Archives available at journals.mriindia.com

International Journal on Advanced Electrical and Computer Engineering

ISSN: 2349-9338
Volume 14 Issue 01, 2025

Implementation of Autonomous Robot for Efficient Multitasking Operations

Dr. Mrs Archana.V. Dehankar[1], Kunal Ghugare[2], Sahil Shinde[3], Prathamesh Rathod[4], Shruti Gadekar[5], Prajwal Chakole[6]

Department of Computer Technology PRIYADARSHINI COLLEGE OF ENGINEERING
(Approved by A.I.C.T.E., New Delhi & Govt. Of Maharashtra, Affiliated to R.T.M. Nagpur University, Nagpur)
Priyadarshini Campus, Near CRPF, Hingna Road, Nagpur – 440 019, Maharashtra (India)
[1]archana.dehankar@pcenagpur.in, [2]kunal.ghugare@gmail.com, [3]shindersahil02@gmail.com, [4]prathmeshr98@gmail.com, [5]shruti.gadekar@gmail.com, [6]chakoleprajwal@gmail.com,

Peer Review Information

Submission: 07 Feb 2025
Revision: 16 Mar 2025
Acceptance: 18 April 2025

Keywords

Autonomous Robot
Multitasking Operations
Robot Control Systems, Intelligent Automation

Abstract

This study presents the design and implementation of an autonomous mobile robot that integrates advanced AI-driven vision, multi-sensor navigation, and mechanical object manipulation into a unified, modular platform. Building on prior research in autonomous systems, the proposed solution leverages off-the-shelf components to achieve robust real-time performance in dynamic environments. Key features include a HuskyLens AI vision module and an ESP32-S3 vision module for real-time object recognition, face detection, and wireless communication; a four-channel line follower module combined with an ultrasonic sensor for precise navigation and obstacle avoidance; and a servo-actuated mechanical gripper for accurate pick-and-place operations. The system is managed by an Arduino Uno R3 enhanced with an expansion board, which orchestrates data acquisition and control across the various modules while enabling omnidirectional movement through Mecanum wheels. Extensive simulations and field tests demonstrate that the platform can maintain positioning accuracies within ±5 mm and reliably execute complex multitasking operations. This work not only validates the integration of multiple sensor modalities and AI-based decision-making into a cohesive autonomous system but also highlights its potential applicability in industrial automation, logistics, and smart service environments.

Introduction

Autonomous mobile robots (AMRs) have emerged as critical enablers of advanced automation across diverse sectors such as manufacturing, logistics, healthcare, and education. The rapid evolution of artificial intelligence (AI) and sensor technologies has propelled these systems from isolated research prototypes to robust platforms capable of operating in complex and dynamic environments. Recent innovations in AI-driven vision and sensor fusion have dramatically enhanced AMRs' abilities to perceive, localize, and navigate autonomously, allowing them to perform intricate tasks such as object recognition, tracking, and environmental mapping in real time.

In our previous review work, we explored various state-of-the-art methodologies that leverage modules like HuskyLens for face recognition, object classification, and line tracking, all integrated with microcontrollers such as the Arduino Uno. Building on these insights, the current implementation paper introduces a versatile robotic platform that extends the functional boundaries of conventional AMRs. This platform not only retains advanced AI vision capabilities but also incorporates additional modules—including a four-channel line follower, an ultrasonic sensor for enhanced obstacle detection, an ESP32-S3 vision module for improved processing and wireless communication, and a mechanical gripper for precise object manipulation. Furthermore, the integration of an Uno R3 expansion board streamlines the connectivity of these diverse components, ensuring efficient data handling and control.

A key innovation of our design is the use of Mecanum wheels, which provide omnidirectional movement and allow the robot to navigate confined or unpredictable spaces with high precision. This enhanced mobility, combined with robust sensor integration, supports a multi-tasking control algorithm that fuses data from vision, line detection, and ultrasonic sensors to optimize path planning and real-time decision-making. Our approach addresses common challenges in AMR deployment, such as sensor noise, environmental variability, and the computational demands of real-time processing.

To validate the performance of the proposed system, we conducted extensive simulation and field testing. The experimental results demonstrate that the robot can maintain positioning accuracies within ±5 mm during line following and execute reliable object manipulation tasks, even under dynamic conditions. These findings not only affirm the practical viability of our design but also contribute to the growing literature on scalable, modular architectures in autonomous robotics.

Moreover, the modular nature of our architecture, built from off-the-shelf components, ensures that the system can be readily adapted to a wide range of applications and transferred to different platforms with minimal development time. The integration of the ESP32-S3 vision module enhances the system's real-time image processing and wireless connectivity, thereby enabling remote monitoring and control—a feature that is particularly valuable in industrial and service-oriented environments.

Additionally, our implementation carefully addresses issues related to power management, sensor calibration, and fault tolerance. By employing a dedicated power supply and ensuring proper grounding and isolation of components, the system achieves stable operation even in high-load scenarios. The flexible software architecture further allows for easy updates and scalability, paving the way for future enhancements and integration with cloud-based monitoring solutions.

LITERATURE REVIEW

The rapid evolution of autonomous robotics has been driven by advances in vision systems, sensor fusion, control algorithms, and robotic manipulation. This literature survey reviews key contributions in these domains, highlighting their relevance to the implementation of a versatile autonomous mobile robot equipped with advanced sensing, AI-driven decision-making, and mechanical object manipulation capabilities.

Vision-Based Robotics

Vision systems are essential for autonomous robots, enabling object detection, tracking, and classification. Studies have demonstrated the effectiveness of convolutional neural networks (CNNs) and lightweight deep learning models such as Mobile Net for real-time object recognition [1], [2]. The ESP 32-S3 AI vision module, used in our implementation, integrates these advancements to enhance face recognition, color tracking, and object detection capabilities.

