Project-B Report

On

IOT BASED AUTONOMOUS AIR QUALITY ANALYZER AND PURIFIER

Submitted in partial fulfillment of the requirement of
University of Mumbai for the Degree of

Bachelor of Technology

In

Electronics & Computer Science

Submitted By

Anusha Aherwar

Prerna Singh

Shreeya Ghegad

Ajay Zodge

Supervisor

Prof. Swati Patil

Department of Electronics & Computer Science

PILLAI COLLEGE OF ENGINEERING

New Panvel – 410206

UNIVERSITY OF MUMBAI

Academic Year 2025 - 26

DEPARTMENT OF ELECTRONICS & COMPUTER SCIENCE
Pillai College of Engineering
New Panvel – 410206

CERTIFICATE

This is to certify that the requirements for the B.Tech. Project B entitled '**IoT based Autonomous Air Quality Analyzer and Purifier**' have been successfully completed by the following students:

Name	**Roll No.**
Anusha Aherwar	A717
Prerna Singh	A741
Shreeya Ghegad	A749
Ajay Zodge	A701

in partial fulfillment of Bachelor of Engineering of Mumbai University in the Department of Electronics & Computer Science, Pillai College of Engineering, New Panvel – 410206 during the Academic Year 2025 – 2026.

Supervisor
(Prof. Swati Patil)

Project Coordinator
Prof. Ajit Saraf

Head of Department

Dr. Monika Bhagwat

Principal

Dr. Sandeep M. Joshi

DEPARTMENT OF ELECTRONICS & COMPUTER SCIENCE
Pillai College of Engineering
New Panvel – 410206

SYNOPSIS APPROVAL FOR B.TECH.

This project synopsis entitled "**IoT based Autonomous Air Quality Analyzer and Purifier**" by **Anusha Aherwar , Prerna Singh , Shreeya Ghegad and Ajay Zodge** are approved for the degree of B.Tech. in **the department of Electronics & Computer Science.**

.

Examiners:

1. ________________

2. ________________

Supervisors:

1. ________________

2. ________________

Chairman:

1. ________________

Date:

Place:

Declaration

We declare that this written submission for the B.Tech. The project entitled "**IoT based Autonomous Air Quality Analyzer and Purifier**" represents our ideas in our own words and wherever others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any ideas / data / fact / source in our submission. We understand that any violation of the above will cause disciplinary action by the institute and also evoke penal action from the sources which have thus not been properly cited or from whom paper permission has not been taken when needed.

Project Group Members:

Anusha Aherwar: ___________________________

Prerna Singh: ___________________________

Shreeya Ghegad: ___________________________

Ajay Zodge: ___________________________

Date:

Place:

INDEX

ABSTRACT

Internet of Things (IoT) is a network of interconnected devices that collect, transmit, and process real-time data without human intervention. Embedded with sensors, softwares and communication technologies, IoT enhances automation across various sectors, including healthcare, transportation,smart homes, and industrial operations. It incorporates real-time monitoring, automation, cost reduction, and data-driven decision-making, improving efficiency and resource optimization. IoT also enhances safety & security. This project proposes an automatic and mobile Air Purifier that integrates IoT and AI-driven automation to enhance air purification efficiency. Air pollution is a major environmental concern affecting human health, necessitating efficient indoor air purification systems. Traditional air purifiers are stationary, limiting their effectiveness in multi-room environments. Various robotic navigation techniques, such as Line Following, SLAM (Simultaneous Localization and Mapping), and LiDAR-based Mapping, exist for autonomous movement. However, these methods can be costly and computationally complex. To balance affordability and effectiveness, this project employs a Raspberry Pi-based system with sensor-guided movement and a mobile app for user control. The purifier autonomously navigates based on real-time air quality data and user commands, optimizing purification across multiple rooms.

This work highlights the integration of low-cost sensors, IoT communication, and motor control algorithms to create an intelligent, self-moving air purifier while ensuring budget-friendly Implementation.

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 SIGNIFICANCE

The goal of this project is to develop an IoT-based Automatic and Mobile Air Purifier that efficiently enhances indoor air quality through intelligent automation and mobility. The project aims to study various robotic navigation techniques to identify their limitations and propose a cost-effective, efficient movement strategy. It also focuses on the integration of IoT-based sensors for real-time air quality monitoring and autonomous movement control. Additionally, the project seeks to explore mobile app-based automation for convenient user control and remote operation of the purifier. A Raspberry Pi-based platform will be used to implement a sensor-guided movement system that enables budget-friendly self-navigation. Finally, the system's performance will be evaluated based on air purification efficiency, movement accuracy, and user interaction, ensuring a smart and user-friendly solution for improving indoor environments.

1.2 BACKGROUND

Air pollution has become one of the most pressing environmental challenges, significantly affecting human health and overall well-being. Indoor air, which people are exposed to for most of their day, often contains high levels of dust, smoke, carbon dioxide, and other harmful pollutants. Traditional air purifiers, though effective in filtering airborne contaminants, are typically stationary and lack the intelligence to adapt to varying air quality conditions across different areas of a room or building. With advancements in Internet of Things (IoT) and robotics, it has become possible to create smart systems capable of monitoring, analyzing, and improving indoor air quality autonomously.

