ABSTRACT

The Smart Cleaning Robot is a sophisticated IoT-enabled system designed to automate and optimize household cleaning. Equipped with various sensors like IR obstacle detectors, water level sensors, and motor encoders, the robot continuously senses its surroundings and internal state for effective and consistent operation. In this design, ESP32 acts as the central communication platform, forming the backbone of Wi-Fi communication, WebSocket messaging, and real-time telemetry broadcasting. At the same time, Arduino is used for high-precision motor control via the PWM signal for smooth movement, precise turns, and increased stability on the floor during navigation. A mobile application built in Flutter assists users in monitoring the robot status, adjusting cleaning mode, controlling the direction, and alerting them instantly. The information from the telemetry view consists of data such as water level, sensors, and runtime, which is portrayed in real time to the user for better engagement. The integration of IoT, embedded systems, and robotics in this work eventually aims to provide an intelligent cleaning solution to reduce manual efforts with respect to performance consistency. Smart Cleaning Robot is thus one of the most feasible methods of modern home automation, as it improves convenience and performance while extending user control through a united and responsive digital space.

TABLE OF CONTENT

CONTENTS **Page No.**

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ESP32	Espressif Systems 32-bit Microcontroller
IR	Infrared
TTL	Time To Live
RX	Receiver
TX	Transmitter
COM	Communication Port
LED	Light Emitting Diode
TWI	Two-Wire Interface
PWM	Pulse Width Modulation
IDE	Integrated Development Environment
UART	Universal Asynchronous Receiver–Transmitter
SPI	Serial Peripheral Interface
I2C	Inter-Integrated Circuit
BLE	Bluetooth Low Energy
GPIO	General Purpose Input/Output
GND	Ground
VIN	Voltage Input
EMF	Electromotive Force
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
JS	JavaScript

CHAPTER 1

INTRODUCTION

Rapid urbanization and busy lifestyles have increased the demand for automated solutions that reduce human effort in routine household and industrial tasks. Floor cleaning is one such repetitive and time-consuming activity that requires regular attention to maintain hygiene and cleanliness. Conventional manual cleaning methods are labour-intensive, inconsistent, and inefficient, especially in large areas such as offices, hospitals, shopping malls, and educational institutions. To address these challenges, robotic floor cleaning systems have emerged as an effective alternative, offering automation, consistency, and improved efficiency.

Fig 1.1: Cleaning Robot Using Advancements

The Smart Cleaning Robot project focuses on designing and developing an intelligent, cost-effective robotic system capable of performing floor cleaning operations with minimal human intervention. The proposed system integrates embedded electronics, sensors, motor drivers, and wireless communication to enable autonomous navigation, obstacle detection, and real-time user control as shown in Fig 1.1. Unlike traditional standalone robotic cleaners, this project emphasizes a modular architecture using two microcontrollers—Arduino UNO and ESP32—to distribute control tasks efficiently and enhance system reliability.

In this project, the Arduino UNO is dedicated exclusively to controlling the movement motors of the robot, ensuring precise motion control and smooth navigation. Motor driver circuits are interfaced with the Arduino to regulate direction and speed of the drive motors. The ESP32 microcontroller, on the other hand, handles advanced functionalities such as hosting a web server, managing user commands through a browser-based control panel, and controlling additional L293D motor drivers used for cleaning mechanisms like brushes and water pump motors. This separation of responsibilities improves real-time performance and simplifies system debugging and scalability.

Fig 1.2: Physical Input Invested by Individuals

Wireless control is achieved through an ESP32-based web server, allowing users to operate the robot using any Wi-Fi-enabled device without the need for a dedicated mobile application. Commands such as start, stop, directional movement, and cleaning operations are transmitted via UART serial communication from the ESP32 to the Arduino UNO. The robot is equipped with ultrasonic and infrared sensors to detect obstacles and avoid collisions, ensuring safe navigation in indoor environments. A water level sensor monitors the cleaning fluid, while regulated power supply modules maintain stable voltage levels for all components.

The proposed Smart Cleaning Robot demonstrates how IoT concepts can be effectively applied to real-world automation problems. By combining low-cost hardware, open-source development tools, and a web-based interface, the system provides an accessible solution suitable for academic research, small-scale commercial use, and future enhancements. Overall, this project contributes to the growing field of service robotics by offering a practical, scalable, and efficient automated cleaning solution.

Furthermore, the proposed system emphasizes ease of use, flexibility, and scalability, making it suitable for both domestic and commercial cleaning applications. The web-based control interface eliminates dependency on proprietary mobile applications and enables platform-independent operation. The modular hardware design allows future integration of advanced features such as scheduled cleaning, path optimization, and sensor fusion for improved autonomy. By leveraging widely available components and open-source development platforms, the Smart Cleaning Robot serves as an effective educational model while addressing practical challenges in automated cleaning. This project highlights the potential of embedded and IoT-based robotic systems in improving efficiency and hygiene standards as shown in Fig 1.2.

1.1 Motivation For Project

The motivation for the development of the Smart Cleaning Robot arises from an escalating need for simplification and automation of day-to-day household cleaning activities. In today's modern lifestyle, people cannot allocate enough time and energy to regular house chores like sweeping, vacuuming, and mopping. Classic methods of cleaning require continuous physical effort and regularity, which are very difficult to observe; at the same time, working professionals, elderly people, and those with limited mobility find it very difficult to be involved in cleaning activities. While commercial cleaners using robots are available, most of these are high-end, less customizable, and inflexible for educational or personal development purposes. This project will design an inexpensive, efficient, and user-friendly cleaning robot that will have IoT connectivity, sensors, and a website-based controller integrated into it. The designed system will leverage convenience, raise hygiene standards, and illustrate well that accessible technology can prove quite effective in solving problems encountered in everyday life. This project is motivated to merge innovation with practical utility.

1.2 Goals Of the Project

The main aim of the Smart Cleaning Robot project is to design an efficient and intelligent cleaning system with reduced human effort while maintaining proper hygiene in indoor environments. This project is envisioned to encompass designing a complete robot that would be able to sweep and mop autonomously or when instructed by a web application. Another significant objective is designing it with IoT-based connectivity with the ESP32 for real-time monitoring, updating telemetry, or user interactions within the system. It will also be designed to connect sensors that detect obstacles and the level of water for enhanced safety, precision, and reliability. Furthermore, it will be designed to provide smooth and accurate movement of the robot via motor control using Arduino and motor driver modules. In addition, emphasis will be placed on offering a cost-effective and easy-to-modify platform suitable for learning in academics and prototyping, which may further be expanded. The idea behind this project is to show how embedded systems and IoT technologies can be integrated for the automation of daily tasks.

1.3 Objectives Of Project

- To design and develop a robot to navigate autonomously without human control.
- To detect obstacle using ultrasonic sensor and IR Sensor.

- To integrate vacuum, motor and filter to effectively collect visible dust and small debris.
- To evaluate the performance of autonomous cleaning robot using some performance parameters.

1.4 Problem Statement

Traditional floor cleaning methods involve much time, physical effort, and frequency, which is hard to maintain in the busy lifestyles of today. In most cases, this manual cleaning is inefficient, irregular, and requires a lot from an individual, especially for elderly people and those with mobility problems. While commercial robotic cleaners exist, they are quite expensive, with very limited scope for customization. An inexpensive, intelligent, user-controlled cleaning solution that automates routine floor cleaning tasks is needed. This project discusses a smart cleaning robot by integrating sensor-based navigation, IoT connectivity, and control to ensure efficient, safe, and regular indoor cleaning.

1.5 Applications Of the Project

The Smart Cleaning Robot Can be used in various applications:

- **Residential Cleaning:** The residential cleaning model assists in automating daily sweeping and mopping tasks, thus helping homes and apartments maintain consistent cleanliness, even when users are busy or away.

- **Offices and Commercial Workspaces:** Keeps the floors clean in offices, clinics, shops, and studios. Can work outside of working hours to avoid disturbing employees or customers.

- **Health Care/Small Medical Facilities:** Assists in keeping clean, clinics, physiotherapy centres, and diagnostic laboratories. Dust accumulation is reduced, helping to keep non-critical areas hygienically clean.

- **Support for the Elderly and Physically Challenged:** Suitable for elderly and physically challenged people when they might find it difficult to do manual cleaning. It is also compatible with a web interface to clean your house while sitting comfortably inside it.

CHAPTER 2

LITRATURE SURVEY

2.1 Objective 1: To design and develop a robot to navigate autonomously without human control.

Mohamed S. Saleh, et al., [1] investigated optimal mobile robot navigation and obstacle avoidance using an Adaptive Neuro-Fuzzy Inference System (ANFIS) controller. Saleh introduced a multi-layer ANFIS controller to manage both movement and obstacle avoidance in cluttered environments by using sensor data from infrared and ultrasonic distance sensors. Through MATLAB simulation, the proposed controller was tested across multiple scenarios with different obstacle configurations, enabling the robot to reach targets without collision and in optimized paths.

S. Kumar, et al., [2] developed a method for autonomous navigation and real-time mapping using ultrasonic sensors in a humanoid robot. In this work, S. Kumar focused on enabling a robot to navigate unknown environments without human control by using ultrasonic distance measurements to build a 2-D map of the surroundings and estimate its own position. The research employed the Webots simulator for implementation and testing, relying solely on ultrasonic sensors to detect obstacles and compute the robot's motion path.

Naresh Tuppathi, et al., [3] designed and implemented an autonomous robotic vehicle capable of detecting and avoiding obstacles using ultrasonic sensors interfaced with an ATmega328 microcontroller. Tuppathi's primary contribution was the development of the reactive navigation algorithm that processes ultrasonic sensor data to make real-time decisions about movement direction. When the sensor detects an object within a predefined threshold distance, the control logic automatically triggers steering adjustments to avoid collisions.

Imran Mokashi, et al., [4] developed a smart vacuum cleaner robot that operates autonomously in diverse indoor environments using an Arduino Uno microcontroller as the central control unit. The system integrates ultrasonic sensors for real-time obstacle detection and motor drivers for movement control to ensure smooth, collision-free navigation without human control. Mokashi's work focused on both hardware and software optimization—configuring sensors for reliable distance measurement and programming the control logic to manage decision-making for motion and cleaning tasks.

2.2 Objective 2: To detect obstacle using ultrasonic sensor and IR Sensor.

A. Pandey, et al., [5] developed an autonomous mobile robot system for obstacle detection and avoidance using a combination of ultrasonic and infrared (IR) sensors. A. Pandey focused on integrating ultrasonic sensors for long-range distance measurement and IR sensors for short-range obstacle detection to improve reliability in indoor environments. The sensor data were processed by a microcontroller to determine obstacle presence and trigger appropriate movement decisions. Experimental results showed that the combined use of ultrasonic and IR sensors enhanced obstacle detection accuracy and reduced collision probability.

S. Kumar, et al., [6] proposed an obstacle detection and navigation system based on ultrasonic and infrared sensing techniques for mobile robotic applications. S. Kumar concentrated on designing a sensor-based control algorithm that continuously monitored the robot's surroundings using ultrasonic sensors for distance estimation and IR sensors for edge and object detection. The robot adjusted its motion in real time to avoid obstacles without human intervention. Performance evaluation demonstrated smooth navigation, reliable obstacle detection, and improved safety in cluttered environments.

L. N. Prashanth, et al., [7] developed an obstacle detection and navigation system for a smart cleaning robot using ultrasonic and infrared sensors. L. N. Prashanth focused on employing ultrasonic sensors to measure obstacle distance and IR sensors to detect nearby objects and edges in indoor environments. The sensor data were continuously processed by a microcontroller to determine safe navigation paths. When obstacles were detected within a predefined threshold, the robot altered its direction automatically. Experimental evaluation confirmed reliable obstacle detection, smooth navigation.

K. Sathish, et al., [8] designed an autonomous robotic platform for obstacle detection and avoidance using ultrasonic and IR sensing techniques. K. Sathish concentrated on integrating ultrasonic sensors for long-range obstacle detection and IR sensors for short-range and edge detection. The microcontroller processed real-time sensor inputs to control motor movement and prevent collisions. Testing in indoor environments showed accurate obstacle recognition and responsive navigation behaviour. The study concluded that the combined use of ultrasonic and IR sensors improves detection reliability and ensures safer autonomous operation in cleaning and service robots.

2.3 Objective 3: To integrate vacuum, motor and filter to effectively collect visible dust and small debris.

Tri Cuong Do, et al., [9] developed a smart autonomous vacuum cleaning robot that enhances debris removal efficiency using a dual-stage centrifugal vacuum system integrated within a mobile platform. Tri Cuong Do's work concentrated on configuring the vacuum unit to generate sufficient suction while maintaining a compact form factor appropriate for residential use. The robot employed differential-drive kinematics and a ROS 2 autonomy framework to coordinate navigation and cleaning actions.

Soorya Ramdas, et al., [10] proposed a floor vacuum cleaning robot that integrates a robotic chassis, motion controller, and vacuum collection mechanism to enable autonomous debris pickup during navigation. The first author designed the vacuum subsystem to work alongside the robot's locomotion such that particles are gathered as the robot moves along preprogrammed paths. The system included coordinated motor control for wheel motion and vacuum fan operation, allowing suction to occur simultaneously with navigation. Testing results confirmed that the robot effectively collected small debris along its path without human intervention.