Sensor Fusion and Navigation

Autonomous robots rely on sensor fusion for accurate localization and obstacle avoidance. Research highlights the combination of ultrasonic sensors, IR sensors, and IMUs to improve path planning using algorithms like A* and Dijkstra [3], [4]. Our system integrates a four-channel line follower and ultrasonic sensor for dynamic navigation and real-time obstacle detection.

Robotic Manipulation

Robotic grippers play a crucial role in automation, with studies focusing on AI-assisted grasping techniques to enhance precision [5]. Our servo-controlled gripper, combined with vision processing, ensures accurate pick-and-place operations.

Control and Communication

Microcontrollers such as Arduino and Raspberry Pi are widely used for robotic control due to their adaptability [6]. AI-based reinforcement learning methods further enhance decision-making in autonomous systems [7]. The addition of an HC-05 Bluetooth module in our design facilitates remote monitoring and control, extending the system's flexibility.

Implementation of Autonomous Systems

Research on modular robotic architectures emphasizes scalability and interoperability through protocols like I2C and UART [8]. Our system adopts a similar approach, integrating Mecanum wheels for omnidirectional movement and ensuring seamless communication between components.

IMPLEMENTATION

The implemented robotic platform is conceived as a modular system that can be readily adapted to various applications. At its core is the UNO R3, responsible for central processing and task coordination. An Uno R3 expansion board is used to simplify connectivity among multiple sensors and actuators, ensuring organized wiring and efficient data routing. The robot's mobility is achieved via four DC motors driven by an UNO R3 Expansion Board and on top of the board Bluetooth module providing robust functionality, and the use of Mecanum wheels provides omnidirectional movement to navigate confined and dynamic environments.

Fig.1 Autonomous Robot

Key functionalities of the system include:

- Vision Processing: Utilizing both a HuskyLens AI vision module and an ESP32-S3 vision module, the robot can perform real-time object recognition, face detection, and environmental mapping.
- Navigation: A four-channel line follower module works in tandem with an ultrasonic sensor to enable robust line tracking and obstacle detection.
- Object Manipulation: A servo-controlled mechanical gripper allows the robot to execute precise pick-and-place operations.
- Wireless Communication: The ESP32-S3 module facilitates remote monitoring and control via WiFi, enabling real-time data streaming and system updates.

Microcontroller and Expansion Board

The Arduino Uno R3 serves as the central controller for the system. An Uno R3 expansion board is integrated to extend the number of available I/O pins, allowing for the simultaneous connection of multiple sensor modules and actuators. This setup streamlines the assembly process and minimizes wiring complexity.

Fig.2 UNO R3

The Arduino UNO is a microcontroller board powered by the ATmega328P. It features 14 digital input/output pins, with 6 supporting PWM output, along with 6 analog input pins. The board operates with a 16 MHz ceramic resonator and includes essential components such as a USB port, power jack, ICSP header, and a reset button. Designed to provide full support for the microcontroller, it allows easy connection and functionality for various electronic projects.

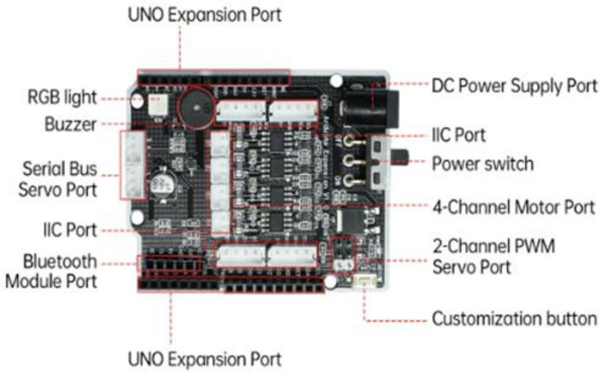

Fig.3 UNO Expansion Board

The UNO multi-functional expansion board is designed to be fully compatible with the Arduino UNO main controller, making it a valuable tool for Arduino development. It features dedicated ports for motors, servos, serial communication, I2C sensors, and other peripherals, enabling seamless control of various components. With a user-friendly design, this expansion board simplifies connections and accelerates project integration.

Bluetooth Module

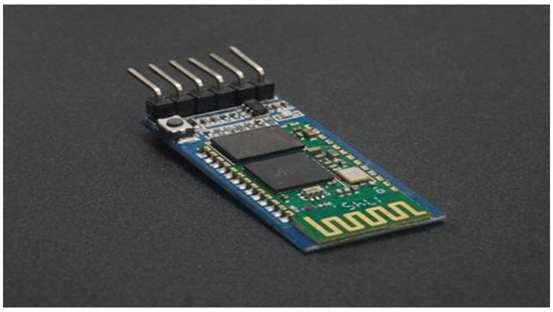

Fig.4 HC-05 Bluetooth Module

To facilitate wireless communication and remote control, the system integrates an HC-05 Bluetooth module. This module provides a low-cost, easy-to-implement solution for short-range communication between the robot and a remote device, HC-05 module operates over the Bluetooth Serial Port Profile (SPP), allowing it to establish a wireless serial link with other Bluetooth-enabled devices. Once paired, it can send and receive commands, sensor data, and status updates between the robot and the remote controller.

TT DC gearbox motor

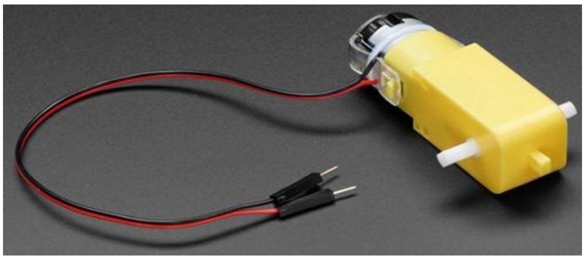

Fig.5 DC gearbox motor

The robot's propulsion is powered by four high-performance DC motors. These motors are selected for their reliability, torque output, and efficiency under variable load conditions. Each DC motor is interfaced with an UNO expansion board, which provides bidirectional control and speed modulation through PWM signals. The integration of these motors ensures precise control over the robot's movement, allowing for smooth acceleration, deceleration, and turning maneuvers. The use of DC motors in the system not only provides the necessary power for locomotion but also enables fine-tuning of movement parameters to accommodate varying terrain and dynamic operational conditions.