The proposed IoT-based Automatic and Mobile Air Purifier addresses these limitations by combining air purification technology with mobility and intelligent control. Using IoT-enabled sensors, the system continuously monitors air quality parameters such as PM2.5, CO_2, temperature, and humidity in real time. Based on this data, the purifier autonomously navigates toward areas with poor air quality using a Raspberry Pi-based movement control system. Furthermore, the integration of a mobile app interface allows users to remotely monitor and control the purifier's operation, providing both convenience and efficiency.

1.3 SCOPE

This project has a wide scope as it not only focuses on improving air quality but also introduces automation and intelligent control through IoT integration. At present, the system monitors air quality in real time and purifies the air automatically while being mobile. In the future, the project can be made more advanced by integrating AI-based automation and cloud connectivity to predict pollution levels and optimize purifier movement. Additional features such as voice alerts or voice control commands can also be implemented, allowing users to interact with the purifier more conveniently. A mobile application can also be developed to display the real-time location and movement of the robot purifier, enabling users to track and control it remotely. Furthermore, the system can be connected to multiple user devices through the cloud so that air quality notifications and purifier status updates can be sent to several users simultaneously, rather than just one.

Chapter 2

LITERATURE REVIEW

2.1 HISTORY:

Air purification has evolved from simple mechanical filters to advanced intelligent systems. Early methods relied on natural ventilation and basic filters that trapped dust but were ineffective against fine pollutants and gases. With industrialization, electric purifiers using HEPA, activated carbon, and electrostatic technologies were developed. However, these were mostly stationary, manually operated, and lacked the ability to detect pollution levels or adjust automatically.

2.2 COMPARISON WITH EXISTING IMPLEMENTATIONS:

Arpita Choudhary et al. [1] introduced the "Design and Fabrication of an IoT-based Air Purifier using HEPA Filter." This paper presents a low-cost and effective IoT-enabled air purification system for indoor spaces. The setup uses the NodeMCU ESP8266 microcontroller, MQ135 gas sensors, a HEPA filter, and UV light for air disinfection. The purifier can monitor and control air quality in real time through a mobile app, allowing users to operate it remotely. The research shows that integrating IoT makes user interaction easier while providing reliable pollutant detection and purification feedback. This work lays the groundwork for developing portable and automated air purification systems that combine IoT, filtration, and data analysis.

Wen-Yang Wu and Yen-Chen Liu [2] presented "Autonomous Guided Robotic Systems in Regulating Indoor Environmental Quality." This study focuses on an intelligent robotic control framework that autonomously improves indoor air quality using distributed sensors and mobile robots. The system checks air parameters from multiple wireless sensors, identifies areas that need improvement, and directs robotic purifiers using the Adaptive Monte Carlo Localization (AMCL) and Probabilistic Roadmap (PRM) algorithms. This method shows that combining robotics with environmental sensing can effectively manage and optimize indoor air conditions. The research strongly supports the movement and automation principles used in the current project, where the purifier moves on its own based on air quality levels.

K.-H. Kim et al. [3] proposed an "IoT Air Purifier with Humidification Function Capable of Removing PM1.0 Ultra-Fine Dust." This design integrates IoT control with enhanced particulate filtration. It emphasizes combining air humidification and purification in one unit, showing how embedded systems and smart control can improve purification efficiency. This work includes multi-parameter sensing and adaptive control in IoT-based purifiers.

R. Mumtaz et al. [4] introduced “Internet of Things (IoT) Based Indoor Air Quality Sensing and Predictive Analytics — A COVID-19 Perspective.” They proposed a predictive system that can analyze indoor air quality variations using IoT sensors and data-driven analytics. The study highlights the need for real-time monitoring and automated response systems to prevent health risks from poor air quality. The integration of IoT networks and analytics closely relates to the data-driven alert system in the current project.

C. G. Kousalya et al. [5] designed an “IoT-Based Indoor Air Purifier” that combines air quality sensing with purification control through wireless communication. They used multiple sensors and Wi-Fi-enabled microcontrollers for efficient air monitoring, similar to the proposed system’s design that involves Raspberry Pi and gas sensors. It shows that real-time monitoring and actuator control can lead to effective closed-loop purification systems.

H. Dai and B. Zhao [6] explored “Reducing Airborne Infection Risk by Locating Air Cleaners at Proper Positions Indoor.” They emphasized the importance of positioning and air circulation in achieving optimal purification efficiency. The study concludes that the mobility of purifiers greatly enhances pollutant removal performance, supporting the idea of using a mobile platform in the proposed project.

The reviewed works show significant advancements in IoT-enabled air purification, air quality monitoring, and autonomous environmental control. The combination of wireless sensing, motion control, and real-time data visualization in these studies illustrates how IoT can effectively improve environmental awareness and automation. These advancements create a solid foundation for developing smart and responsive purification systems.

