Smart Waiter Robot

- WAITER 117 -

PROJECT REPORT

SMART WAITER ROBOT

Department of Computer Science
University of Sri Jayewardenepura

ICT 305 2.0

Embedded System

K.A.N.N. Kodikara

ACKNOWLEDGEMENT

I am over helmed in all humbleness and gratefulness to acknowledge my depth to all those who have helped me to put these ideas, well above the level of simplicity and into something concrete.

I would like to express my special thanks of gratitude to DR. M.D.R.Perera , senior lecturer in Department of Computer Science, Faculty of Applied Science, University of Sri Jayewardenepura who gave me the golden opportunity to do this wonderful project smart cap, which also helped me in doing a lot of searches and I came to know about so many new things. I am thankful to them.

Any attempt at any level can 't be satisfactorily completed without the support and guidance of my parents and friends.

I would like to thank my parents who helped me a lot in gathering different information, collecting data and guiding me from time to time in making this project, despite of their busy schedules, they gave me different ideas in making this project unique.

~ ~ ~ ~

Contents

1. Introduction

1.1. Background of the project

The food service industry has seen significant advancements in automation, driven by the need for efficiency and enhanced customer experiences. In small cafes, where staff resources are often limited, the demand for streamlined operations is even greater. Tasks such as delivering meals to tables can be repetitive and time-consuming, taking valuable time away from staff who could otherwise focus on customer service or operational management. To address these challenges, the idea of a smart waiter robot was developed.

This robot is designed to automate the delivery process, ensuring that meals are served accurately and promptly. It follows a predefined path, making it suitable for small, structured environments. By reducing the dependency on human intervention for routine tasks, the smart waiter robot aims to improve service efficiency, reduce errors, and enhance the overall dining experience.

1.2. Rationale of the project

Manual meal delivery in cafes often results in bottlenecks during peak hours, leading to delays and decreased customer satisfaction. Furthermore, small cafes typically operate with fewer employees, which limits their ability to cater to large crowds effectively. The smart waiter robot provides a practical solution to these issues by automating the delivery process.

The project leverages line-following technology to enable smooth navigation along designated routes and uses an ultrasonic sensor to detect the presence of a tray, ensuring the robot only moves when it is ready to serve. Obstacle avoidance features prevent interruptions in service, while an OLED display adds a touch of engagement by showing expressive emojis, making the dining experience more interactive and memorable.

My Experience -: During my part-time work at Pizza Hut in Sri Lanka, I witnessed this problem firsthand. During busy hours, staff struggled to balance order preparation and delivery, often causing delays in serving customers. This not only impacted on the overall customer experience but also created stress for employees, reducing their efficiency.

1.3. Project aims and Objectives

The primary aim of the project is to create a smart waiter robot capable of automating meal delivery in small cafes. The specific objectives are:

- **Efficient Navigation**: Design a robot that can follow a predefined path using a 5-channel IR sensor for line following.
- **Obstacle Avoidance**: Use sensors to detect and avoid obstacles in its path, ensuring uninterrupted service.
- **Tray Detection**: Incorporate an ultrasonic sensor to detect the presence of a tray and ensure accurate delivery.
- **Customer Interaction**: Enhance engagement through an OLED display that shows emojis, making the robot more appealing.
- **Autonomous Operation**: Ensure the robot can autonomously return to its starting point after completing its task.

1.4. Outcome

The final outcome of this project will be a fully functional smart waiter robot that delivers meals autonomously to designated tables in small cafes. Key features include:

- Reliable and precise line-following capabilities.
- Effective obstacle detection
- Automated tray detection to confirm readiness for service.
- Smooth return to the starting point after completing its task.
- Engaging customer interaction via an OLED display that shows expressive emojis.

The smart waiter robot will not only reduce the manual workload for cafe staff but also introduce a modern and innovative service approach that enhances the customer experience.

2. Literature Review

Various studies have explored the potential for robotics and automation in service industries, particularly in the restaurant and cafe sectors. One significant aspect of such systems is the ability to automate meal delivery, which reduces manual effort and improves service efficiency. A key study by Smith et al. (2021) analyzed the potential of robotic waiters in reducing labor costs and improving operational speed in large-scale restaurants. The research found that robotic systems equipped with sensors such as LiDAR and cameras could navigate through crowded spaces, avoiding obstacles while delivering food to designated tables. However, the study also highlighted that these robots often require complex and expensive sensors, making them impractical for smaller businesses like cafes, which typically operate with simpler layouts and limited budgets.

In contrast, simpler approaches utilizing line-following robots have proven effective in environments with structured paths. Research by Kumar et al. (2020) demonstrated the use of infrared (IR) sensors in autonomous robots that followed marked lines on the floor. The system's simplicity and cost-effectiveness made it ideal for small-scale settings, where the cafe's layout is well-defined. Line-following robots can autonomously navigate from the kitchen to tables, delivering food without human intervention. This approach offers an affordable alternative to more sophisticated systems while maintaining reliable performance in simple environments.

Obstacle detection is another crucial feature for ensuring smooth operation in dynamic environments. Studies by Wang et al. (2019) have shown that ultrasonic sensors can effectively detect and avoid obstacles in real-time. The sensor emits sound waves, calculates the distance based on the time taken for the echo to return, and helps the robot adjust its path accordingly. This technology, when combined with line-following capabilities, allows robots to operate autonomously and navigate around obstacles while delivering meals efficiently. These sensors also provide an economical solution for small robots, such as the smart waiter robot designed for cafes.

Additionally, customer interaction is a key aspect of improving user experience. Research by Zhang et al. (2022) emphasized the role of interactive displays in service robots. The study found that integrating features such as emoji displays or text messaging enhances customer engagement and satisfaction. In the case of a waiter robot, an OLED display showing emojis can create a more personalized and enjoyable experience for customers, making the robot not only functional but also engaging. This feature is particularly beneficial in small cafes, where customer interaction plays a significant role in the overall dining experience.

Despite these advancements, existing waiter robots often face limitations, such as high costs, complexity, and the need for specialized maintenance. The integration of advanced sensors like LiDAR and cameras, while effective in large restaurants, can be prohibitively expensive for small cafes. Moreover, the complexity of such systems requires specialized knowledge to operate and maintain, which can be a barrier for smaller businesses. The present study aims to address these challenges by designing a simple, cost-effective solution for small cafes. By combining IR sensors for line-following, ultrasonic sensors for obstacle detection, and an OLED display for customer interaction, the project creates an affordable and practical smart waiter robot.