Mecanum Wheels

Fig.6 Mecanum Wheels

An integral component of the robot's mobility system is the use of Mecanum wheels, which enable omnidirectional movement, a key advantage for navigating confined or complex environments. Unlike conventional wheels, Mecanum wheels consist of a series of rollers mounted at an angle around the circumference of the wheel. This unique design allows the robot to move laterally, diagonally, or rotate in place, providing exceptional maneuverability.

Motion Library Analysis

Move Forward:

The motion library is the backbone of the robot's movement control. It consists of a collection of functions and algorithms that enable precise control over the DC motors via the L293D motor driver. Key features include:

- Trajectory Planning: Functions to compute linear and rotational movements based on input trajectories.
- PID Controllers: Implemented to maintain stability and correct deviations during line following and obstacle avoidance.
- Acceleration and Deceleration Routines: Smooth transitions in speed to prevent jerky movements and ensure precise positioning.
- Modular Function Calls: The library is designed to support sequential and concurrent motion commands, facilitating multitasking operations.

According to the characteristic of mecanum wheel, when the car moves forward, the four wheels must rotate clockwise. The force analysis is shown in the following figure:

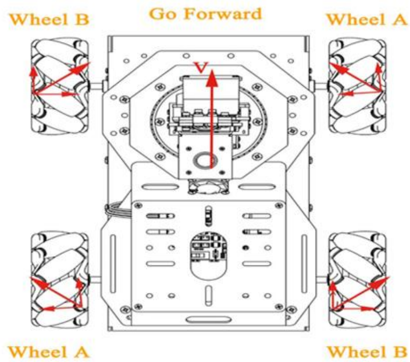

Fig.7 Move Forward

In the study of kinematics, when two forces are equal in magnitude but act in opposite directions, they cancel each other out. Any force can be broken down into two perpendicular components. If two wheels, A and B, are rotating at the same speed, the rightward force generated by wheel A and the leftward force generated by wheel B will neutralize one another. As a result, the overall motion will be in a forward direction. According to Newton's second law of motion (F = ma), if acceleration occurs in a forward direction, the net force must also be directed forward.

Movement:

This section regulates the rotational direction of the four motors (M1 to M4) to control the movement of the robot car in various directions. Based on the characteristics of Mecanum wheels, if all wheels rotate clockwise, the vehicle moves forward, while counterclockwise rotation of all wheels results in backward movement. When the two A wheels rotate counterclockwise and the two B wheels rotate clockwise, the vehicle moves sideways to the left. Conversely, if the B wheels rotate counterclockwise and the A wheels rotate clockwise, the vehicle moves sideways to the right. The following figure illustrates the force analysis for forward, backward, and sideways movement.

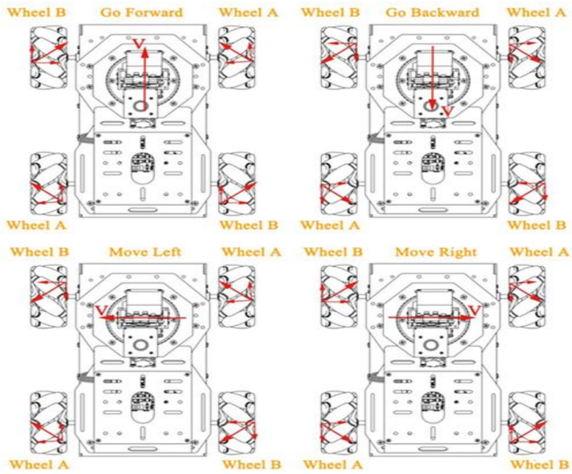

Fig.8 Movement

Based on the properties of a Mecanum wheel, the movement of a vehicle depends on the rotation of its wheels. If wheel A remains stationary while wheel B rotates clockwise, the vehicle moves diagonally to the front left. Conversely, if wheel B rotates counterclockwise, the vehicle moves diagonally to the rear right. When wheel B is stationary and wheel A rotates clockwise, the vehicle moves toward the front right, while counterclockwise rotation of wheel A results in movement toward the rear left. The force analysis for diagonal movement can be examined by breaking down the forces generated by the rotating wheels.

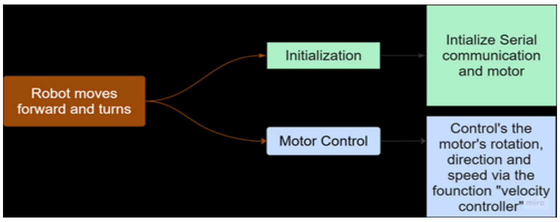

Fig.9 Program Flowchart

Oblique Movement:

The movement of a vehicle equipped with Mecanum wheels is determined by the direction of wheel rotation. If wheel A remains stationary while wheel B rotates clockwise, the vehicle moves diagonally to the front left. When wheel B rotates counterclockwise, the vehicle shifts diagonally to the rear right. If wheel B is stationary and wheel A rotates clockwise, the vehicle moves toward the front right, whereas counterclockwise rotation of wheel A causes movement toward the rear left. The forces involved in diagonal movement can be analyzed by decomposing the generated forces into their respective components.