CHAPTER 3

SYSTEM REQUIREMENT AND ANALYSIS

3.1 SOFTWARE REQUIREMENTS

:.No	Description
1	Raspberry Pi OS
2	Python 3
3	Python Libraries
4	Blynk App (Mobile)

Table 3.1-Software Requirement

3.1.1 Detail Description of software

The primary software platform for this project is Raspberry Pi OS, which serves as the operating system for the Raspberry Pi microcontroller. Python 3 is used as the main programming language due to its simplicity, extensive libraries, and strong support for IoT applications.

Several Python libraries are required to implement the project's functionality:

- gpiozero or RPi.GPIO: These libraries are used to interface with GPIO pins on the Raspberry Pi, enabling control of motors for autonomous movement, and relays for switching the fan ON or OFF.
- blynklib (Blynk Python library): This library facilitates communication between the Raspberry Pi and the Blynk mobile application, allowing the system to send real-time AQI readings and receive manual ON/OFF commands from the user.
- time: Provides timing functions to create delays and control the frequency of sensor readings and fan activation.

The mobile application uses Blynk to create an easy-to-use interface. This interface shows real-time AQI readings and lets users control the purifier manually. Since the project emphasizes immediate monitoring instead of tracking historical data, there is no data storage or logging of past information.

The app also gives users the option to check the robot's current status, whether it is moving or stopped, and to interact with the system from a distance.

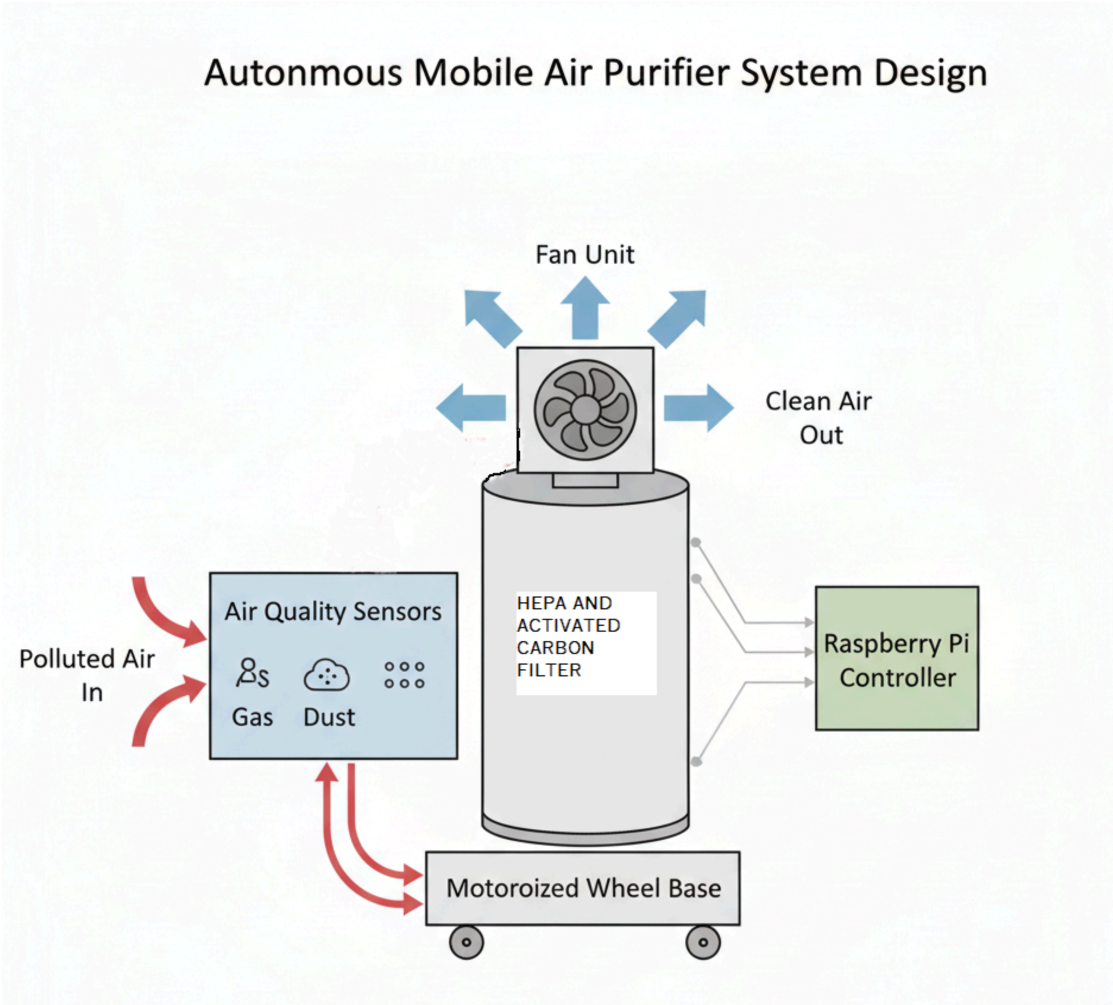

Fig. 3.1.1.System Design

3.2 HARDWARE REQUIREMENTS

:.No	Description	Qty.
1	Raspberry pi 4b	1
2	MQ-7	1
3	MQ-135	1
4	dht22	1
5	GP2Y1010F48	1
6	ADS-1115	1
7	Resistors	1
8	capacitors	2
9	Relay 4 channel	1
10	Ultrasonic sensor HC-SR04	2
11	IR sensor	2
12	L298N driver	1
13	HEPA filter	1
14	Activated Carbon filter	1
15	12V DC Fan	1
16	12V li-on Battery	1
17	DC-DC buck converter	1
18	Chassis	1