W. A. Heitbrink, et al., [11] investigated the performance of commercial vacuum cleaners by analysing the interaction between the motor, airflow, and filtration system under debris loading conditions. The first author focused on evaluating how dust accumulation affected airflow resistance and suction capability. The system design emphasized the role of filter type and motor power in maintaining consistent debris collection. Experimental testing demonstrated that vacuum systems using cyclonic separation maintained higher suction efficiency compared to traditional bag filters.

J. V. Fayzullayevich, et al., [12] presented a numerical study on a regenerative air vacuum sweeper designed to improve particle pickup efficiency. The authors integrated a suction motor, pickup head, and particle separation mechanism to analyze debris collection behavior. The first author modeled particle trajectories under varying suction pressures and vehicle speeds using CFD techniques. Results indicated that optimized suction geometry and airflow distribution significantly improved the collection of visible dust and small debris, demonstrating the effectiveness of coordinated motor and vacuum system design.

2.4 Objective 4: To evaluate the performance of autonomous cleaning robot using some performance parameters.

A. Kumar, et al., [13] conducted a comprehensive performance evaluation of an autonomous cleaning robot using key parameters such as cleaning efficiency, area coverage, navigation accuracy, and power consumption. The first author developed a structured experimental setup to quantify debris pickup rate across different floor conditions and obstacle layouts. Sensor feedback, motor speed, and battery discharge characteristics were analysed to assess system reliability and operational stability. Experimental results demonstrated consistent cleaning performance with high coverage efficiency and effective energy utilization, validating the proposed evaluation methodology.

S. Park, et al., [14] proposed a detailed performance assessment framework for an autonomous vacuum cleaning robot based on parameters including coverage ratio, obstacle avoidance success rate, path redundancy, and task completion time. The first author focused on evaluating the robot's localization accuracy and motion control behavior during autonomous navigation. Real-time sensor data and trajectory logs were used to analyze navigation efficiency. The results showed that optimized path planning significantly reduced overlap and improved cleaning effectiveness, confirming the usefulness of performance-based evaluation metrics.

R. Singh, et al., [15] analysed the performance of an intelligent floor-cleaning robot by evaluating suction efficiency, battery endurance, cleaning duration, and surface adaptability. The first author designed controlled experiments to test robot performance on different floor types such as tile, wood, and carpet. Motor load variation and vacuum airflow performance were monitored during operation. The findings indicated that adaptive speed and suction control enhanced debris collection while extending battery life.

L. Chen, et al., [16] presented a systematic evaluation of an autonomous cleaning robot using performance indicators such as coverage efficiency, localization precision, system reliability, and fault tolerance. The first author emphasized benchmarking robot performance under varying obstacle densities and dynamic environmental conditions. Long-duration tests were conducted to analyze consistency and repeatability of cleaning behavior.

2.5 Scope Of the Project

1. Technical Scope

- Microcontroller Integration: This utilizes ESP32 for Wi-Fi communication, WebSocket telemetry, and remote control. Arduino drives the motor control with precision, such as PWM, and manages real-time operations like running actuators.

- Sensor-Based Automation: Integrates IR sensors to detect obstacles and navigate safely. The water level sensor ensures that the mopping mechanism works appropriately.

- Motor and Driver System: Supports DC motors controlled by L293D motor drivers. Smooth navigation by enabling movement forward, reverse, turning, and at variable speeds.

- IoT and Networking: Operates on local Wi-Fi networks for stable communication. Real-time telemetry and bidirectional commands via WebSocket protocol.

- Power System: Includes rechargeable battery support for autonomous operation. Power regulation ensures safe operation of microcontrollers, sensors, and motors.

2. Functional Scope

- Automatic Cleaning Mode: Automatically navigates and cleans rooms without user intervention. Obstacle avoidance, speed regulation, and safe stops are done automatically.

- Manual Cleaning Mode: The users can control the movement of the robot through the web interface using controls. Ideal for cleaning specific spots or manoeuvring manually through complicated spaces.

- Sweeping and mopping capabilities: It supports dual-mode cleaning: dust removal and floor mopping.

CHAPTER 3

OBJECTIVES AND METHADOLOGY

3.1 Objectives Of Project

- To design and develop a robot to navigate autonomously without human control.
- To detect obstacle using ultrasonic sensor and IR Sensor.
- To integrate vacuum, motor and filter to effectively collect visible dust and small debris.
- To evaluate the performance of autonomous cleaning robot using some performance parameters.

3.2 Methodology Of Project

The complete operational flow of the Smart Cleaning Robot is represented by Fig. 3.1, showing the methodology. First of all, the robot is powered on and all the hardware components are initialized, which include the ESP32, Arduino, motor driver, sensors, and Wi-Fi module. Then, the ESP32 initializes the web server and attempts to connect to the local Wi-Fi. If the connection isn't established, the system will try again until a stable connection is achieved. Once connected, the web-based control interface gets launched and becomes accessible with a browser. Sensor data is continuously monitored and transmitted, while user commands received from the website will drive the robot for its movement and cleaning operations.

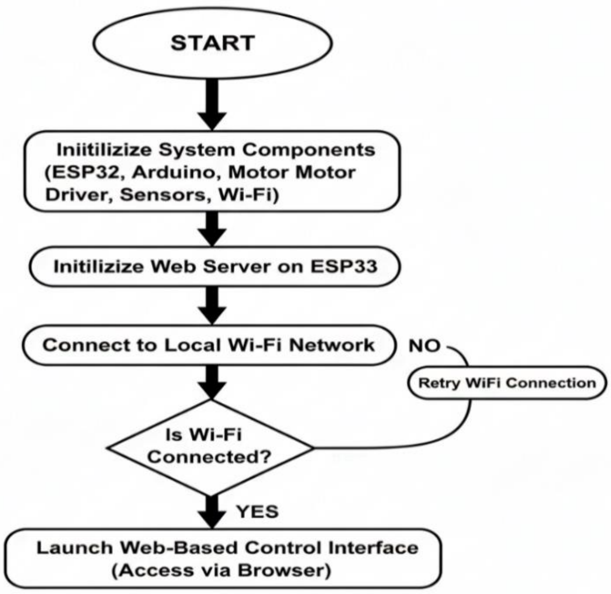

Fig 3.1: Flowchart Representing the Overall Methodology

Fig 3.2 represents the methodology of the Smart Cleaning Robot in manual mode. After the robot is powered on, the operating mode is chosen through the web-based interface. Once the manual mode has been selected, the system waits for a command to start/stop or change the direction and speed from the website. Such commands are forwarded to the motor driver to control the movement of the robot. Simultaneously, IR sensors and water level sensors monitor the environment incessantly. On detecting an obstacle, motors stop, changing the direction. Live status is posted to the website dashboard until a stop command is issued following which the robot turns into the idle state.

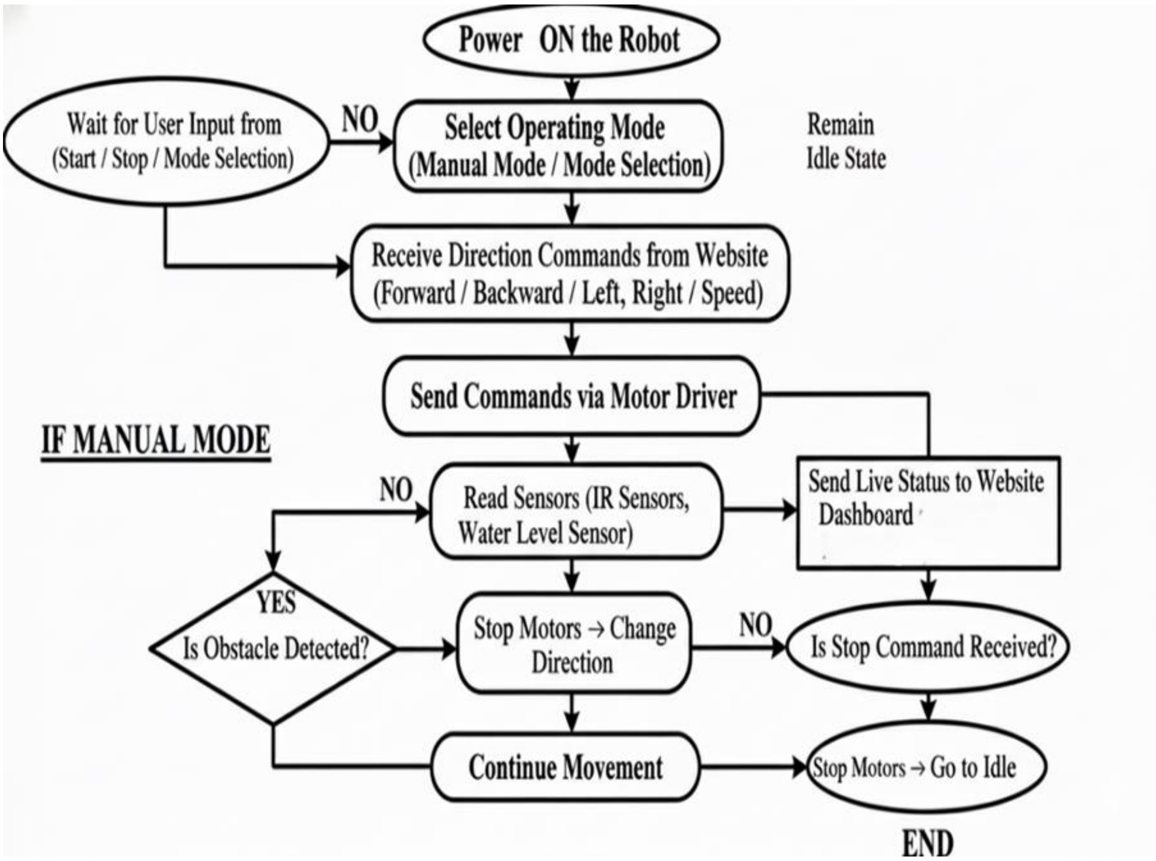

Fig 3.2: Flowchart Representing Manual Mode Operation

The smart cleaning robot starts operating in autonomous mode when this option is selected, as shown in Fig 3.3. The robot begins its cleaning routine and constantly checks for obstacles with IR sensors. If it detects an obstacle, the robot stops, turns left or right, and then moves forward again. The sweeping and mopping mechanisms turn on while it moves. A water level sensor keeps track of the cleaning fluid and shows alerts on the website if the level is low. Live data, including speed, obstacle status, and water level, is sent to the web interface. When the cleaning area is finished, the motors stop, the system status is updated, and the robot either goes into idle mode or restarts if needed.

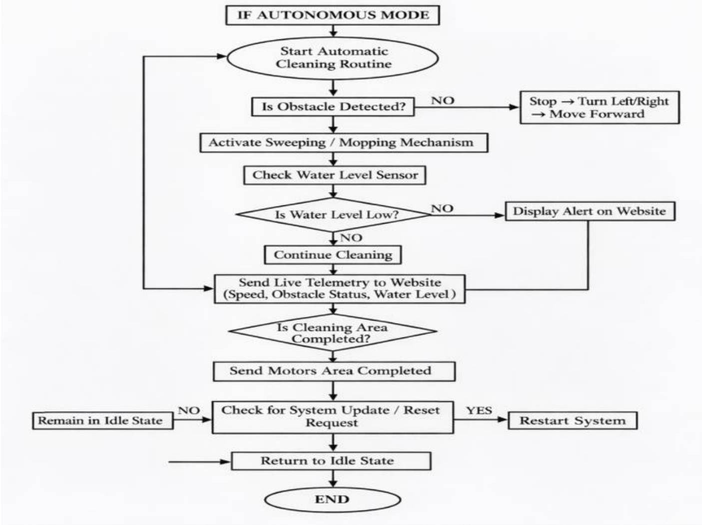

Fig 3.3: The Autonomous cleaning methodology robot

Once the smart cleaning robot is turned on, the first stage it goes through is powering and initializing all of its different system components (such as ESP32, sensors, motor drivers, and web server), as shown in Fig 3.4. The robot connects to a local Wi-Fi network before allowing users to select between manual & autonomous operation modes via a website interface. In manual mode, the user can give the robot specific movement commands and those movements will continue until a stop signal is sent. In autonomous mode, the robot will clean automatically while still detecting obstacles with sensors; should an obstacle be detected, the robot will avoid hitting it by changing direction but continuing to clean. Upon completing its cleaning duties, the robot stops working and goes back into an idle state.