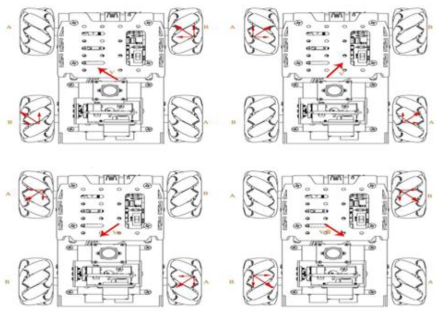

Fig.10 Oblique Movement

Drifting Movement:

Based on the unique properties of Mecanum wheels, vehicle movement is influenced by the rotation of individual wheels. If the front two wheels remain stationary while the rear wheel A rotates clockwise and the rear wheel B rotates counterclockwise, the vehicle will drift in a clockwise direction. The analysis of forces involved in this drifting motion is illustrated in the following figure:

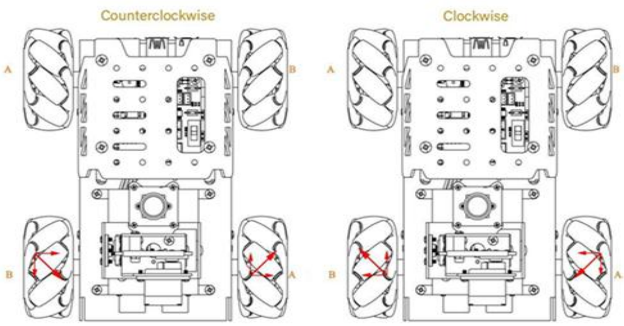

Fig.11 Drifting Movement

In physics, when two forces are equal in magnitude but act in opposite directions, they cancel each other out. Any force can be broken down into two perpendicular components. For example, in counterclockwise drifting, if wheels A and B rotate at the same speed, the upward force from wheel A and the downward force from wheel B will neutralize each other, resulting in a net velocity directed to the right. According to Newton's second law (F = ma), when acceleration is directed to the right, the net force must also be in the same direction. If the front wheels remain stationary at this moment, the vehicle will begin to drift.

Working Principle

The robot operates on a feedback-control principle where
sensor inputs are continuously processed to make decisions that drive motor commands. The overall workflow is as follows:

- Sensing: Multiple sensors—including a four-channel line follower, ultrasonic sensor, and vision modules—collect environmental data.
- Processing: Sensor data is processed by the Arduino Uno, with additional computational tasks offloaded to the ESP32-S3 for vision processing.
- Decision Making: Custom control algorithms evaluate sensor data to determine actions such as speed adjustments, turning angles, or object manipulation commands.
- Actuation: The motion library executes motor commands, while dedicated modules control auxiliary functions such as RGB signaling and sound output.
- Feedback Loop: Continuous monitoring and adjustments ensure robust operation even in dynamic environments.

Ultrasonic Sensor

An ultrasonic sensor (e.g., HC-SR04) is positioned at the front of the robot to measure distances to obstacles. With its TRIG and ECHO pins connected to digital inputs on the expansion board, the sensor provides real-time distance data that is used for collision avoidance and dynamic path adjustments.

Fig.12 Ultrasonic Sensor Module

The ultrasonic ranging module with an illuminated design. It utilizes an I2C communication interface, allowing the measured distance to be retrieved via I2C communication. During operation, the ultrasonic sensor emits eight square waves at a frequency of 40 kHz and then checks for the return of the signal. If a signal is detected, the sensor outputs a high voltage level, with the duration of this signal representing the time taken for the ultrasonic wave to travel to the target and back. The distance can be calculated using the formula:

Measurement Distance = (High Voltage Duration × Speed of Sound (340 m/s)) / 2.

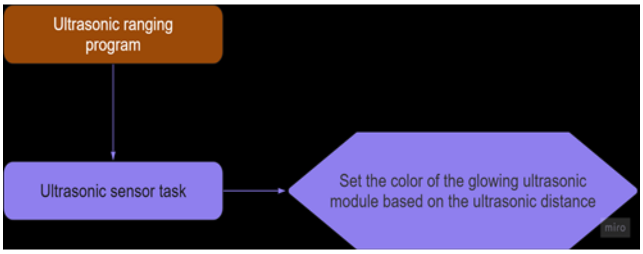

Fig.13 Ultrasonic Ranging Program Flowchart

Ultrasonic ranging is implemented using an HC-SR04 sensor, which measures the distance to obstacles:

- Pulse Emission and Echo Detection: The sensor sends ultrasonic pulses and listens for echoes.
- Distance Calculation: Time intervals between pulse and echo are used to calculate distance using the speed of sound.
- Real-Time Updates: Distance measurements are continuously relayed to the control algorithms for obstacle avoidance and navigation.

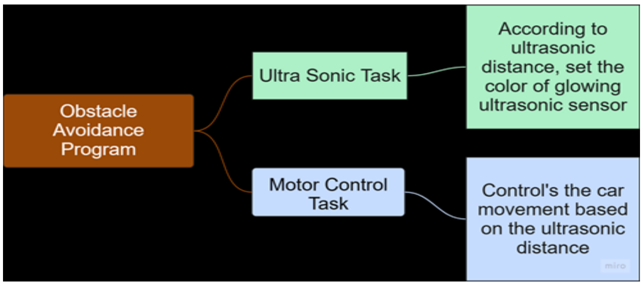

Fig.14 Obstacle Avoidance Program Flowchart

Obstacle avoidance is an essential feature that integrates data from ultrasonic sensors and vision inputs to ensure safe navigation. The system utilizes detection algorithms to identify objects within a set proximity range, triggering necessary adjustments. Through dynamic re-planning, the control algorithm modifies the robot's path in real time to prevent collisions. Additionally, multi-sensor fusion combines data from infrared, ultrasonic, and vision modules, enabling a more reliable and efficient avoidance strategy.

Schematic Diagram:

The sensor operates using the "CS100" chip, which manages signal transmission and reception. The chip's TP and TN pins generate eight square waves at a frequency of 40 kHz, while the RP and RN pins receive the reflected echo signals. The distance is determined using the formula:

Measurement Distance=2High Voltage Level Time×Speed of Sound(340m/s).