Table 3.2-Hardware Requirement

3.2.1 RASPBERRY PI 4B

The Raspberry Pi is a low-cost, high-performance single-board computer designed for a wide range of applications, from learning programming to embedded systems and IoT projects. It integrates a powerful ARM-based CPU, GPU, memory, and I/O interfaces on a compact board, providing a versatile platform for both education and prototyping. The Raspberry Pi supports standard Linux-based operating systems and can run a wide variety of software, making it ideal for hobbyists, students, and professionals alike. With HDMI, USB, GPIO, and network interfaces, it enables connectivity with sensors, motors, displays, and external devices, providing a flexible solution for many embedded and computing projects.

Features

- Broadcom BCM2711, Quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- 2GB / 4GB / 8GB LPDDR4-3200 SDRAM (depending on variant)
- Gigabit Ethernet port
- Dual-band 802.11ac Wi-Fi, Bluetooth 5.0, BLE
- 2 × USB 3.0 ports, 2 × USB 2.0 ports
- 2 × micro-HDMI ports supporting up to 4Kp60 video output
- VideoCore VI GPU supporting OpenGL ES 3.x, 4K H.265/H.264 video decode
- 40-pin GPIO header for digital interfacing
- CSI camera interface, DSI display interface
- USB-C power supply input, 5V DC, 3A
- MicroSD card slot for OS and storage
- Hardware support for SPI, I2C, UART, PWM
- Low-power operation with idle and sleep modes
- Compact form factor with robust community and accessory support

3.2.2 SENSORS

MQ-7 Sensor (Carbon Monoxide Sensor):

The MQ-7 sensor is designed to detect carbon monoxide (CO), a colorless and odorless toxic gas, in the air. It has a sensitive tin dioxide (SnO_2) layer whose electrical resistance changes in the presence of CO. The sensor requires a heating cycle to operate optimally and provides an analog voltage output proportional to the concentration of CO in the surrounding environment. This makes it suitable for applications like indoor air quality monitoring, safety alarms, and gas leak detection in homes and industries.

MQ-135 Sensor (Air Quality Sensor):

The MQ-135 is a versatile gas sensor capable of detecting multiple gases, including ammonia (NH_3), sulfur (SOx), benzene, smoke, and carbon dioxide (CO_2). It operates on a similar principle as other MQ series sensors, where the resistance of its sensing material changes depending on the concentration of surrounding gases. The sensor produces an analog voltage that can be read by microcontrollers to estimate air pollution levels or calculate an approximate Air Quality Index (AQI). It is widely used in indoor and outdoor air quality monitoring systems.

DHT22 Sensor (Temperature and Humidity Sensor):

The DHT22 is a digital sensor that provides precise measurements of ambient temperature and relative humidity. It has a capacitive humidity sensor and a thermistor to detect environmental changes, and it communicates with microcontrollers using a single-wire digital protocol. DHT22 can measure temperatures from –40°C to 80°C and humidity from 0% to 100%, making it ideal for weather stations, smart homes, and HVAC systems. Its reliability and digital output make it easy to integrate into IoT and environmental monitoring projects.

PM2.5 Sensor (Particulate Matter Sensor):

PM2.5 sensors are used to measure fine airborne particles with diameters less than 2.5 micrometers, which are small enough to penetrate the lungs and cause health problems. These sensors, like the SDS011 or PMS5003, typically use a laser scattering method to detect and count particles in the air. The output is usually given in micrograms per cubic meter ($\mu g/m^3$), which helps evaluate air pollution levels. PM2.5 sensors are critical components of smart air purifiers, pollution monitoring stations, and health-focused IoT systems.

3.2.3 L298N Motor Driver and DC Motors

The L298N is a dual H-bridge motor driver that controls the speed and direction of two DC motors. These motors are connected to mecanum wheels, allowing the purifier to move omnidirectionally.
Specifications:

- Operates with 5V–12V DC motors
- Dual H-bridge driver supports two motors simultaneously
- PWM control for speed regulation

The Raspberry Pi sends PWM signals to the L298N, controlling motor speed and direction to navigate the purifier autonomously toward areas with higher pollution levels.

3.2.4 Ultrasonic Sensors and IR sensors

Two ultrasonic sensors and two IR sensors are used for obstacle detection and navigation. Ultrasonic sensors measure distance by sending ultrasonic waves, while IR sensors detect proximity of nearby objects.These sensors allow the purifier to avoid collisions, map its surroundings, and navigate efficiently in indoor environments.

3.2.5 Power Supply

The system is powered by a 12V Li-ion rechargeable battery. A voltage regulator converts 12V to 5V for the Raspberry Pi and sensors and direct 12v is given to motor driver and fan.The power supply ensures uninterrupted operation of motors, sensors, and controller.