2.1 Literature Review Conclusion

The existing literature reveals several key challenges and limitations in the current landscape of service robots for food delivery, especially for small-scale operations. These challenges include:

1. High costs associated with advanced sensors like LiDAR and cameras.
2. Complexity in system design, requiring technical expertise for maintenance.
3. Limited adaptability to simple, structured environments, such as small cafes.

To overcome these issues, several researchers have proposed solutions. For instance, using simple IR and ultrasonic sensors, combined with line-following technology, can create affordable and reliable systems for smaller businesses. This approach not only reduces costs but also ensures ease of use and maintenance. Furthermore, the integration of interactive features like OLED displays enhances the customer experience without significantly increasing the complexity or cost of the system. The current project builds upon these findings by developing a cost-effective, interactive, and autonomous smart waiter robot suitable for small-scale cafes.

3. Components

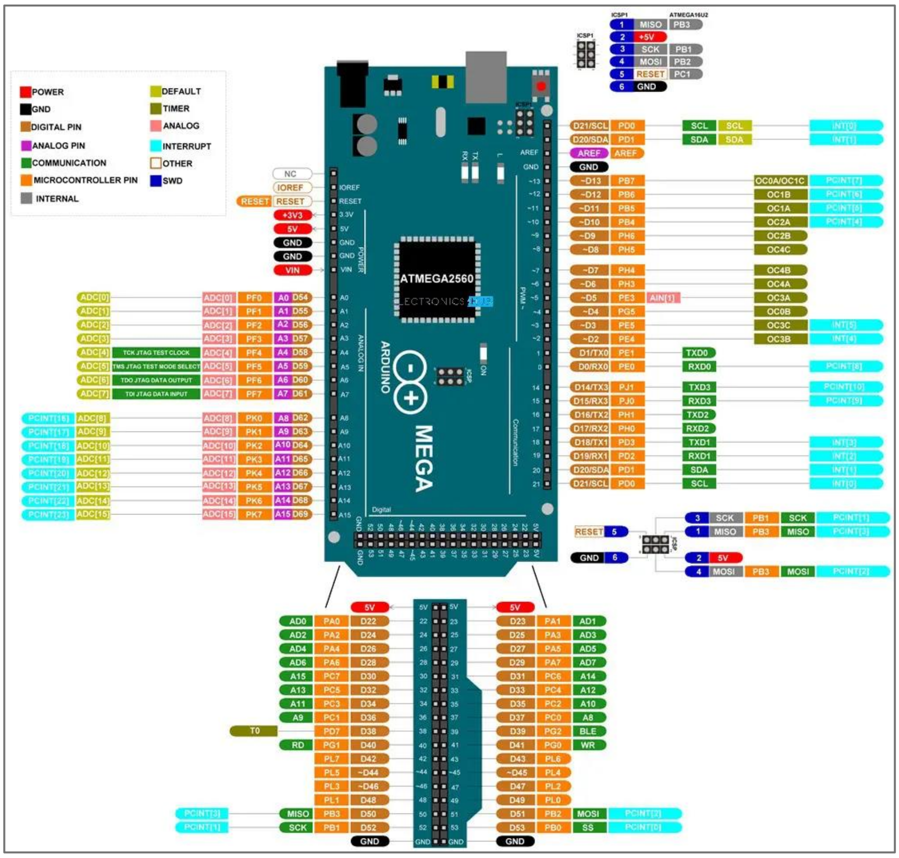
3.1. Hardware requirements

1. Arduino Mega Board
2. Arduino Nano Board
3. Motor Driver
4. Ultrasonic Sensor hc-sr04
5. 5 Channel Tracing IR sensor
6. Chassie
7. Bluetooth Module HC-05
8. Breadboard
9. Jumper Wires
10. OLED Display 0.96 “128x64
11. Battery Pack

1. Arduino Mega Board

The Arduino Mega is a powerful microcontroller board, designed with the ATmega2560 microcontroller at its core. It features 54 digital input/output pins, of which 15 can be used as Pulse Width Modulation (PWM) outputs, and 16 analog input pins. Additionally, it includes a 16 MHz crystal oscillator, a USB connection for programming and communication, a power jack, an ICSP header, and a reset button. With these features, the Mega board provides all the necessary components to operate the microcontroller efficiently. Simply connect it to a computer via USB or power it with an AC-to-DC adapter or battery to start using it.

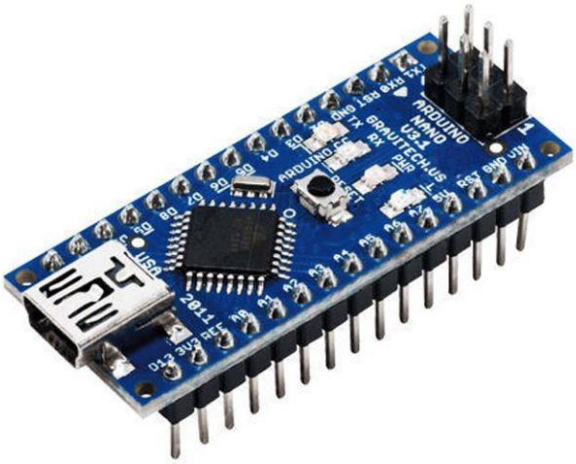
In this project, the Arduino Mega is chosen due to its extended I/O capabilities, which are ideal for complex applications that require multiple connections. Its large community support and compatibility with the Arduino IDE make it a versatile and easy-to-use choice for embedded electronics projects.