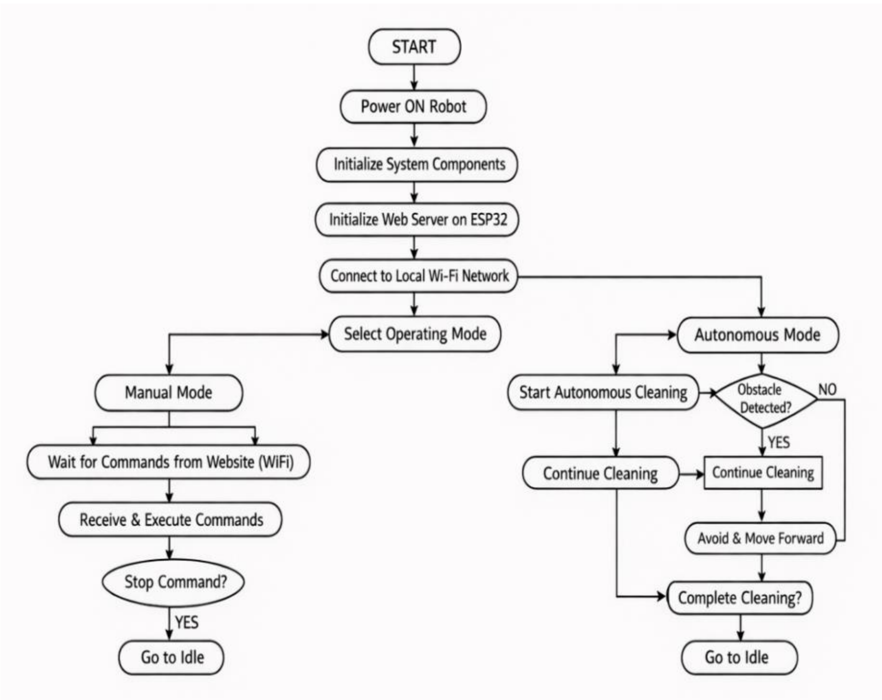

Fig 3.4: Shortened flowchart illustrating the working methodology

Fig 3.4 illustrates the operation of the smart cleaning robot. It begins by powering on and initializing the microcontroller, motor driver, sensors, and cleaning devices. The ESP32 then sets up a web server via the local Wi-Fi network, allowing users to control and monitor the robot through a web interface. Users can choose between manual and autonomous modes. In manual mode, users send movement commands—forward, backward, left, right, and stop—through the interface, with the motor driver executing these actions. Real-time motor status and sensor readings are displayed for monitoring. In autonomous mode, once initiated, the robot performs touchless cleaning by moving forward while scanning for obstacles with IR sensors. Upon detecting an obstacle, it stops, changes direction, and continues cleaning. The sweeping and mopping mechanisms work simultaneously for thorough cleaning. After covering the entire area, the robot shuts down its motors and enters an idle state, as shown in Fig 3.1.

3.3 Circuit Diagram

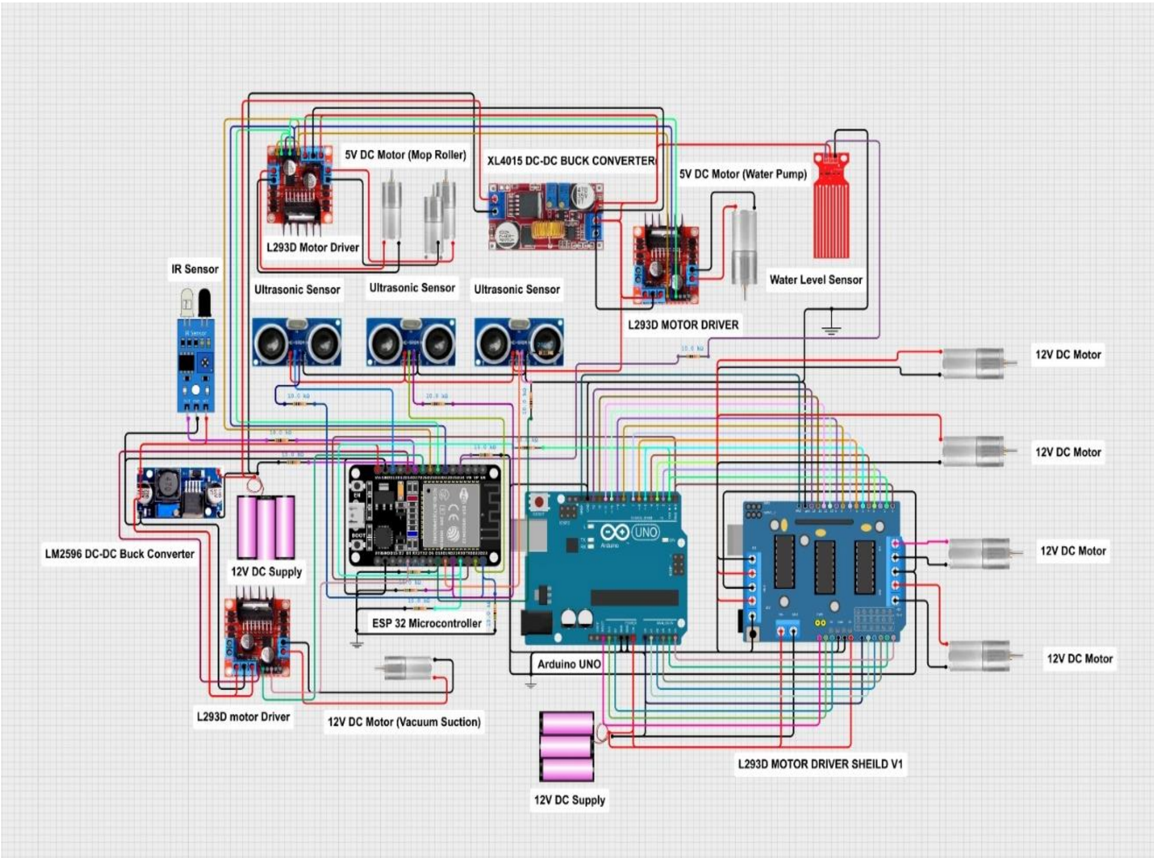

Fig 3.5: Complete circuit diagram of the Cleaning Robot

The complete hardware implementation of the smart cleaning robot consists of control, sensing, actuation, and power management units integrated into a single system. The overall circuit configuration is illustrated in Fig 3.5. The system uses two controllers, namely the ESP32 microcontroller and the Arduino UNO. The ESP32 is responsible for hosting the web server and handling wireless communication, allowing users to control the robot through a browser-based interface. High-level commands received from the web interface are transmitted to the Arduino UNO using UART serial communication.

The Arduino UNO performs real-time processing and controls all peripheral devices. As shown in Fig 3.5, ultrasonic sensors are connected to the Arduino to detect obstacles and measure distances during navigation. An IR sensor is used for short-range obstacle detection, improving accuracy near walls and objects. A water level sensor continuously monitors the water tank and ensures safe operation of the pump by preventing dry running.

Motor control is achieved using multiple L293D motor driver modules and an L293D Motor Driver Shield V1. These drivers control the wheel motors, vacuum suction motor, mop roller motor, and water pump motor. Pulse Width Modulation (PWM) signals from the Arduino allow smooth speed control, while directional pins enable forward, reverse, and turning motions.

Power is supplied by a 12V lead-acid battery. Voltage regulation is handled using LM2596 and XL4015 DC–DC buck converters, which step down the voltage to 5V and other required levels for sensors and controllers. A common ground is maintained across all components to ensure stable and reliable operation. The integrated design shown in Fig 3.5 enables efficient cleaning, safe navigation, and real-time monitoring through the web interface.

- When power is supplied, the 12V battery energizes the system, and the DC–DC buck converters regulate the voltage for controllers, sensors, and motors.
- The ESP32 initializes its Wi-Fi module and hosts a web server, enabling wireless communication with the user through a web-based control panel.
- User commands such as start, stop, direction, and mode selection are received by the ESP32 via a web browser and forwarded to the Arduino UNO using UART serial communication.
- The Arduino UNO processes the received commands and controls the robot’s movement and cleaning mechanisms in real time.
- Ultrasonic sensors continuously measure the distance to obstacles, while the IR sensor provides short-range object detection to avoid collisions.

CHAPTER 4

DESIGN AND DEVELOPMENT

4.1 Hardware Description

Hardware components required for the proposed system is as listed below:

Hardware Requirements

- Arduino UNO
- L293D Motor Driver Shield V1
- ESP32 DEVKIT V1
- Ultrasonic Sensor (HC-SR04)
- IR Sensor
- Water Level Sensor
- L293D Motor drivers
- XL4015 DC to DC Buck Converter
- LM2596 DC to DC Buck converter
- DC Motor
- BO Motor
- Water Pump Motor
- 12V 7ah Lead Acid Battery

4.1.1 Arduino UNO

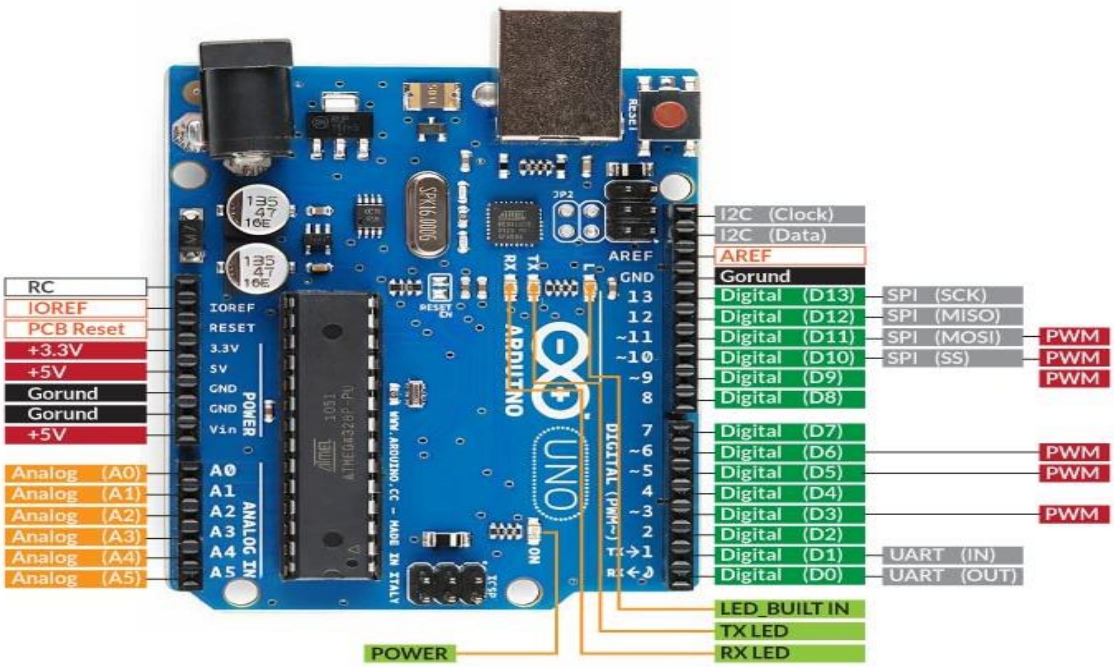

Fig 4.1: Description of Arduino uno

Arduino UNO a microcontroller that is perfect for robotic and IoT projects, It has an ATMega328p chip and 14 I/O pins, among those 6 are pins which support PWM which is also, for controlling Voltage and sensors for their broad applications as illustrated in Fig 4.1. The necessary features, such as a 16 MHz quartz crystal for accurate timekeeping, a USB port for easy communication with your computer, a power jack for power and push buttons for the carrier the user communicates. With 32 KB of storage and 2 KB of memory, you’ve got enough room to store your programs and data. Plus, the Arduino IDE makes programming a breeze, delivering a simple and easy to-understand language, even for beginners.

- The serial pins 0 (Rx) and 1 (Tx) serve as conduits for TTL serial data transmission and reception. These pins establish communication with the ATmega328P to ESP32.

- The Pulse Width Modulation (PWM) pins, namely Pins 3, 5, 6, 9, and 11, are capable of generating 8-bit PWM signals using the analogWrite() function, enabling precise control of motor speed and actuator intensity.

- The built-in LED Pin TX and RX is directly linked to a built-in LED. A logic high signal on pin 0 and pin 1 activates the LED, while a logic low signal turns it off.

Arduino UNO Pin Numbers Used in the Project

- **Pin 0 (RX)** – Receives commands from ESP32
- **Pin 1 (TX)** – Sends status data to ESP32
- **Pins 3, 5, 6, 9** – PWM signals to L293D motor driver shield
- **Digital Pins 10–13** – Motor direction control
- **5V & GND** – Power supply for sensors and motor driver

Communication

Arduino makes the process simpler to interact with other microcontrollers, computers, and devices other than boards such as Arduino. Digital pins 0 (Rx) and 1 (Tx) on the ATmega328P microcontroller enable UART TTL (5V) serial connection. The ESP32 microcontroller handles this serial communication through Pin D16 (RX2) And Pin D17 (TX0), presenting to programs on the computer as a virtual COM port. External drivers

(CH340) are Necessary since the Microcontrollers like Arduino UNO and esp32 cannot communicate directly to the computer through USB this helps in functioning effectively. Simple text communication with the Arduino board can be made simple with the serial monitor which has been integrated in the Arduino software. In addition to that, the board includes RX and TX LEDs which light up when data is transmitted utilizing the pin0 and pin1 and the USB connection to the computer, having pins 0 and 1 without being utilized in serial communication. Users may interact repeatedly on any of the Uno's digital pins via the Software Serial library. The ATmega328P chip also supports SPI and I2C (TWI) communication. A Wire library has been built in the Arduino software to help with making use of the I2C bus.

4.1.2 L293D Motor Driver Shield V1

The L293D Motor Driver Shield V1 is a dual H-bridge motor driver module designed to control DC motors, stepper motors, and relays in robotic applications as shown in Fig 4.2. It is based on the L293D integrated circuit, which allows bidirectional control of motors with a supply voltage range of 4.5 V to 36 V and a continuous output current of up to 600 mA per channel. This shield is specifically designed to be mounted directly on the Arduino UNO, simplifying hardware connections and reducing wiring complexity.