Components:

- 12V Li-ion Battery
- Voltage Regulator (5V)
- Connecting wires and safety fuses

Provides stable and regulated power to all components, enabling continuous autonomous operation of the purifier.

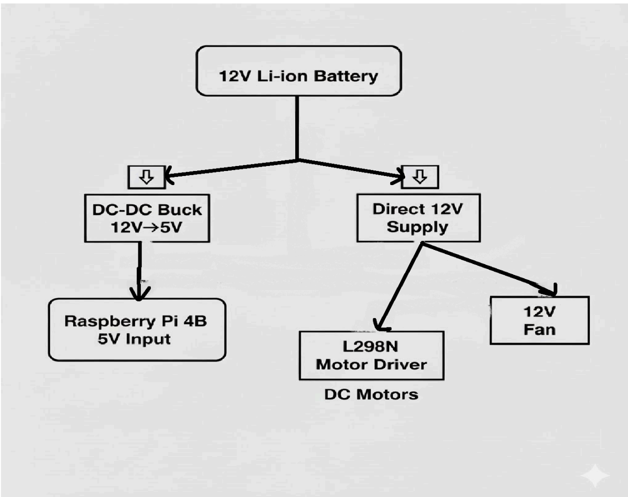

Fig. 3.2.1.Power Supply

CHAPTER 4

METHODOLOGY

4.1 BLOCK DIAGRAM

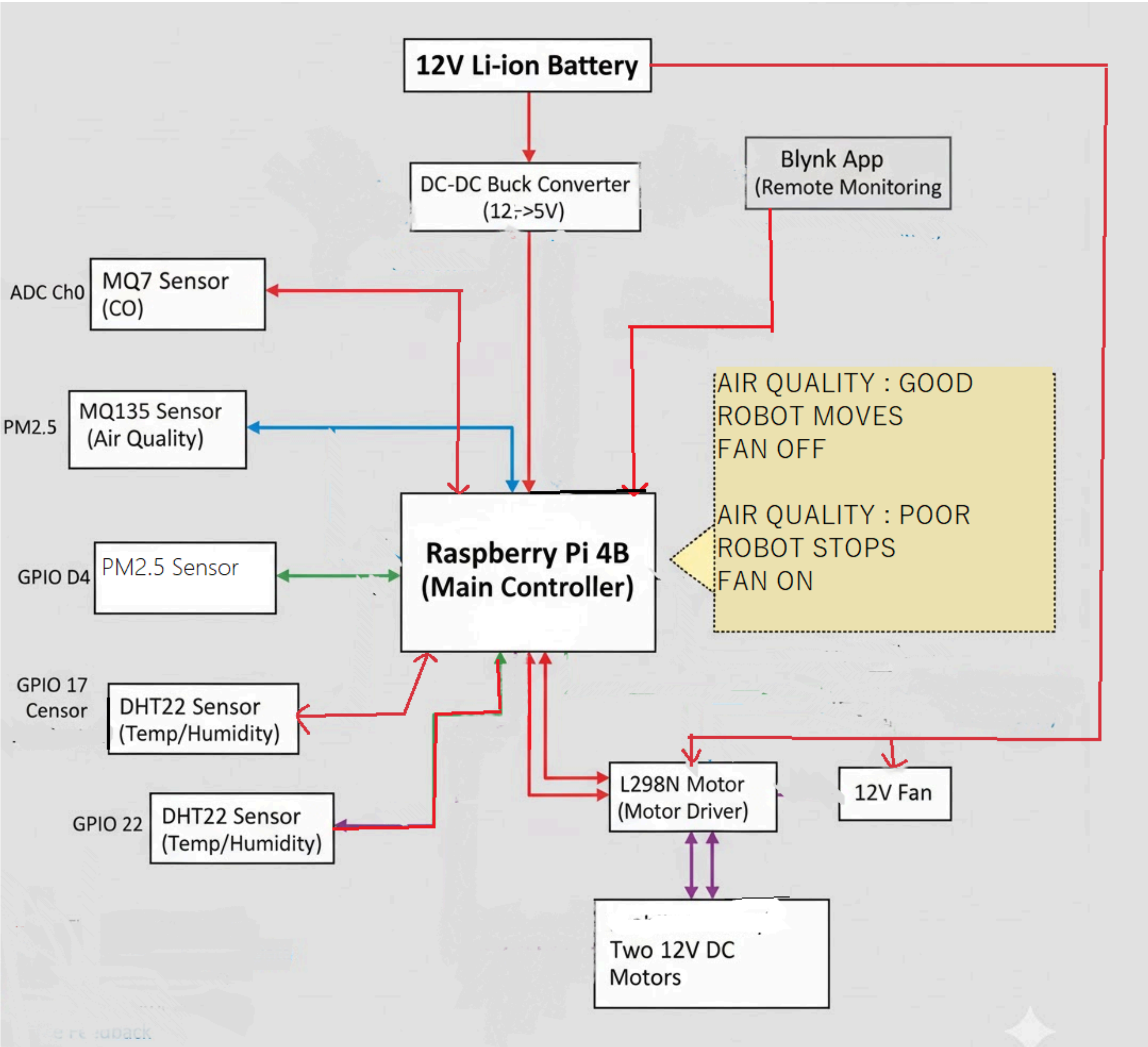

Fig 4.1.1: BLOCK DIAGRAM

4.2 WORKING

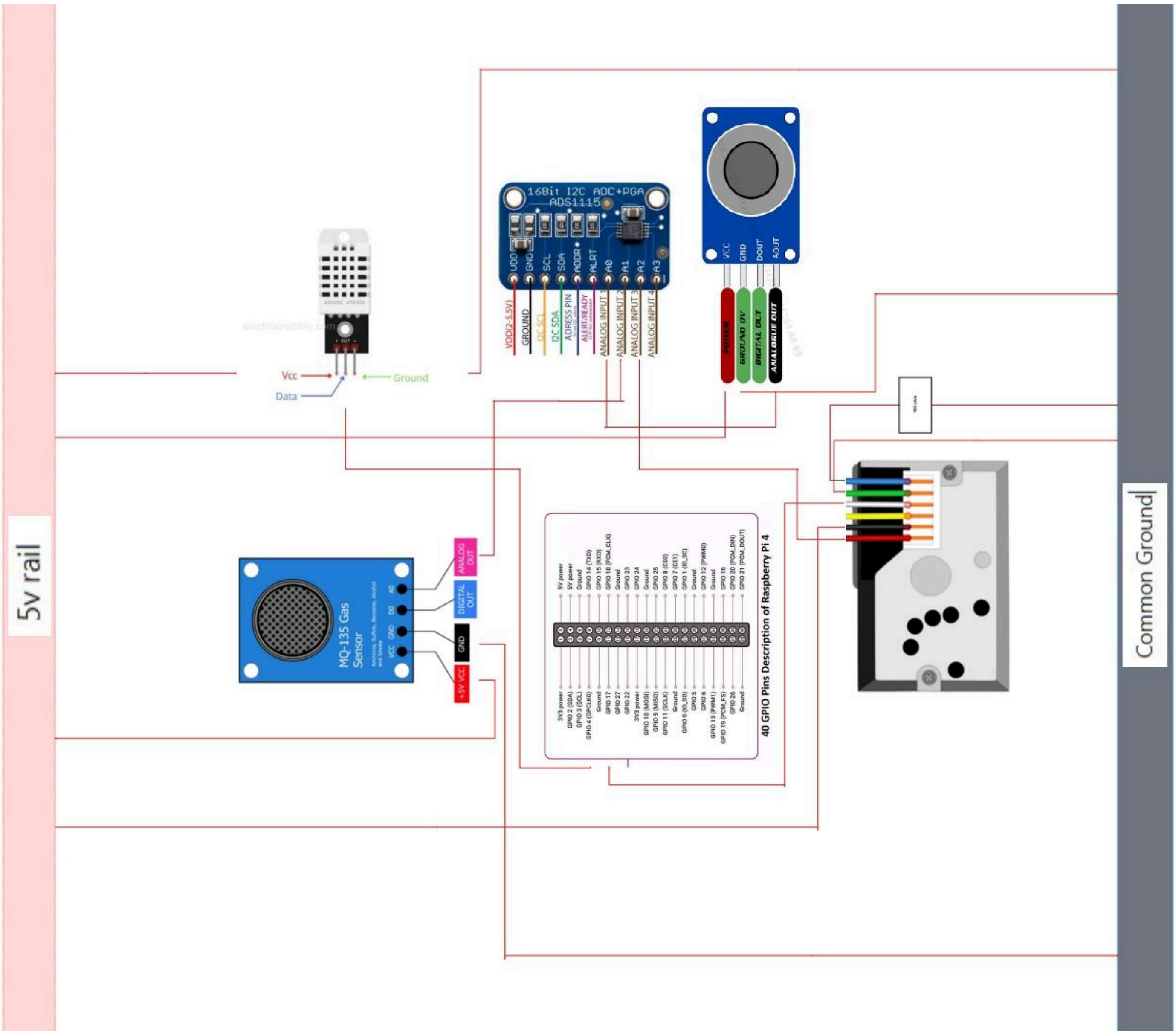

Fig 4.2.1 CIRCUIT DIAGRAM OF SENSORS

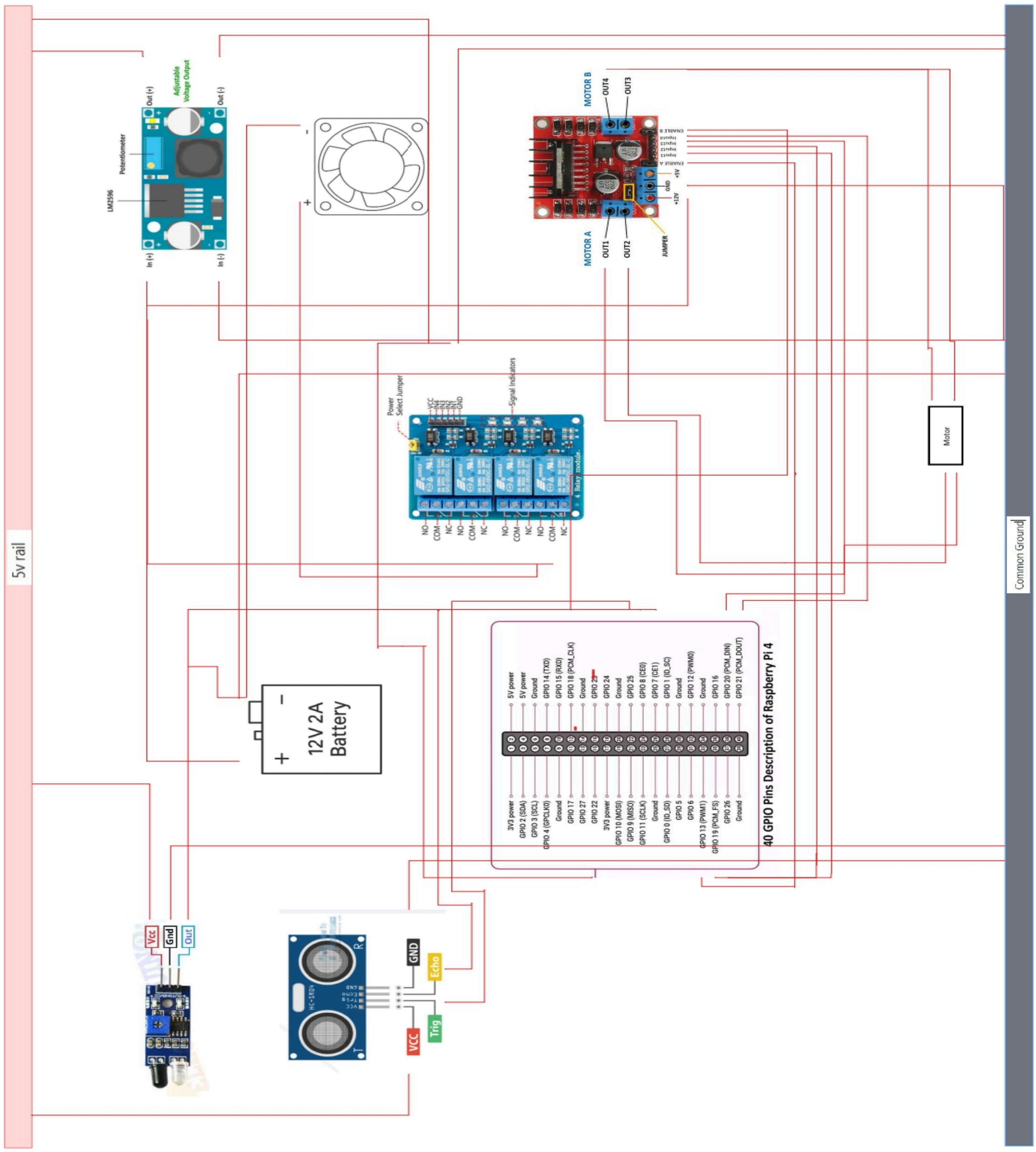

Fig 4.2.2 OBSTACLE AVOIDANCE CONTROL CIRCUIT DIAGRAM

When the system is powered on, the Raspberry Pi 4B initializes all its GPIO pins, the ADS1115 ADC module, and the DHT22 temperature and humidity sensor. To ensure accurate and stable readings, a warm-up period of approximately 30 seconds is maintained, during which the gas sensors stabilize their internal heating elements. Once initialization is complete, the system enters continuous monitoring mode, where all sensors operate in a sequential loop controlled by the Raspberry Pi.