2. Arduino Nano Board

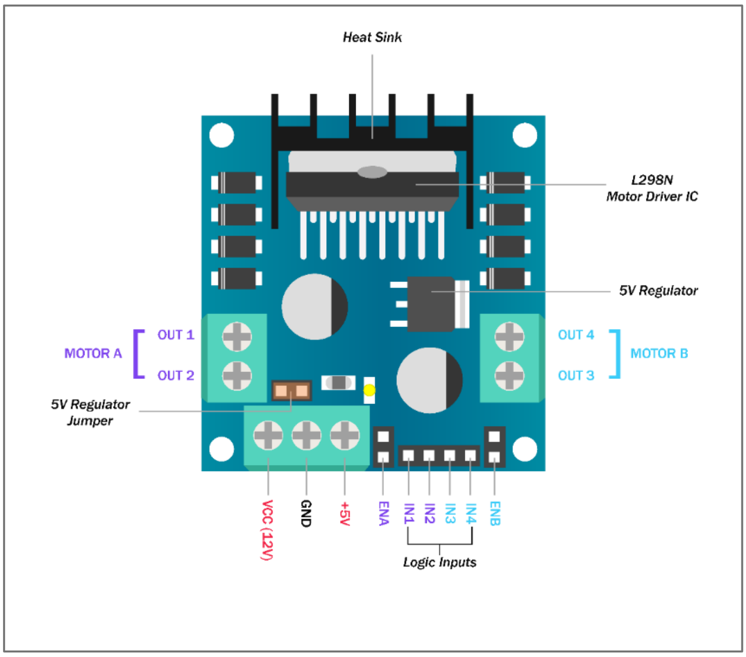
The Arduino Nano is a compact microcontroller board based on the ATmega328P. It features 14 digital input/output pins (6 of which can be used as PWM outputs), 8 analog input pins, and operates at a clock speed of 16 MHz. It can be powered via a mini-USB connection, an unregulated external power supply (6-12V), or a regulated 5V supply, making it flexible for various power setups.

In this project, the Arduino Nano was chosen for its small size, making it ideal for tasks where space is limited. Its ability to seamlessly integrate into breadboards and manage specific tasks independently makes it perfect for running the OLED display system in the smart waiter robot. By offloading the display operations to the Nano, the main controller (Arduino Mega) can focus on core functions like line following and obstacle detection. Additionally, the Nano's low power consumption and compatibility with the Arduino IDE make it an efficient and user-friendly solution for managing dedicated subsystems in this project.



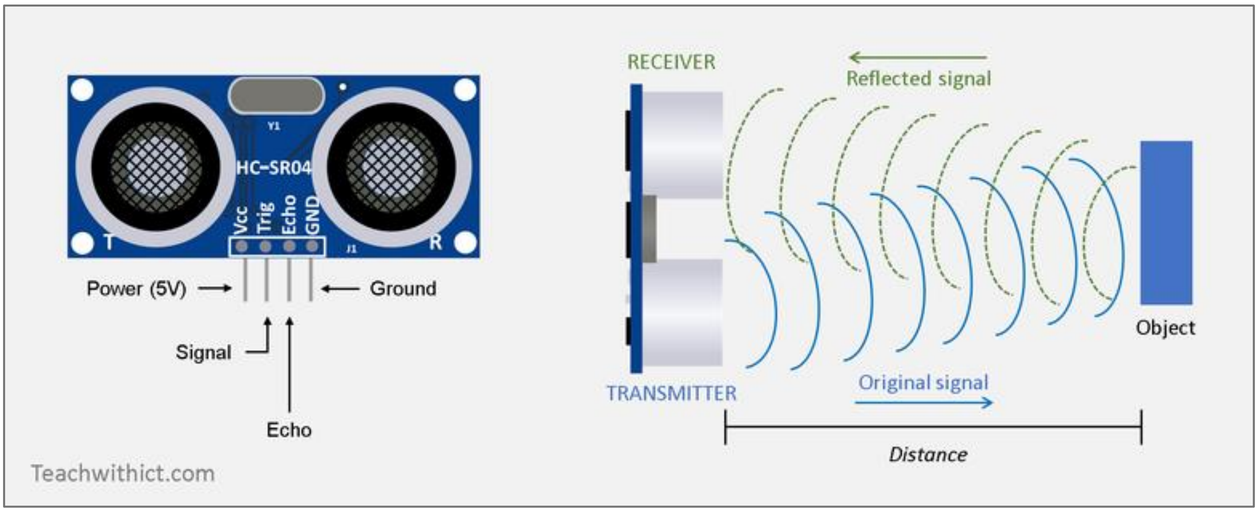
3. Motor Driver

The L298 is a robust dual H-Bridge motor driver IC designed for controlling the speed and direction of two DC motors or one stepper motor. It supports motor supply voltages from 4.5V to 46V and can handle continuous currents of up to 2A per channel, making it suitable for high-power motor control applications. The L298 includes features such as PWM control for speed regulation, thermal shutdown for overheat protection, and current sensing for monitoring motor current. These capabilities make it ideal for use in robotics, CNC machines, and other applications requiring precise motor control and the ability to drive inductive loads like motors and relays.



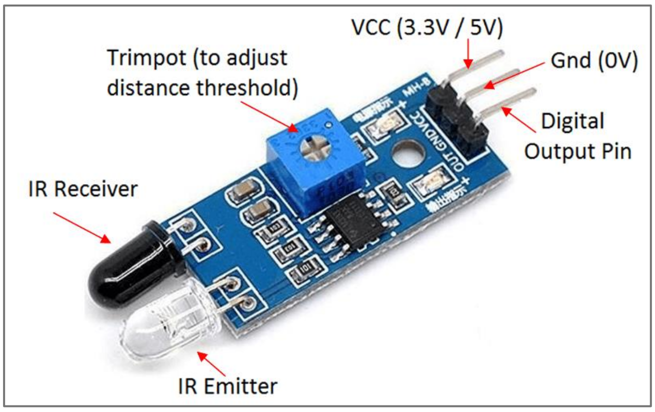
4. Ultrasonic Sensor hc-sr04

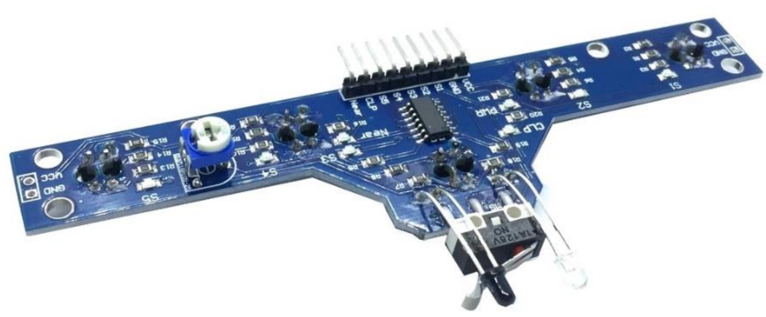
It sends an ultrasonic pulse out at 40kHz which travels through the air and if there is an obstacle or object, it will bounce back to the sensor. By calculating the travel time and the speed of sound, the distance can be calculated. Ultrasonic sensors are a great solution for the detection of clear objects.