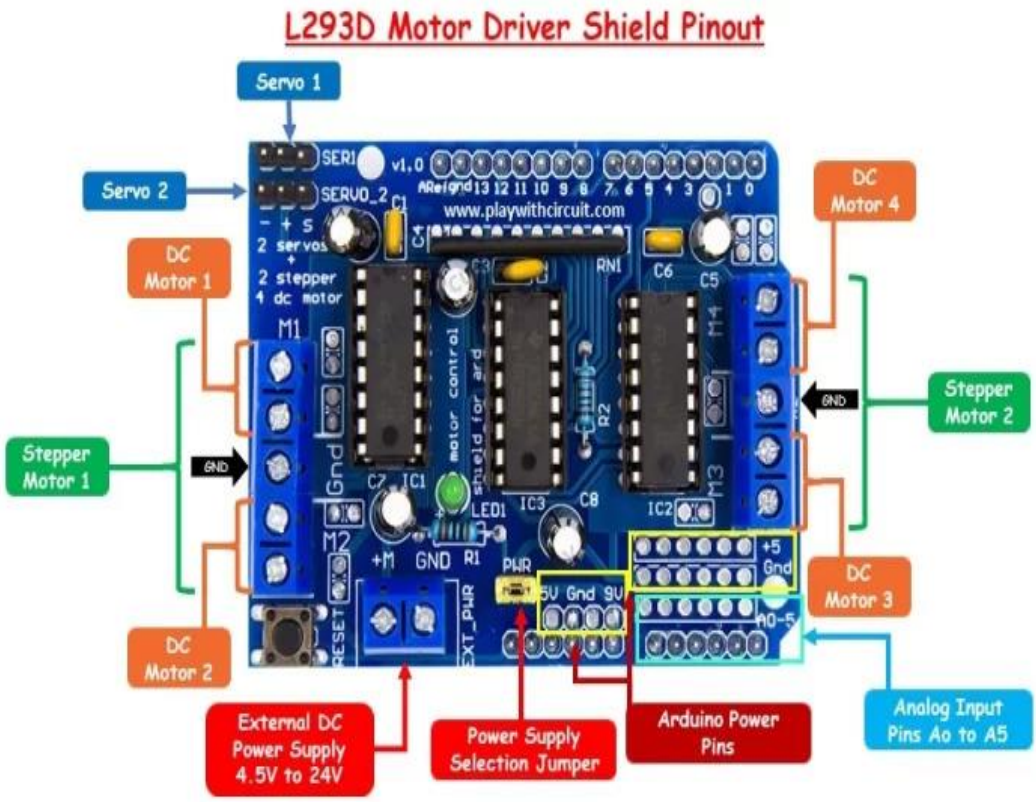

Fig 4.2: Description of L293D Motor Driver Shield V1

The shield supports control of up to four DC motors or two stepper motors simultaneously. Motor speed is controlled using Pulse Width Modulation (PWM) signals generated by the Arduino, while motor direction is managed through digital control pins. In this smart cleaning robot project, the L293D motor driver shield is used to drive the DC motors responsible for robot movement, enabling forward, backward, left, and right navigation.

The motor driver shield includes built-in flyback diodes, which protect the circuit from voltage spikes generated by inductive motor loads. It also provides separate power terminals for motor supply and logic supply, ensuring stable operation and preventing damage to the microcontroller. The simplicity, reliability, and compatibility with Arduino IDE make the L293D Motor Driver Shield V1 suitable for implementing real-time motor control.

Characteristics of L293D Motor Driver Shield V1

- Dual H-bridge motor driver based on the L293D integrated circuit
- Supports control of up to four DC motors or two stepper motors
- Provides bidirectional motor control (forward and reverse rotation)
- Operates with a motor supply voltage range of 4.5 V to 36 V
- Delivers a continuous output current of up to 600 mA per channel
- Supports Pulse Width Modulation (PWM) for precise motor speed control
- Designed as a stackable shield compatible with Arduino UNO
- Provides separate power connections for logic and motor supply
- Enables independent control of multiple motors
- Suitable for robotic navigation and automation applications
- Simple integration with the Arduino IDE and Embedded C/C++ programming

Communication

- Receives control signals directly from the Arduino UNO digital and PWM pins
- Uses logic-level (5 V TTL) signals for motor direction and speed control
- Communicates indirectly with the ESP32 through the Arduino UNO, enabling web-based command execution
- Supports PWM-based communication for real-time motor speed variation
- Allows seamless integration with UART-based command processing handled by the Arduino UNO.

4.1.3 ESP32 DEVKIT V1

The ESP32 is a high-performance microcontroller widely used in IoT and robotic applications due to its integrated wireless communication capabilities. It is based on a 32-bit dual-core Xtensa LX6 processor operating at a clock frequency of up to 240 MHz, enabling efficient multitasking and real-time control. The ESP32 supports both Wi-Fi (802.11 b/g/n) and Bluetooth (Classic and BLE), making it suitable for web-based monitoring and control systems as shown in Fig 4.3.

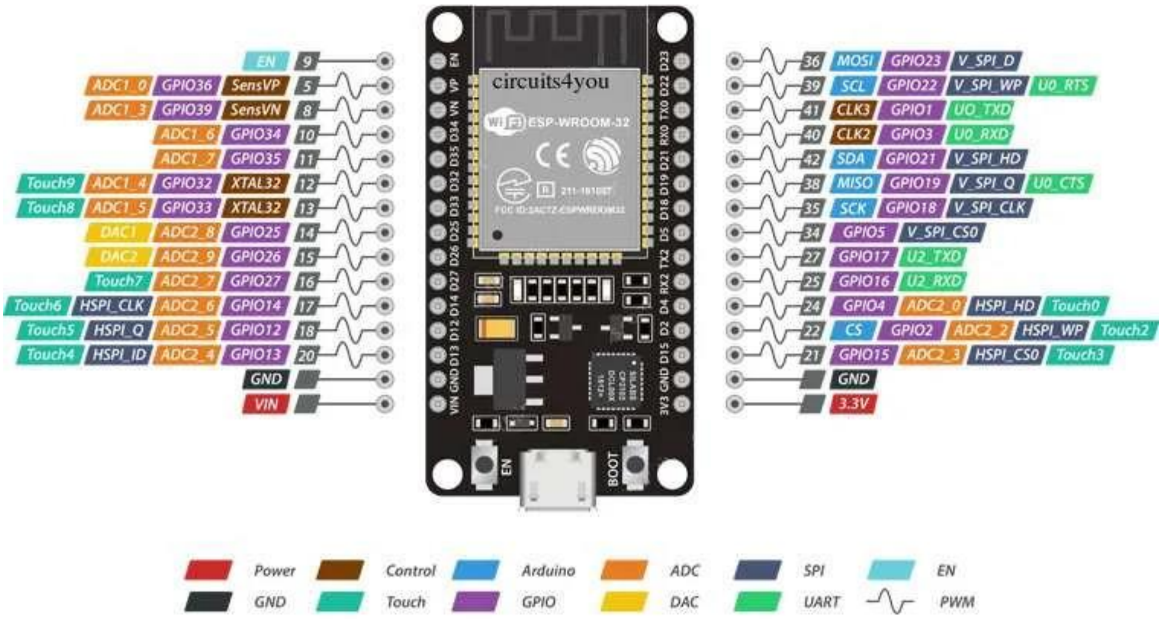

Fig 4.3: Description of ESP32 Devkit V1

- The ESP32 (30-pin) development board provides 30 physical pins, including power, ground, and GPIO pins.
- It includes up to 25 usable GPIO pins for digital input/output operations. All GPIO pins operate at a 3.3 V logic level, ensuring low power consumption.
- Multiple pins support Pulse Width Modulation (PWM), enabling motor speed and actuator control. Dedicated pins are available for UART, SPI, and I2C communication.
- Supports UART serial communication for interfacing with the Arduino UNO.
- Uses Wi-Fi (802.11 b/g/n) for hosting a web server and receiving control commands.
- Supports simultaneous client access through the web interface. Enables bidirectional data communication between the robot and the website server.

Note: ESP32 operates at 3.3 V logic, while serial communication with Arduino UNO is handled safely via UART, it is handled by using voltage divider circuit to match the ESP32 voltage requirements.

ESP32 Pin Numbers Used in the Project

- GND – Common ground connection
- VIN (5V) – External power input
- GPIO 16 (RX2) – Receives serial data from Arduino UNO (TX)
- GPIO 17 (TX2) – Transmits serial commands to Arduino UNO (RX)
- GPIO 34 – Water Level Sensor
- GPIO 35 – Charge Left in Battery
- GPIO 27 – IR Sensor Output signal
- GPIO 13 – Ultrasonic Sensor (Front) Trigger pin
- GPIO 23 – Ultrasonic Sensor (Front) Echo pin
- GPIO 22 – Ultrasonic Sensor (Front) Trigger pin
- GPIO 21 – Ultrasonic Sensor (Front) Echo pin
- GPIO 19– Ultrasonic Sensor (Front) Trigger pin
- GPIO 18 – Ultrasonic Sensor (Front) Echo pin
- GPIO 14 – Vacuum Motor Enable 1
- GPIO 26 – Vacuum Motor Control 1
- GPIO 4 – Vacuum Motor Control 2
- GPIO 25 – Mop Control Motor Enable1
- GPIO 33 – Mop Control Motor Control 1
- GPIO 32 – Mop Control Motor Control 2

ESP32 to Arduino UNO Pin Mapping Table

Table 4.1: Pin Mapping Between ESP32 and Arduino UNO

ESP32 (30-Pin)	Function	Arduino UNO	Purpose
GPIO 17 (TX2)	UART Transmit	Pin 0 (RX)	ESP32 → Arduino commands
GPIO 16 (RX2)	UART Receiver	Pin 1 (TX)	Arduino → ESP32 status
GND	Ground	GND	Common Ground
VIN/5V	Power input	5V	Power reference

4.1.4 Ultrasonic Sensor (HC-SR04)

The ultrasonic sensor is used for obstacle detection and distance measurement in the smart cleaning robot. It operates on the principle of ultrasonic sound wave reflection, where high-frequency sound waves are transmitted and the reflected echo is received to calculate the distance of an object. This sensor enables the robot to detect obstacles in real time and avoid collisions during autonomous navigation, as illustrated in Fig 4.4.

In this project, the ultrasonic sensor continuously monitors the surrounding environment and provides distance data to the ESP32. Based on this information, the control algorithm adjusts motor direction and speed to ensure safe movement. Due to its accuracy, reliability, and ease of interfacing, the ultrasonic sensor is widely used in robotic navigation systems.

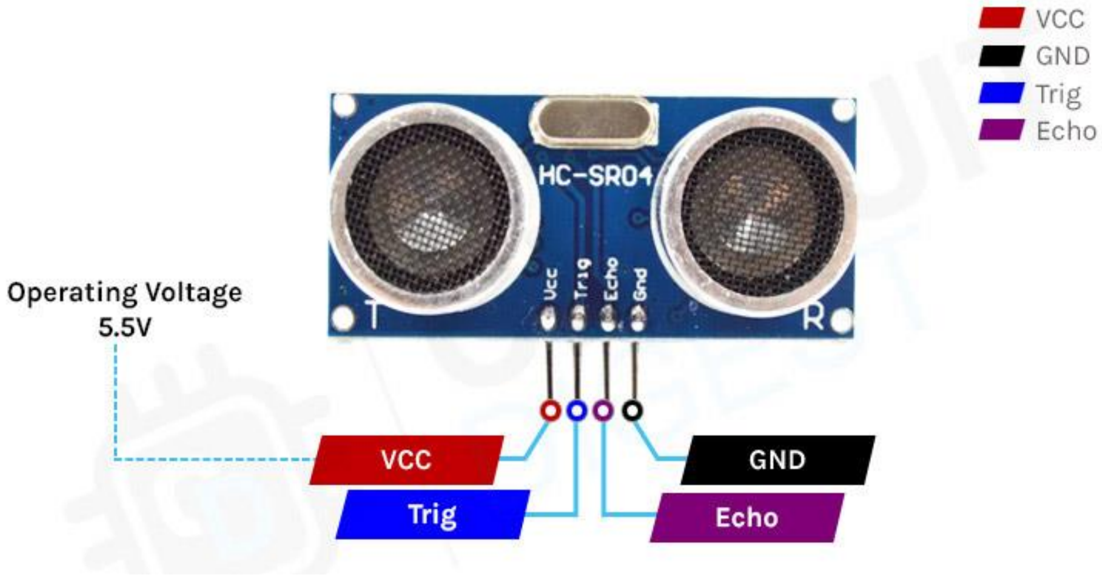

Fig 4.4: Ultrasonic Sensor

The HC-SR04 Ultrasonic Sensor is a 4pins module with main pins being VCC, Ground, Echo, Trigger. It performs the function of measuring distance of obstacle it detects, to perform the function it uses ultrasonic waves as shown in Fig 4.5. ultra sonic sensor converts electric energy to sound waves.

The build design of this sensor is, it resembles two eyes which does the function of transducer. This transducer has the frequency ranging above 18kHz for navigating the objects. HC-SR04 sensor works on the sonar principle like bat navigation and can detect the distance of objects without the need for body contact.

It provides reliable, high-precision readings covering non-contact measurements from 2 cm to 400 cm or 1 inch to 13 feet in a convenient package.