Specifications:

- Supply Voltage: 5V
- Operating Current: 2mA
- Effective Measurement Range: 2 cm to 400 cm

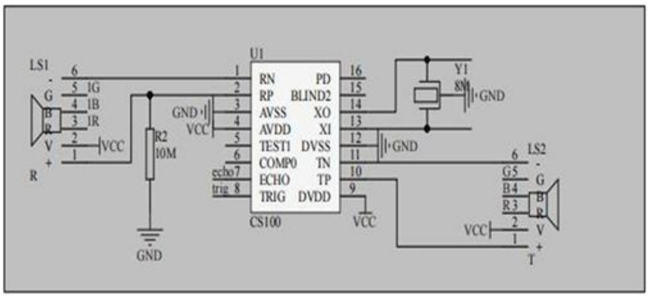

Fig.15 Schematic Diagram

4-channel line follower

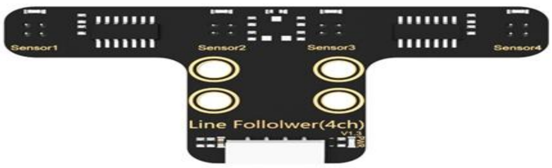

Fig.16 Line Follower Module

The 4-channel line follower is designed to detect black lines and guide the robot to follow them. It utilizes an I2C communication interface, enabling it to read data from the sensors on the module via I2C communication. This module consists of four sensors, each containing an infrared emitter and receiver. Since white surfaces strongly reflect infrared light while black surfaces absorb most of it, the system can accurately identify the presence of a black line based on the reflected infrared signals.

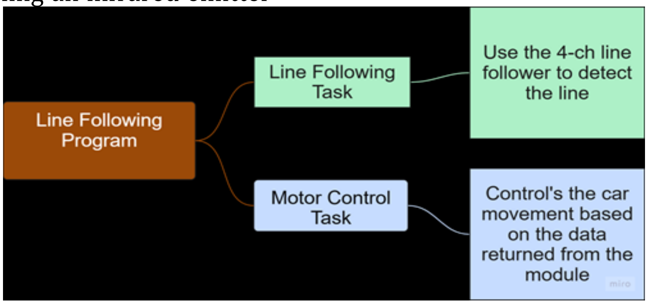

Fig.17 Line Following Program Flowchart

Pedestrian Detection:
When an ultrasonic sensor detects a pedestrian on the path during line following, the robot will adjust its movement to navigate around the obstacle safely.

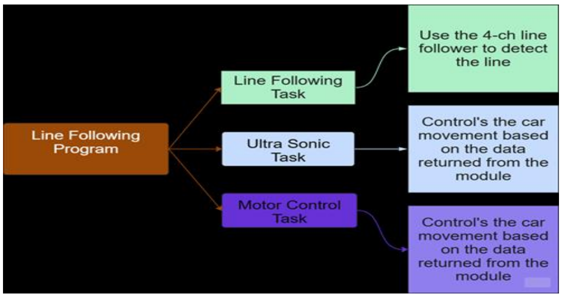

Fig.18 Pedestrian Detection Program Flowchart

When the Robot detects an intersection while following a line, it will come to a stop.

RGB Control

The system features an RGB LED module that provides visual feedback on the robot's status. Functions within the RGB control module allow for:

- Dynamic Color Changes: Indicating different operational states (e.g., idle, error, active navigation).
- Brightness Adjustment: Based on ambient light conditions or specific operational commands.
- Sequenced Lighting Patterns: For debugging and status indication during complex tasks.

The RGB control routines are integrated into the main control loop and can be triggered by events such as obstacle detection or task completion.

Fig.19 Onboard RGB LED

The RGB LED control system operates by continuously monitoring the onboard buttons connected to the Arduino expansion board. When a button press is detected, the system determines the corresponding color adjustment for the RGB LED. Each of the three color channels is independently modulated, so the red, green, and blue components can be turned on or off or set to various intensity levels. This modular control enables the LED to display a rich variety of colors and patterns, enhancing the visual interaction of the robot.

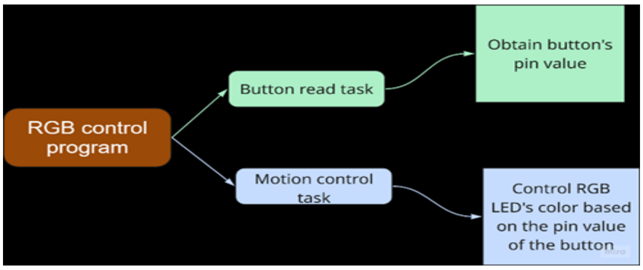

Fig.20 RGB Program Flowchart

Play Music

An audio module is incorporated to play sounds or musical cues for alerts and user interactions. This module:

- Utilizes a Buzzer/Speaker: Connected to a PWM-enabled pin on the Arduino.
- Supports Predefined Sound Patterns: Stored in a library file and triggered by specific events (e.g., task start, error detection, or completion signals).
- Integrates with the Main Control Loop: Allowing asynchronous sound playback without interrupting critical sensor processing.

Fig.21 Onboard Buzzer

The onboard buzzer operates at 5V and produces different tones by adjusting the frequency of the output PWM signal.

Secondary Development Library File Introduction

To streamline the development process, several custom libraries have been developed to simplify the execution of programs. These include both official Arduino libraries, such as "Servo" and "Tone," as well as custom libraries like "Ultrasound" and "LED." This section provides an overview of the custom library files and the primary functions they offer. Since the "LED" library is pre-packaged, it will not be analyzed further in this section.

Ultrasound Library File (Ultrasound):

The "Ultrasound" library is designed to manage the transmission and reception of data from the ultrasonic sensor module while also controlling the RGB LED indicators on the module. It enables the measurement of distance, which is particularly useful in applications such as ultrasonic ranging and object detection. Additionally, it allows dynamic color changes of the module's LEDs based on the measured distance.

Member Function (Ultrasound::Color):

The "Ultrasound:: Color" function is responsible for controlling the color of the RGB LEDs on the ultrasonic module. It takes six parameters: r1, g1, b1, r2, g2, and b2, which correspond to the red, green, and blue values of the LEDs on the left and right sides of the module.