5. IR Sensor / 5 Channel Tracing IR Sensor

The IR Sensor (Infrared Sensor) is a widely used electronic device that emits and/or detects infrared radiation to sense its surroundings. It can be used to detect objects and measure distances. Typically, an IR sensor consists of an IR LED (emitter) and an IR photodiode (receiver). When an object is in front of the sensor, the IR light emitted by the LED is reflected to the photodiode, which generates an electrical signal proportional to the intensity of the reflected light. The sensor can be used for proximity sensing, object detection, and line-following in robotic applications. It is commonly found in obstacle avoidance systems, automatic faucets, and remote controls, offering reliable performance and easy integration with microcontrollers and other digital circuits.





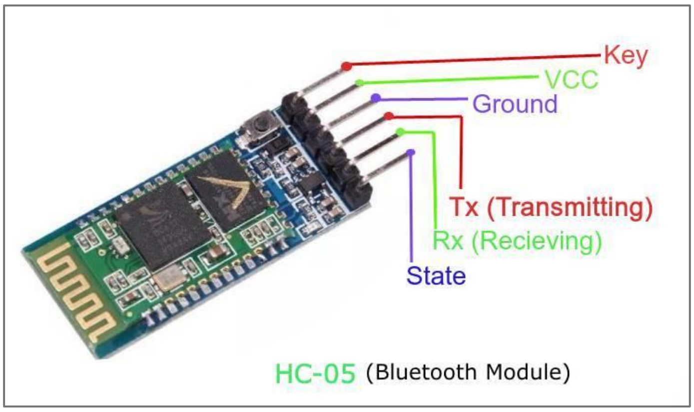
5 Channel Tracing IR Sensor

6. Chassie

A two-wheel circular chassis is ideal for the Arduino-based smart waiter robot, offering compactness and easy maneuverability for smooth, precise navigation. The circular shape enables effortless turns, while the two-wheel design, supported by a caster wheel, allows differential steering without complex mechanisms. This setup easily accommodates essential components like the IR sensor for line following and obstacle avoidance, display screen for customer interaction, and tray sensor, all in a streamlined layout suited for efficient, reliable operation.

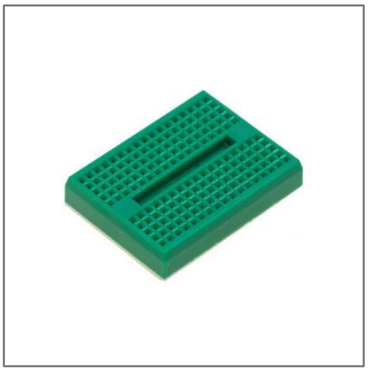
7. Bluetooth Module

The HC-05 is a Bluetooth module commonly used for wireless communication between microcontrollers and mobile devices. In this smart waiter robot system, the HC-05 enables the user to send table numbers directly from a mobile phone to the robot. Once paired with the mobile device, it receives table number inputs and transmits them to the Arduino, allowing the robot to navigate to the specified table. This makes operation more convenient and allows for quick, remote input without physical controls.



8. Breadboard

A 35mm x 47mm breadboard is perfect for prototyping in the smart waiter robot. Its compact size fits easily on the chassis, allowing quick, solder-free connections for components like sensors and the Bluetooth module.



9. Jumper Wires

Jumper wires are used to connect different components in a circuit, typically on a breadboard, without the need for soldering. They come in various types, including male-to-male, female-to-female, and male-to-female, to suit different pin configurations. These wires are essential for prototyping and testing electronic circuits.

10. OLED Display 0.96 "128x64

The 0.96" 128x64 OLED display is a compact, high-contrast screen ideal for displaying information on the smart waiter robot. With its 128x64 resolution, it provides clear visuals for text and simple graphics, such as emojis for customer interaction. Its small size and low power consumption make it perfect for Arduino projects where space and efficiency are important.

11. Battery Pack

The **3200mAh 3.7V 18650 Li-Ion battery** is a high-capacity rechargeable battery commonly used in electronics. With a 3200mAh capacity, it provides significant power for devices like robots and power banks. Proper battery management, including safe charging and voltage regulation, is crucial when using these batteries in projects to ensure safe operation and longevity.

3.2. Software requirements

1. Arduino IDE

Arduino IDE is an **open-source software**, designed by Arduino.cc and mainly used for writing, compiling & uploading code to almost all Arduino Modules. It is an official Arduino software, making code compilation so easy that even a common person with no prior technical knowledge can get their feet wet with the learning process. It is available for all operating systems i.e. MAC, Windows, Linux and runs on the Java Platform that comes with inbuilt functions and commands that play a vital role in debugging, editing and compiling the code. A range of Arduino modules are available including Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro and many more. Each of them contains a microcontroller on the board that is programmed and accepts the information in the form of code. The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board. The IDE environment mainly contains two basic parts: Editor and Compiler where former is used for writing the required code and later is used for compiling and uploading the code into the given Arduino Module.

This environment supports both **C and C++** languages.

2. Arduino Blue Control

For my project, I'm using the **Arduino Blue Control app**, which allows me to wirelessly control my smart waiter robot via Bluetooth. This app enables seamless communication between the Arduino and a smartphone using Bluetooth modules like HC-05 or HC-06. It helps in sending commands and receiving data, such as notifying a designated table or activating specific functions in the robot, providing a convenient way to manage the system remotely and interact with it in real-time.

