Accessed 13 Apr. 2024

[3] A. Deep, “Interfacing HC-SR04 Ultrasonic Distance Sensor with ATmega32 microcontroller,” electroSome, 16-Mar-2018. [Online]. Available: https://electrosome.com/ultrasonic-distance-sensor-atmega32/amp/. [Accessed: 03-Apr2024].

[4] “DC motor interfacing with AVR ATmega16/ATmega32,” Electronicwings.com. [Online]. Available: https://www.electronicwings.com/avr-atmega/dc-motorinterfacing-with-atmega-1632. [Accessed: 03-May-2024].

[5] “IR sensor with atmega32 and atmel studio 4,” AVR Freaks, 21-Mar-2017. [Online]. Available: https://www.avrfreaks.net/forum/ir-sensor-atmega32-and-atmel-studio-4. [Accessed: 03-Apr-2024].

[6]“Tutorial 6 Simple Water Level Sensor Circuit With Led Display And Buzzer Using Arduino Uno.” YouTube, uploaded by CirKit3, 25 October 2020, www.youtube.com/watch?v=gUBDmE27F_U.

[7]“Alldatasheet.Com” Electronic Parts Datasheet Search, www.alldatasheet.com/. Accessed 13 June. 2024