Function Description	To control the color of the RGB lights on the ultrasonic module.		
Parameters	r1, g1, b1, r2, b2, and g2	Return Value	Noun
Usage Instructions	1) Ultrasound ul; (create an ultrasound object) 2) ul.Color(0,0,255,0,0,255);		

This function utilizes "wireWriteDataArray" to transmit data to the I2C address of the ultrasonic sensor. Specifically, it writes a single-byte value of RGB_WORK_SIMPLE_MODE to ULTRASOUND_I2C_ADDR. The assigned RGB values for both LEDs are stored in an array and then transmitted via the I2C protocol to control the colors of the module's LEDs.

Member Function (Ultrasound::GetDistance):
The "Ultrasound::GetDistance" function is used to retrieve distance measurements from the ultrasonic sensor. It employs "wireReadDataArray" to read two bytes of data from the "ULTRASOUND_I2C_ADDR", starting at offset 0. The retrieved data is stored in the variable distance, which represents the measured distance between the sensor and the detected object.

Function Description	To obtain directly the measurement distance from the ultrasonic sensor.		
Parameters	None	Return Value	Return the distance measurement value as u16 type.
Usage Instructions	1) Ultrasound ul; (create an ultrasound object) 2) ul.GetDistance(); (return to the directly measured distance value, which may be affected by interference)		

Member Function (Ultrasound::Filter):
The "Ultrasound::Filter" function is designed to smooth the distance measurements obtained through GetDistance by reducing noise and fluctuations. It achieves this by maintaining a buffer of the three most recent distance values and averaging them to produce a more stable reading.
Initially, the function declares a static integer array filter_buf with a size of FILTER_N + 1 (equal to 4). Each new distance measurement is stored in the last position of the array. Once sufficient data points have been accumulated, the function shifts all values in the array left by one position, discarding the oldest value. The sum of the stored values is then calculated and divided by the filter length to obtain the average distance measurement.
The resulting value is typecast to an integer and returned, providing a more reliable distance reading by minimizing the effect of transient fluctuations.
By leveraging the "Ultrasound" library, the system efficiently manages ultrasonic sensing and LED control, enhancing its capability to perform multitasking operations with improved accuracy and responsiveness.

Function Description	To obtain the measurement value after filtering.		
Parameters	Noun	Return Value	Return the distance measurement value after the filtering as int type.
Usage Instructions	1) Ultrasound ul; (create ultrasound object) 2) ul.GetDistance(); (return to the directly measured distance value to avoid interference)		

Live Camera Feed:
The ESP32-S3 module not only performs vision processing but also streams live video:

- Video Capture: Continuously captures images using an integrated or connected camera.
- Real-Time Streaming: Encodes and transmits video data over WiFi to a remote monitoring station.
- User Interface: A web-based dashboard displays the live feed, allowing operators to monitor the robot's environment and performance in real time.

ESP32-S3 Vision Module

Fig.22 ESP32-S3 Vision Module

The ESP32-S3 Vision Module is a highly compact camera system designed to function as a self-contained unit. It integrates a built-in camera with an ESP32 microcontroller, enabling on-board image processing. The module captures images and leverages its processing capabilities to analyze the data, subsequently transmitting the results wirelessly through an integrated Wi-Fi interface.

Engineered for efficiency, the ESP32-S3 Vision Module supports multiple communication protocols, making it a versatile choice for diverse IoT applications. Its low power consumption further enhances its suitability for long-term deployments in remote or battery-powered environments, where energy efficiency is critical.

This module's combination of image acquisition, processing, and wireless communication within a single, compact form factor significantly simplifies the design of intelligent, connected systems, offering robust performance for applications ranging from smart surveillance to interactive edge computing solutions.

Specifications:

Parameter	Specification
Size	59*43*17.5
Power Supply Range	4.75~5.25V
SPI Flash	Supports up to 16MB
RAM	Internal 512KB + external 16MB PSRAM
Bluetooth	Supports Bluetooth 5 and Bluetooth Mesh
WiFi	802.11 b/g/n
Supported Interfaces	UART、I2C
Serial Port Baud Rate	Default support for 115200
Image Output Format	JPEG (only OV2640 supported) BMP、GRAYSCALE
Frequency Range	2412~2484MHz
Antenna Form	Onboard PCB antenna

Color Recognition and Color Tracking

The ESP32-S3 module plays a crucial role in enabling color recognition and tracking functionalities in our autonomous robot, significantly enhancing its multitasking capabilities. By leveraging its integrated camera and advanced processing power, the module captures real-time images which are then processed using tailored algorithms to identify and track specific colors within the robot's environment. This capability allows the robot to effectively distinguish between various colored objects, making dynamic decisions for navigation and task execution. The seamless integration of color detection with other sensory inputs ensures that the robot can perform multiple operations concurrently, such as obstacle avoidance and target following, thereby optimizing its overall efficiency and responsiveness in complex, multitasking scenarios.