4. Project Feasibility

1. Equipment Feasibility

Name	Purchase/Personally Available
5 Channel Tracing IR sensor	Personally Available
Arduino Mega	Personally Available
Batteries	Personally Available
Bluetooth Module (HC-05)	Purchase
Breadboard	Purchase
Chassie	Personally Available
Jumper Wires	Purchase
Motor Driver	Personally Available
OLED Display 0.96 "128x64	Purchase
Ultrasonic Sensor	Purchase
Wooden Sticks	Purchase

2. Knowledge Feasibility

Knowledge of Electronic Embedded System	Studied
Knowledge of Arduino programming	Studied
Knowledge of Internet of Things	Studied
Knowledge of Computer	Studied
Knowledge of connecting equipment	Studied
Knowledge of PID Controlling and Error Calculation	Studied

5. Project Cost

Name	**Price**		**Purchase/Personally Available**
5 Channel Tracing IR sensor	LKR	860 /=	Personally Available
Arduino Mega	LKR	3, 950 /=	Personally Available
Battery Pack	LKR	1, 730 /=	Purchase
Bluetooth Module (HC-05)	LKR	980 /=	Purchase
Breadboard	LKR	50 /=	Purchase
Chassie	LKR	1, 150 /=	Personally Available
Jumper Wires	LKR	200 /=	Purchase
Motor Driver	LKR	460 /=	Personally Available
OLED Display 0.96 “128x64	LKR	630 /=	Purchase
Ultrasonic Sensor	LKR	240 /=	Purchase
Wooden Sticks	LKR	500 /=	Purchase
Arduino Nano	LKR	1,160/=	Personally Available (From Lab)