The Pi first reads data from the DHT22 sensor due to its strict timing requirements, followed by the MQ7, MQ135, and PM2.5 sensors through the ADS1115 analog-to-digital converter. Each sensor provides critical environmental data used to evaluate air quality. The MQ7 detects carbon monoxide concentration, the MQ135 measures the presence of general air pollutants such as ammonia and benzene, and the PM2.5 sensor captures fine particulate matter suspended in the air. These analog readings are converted into digital values by the ADC and then compared against predefined threshold levels to determine the surrounding air quality condition.

The system uses the following reference thresholds for decision-making:

- **MQ7 (CO):** Warning ≥0.50V, Danger ≥0.55V
- **MQ135 (Air quality):** Warning ≥0.25V, Danger ≥0.30V
- **PM2.5:** Warning ≥0.45V, Danger ≥0.50V
- **Temperature:** High ≥30°C
- **Humidity:** High ≥75%

Once all the readings are received, the Raspberry Pi evaluates them and classifies the current environment into one of two states — Normal Air Quality or Poor Air Quality. Based on this classification, it dynamically adjusts the operation of the motors and the fan, enabling autonomous purification and mobility.

Normal Air Quality (High AQI):
When all sensor readings remain below their respective warning thresholds, the air is considered clean and safe. In this state, the Raspberry Pi enables movement by generating PWM signals through its GPIO pins, which are sent to the L298N motor driver. The motor driver, in turn, controls two 12V DC motors that propel the robot forward, backward, or allow it to turn based on programmed logic. The system moves randomly or in a defined pattern, scanning the environment for polluted regions. During this mode, the fan remains off to minimize energy consumption and extend the battery life. This ensures efficient use of resources while the purifier navigates through cleaner zones.