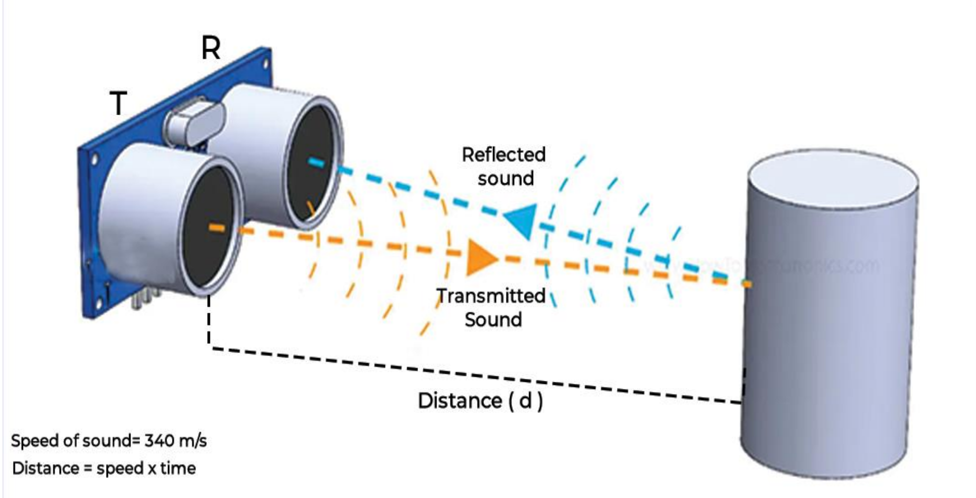

Fig 4.5: Working of Ultrasonic Sensor

Specifications:

- Power Supply :5V DC
- Quiescent Current: <2mA
- Effectual Angle: <15°
- Ranging Distance: 2cm – 500 cm/1" - 16ft
- Resolution: 0.3 cm

The HC-SR04 Ultrasonic sensor is a widely used 4-pin module with pins named VCC, Trigger, Echo, and Ground. It serves essential functions in distance measurement and object detection through the emission and reception of ultrasonic waves. By employing the basic formula Distance = Speed × Time, with the speed of ultrasonic waves in air typically around 330m/s, the sensor determines the distance by measuring the time taken for the wave to return after reflecting off an object

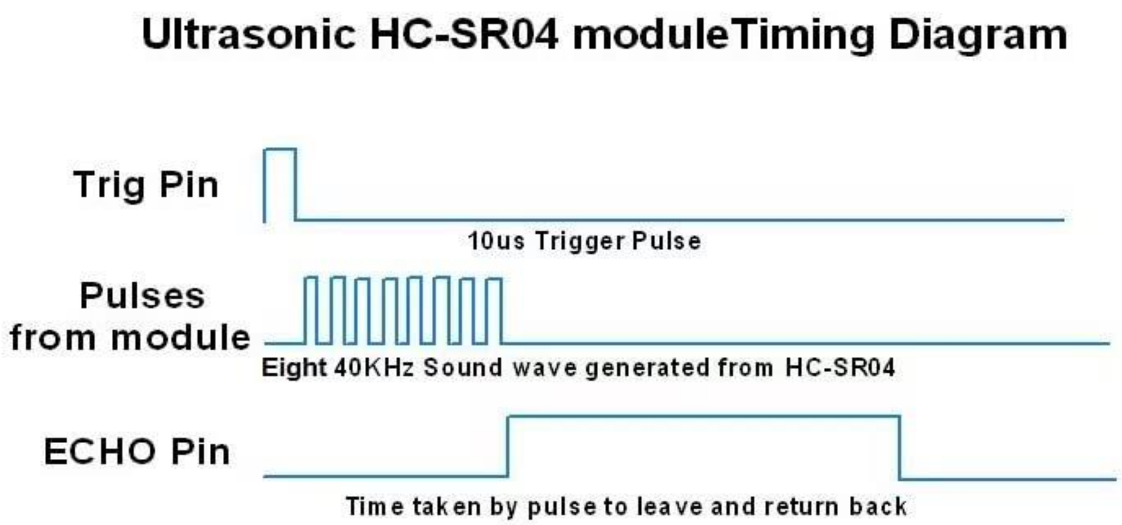

Fig 4.6: Ultrasonic Sensor Module Timing Diagram

4.1.5 IR Sensor

The infrared (IR) sensor is used for object detection and edge sensing in the smart cleaning robot. It operates on the principle of infrared light emission and reflection, where an IR transmitter emits infrared rays and an IR receiver detects the reflected signal from nearby objects or surfaces. This enables the robot to identify obstacles, walls, and edges, such as staircases, ensuring safe navigation, as shown in Fig 4.7.

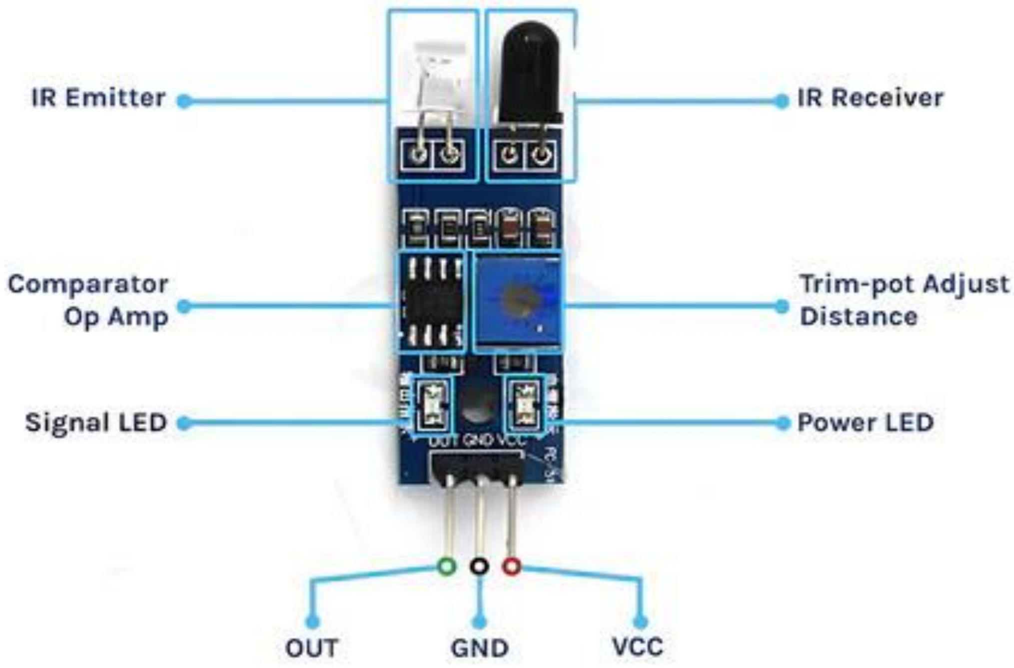

Fig 4.7: Description of IR Sensor

In this project, the IR sensor continuously monitors the robot's immediate surroundings and provides digital output to the ESP32. Based on the sensor feedback, the control algorithm initiates appropriate actions such as stopping or changing direction. The IR sensor is commonly used in robotic systems due to its fast response, simplicity, and cost-effectiveness.

IR Sensor – Pin Description

- **VCC** – Supplies operating voltage (typically 3.3 V or 5 V)
- **GND** – Ground connection
- **OUT** – Digital output indicating object detection

IR Sensor – Characteristics

- Operates using infrared light reflection principle
- Typical operating voltage of 3.3 V to 5 V DC

- Provides digital output for easy interfacing
- Fast response time for real-time detection
- Effective for short-range object and edge detection
- Adjustable sensitivity using onboard potentiometer
- Low power consumption
- Suitable for indoor robotic applications

IR Sensor – Communication

- Communicates with the ESP32 using a digital output signal
- Output changes based on the presence or absence of an object
- Enables quick decision-making for obstacle avoidance and edge detection

4.1.6 Water Level Sensor

The water level sensor is used to monitor the water level in the cleaning tank of the smart cleaning robot. It operates by detecting the presence and height of water through conductive sensing tracks, enabling the system to determine whether sufficient water is available for wet mopping operations. This ensures efficient water usage and prevents the pump from operating under dry conditions, as illustrated in Fig 4.8.

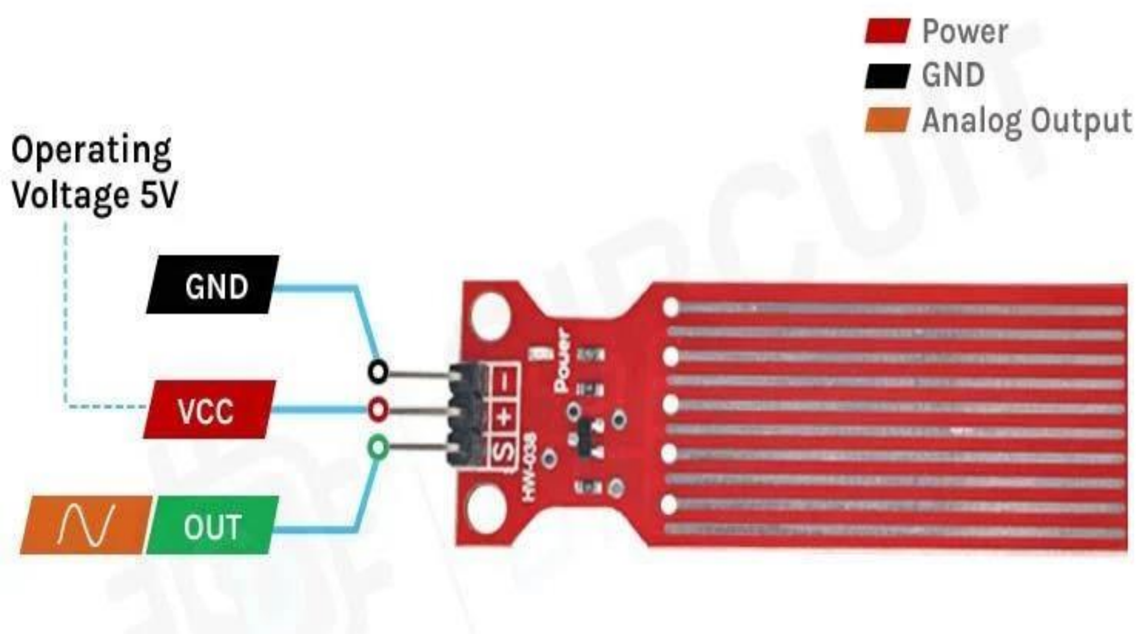

Fig 4.8: Water Level Sensor Used for Monitoring Water Tank Level

In this project, the water level sensor continuously provides feedback to the ESP32. When the water level falls below a predefined threshold, the control system can stop the water pump or alert the user through the web interface. Due to its simple design, reliability, and ease of interfacing, the water level sensor is well suited for smart cleaning applications.

Water Level Sensor – Pin Description

- VCC – Supplies operating voltage (typically 3.3 V or 5 V)
- GND – Ground connection
- Signal (S) – Outputs analog or digital signal proportional to water level

Water Level Sensor – Characteristics

- Operates on conductive sensing principle
- Typical operating voltage of 3.3 V to 5 V DC
- Provides analog output for measuring water level
- Fast response time for real-time monitoring
- Simple structure and low power consumption
- Easy interfacing with ESP32
- Suitable for small water tanks and reservoirs

Water Level Sensor – Communication

- Communicates with the ESP32 using an analog signal
- Water level is determined by reading the sensor voltage value
- Enables automatic control of water pump and cleaning operation

4.1.7 L293D Motor Driver

The L293D is a widely used motor driver IC that enables the control of DC motors in both directions. With its 16-pin configuration, it has the capability to manage two DC motors concurrently, offering versatility in motor control. Serving as a dual H-bridge motor driver, the L293D facilitates the operation of various sizes of motors, ranging from small to relatively large ones, as illustrated in Fig 4.9.

In this project, the ESP32 sends direction and speed control signals to the L293D motor driver board. The board receives logic-level inputs and PWM signals to regulate motor rotation and speed. This motor driver board ensures reliable motor operation, electrical isolation, and protection of the microcontroller from back EMF generated by the motors.