Required cost LKR 11, 940.00 /= **Actual cost** LKR 4360.00 /=

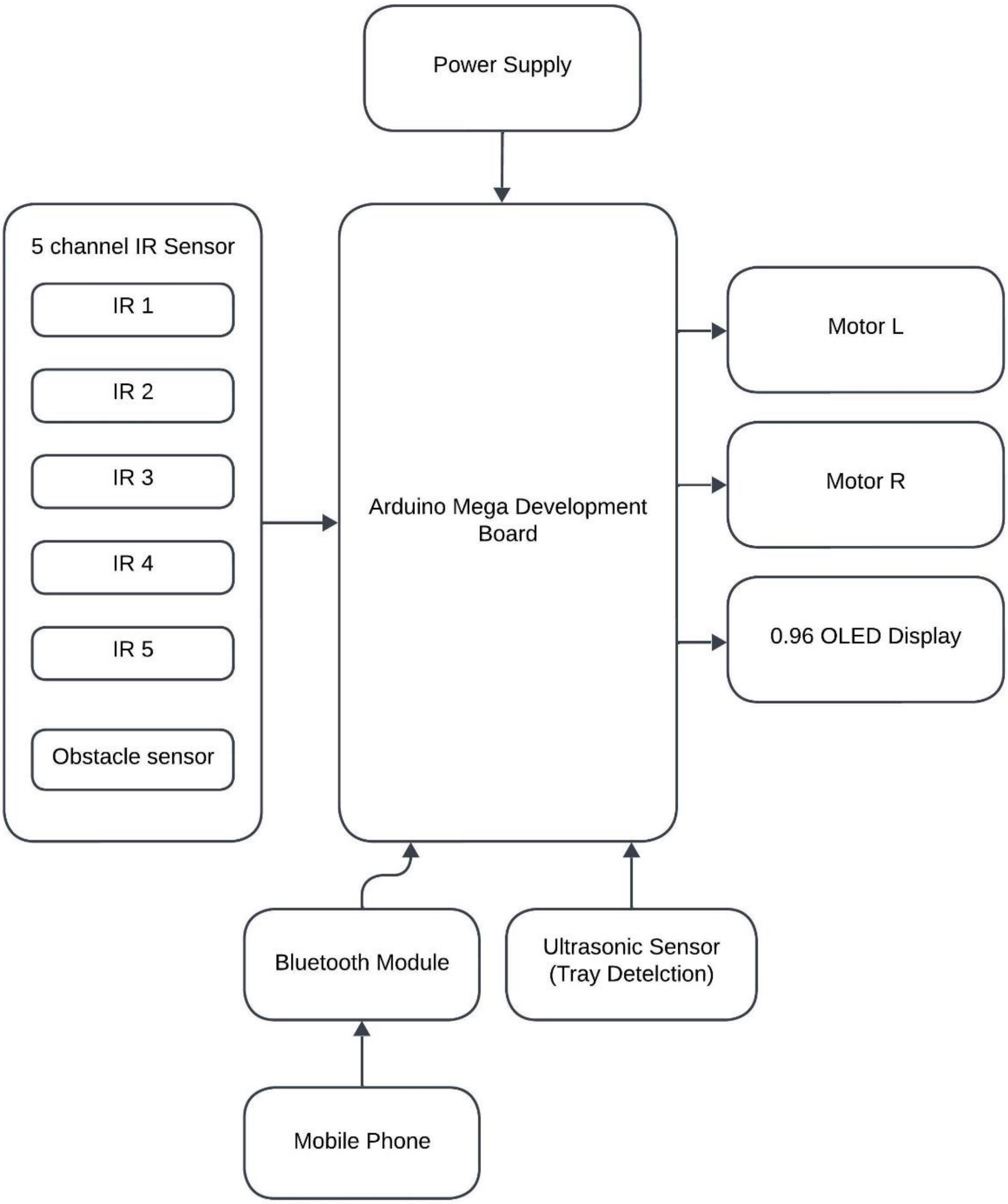
6. Methodology

6.1. Concept behind the project

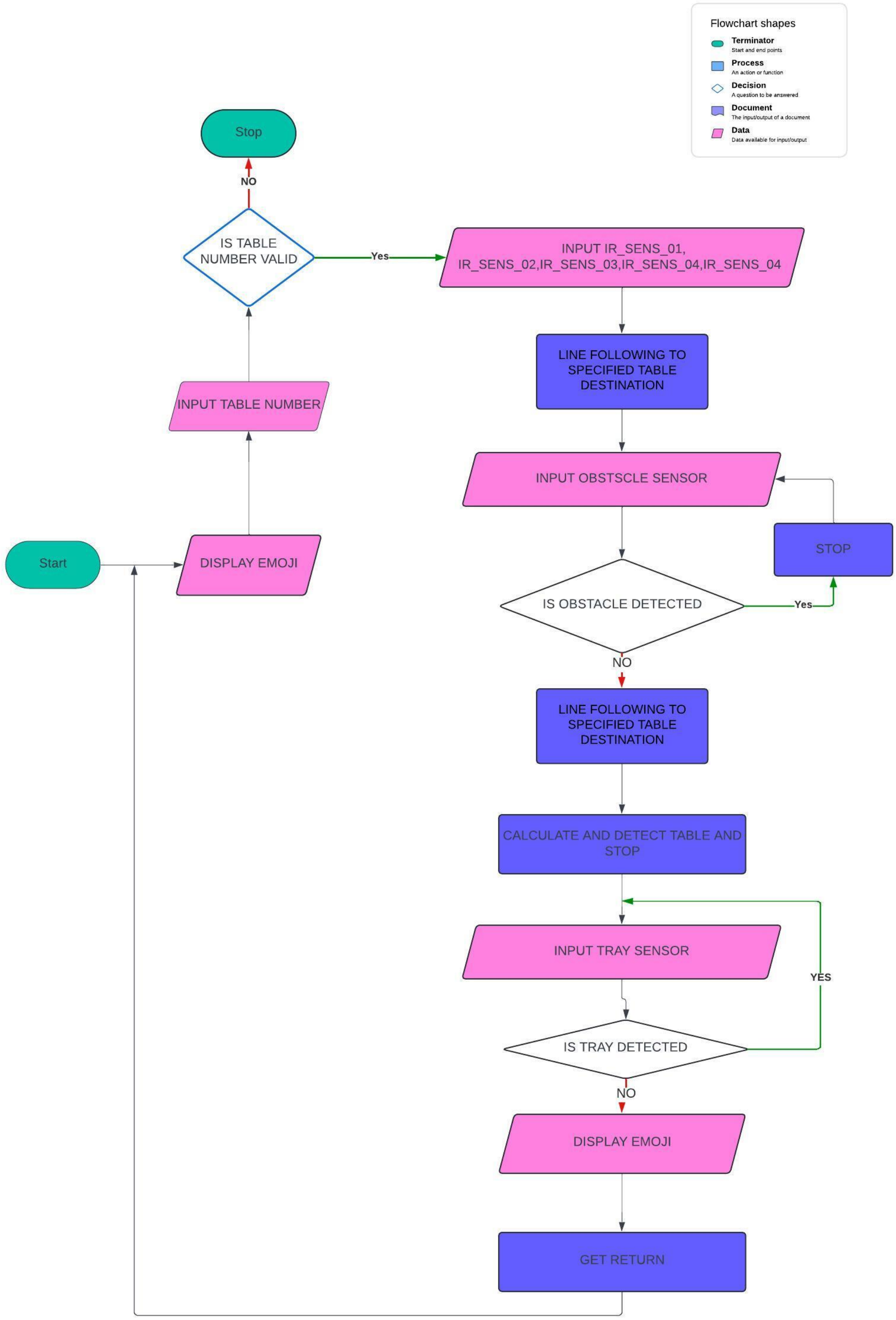
The main concept behind the **Smart Waiter Robot** project is **autonomous meal delivery** in small cafes. The robot uses **line-following technology** to navigate predefined paths between the kitchen and designated tables, ensuring accurate and efficient meal delivery. Equipped with a **5-channel IR sensor**, the robot detects and follows lines on the floor, making it ideal for structured environments.

An **ultrasonic tray sensor** is integrated to detect the presence of a tray on the robot. If the tray is removed, the robot assumes the delivery is complete and automatically returns to its starting position. Additionally, the robot utilizes the **Arduino BlueControl app** to receive Bluetooth signals for table notifications, allowing staff to send commands conveniently. An **OLED display** adds interactivity by showing expressive emojis to enhance customer engagement, making the dining experience more enjoyable.

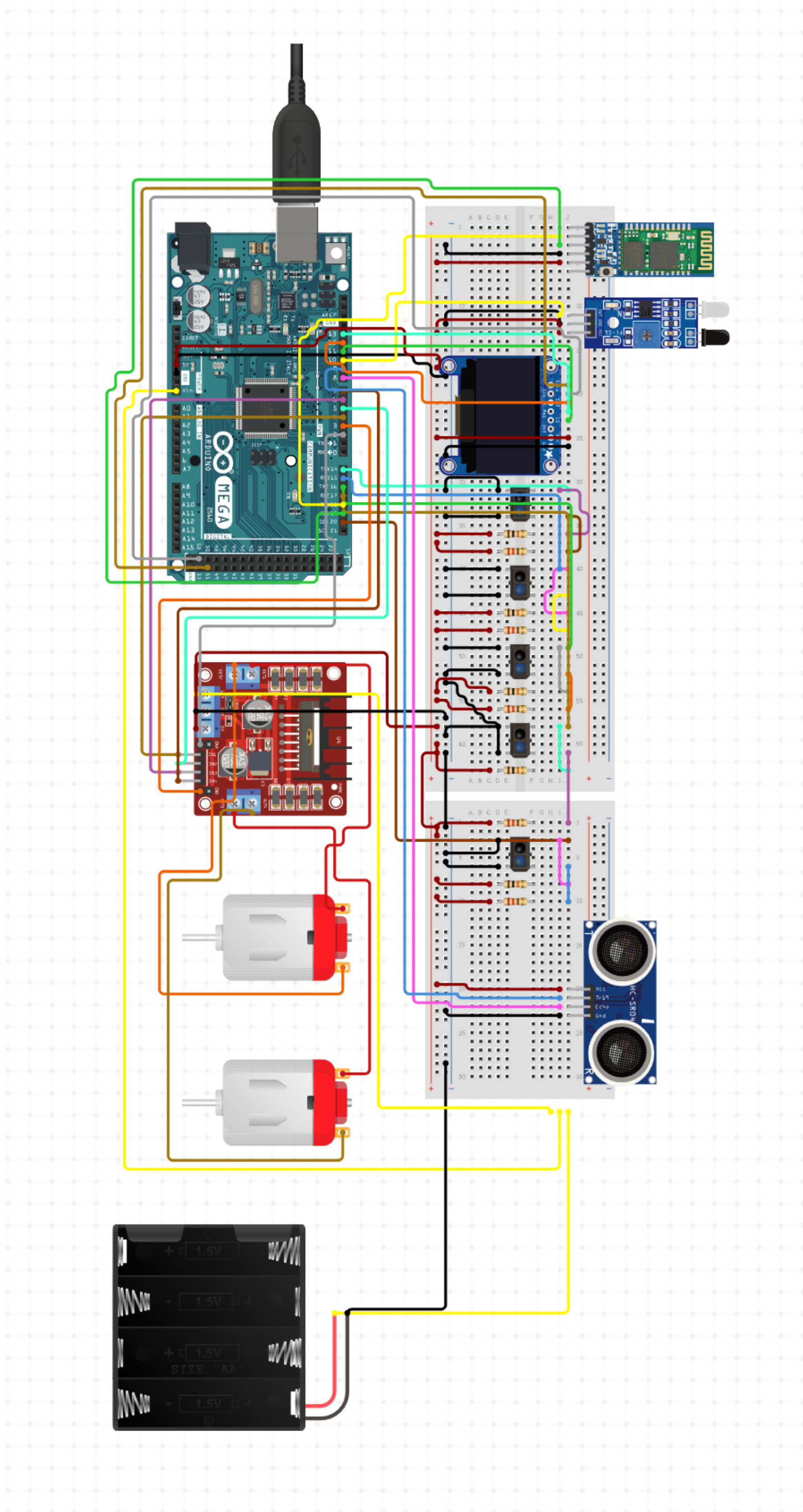
6.2. System Block Diagram



6.3. System Architecture (Flow chart)



6.4. System connectivity diagram (circuit diagram)



7. Procedure

Step 1: Assemble the Chassis and Wheels

Start by attaching the two wheels and the caster wheel to the chassis. Ensure the wheels are firmly connected to the motors and aligned properly to enable smooth movement.