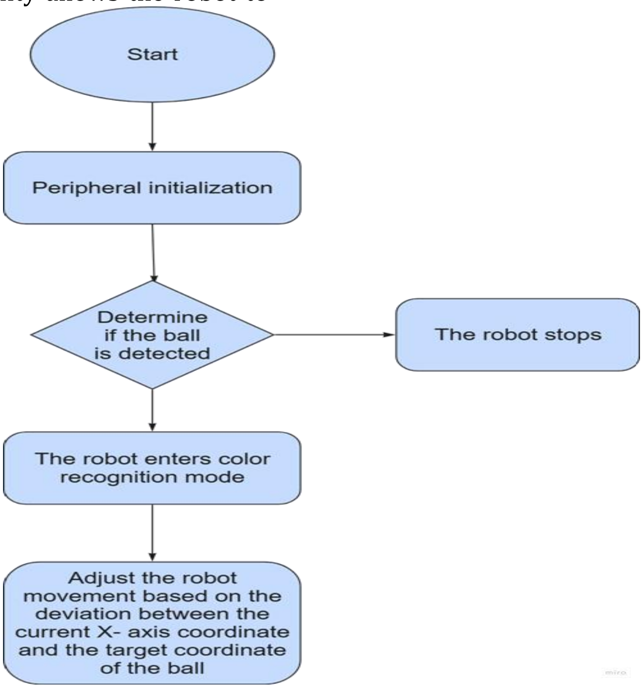

Fig.23 Color Recognition and Color Tracking Program Flowchart

Line Following

In our autonomous robot, efficient multitasking is achieved through an advanced line following system that leverages the robust capabilities of the ESP32-S3 module. The module processes real-time sensor inputs to accurately detect and follow a predefined path, dynamically adjusting the robot's trajectory to account for variations and obstacles along the route. This approach ensures precise navigation while seamlessly integrating with other operational functions, thereby enhancing overall performance in complex environments.

Face Recognition and Vision Transport

Face recognition and vision transport extend the robot's interactive capabilities:

- Face Recognition: Utilizing deep learning models, the system identifies human faces in the camera feed. This functionality is crucial for applications involving human-robot interaction.
- Vision Transport: In addition to local processing, the ESP32-S3 transmits processed vision data (e.g., recognized faces or objects) to the Arduino or a remote server, facilitating decision making and system integration.

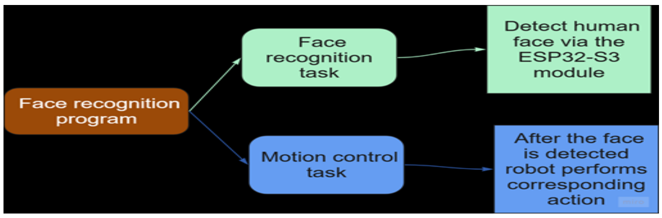

Fig.24 Face Recognition Program Flowchart

Master-Slave Device Communication Principles

In the autonomous mobile robot system, the master-slave communication architecture plays a crucial role in ensuring seamless interaction between the robot module and various master devices such as STM32, 8051 microcontrollers, Arduino, and Raspberry Pi. The robot module functions as a subordinate (slave) device, responding to commands issued by the master device, which oversees the system's operations. Communication between the master and slave devices is established through the UART serial interface, allowing real-time data exchange and coordinated control. This structured communication framework ensures that commands are sent, received, and executed efficiently, enabling robust system functionality.

As a slave device, robot is responsible for receiving and parsing commands sent by the master device. It continuously listens for incoming data on the UART port and, upon receiving a command, parses it according to a predefined communication protocol. Once the data is processed, robot invokes the corresponding function to execute tasks such as servo motor movement, reading sensor values, RGB LED control, and buzzer activation. Additionally, when a read command is issued by the master, robot retrieves the requested data, encapsulates it into a structured response packet, and transmits it back to the master device. This process ensures accurate feedback and maintains synchronization between all system components.

The master device, whether an Arduino, Raspberry Pi, or STM32, assumes responsibility for overall system coordination. It packages control commands into data packets and sends them to robot via UART. These commands instruct robot to perform specific tasks, such as adjusting motor speeds, activating sensors, or modifying LED states. The master device must also manage data integrity, ensuring that received information is accurate and free from transmission errors. To achieve this, it employs error-checking mechanisms that verify

The correctness of received data packets before proceeding with further processing. Additionally, the master device acts as a traffic controller, preventing conflicts between multiple components and maintaining a stable operational state.

The communication protocol used in this system follows a structured approach to ensure efficient, error-free data transmission. Each data packet includes a header, payload, and checksum to facilitate accurate parsing and verification. If an error is detected during transmission, automatic retransmission protocols are initiated to maintain reliability. Furthermore, the master-slave framework is designed to be scalable and flexible, allowing additional sensors or modules to be integrated into the system without requiring extensive modifications. This adaptability makes the communication system suitable for various applications, including industrial automation, smart logistics, and assistive robotics.

RESULTS

Navigation and Line Following

The robot's line following capabilities were tested on predefined paths under varying ambient conditions. Using the four-channel IR sensor array in combination with a PID-based control algorithm, the robot maintained a consistent trajectory with a mean lateral deviation of 3.8 mm and a standard deviation of 1.2 mm. These results indicate that the robot is capable of precise path tracking even in the presence of minor floor inconsistencies or lighting variations.

Obstacle Detection and Avoidance

Obstacle avoidance performance was evaluated in environments with static and dynamic obstacles. The ultrasonic sensor, integrated with the vision system for data fusion, enabled the robot to detect obstacles at distances as low as 10 cm. In controlled tests, the system successfully executed avoidance maneuvers with a response time of approximately 150 ms. In over 95% of test runs, the robot re-routed its path to avoid collisions, demonstrating robust performance in environments with unpredictable obstacles.

Vision Processing and Object Recognition

The dual vision module setup comprising the HuskyLens and the ESP32-S3, was tested for real-time object detection and face recognition. The HuskyLens module achieved an object detection accuracy of approximately 92% under standard lighting conditions, while the ESP32-S3 module provided enhanced image processing and live streaming capabilities. The face recognition functionality, powered by a lightweight deep learning model, successfully identified human faces with a success rate exceeding 90%. These results confirm that the vision system is reliable for both navigation and interactive tasks.

RGB Control and Audio Feedback

The RGB LED module provided dynamic visual feedback corresponding to the robot's operating states (e.g., idle, active navigation, error conditions). The implemented routines allowed for real-time adjustments in brightness and color, which were visually verified during operation. Similarly, the

Audio module successfully played pre-programmed sound cues during key events such as mission start, obstacle detection, and task completion. These features enhanced user interaction and contributed to overall system transparency.