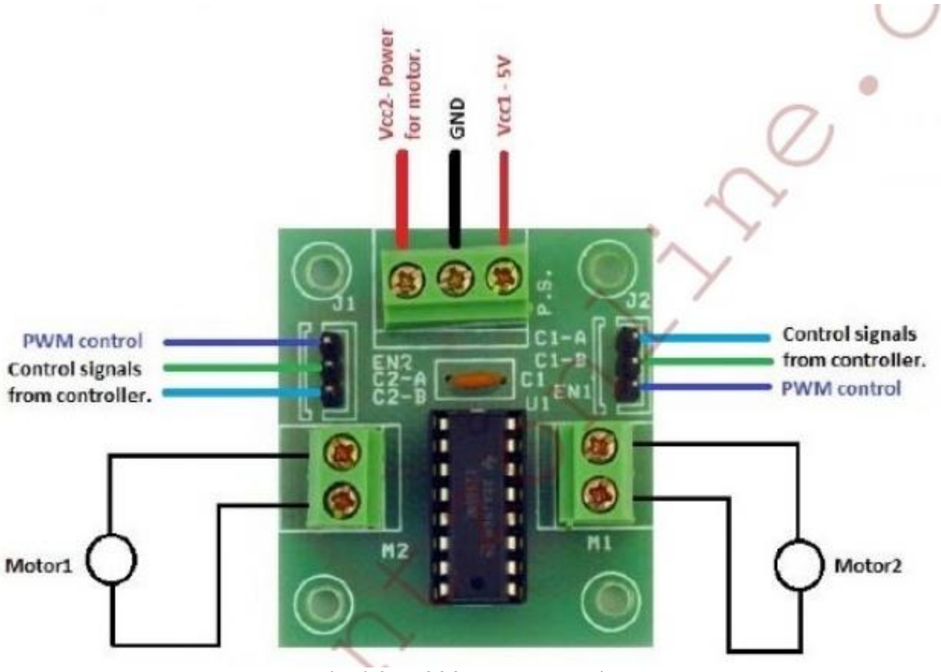

Fig 4.9: L293D Motor Driver

Working of the L293D Motor Driver Board

- Direction of motor rotation is controlled using logic inputs A and B
- Input combination ‘10’ rotates the motor in one direction
- Input combination ‘01’ rotates the motor in the opposite direction
- EN1 and EN2 must be enabled (logic HIGH) for motor operation
- PWM signals applied to EN pins control motor speed
- Vcc2 supplies power directly to the motors
- Common ground ensures stable and reliable operation

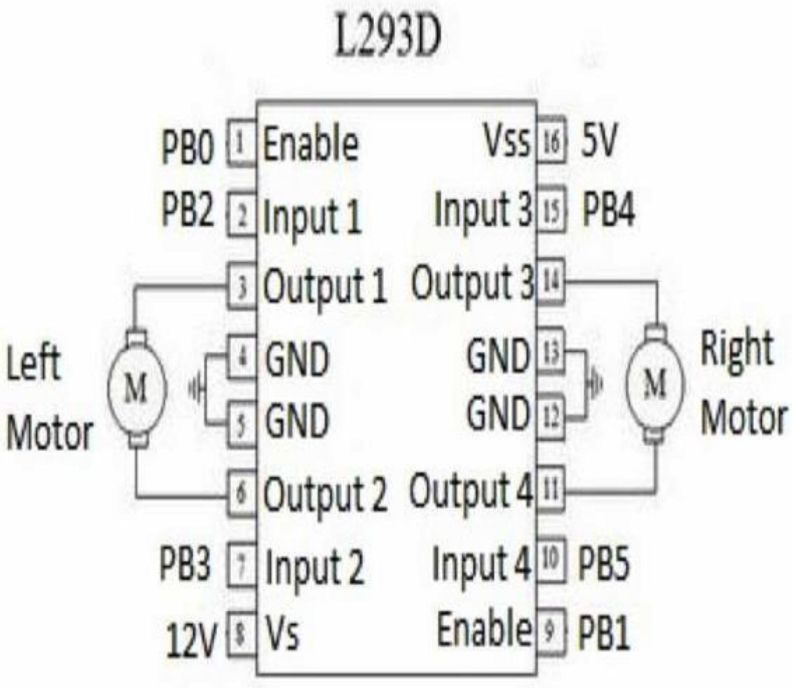

Fig 4.10: Pin Connection of L293D

L293D Motor Driver Board – Pin Description

- **EN1, EN2** – Enable pins for Motor 1 and Motor 2; used for speed control via PWM
- **A, B (Input Pins)** – Direction control signals from the controller
- **OUT1, OUT2** – Output terminals connected to Motor 1
- **OUT3, OUT4** – Output terminals connected to Motor 2
- **VCC1 (5 V)** – Logic supply voltage for the driver IC
- **VCC2** – Motor power supply (e.g., 6 V–12 V depending on motor rating)
- **GND** – Common ground for controller, motor driver, and power supply

4.1.8 XL4015 DC to DC Buck Converter

The XL4015 DC to DC buck converter is used in the smart cleaning robot to efficiently step down a higher DC input voltage to a stable lower output voltage required by the system components. It operates on the buck (step-down) conversion principle, providing high efficiency and reliable voltage regulation. The converter ensures safe and consistent power delivery to sensitive electronic modules such as the Arduino UNO, ESP32, sensors, and motor driver circuits, as shown in Fig 4.11.

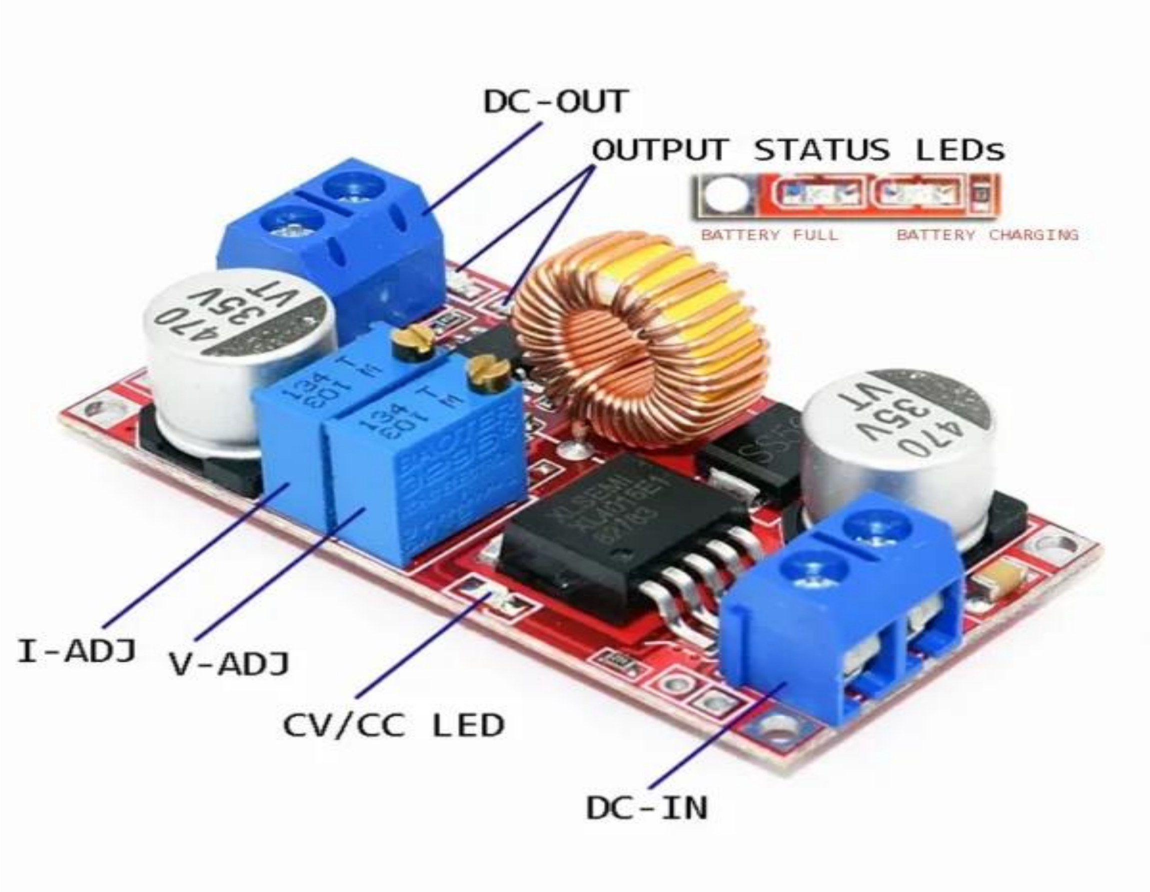

Fig 4.11: XL4015 DC to DC Buck Converter Module Used for Voltage Regulation

In this project, the XL4015 buck converter receives power from the main battery supply (12V) and regulates it to required voltage levels (such as 5V) using an adjustable onboard potentiometer. This prevents overvoltage damage and reduces power loss compared to linear regulators. The high current handling capability of the XL4015 makes it suitable for robotic applications where multiple modules operate simultaneously.

Characteristics

- DC–DC buck (step-down) converter
- Input voltage range: 4 V – 38 V
- Adjustable output voltage: 1.25 V – 36 V
- Maximum output current: 5 A
- High efficiency (up to 95%)
- Low heat dissipation
- Built-in thermal and short-circuit protection
- Compact and robust design

4.1.9 LM2596 DC to DC Buck Converter

The LM2596 DC to DC buck converter is employed in the smart cleaning robot to efficiently convert a higher DC input voltage into a stable lower output voltage suitable for low-power electronic components. It operates using a high-frequency switching regulator, offering improved efficiency compared to linear voltage regulators. The LM2596 module plays a critical role in maintaining voltage stability for microcontrollers and sensors, as illustrated in Fig 4.12.

Fig 4.12: LM2596 DC to DC Buck Converter Module Used for Voltage Regulation

In this project, the LM2596 receives power from the main battery source and steps it down to a regulated voltage level, typically 5 V, required by modules such as the Arduino UNO, ESP32, and sensor circuits. An onboard multi-turn potentiometer allows precise adjustment of the output voltage, ensuring safe operation and protection against overvoltage conditions.

Pin Description / Terminals

- **VIN+** – Positive input voltage terminal
- **VIN−** – Input ground terminal
- **VOUT+** – Regulated output voltage
- **VOUT−** – Output ground terminal
- **Adjust Potentiometer** – Used to fine-tune output voltage

Characteristics

- Buck (step-down) DC–DC converter
- Input voltage range: 4 V – 40 V
- Adjustable output voltage: 1.23 V – 37 V
- Maximum output current: 3 A
- Conversion efficiency up to 92%
- Integrated thermal shutdown protection
- Short-circuit and over-current protection
- Compact and lightweight module

4.1.10 DC Motors

The Fig 4.13 shows the DC motor. Motor is an electromechanical device that converts direct current (DC) electrical energy into mechanical rotational motion. It works on the principle that a current-carrying conductor placed in a magnetic field experiences a force, which causes the motor shaft to rotate.

DC motors are widely used in robotics and automation projects because of their simple control, high efficiency, and reliable performance. The speed and direction of a DC motor can be controlled by varying the input voltage and changing the polarity of the supply.

In this project, DC motors are used to move the smart trolley forward, backward, and to turn left or right.

Fig 4.13: DC Motor

Advantages:

- Simple construction
- Easy speed and direction control
- High torque at low speed
- Suitable for battery-powered systems

4.1.11 BO Motor

The BO motor is a compact DC geared motor widely used in robotic applications due to its lightweight structure, low power consumption, and ease of control. In the smart cleaning robot, BO motors are employed to drive the wheels, enabling smooth forward, backward, and turning movements. The integrated gearbox reduces speed while increasing torque, making the motor suitable for carrying the robot load efficiently and sufficiently, as shown in Fig 4.14.

Fig 4.14: BO Motor

Characteristics

- Type: DC Geared BO Motor
- Operating voltage: 6 V – 12 V
- Rated speed: 100–300 RPM (model dependent)
- High torque with low power consumption

- Lightweight plastic gearbox
- Shaft type: Single side shaft
- Compact size suitable for small robots
- Smooth and low-noise operation

4.1.12 Water Pump Motor

The water pump motor is used in the smart cleaning robot to dispense water or cleaning solution onto the floor during the cleaning process. It plays a crucial role in wet cleaning by supplying a controlled amount of liquid to the cleaning area. The pump is activated only when required, thereby reducing water wastage and improving cleaning efficiency, as illustrated in Fig 4.15.

Fig 4.15: Water Pump Motor

Characteristics

- Type: Mini DC Water Pump Motor
- Operating voltage: 6 V – 12 V DC
- Low current consumption
- Compact and lightweight design
- Continuous and stable water flow
- Suitable for small-scale robotic applications

4.1.13 12V 7ah Lead Acid Battery

The lead acid battery is used as the primary power source for the smart cleaning robot. It supplies electrical energy to the motors, control units, sensors, and power regulation modules. Due to its high current delivery capability, reliability, and low cost, the lead acid battery is well suited for robotic applications and requiring sustained power, as shown in Fig 4.16.

Fig 4.16: 12V 7ah Lead Acid Battery

Characteristics

- Type: Sealed Lead Acid (SLA) Battery
- Rated voltage: 12 V DC
- Capacity: 7 Ah

Table 4.2: Specifications of Lead Acid Battery Used in the Project

Parameter	**Specification**
Battery Type	Sealed Lead Acid (SLA)
Nominal Voltage	12 V DC
Capacity	7 Ah
Energy Capacity	84 Wh
Chemistry	Lead–Acid
Rechargeable	Yes
Maximum Discharge Current	High (motor suitable)
Operating Temperature	−20 °C to 50 °C
Charging Method	Constant voltage charging
Typical Charging Voltage	13.6 V – 14.4 V
Weight	~2.0 – 2.5 kg
Application	Robotic systems, backup power
Used In Project For	Motors, pump, controllers, sensors

4.2 Software Description

Software required for the proposed system is as listed below:

- **C++**
- **HTML (HyperText Markup Language)**
- **CSS (Cascading Style Sheets)**
- **JS (JavaScript)**

4.2.1 C++

C++ is a powerful, high-performance, general-purpose programming language widely used in system-level, embedded, and real-time applications. It supports both procedural and object-oriented programming paradigms, enabling developers to design modular, reusable, and efficient code. C++ provides direct access to hardware resources, allowing precise control over memory and execution speed, which makes it highly suitable for microcontroller-based and robotic systems. Its strong type checking, rich set of operators, and extensive standard library enhance program reliability and performance.

C++ was developed by Bjarne Stroustrup in the early 1980s as an extension of the C programming language, originally named “C with Classes.” The language was officially renamed C++ in 1983. Over time, C++ evolved through standardized versions such as C++98, C++11, C++14, C++17, and C++20, introducing modern features like improved memory management, multithreading support, and enhanced performance optimization. Due to its efficiency, flexibility, and close-to-hardware capabilities, C++ is extensively used in embedded systems, robotics, control systems, and real-time applications, making it an ideal choice for microcontroller programming in this project.