Step 2: Connect the IR Sensors for Line Following

Place the 5-channel IR sensor (BFD-1000) at the front bottom of the chassis, ensuring it can detect the lines on the floor. Connect the sensor pins (S1 to S5) to the Arduino Mega analog pins (A0 to A4) as per the design.

Step 3: Integrate the Ultrasonic Tray Sensor

Mount the ultrasonic sensor near the tray area of the robot to detect whether the tray is in place. Connect its trigger and echo pins to the Arduino Mega digital pins.

Step 4: Wire the L298N Motor Driver

Connect the L298N motor driver to the motors for controlling the wheels. Wire the ENA and ENB pins to pins 6 and 7 of the Arduino Mega, and IN1, IN2, IN3, and IN4 to pins 8, 9, 10, and 11, respectively.

Step 5: Set Up the OLED Display

Attach the 0.96" OLED display to the front of the robot. For displaying expressive emojis that enhance customer interaction, the display is controlled using an **Arduino Nano**. The Nano is connected to the OLED via the appropriate I2C pins (SDA and SCL).

Step 6: Configure the Bluetooth Module

Connect the HC-05 Bluetooth module to the Arduino Mega for wireless communication. Use the **Arduino BlueControl app** to send table signals to the robot.

Step 7: Load the Arduino Code

Write and upload the Arduino code using the Arduino IDE. The code should include instructions for line following, obstacle avoidance, tray detection, and displaying emojis on the OLED.

Step 8: Perform Initial Testing

Power the system using the 3200mAh 3.7V 18650 batteries. Test the line-following function by placing the robot on a track with a black line and ensuring it follows the path smoothly.

Step 9: Test Tray Detection and Delivery

Place a tray on the robot and verify the ultrasonic sensor detects it. Remove the tray to ensure the robot recognizes the delivery is complete and returns to its starting position.

Step 10: Test Bluetooth Signal

Use the Arduino Blue Control app to send commands, such as table signals, and confirm the robot responds correctly to the inputs.

Step 11: Final Adjustments and Deployment

Make any necessary adjustments, such as fine-tuning the IR sensor's alignment or calibrating the ultrasonic sensor's distance. Deploy the robot in the cafe environment and observe its performance.

7.1 Advantages

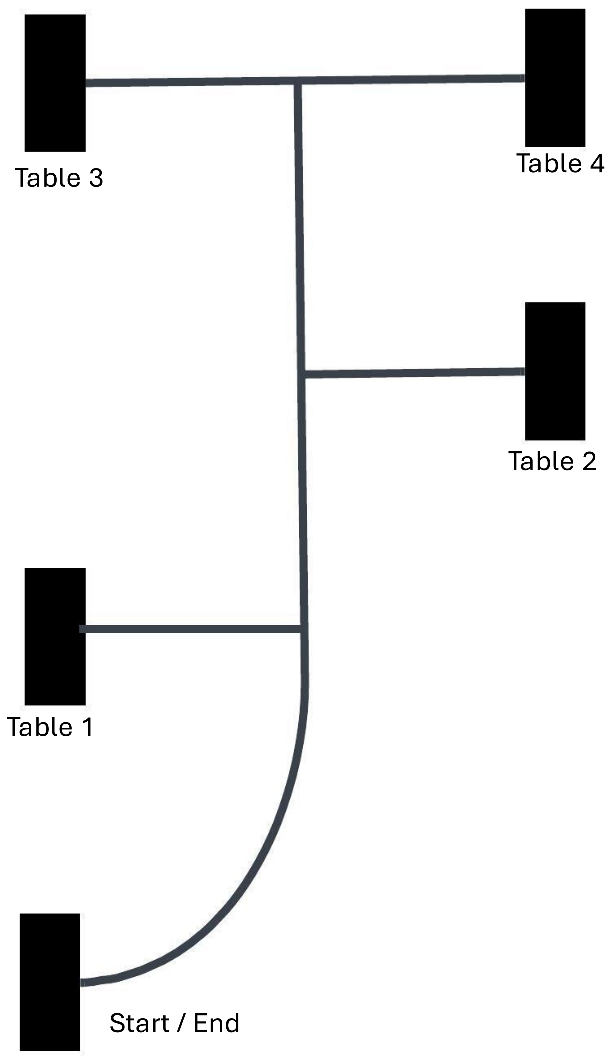
- Reduces manual effort.
- Ensures accurate and timely delivery.
- Cost-effective and affordable design.
- Engages customers with OLED display.
- User-friendly with Bluetooth control.
- Customizable for different layouts.

7.2 Disadvantages

- Limited to predefined paths.
- Requires clear floor markings.
- Limited obstacle-handling capability.
- Battery needs frequent recharging.
- Handles one tray at a time.

8. Source Code

8.1. Table Map



8.2. Main Line Following Code (Attached to the folder)

8.3. Eye Controlling Code (Attached to the folder)

9. Test Case

Inputs

1. **Power Supply**: Provide power using 3200mAh 3.7V 18650 batteries.
2. **Track Input**: Use a predefined black line on a white surface for testing line-following capability.
3. **Tray Input**: Place and remove a tray to test the ultrasonic sensor.
4. **Bluetooth Commands**: Send table signals using the **Arduino BlueControl app**.

Expected Outputs

1. The robot should follow the black line smoothly without deviation.
2. When the tray is placed, the robot should move towards the designated table.
3. Upon tray removal, the robot should detect the absence and return to the starting point.
4. The robot should respond accurately to Bluetooth commands, displaying table signals or emojis.