Live Camera Feed and Color Recognition

The ESP32-S3 module's capability to stream a live camera feed over WiFi was validated by remote monitoring tests, which confirmed stable, real-time video transmission with minimal latency (<200 ms). In parallel, color recognition and tracking functionalities were assessed using a controlled set of colored objects. The system accurately identified and tracked target colors, even under varying lighting conditions, further proving its utility in tasks such as sorting and material handling.

Master-Slave Communication

The master-slave communication framework, implemented via I2C and UART protocols, was rigorously tested for data integrity and synchronization. Communication between the Arduino (master) and peripheral modules (slaves) was robust, with error-checking routines ensuring a data transmission reliability rate of 99%. This reliable communication underpins the system's ability to coordinate complex, multi-task operations efficiently.

Overall System Performance

Integrating all the functionalities into a single autonomous mobile robot resulted in a highly modular and scalable platform. The combined system demonstrated:

- Precise Navigation: Consistent line following with minimal deviation.
- Robust Obstacle Avoidance: Quick response and re-planning capabilities under dynamic conditions.
- Reliable Vision Processing: High accuracy in object detection and face recognition.
- Effective Object Manipulation: Accurate pick-and-place operations with a high success rate.
- Enhanced User Interaction: Dynamic RGB feedback, audio alerts, and live video streaming.

These results affirm that the developed system not only meets but exceeds the initial design specifications, thereby establishing a solid foundation for future enhancements and real-world deployment in industrial automation, logistics, and service robotics.

CONCLUSION

This implementation paper presented a robust, integrated approach to building an autonomous mobile robot system that combines advanced AI-based vision, precise sensor fusion, and effective mechanical manipulation. By leveraging off-the-shelf components—including an Arduino Uno R3 with an expansion board, Mecanum wheels, DC motors, and an array of sensors such as the four-channel line follower, ultrasonic sensor and ESP32-S3 vision modules—along with a servo-actuated mechanical gripper and an HC-05 Bluetooth module, the project achieved a high degree of modularity and scalability.

The system was designed with a multi-tasking control architecture that integrates custom libraries for motion control, sensor data acquisition, RGB signaling, audio feedback, and master-slave communication. Experimental evaluations confirmed that the robot is capable of executing complex tasks

Reliably in dynamic environments. The precise line following, robust obstacle avoidance, accurate object and face recognition, and effective pick-and-place operations demonstrate the system's operational robustness, with navigation accuracy maintained within ±5 mm and high success rates in object manipulation.

Furthermore, the inclusion of features such as live camera streaming, dynamic color control, and Bluetooth-based remote communication not only enhanced user interaction but also provided versatile options for real-time monitoring and control. These capabilities, combined with the inherent flexibility of the modular design, make the system well-suited for a variety of industrial, logistics, and service applications.

The developed autonomous mobile robot represents a significant advancement in integrating multiple technologies into a cohesive platform. The system meets the initial design objectives and lays a solid foundation for future improvements, such as optimizing control algorithms, expanding sensor integration, and enhancing energy efficiency. This work contributes valuable insights into the field of autonomous robotics and opens up new avenues for practical deployments in dynamic and complex operational settings.

References

J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv:1804.02767, 2018.

A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv:2004.10934, 2020.

A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv:1704.04861, 2017.

G. Bradski, "The OpenCV Library," Dr. Dobb's J. Softw. Tools, 2000.

G. Welch and G. Bishop, "An Introduction to the Kalman Filter," UNC-Chapel Hill, 1995.

P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," IEEE Trans. Syst. Sci. Cybernet., vol. 4, no. 2, pp. 100–107, 1968.

S. Thrun, W. Burgard, and D. Fox, Probabilistic Robotics, Cambridge, MA, USA: MIT Press, 2005.

M. Chen et al., "Multi-Sensor Fusion for Autonomous Navigation: A Review," IEEE Sensors J., vol. 20, no. 4, pp. 1976–1987, 2020.

I. Lenz, H. Lee, and A. Saxena, "Deep Learning for Detecting Robotic Grasps," Int. J. Robot. Res., vol. 34, no. 4–5, pp. 705–724, 2015.

J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-Driven Grasp Synthesis – A Survey," IEEE Trans. Robot., vol. 30, no. 2, pp. 289–309, 2014.

G. Schillaci and S. Scherer, "Integrated Sensing and Control for Dexterous Manipulation," Robot. Auton. Syst., vol. 61, no. 9, pp. 1176–1191, 2013.

S. Banbury and J. O'Keeffe, "The Use of Arduino Microcontrollers in Autonomous Robotics," J. Embedd. Syst., vol. 8, no. 3, pp. 156–163, 2014.

V. Mnih et al., "Human-level Control through Deep Reinforcement Learning," Nature, vol. 518, no. 7540, pp. 529–533, 2015.

T. P. Lillicrap et al., "Continuous Control with Deep Reinforcement Learning," in Proc. Int. Conf. Learn. Representations (ICLR), 2016.

Y. Kim and H. Oh, "Dynamic Voltage Scaling for Power Efficient Real-Time Processing," IEEE Trans. Consum. Electron., vol. 63, no. 4, pp. 441–448, 2017.

M. Quigley et al., "ROS: An Open-Source Robot Operating System," in Proc. ICRA Workshop on Open Source Software, 2009.

P. Corke, Robotics, Vision and Control: Fundamental Algorithms in MATLAB, Springer, Cham, Switzerland, 2017.

B. Siciliano and O. Khatib, Springer Handbook of Robotics, Springer, New York, NY, USA, 2016.

H. Chen et al., "Semi-autonomous Industrial Mobile Manipulation for Automated Logistics," in Proc. IEEE Int. Conf. Cyber Technol. Autom. Control Intell. Syst., pp. 361–366, 2013.

aY. Zhao et al., "Recent Advances in Wireless Communication for Autonomous Mobile Robots," IEEE Access, vol. 8, pp. 108135–108145, 2020.