C++ supports methods (member functions) through its object-oriented programming paradigm. Methods are functions defined within a class and are used to operate on the data members of that class. They enable modular code design, abstraction, and reusability. In embedded systems, methods are commonly used to define hardware control logic such as motor movement, sensor reading, and communication handling. C++ also supports function overloading, inline functions, and templates, allowing developers to write efficient and flexible code. These features make C++ highly suitable for developing structured, maintainable, and performance-critical embedded applications

4.2.2 Arduino IDE

The Arduino Integrated Development Environment (IDE) features a comprehensive suite including a text editor for code writing, message area, text console, and toolbar with common function buttons and menus. It facilitates connectivity with Arduino hardware for program uploading and communication. Supporting C and C++ languages with specialized code structuring, it leverages a software library from the Wiring project for input/output procedures. User-written code typically comprises two essential functions for sketch initiation and main program loop as shown in fig. These are compiled and linked with a program stub main() into an executable cyclic executive program using the GNU tool chain bundled with the IDE. The cross-platform application, written in Java, caters to Microsoft Windows, macOS, and Linux. Originating from IDEs for Processing and Wiring languages, it boasts a code editor with features like text manipulation, syntax highlighting, and compilation/uploading mechanisms for Arduino boards, ensuring a user-friendly development environment.

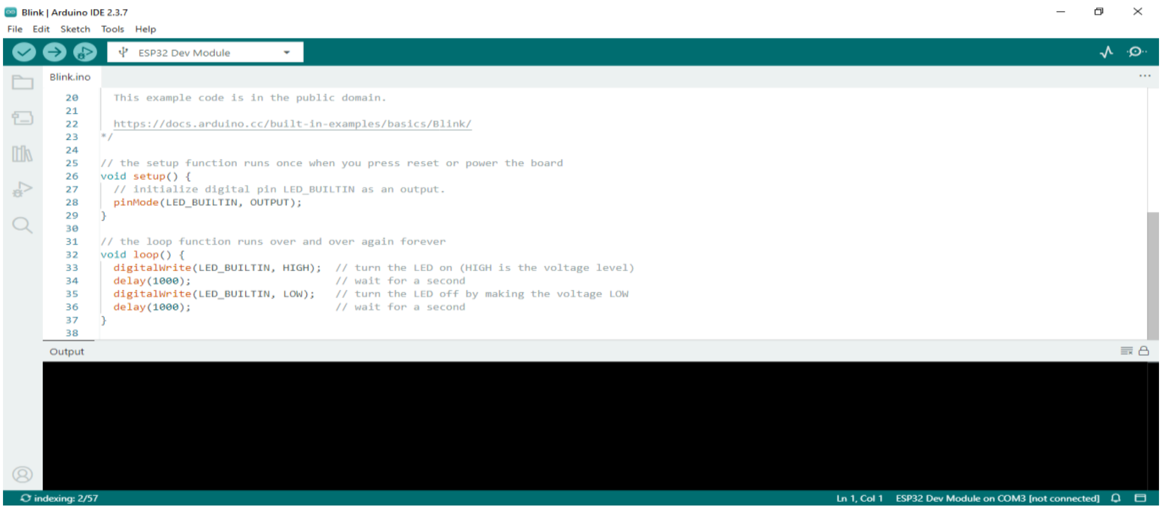

Fig 4.17: Arduino IDE Software

4.2.3 HTML (HyperText Markup Language)

HTML is the standard markup language used for structuring content on the World Wide Web. It defines the layout and organization of web pages using elements such as headings, paragraphs, images, links, forms, and tables. HTML follows a tag-based structure that allows browsers to interpret and render web content accurately. It forms the backbone of web applications by providing a semantic structure that improves readability and accessibility. HTML is platform-independent and works seamlessly across different browsers and devices.

HTML was introduced by Tim Berners-Lee in 1991 as part of the early development of the World Wide Web. Over time, HTML evolved through multiple versions, with HTML5 emerging as a major standard that supports multimedia elements, graphics, and improved form handling without the need for external plugins. In this project, HTML is used to design the structure of the web-based control interface that allows users to monitor and control the smart cleaning robot.

4.2.4 CSS (Cascading Style Sheets)

CSS is a style sheet language used to control the presentation and visual appearance of HTML documents. It enables developers to apply consistent styling such as colours, fonts, layouts, spacing, and animations across web pages. CSS separates content from design, improving maintainability and flexibility. It supports responsive design techniques, allowing web interfaces to adapt to different screen sizes and devices.

CSS was developed by the World Wide Web Consortium (W3C) and first released in 1996. With the evolution of CSS3, advanced features such as transitions, transformations, grid layouts, and flexbox were introduced, significantly enhancing modern web design capabilities. In this project, CSS is used to create a clean, user-friendly, and responsive web interface for controlling and monitoring the robot through a browser-based server hosted on the ESP32.

4.2.5 JS (JavaScript)

JavaScript is a high-level, dynamic scripting language primarily used to add interactivity and logic to web pages. It enables real-time updates, event handling, data validation, and communication between the user interface and backend systems. JavaScript supports multiple programming paradigms, including procedural, object-oriented, and event-driven programming, making it highly flexible for web applications.

JavaScript was developed by Brendan Eich in 1995 and later standardized as ECMAScript. Modern versions of JavaScript support asynchronous programming, APIs, and real-time communication technologies. In this project, JavaScript is used to handle user inputs from the web interface and transmit control commands to the ESP32 web server, enabling real-time communication between the user and the smart cleaning robot.

CHAPTER 5

IMPLEMENTATION AND RESULT

5.1 Implementation

The Cleaning Robot design consists of the following components. Each component was allocated to different team member to research and submit the required design criteria.

- Cleaning Robot mechanical design and web interface
- Connection to the control panel using ESP32 web server
- Serial communication between ESP32 and Arduino UNO
- Sensor integration for obstacle detection and navigation
- Motor control and cleaning mechanism operation
- Power supply, voltage regulation, and battery management
- System testing, integration, and performance evaluation

5.1.1 Cleaning Robot mechanical design and web interface

The mechanical design of the smart cleaning robot emphasizes compactness, stability, and effective cleaning operation. The chassis is designed to securely mount all hardware components such as the Arduino UNO, ESP32, motor driver shield, sensors, battery, and cleaning unit. Proper weight distribution is maintained to ensure smooth movement and balance during operation. BO motors are fixed symmetrically to enable controlled navigation, while the water pump and cleaning mechanism are positioned to provide uniform wet cleaning coverage across the floor surface.

The web interface is implemented using HTML, CSS, and JavaScript and is hosted on the ESP32 microcontroller functioning as a web server. This interface allows the user to control the robot through a web browser without the need for a mobile application. User commands are transmitted wirelessly to the ESP32 and then forwarded to the Arduino UNO via UART communication for real-time execution.

- Compact and lightweight chassis design for easy manoeuvrability Stable mounting of motors, sensors, and cleaning components
- User-friendly interface developed using HTML, CSS, and JavaScript
- Real-time command execution via UART communication between ESP32 and Arduino

5.1.2 Connection to the control panel using ESP32 web server

The ESP32 microcontroller operates as a web server to provide wireless connectivity between the user and the smart cleaning robot. It creates a local network interface that allows users to access the control panel through a standard web browser using a Wi-Fi connection. This eliminates the need for a dedicated mobile application and ensures platform-independent control from any device such as a laptop, tablet, or smartphone.

User commands such as movement control, cleaning activation, and system stop are issued through the web interface and are processed in real time by the ESP32. These commands are then transmitted to the Arduino UNO via UART serial communication for execution. The ESP32 handles high-level communication and user interaction, while the Arduino manages low-level motor and sensor operations. This division of tasks improves system reliability, responsiveness, and modularity.

- ESP32 hosts an embedded web server for robot control
- Enables wireless, browser-based user interaction
- Supports real-time command transmission
- Eliminates dependency on mobile applications
- Communicates with Arduino UNO via UART
- Improves system scalability and ease of use

5.1.3 Serial communication between ESP32 and Arduino UNO

Serial communication is used to establish reliable data transfer between the ESP32 and the Arduino UNO in the smart cleaning robot system. The ESP32 receives user commands from the web-based control panel and forwards these commands to the Arduino UNO using UART (Universal Asynchronous Receiver/Transmitter) serial communication. This communication method ensures real-time command delivery with minimal latency.

The ESP32 handles high-level operations such as Wi-Fi connectivity and web server management, while the Arduino UNO is responsible for low-level control tasks including motor driving, sensor interfacing, and cleaning mechanism operation. Data transmission occurs through the ESP32 TX pin and Arduino UNO RX pin, and vice versa, with a common ground connection to ensure signal integrity. A suitable baud rate is configured on both microcontrollers to maintain synchronized communication.

5.1.4 Sensor integration for obstacle detection and navigation

Sensor integration plays a vital role in enabling obstacle detection and safe navigation of the smart cleaning robot. Ultrasonic and infrared (IR) sensors are interfaced with the Arduino UNO to continuously monitor the robot's surroundings. The ultrasonic sensor measures the distance to obstacles by emitting ultrasonic waves and calculating the time taken for the echo to return, allowing accurate distance estimation. The IR sensor provides short-range detection, which helps identify nearby obstacles and surface variations.

The sensor data is processed by the Arduino UNO in real time to determine the presence and proximity of obstacles. When an obstacle is detected within a predefined threshold distance, the control logic initiates appropriate actions such as stopping the robot, changing direction, or slowing down the motors. This ensures collision avoidance and smooth navigation in indoor environments. The integration of multiple sensors enhances reliability, improves detection accuracy, and ensures safe autonomous movement of the robot during cleaning operations.

5.1.5 Motor Control and Cleaning Mechanism Operation

Motor control is a critical function of the smart cleaning robot, enabling precise movement and effective cleaning performance. The ESP32 controls the BO motors through the L293D motor driver, which acts as an interface between the low-power control signals and the high-current motors. Pulse Width Modulation (PWM) signals generated by the Arduino are used to regulate motor speed, allowing smooth forward, backward, and turning motions.

The cleaning mechanism consists of a water pump motor that supplies water for wet cleaning operations. The pump is activated and controlled by the user commands received via the ESP32. Proper synchronization between the movement motors and the cleaning mechanism ensures uniform floor coverage. This coordinated motor control approach enhances cleaning efficiency, reduces power consumption, and ensures stable robot operation during continuous use.

5.1.6 Power supply, voltage regulation, and battery management

The power supply system is designed to ensure stable and reliable operation of the smart cleaning robot. A sealed lead-acid battery is used as the primary power source, providing sufficient capacity to support motors, sensors, and control electronics. Since different components require different operating voltages, DC–DC buck converters are employed for voltage regulation. The XL4015 and LM2596 buck converters step down the battery voltage to suitable levels required by the Arduino UNO, ESP32, sensors, and motor driver. Proper battery management is implemented to prevent voltage fluctuations, overloading, and unexpected system shutdowns. Regulated power delivery ensures consistent motor performance and reliable sensor operation during continuous cleaning. This power management approach enhances system efficiency, extends battery life, and protects sensitive electronic components from voltage-related damage.

5.1.7 System testing, integration, and performance evaluation

System testing and integration were carried out to ensure proper coordination between all hardware and software modules of the smart cleaning robot. Individual components such as the ESP32 web server, Arduino UNO control logic, sensors, motor driver, and power supply were initially tested independently to verify correct functionality. After successful unit testing, all modules were integrated to form a complete system.

Performance evaluation was conducted under controlled indoor conditions to assess navigation accuracy, obstacle detection reliability, cleaning efficiency, and response time to user commands. The robot demonstrated stable movement, timely obstacle avoidance, and consistent cleaning operation. Communication latency between the ESP32 and Arduino UNO was minimal, confirming reliable serial data transfer. The evaluation results indicate that the system meets the design objectives and performs efficiently for small-scale autonomous cleaning applications.

5.2 Results

To design and develop a robot to navigate autonomously without human control

The objective of this project is to design and develop a robotic system capable of navigating autonomously without human intervention. The robot employs embedded microcontrollers, motor drivers, and multiple sensors to perceive its surroundings and make real-time navigation decisions. Obstacle detection sensors continuously monitor the environment, enabling the robot to avoid collisions and adjust its path dynamically. Autonomous navigation improves operational efficiency, reduces the need for manual control, and ensures consistent cleaning performance as shown in Fig 5.1. This approach allows the robot to operate safely and reliably in indoor environments while performing cleaning tasks with minimal user supervision.

Fig 5.1: Image of Control without human intervention

To detect obstacles using ultrasonic and IR sensors:

Obstacle detection is a critical function in autonomous cleaning robots to ensure safe navigation and prevent collisions. Ultrasonic sensors are widely used to measure the distance between the robot and surrounding objects by transmitting high-frequency sound waves and calculating the echo return time. This enables accurate detection of obstacles such as walls, furniture, and large objects. In addition, infrared (IR) sensors are employed for short-range detection and edge sensing, particularly for identifying nearby obstacles and drop-offs as shown in Fig. 5.2. The combined use of ultrasonic and IR sensors improves detection reliability, enhances navigation accuracy, and supports smooth autonomous movement in indoor environments.