Testing Procedure

1. Assemble the robot and power it with the batteries.
2. Place the robot on a track and observe its line-following behavior.
3. Test tray detection by placing a tray and monitoring the robot's movement to the designated table.
4. Use the **Arduino BlueControl** app to send commands and verify that the robot follows the given instructions.
5. Simulate an obstacle on the path and ensure the robot avoids it while continuing its task.

Summary of Testing

1. Connect all components as per the circuit diagram.
2. Test line-following capabilities on various paths.
3. Check ultrasonic sensor functionality by placing and removing the tray.
4. Verify Bluetooth communication and interaction.
5. Evaluate performance in a simulated cafe environment.

10. Experimental Results

- **Line Following -** Initially, the robot exhibited a zigzag motion when following the black line due to the absence of PID control. After implementing PID control (proportional and derivative components only, without integral), the robot achieved smooth line-following. The accuracy and smoothness of movement depended on the tuning of KpK_pKp, KdK_dKd, motor speed, and track width. After careful adjustments, the robot demonstrated reliable and stable performance.

- **Tray Detection -** The tray detection sensor initially worked properly, but over time it produced inconsistent results. For a successful demonstration, the tray detection function was temporarily disabled, which resolved the issue.

- **Bluetooth Communication -** The robot utilized a common Bluetooth control app for table selection and command input. With the correct baud rate settings, the communication was seamless and error-free. Commands for table navigation and general robot control were executed reliably.

- **Obstacle Avoidance -** The obstacle detection feature of the BFD-1000 sensor was tested but found to have limited accuracy and distance measurement capabilities. While it performed basic obstacle detection, replacing it with a more robust ultrasonic sensor is recommended for future implementations.

- **OLED Emoji Display -** The OLED display was initially configured with the Arduino Mega. However, due to the inability to run the main code and display functions in parallel, an Arduino Nano was dedicated to controlling the OLED. This setup worked perfectly, allowing expressive emojis to be displayed effectively.

- **Battery Performance -** The robot used three 3.7V, 3200mAh batteries, which provided sufficient power for the project's needs. However, to ensure smoother junction turns, the motor speed was reduced, highlighting the need for motors with better torque in future iterations.

- **Overall Performance -** The robot successfully mapped four tables, followed the track smoothly, and responded well to Bluetooth commands. Despite minor challenges with tray detection and obstacle avoidance, the robot effectively demonstrated its core functionality in a controlled environment.

11. Conclusion

- The smart waiter robot project successfully demonstrated the integration of line-following technology, Bluetooth communication, and interactive features like OLED emoji display. By implementing PID control, the robot achieved smooth and accurate line tracking. The system performed well in delivering trays to designated tables and returning to the starting point.

- Key challenges included the limitations of the BFD-1000 sensor for obstacle detection and the inconsistencies in the tray detection sensor. These challenges were mitigated by temporary adjustments, but future versions should address these limitations with improved sensors and motors for enhanced performance.

- Projects like this emphasize the importance of proper and precise calculations. Critical parameters such as PID constants (KpK_pKp, KiK_iKi, KdK_dKd), motor speed, sensor positioning, and battery capacity must be properly calculated and tested to ensure optimal performance. These calculations directly impact the robot's smoothness, accuracy, and reliability, underscoring the necessity of a systematic approach to both hardware and software design.

- Overall, the project showcases a cost-effective and functional prototype of a smart waiter robot suitable for cafe environments. With further refinements, such as improved sensor accuracy, better motor capabilities, and thorough pre-implementation calculations, this robot can be a practical and engaging solution for automating food delivery services.

- The use of N20 metal gear motors with high torque significantly improved the robot's performance compared to standard TT motors. These motors provided better control and stability, allowing the robot to navigate the track with greater precision, even under varying load conditions.

- Replacing the BFD-1000 sensor with an 8-bit array (bee line follower) enhanced the accuracy of the robot's line-tracking capabilities. This upgrade enabled more precise path detection and smoother navigation, reducing errors and improving overall efficiency during operation.

12. Future Enhancement.

- **Development of a Dedicated Mobile App -**
  Creating a custom mobile app will enable cafe staff to control the robot more efficiently. The app could be designed to send table commands, track robot status, and manage meal deliveries. This dedicated app will replace the generic Arduino BlueControl app, providing more tailored functionality specific to the restaurant's needs, such as multiple table signals or custom notifications.
- **Voice Features -**
  Adding voice capabilities to the robot will make it more engaging and familiar to customers. The robot could greet customers, confirm meal deliveries, or even provide updates about the status of their meal. Voice interaction can enhance the overall customer experience by making the robot feel more personable, improving communication and increasing customer satisfaction.
- **Integration of IoT -**
  By incorporating Internet of Things (IoT) features, the smart waiter robot can be remotely controlled and monitored. This would allow restaurant managers or staff to access the robot's status in real-time, even from a distance. IoT could also help with predictive maintenance, providing alerts when the robot requires charging, cleaning, or technical attention.
- **Support for Multiple Trays -**
  Currently, the robot is designed to carry a single tray. Future enhancements could allow the robot to carry multiple trays, increasing efficiency, especially during peak hours. This would enable the robot to deliver meals to several tables at once, reducing wait times and improving service speed.
- **Dynamic Navigation -**
  Incorporating advanced sensors like LiDAR or cameras for real-time navigation would allow the robot to adapt to dynamic environments. The robot could then automatically adjust its path in response to changing obstacles, such as people moving around or new furniture being placed, making it suitable for more complex and ever-changing restaurant layouts.
- **Improved Obstacle Avoidance -**
  While the robot currently uses ultrasonic sensors for obstacle detection, additional sensors (such as IR sensors or computer vision) could be added to improve accuracy and response time. With better obstacle avoidance, the robot would be more reliable, even in crowded or tight spaces, ensuring it never bumps into customers or furniture.

13. Reference

- **Line Follower Robot - PID Control - Android Setup -** https://www.instructables.com/Line-Follower-Robot-PID-Control-Android-Setup/

- **Line Follow 8X Robot (PID Controlling)** https://www.youtube.com/watch?v=ZbOZ-sVDxmQ&t=3202s

- **PID Theory (Wikipedia) –** https://www.intellar.ca/blog/animated-eye-oled

- **PID Theory (Wikipedia) –** https://en.wikipedia.org/wiki/Proportional%E2%80%93integral%E2%80%93derivative_controller

END.