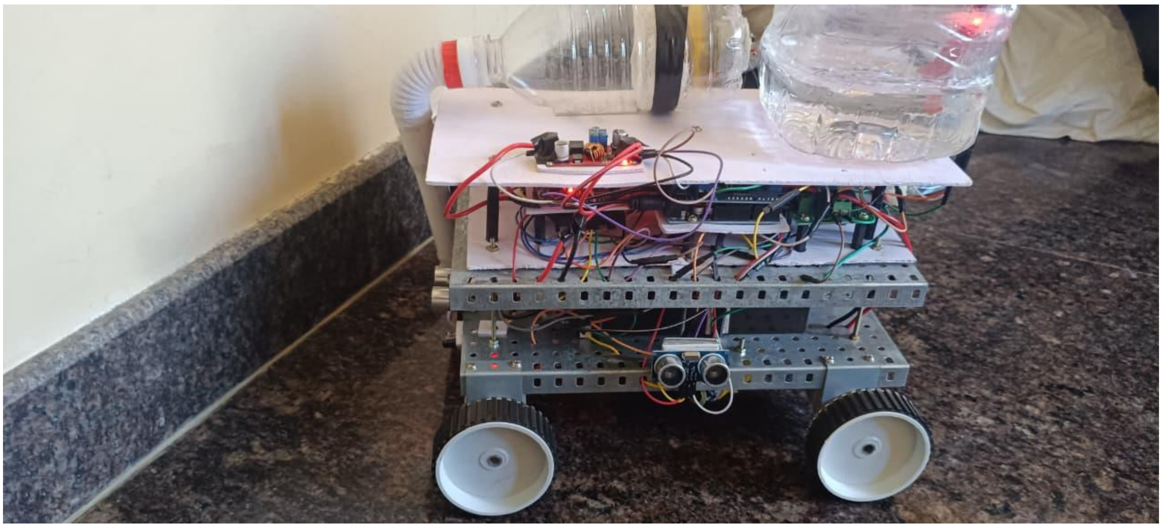

Fig 5.2: Shows the object detection using sensors

To integrate a vacuum motor and filter to effectively collect visible dust and small debris:

The integration of a vacuum motor and filtration system plays a vital role in enhancing the cleaning efficiency of the smart cleaning robot. The vacuum motor generates sufficient suction force to lift visible dust, fine particles, and small debris from the floor surface, while the filter traps these particles to prevent their re-circulation into the environment. This combination ensures improved hygiene and effective dust collection during operation. Similar vacuum-based cleaning mechanisms have been successfully implemented in autonomous cleaning robots to enhance indoor cleanliness and air quality. The working of the vacuum and filtration unit is illustrated in Fig 5.3

Fig 5.3: Working of vacuum and collecting visible dust

To integrate a cleaning mop to clean the surface more effectively:

The integration of a cleaning mop enhances the effectiveness of the autonomous cleaning robot by enabling wet cleaning in addition to dry dust removal. The mop mechanism helps remove stains, fine dust, and adhered particles that cannot be eliminated through vacuum cleaning alone. By maintaining controlled contact with the floor surface, the mop ensures uniform cleaning coverage and improved hygiene. The mop is actuated using a motor-driven mechanism, allowing consistent operation during robot movement. Similar mop-based cleaning mechanisms have been successfully implemented in autonomous floor cleaning robots to improve overall cleaning efficiency. The mop integration and operation are illustrated in Fig 5.4.

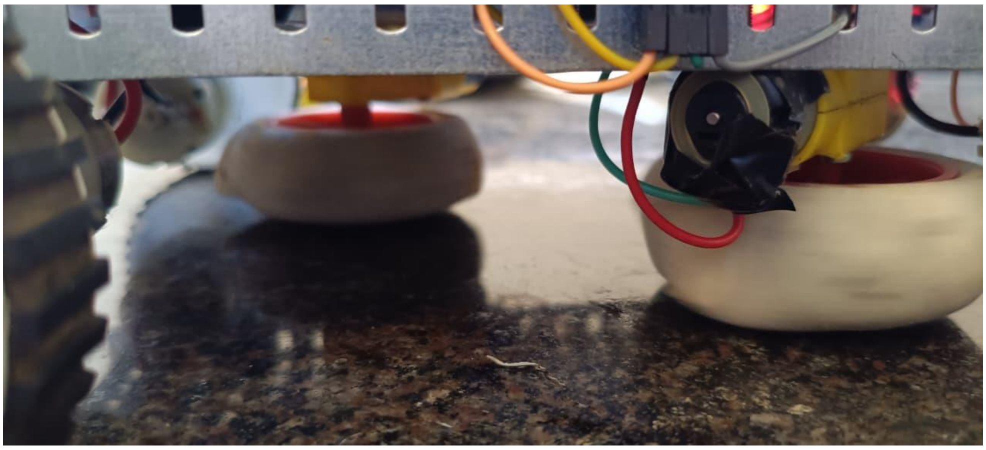

Fig 5.4: Integrated a cleaning mop to clean the surface

To evaluate the performance of the autonomous cleaning robot using suitable performance parameters:

Evaluating the performance of an autonomous cleaning robot is essential to assess its effectiveness and reliability in real-world environments. Key performance parameters such as cleaning efficiency, obstacle avoidance accuracy, response time, battery endurance, and system stability are considered during testing. These parameters help in analyzing how effectively the robot navigates, collects dust, and operates continuously without human intervention. Performance evaluation provides measurable insights into system limitations and areas for improvement. Similar evaluation methodologies have been adopted in autonomous cleaning robot studies to validate system functionality and efficiency. The overall performance evaluation setup is illustrated in Fig 5.5.

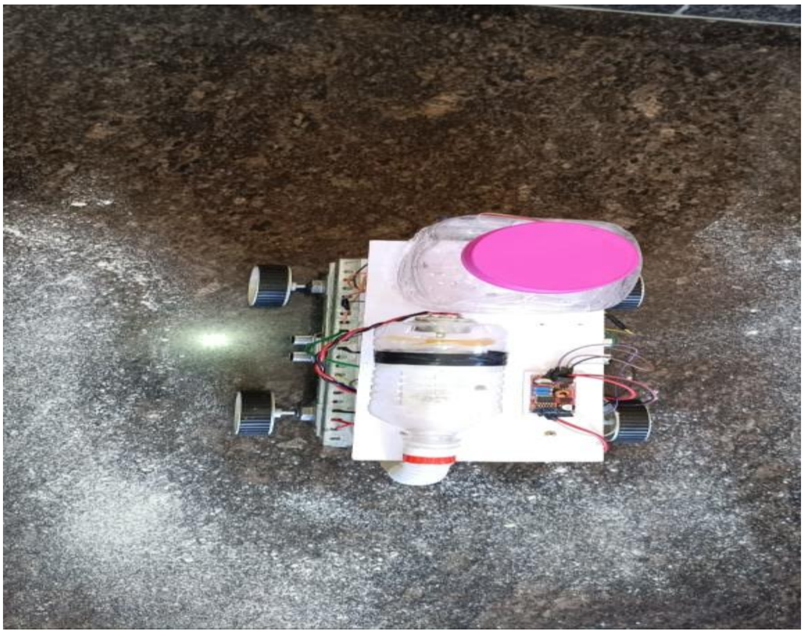

Fig 5.5: Shows the performance of autonomous cleaning robot

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

This project successfully demonstrates the design and implementation of a smart cleaning robot with web-based control and efficient hardware integration. The system combines the strengths of two microcontrollers to achieve reliable operation and modular control. The Arduino UNO is dedicated exclusively to controlling the movement motors, ensuring precise navigation, stable speed regulation, and effective obstacle avoidance through sensor feedback. By assigning only locomotion-related tasks to the Arduino, real-time motor response and control accuracy are significantly improved.

The ESP32 plays a vital role in handling high-level operations. It hosts the web server that enables browser-based user interaction and wirelessly receives control commands. In addition to communication, the ESP32 controls the remaining L293D motor drivers responsible for operating the cleaning mechanisms such as the vacuum motor, mop roller, and water pump. This clear division of responsibilities reduces processing load on the Arduino and enhances overall system reliability.

Sensor integration using ultrasonic, IR, and water level sensors ensures safe navigation and efficient cleaning by preventing collisions and dry running of the pump. Power regulation through DC–DC buck converters provide stable voltage levels, protecting sensitive components and supporting continuous operation. System testing confirmed smooth coordination between the ESP32 and Arduino UNO via serial communication, with minimal latency and consistent performance.

Overall, the proposed smart cleaning robot offers a cost-effective, scalable, and user-friendly solution for indoor cleaning applications. The modular architecture allows future enhancements such as autonomous navigation, improved sensing, and intelligent scheduling, making the system suitable for real-world domestic and commercial environments.

6.2 Future Scope

This project presents the successful development of a smart cleaning robot with efficient hardware integration and web-based control. The system employs a modular control architecture in which the Arduino UNO is dedicated to controlling only the movement motors, ensuring accurate navigation, smooth speed control, and reliable obstacle avoidance. The ESP32 manages high-level operations by hosting a web server that enables browser-based user interaction and controls the remaining L293D motor drivers responsible for the cleaning mechanisms, including the vacuum, mop roller, and water pump.

The integration of ultrasonic, IR, and water level sensors enhances operational safety and cleaning efficiency by preventing collisions and protecting the pump from dry running. Stable power delivery using DC–DC buck converters ensure uninterrupted system performance. Experimental testing confirmed reliable communication between the ESP32 and Arduino UNO, with responsive command execution and consistent cleaning operation. Overall, the developed system offers a cost-effective, flexible, and scalable solution for indoor cleaning applications, with scope for future enhancements such as autonomous navigation and intelligent cleaning modes.

REFERENCES

[1] M. S. Saleh, Y. I. Al Mashhadany, M. Alshaibi, F. Majeed Ameen, and S. Algburi, "Optimal Mobile Robot Navigation for Obstacle Avoidance Based on ANFIS Controller," J. Robot. Control, vol. 6, no. 1, pp. 484–492, Feb. 2025

[2] S. Kumar, A. Smith, and P. Singh, "Autonomous Navigation and Real-Time Mapping Using Ultrasonic Sensors in a Humanoid Robot," Int. Conf. on Robotics and Intelligent Systems (ICRIS), 2024.

[3] N. Tuppathi and M. G. Nizampet, "Design and Fabrication of Obstacle Avoiding Robot Using Ultrasonic Sensor and ATmega328 Microcontroller," Int. J. Sci. Res. Dev., vol. 12, no. 7, pp. 46–50, 2024.

[4] Imran Mokashi, M. K. Delvi, A. S. Kalandar, E. M. Manvith Das, and Y. A. Rahman, "Automated Cleaning Robot with Smart Navigation and Control," J. Mechanical Robotics, vol. 10, no. 1, pp. 1–12, Mar. 2025.

[5] A. Pandey, R. Singh, and M. Gupta, "Autonomous Mobile Robot for Obstacle Detection and Avoidance Using Ultrasonic and Infrared Sensors," in Proc. Int. Conf. on Intelligent Robotics and Automation (ICIRA), 2024, pp. 112–118.

[6] S. Kumar, P. Verma, and L. Sharma, "Obstacle Detection and Navigation System Based on Ultrasonic and IR Sensing for Mobile Robots," Int. J. Robotics and Mechatronics Engineering, vol. 12, no. 3, pp. 152–159, 2025.

[7] L. N. Prashanth, S. Reddy, and T. Subramanian, "Ultrasonic and Infrared Sensor Based Navigation for Smart Cleaning Robots," in Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2025, pp. 220–227.

[8] K. Sathish, A. Kumar, and S. Balakrishnan, "Autonomous Robotic Platform for Obstacle Detection and Avoidance Using Ultrasonic and IR Sensors," Journal of Embedded Systems and Robotics, vol. 5, no. 2, pp. 45–53, Apr. 2025

[9] T. C. Do and T. D. Dang, "Development and Evaluation of a Smart Vacuum Robot with Adaptive Navigation," Engineering, Technology & Applied Science Research, vol. 15, no. 6, pp. 28537–28543, Dec. 2025

[10] S. Ramdas and A. N. N. T, "Floor Bot-Floor Vacuum Cleaning Robot," *Int. J. Soft Computing and Engineering (IJSCE)*, vol. 14, no. 6, pp. 1–4, Jan. 2025

[11] W. A. Heitbrink and J. Santalla-Elias, "The Effect of Debris Accumulation Upon Air Flow and Filter Resistance to Air Flow for Four Commercially Available Vacuum Cleaners," Tech. Rep, The Center to Protect Workers' Rights, Silver Spring, MD, USA, May 2008.

[12] J. V. Fayzullayevich, G. Tan, F. J. Alex, Y. Wu, and P. K. Agyeman, "Numerical Study of Factors Affecting Particle Suction Efficiency of Pick-Up Head of a Regenerative Air Vacuum Sweeper," Processes, vol. 10, no. 7, article 1252, 2022,

[13] A. Kumar, V. S. Rathod, and P. Das, "Performance Evaluation of an Autonomous Cleaning Robot Using Cleaning Efficiency, Area Coverage, and Navigation Accuracy Metrics," Int. J. Robotics and Automation Research (IJRAR), vol. 8, no. 2, pp. 73–82, 2025.

[14] S. Park, J. H. Lee, and D. Y. Kim, "Performance Assessment Framework for Autonomous Vacuum Cleaning Robots Based on Coverage Ratio and Navigation Efficiency," in Proc. IEEE Int. Conf. on Robotics and Automation (ICRA), 2025, pp. 4552–4560.

[15] R. Singh, L. N. Patel, and A. K. Verma, "Analysis of Intelligent Floor–Cleaning Robot Performance: Suction Efficiency, Battery Endurance, and Surface Adaptability," Int. J. Service Robotics and Autonomous Systems, vol. 4, no. 1, pp. 29–38, Mar. 2025.

[16] L. Chen, Y. Zhao, and M. S. Ali, "Systematic Evaluation of Autonomous Cleaning Robots Using Coverage Efficiency, Localization Precision, and Reliability Metrics," *IEEE Trans. Automation Science and Engineering*, vol. 22, no. 1, pp. 105–116, Jan. 2025.