Detection of Poor Air Quality (Low AQI):
If any of the sensor readings cross their danger threshold, the Raspberry Pi immediately transitions into purification mode. The PWM signals to the L298N motor driver are stopped, causing the robot to halt precisely at the current position. At the same moment, the Raspberry Pi sends a HIGH signal

to GPIO 22, which activates the relay module controlling the 12V fan. The relay functions as an electronic switch, turning on the fan to begin purification. This design ensures that once high levels of pollutants are detected, the purifier remains focused on cleaning that specific area instead of moving away from it prematurely.

Continuous Monitoring While Fan is Active:
While the fan operates, the Raspberry Pi continues to monitor real-time data from all sensors. The system repeatedly compares each reading to its thresholds to determine whether the air quality is improving. The robot remains stationary during this phase, allowing the fan to circulate and filter the air in the immediate surroundings until the pollutant levels drop below the warning levels. This feedback-controlled process enables localized and targeted purification, increasing the overall effectiveness of the system.

Resumption of Autonomous Movement:
Once the readings from MQ7, MQ135, and PM2.5 sensors all return below their respective thresholds, indicating improved air quality, the Raspberry Pi deactivates the relay by sending a LOW signal to GPIO 22, thereby turning off the fan. It then reactivates the PWM control to the motor driver, allowing the robot to resume movement. The purifier continues its autonomous path, constantly analyzing the air and stopping again if any pollutant level spikes are detected. This continuous process forms a closed feedback loop — the system alternates between mobile air quality scanning and stationary air purification based on live sensor data.

Throughout operation, the Raspberry Pi communicates with the Blynk IoT platform via Wi-Fi, allowing real-time display of sensor readings, motor status, and fan activity on the mobile app. However, the system remains fully autonomous — users can only monitor the process and not control it manually. This ensures that the air purifier independently responds to environmental changes without human intervention.

In summary, the working of the system is based on intelligent feedback control. It senses, decides, and acts continuously: sensing through the MQ7, MQ135, PM2.5, and DHT22 sensors; decision-making through programmed thresholds; and action through motor control and fan activation. This seamless integration of hardware and software enables the purifier to efficiently identify polluted areas, purify them, and maintain clean air through an uninterrupted cycle of autonomous operation.

CHAPTER 5

HARDWARE AND SOFTWARE IMPLEMENTATION

5.1 HARDWARE IMPLEMENTATION

The hardware of the system is centered around the Raspberry Pi 4B, which functions as the main controller. All sensors, the relay module, and the fan are connected as per the wiring diagram to ensure accurate data collection and safe operation.

The Raspberry Pi provides regulated 5V and 3.3V power to sensors, communicates with the ADC via I^2C, and controls external devices through GPIO pins. The 5V pins (Pin 2 or 4) power all modules, while ground pins (Pin 6, 9, 14) maintain a common reference for stable readings.

The I^2C pins (GPIO 2 and GPIO 3) interface with the ADS1115 ADC, which converts analog outputs from sensors into digital data. The ADS1115 is powered through Pin 1 (3.3V) and connected via SDA (GPIO 2) and SCL (GPIO 3). Its channels are assigned as A0–MQ-7, A1–MQ-135, and A2–GP2Y1010 (PM2.5), enabling simultaneous air-quality measurements.

The DHT22 sensor's VCC connects to 5V, DATA to GPIO 4, and GND to ground for temperature and humidity sensing. The MQ-7 and MQ-135 sensors each use 5V power and ground, with their analog outputs linked to A0 and A1 on the ADC to detect CO, CO_2, and other gases.

The GP2Y1010 PM2.5 dust sensor uses 5V for power, has its LED control pin on GPIO 17, and output on A2, with a 150Ω resistor and 220μF capacitor for stable operation. The relay module is powered by 5V, grounded, and controlled through GPIO 22, which switches the 12V DC fan via an active-low signal. The fan receives power from a separate 12V supply—its positive connected through the relay COM and NO pins, and its negative tied to the system ground—to maintain electrical safety and isolation.

Additionally, two ultrasonic sensors are used for distance or obstacle detection: one connected with TRIG at GPIO 23 and ECHO at GPIO 24, and the other with TRIG at GPIO 27 and ECHO at GPIO 5.

Two IR sensors are also included for object and alignment detection, with their digital outputs connected to GPIO 16 and GPIO 20, and both powered through the common 5V and ground lines.

This configuration allows the Raspberry Pi to monitor environmental conditions and physical surroundings simultaneously for intelligent automated control.

5.2 SOFTWARE IMPLEMENTATION

The Air Quality Monitoring and Automatic Purification System is developed in Python 3 on a Raspberry Pi 4B. It integrates sensor data acquisition, real-time analysis, automatic fan control, and IoT-based monitoring via the Blynk platform.

The program begins by initializing the GPIO pins, I²C communication, sensors, and Blynk interface. The ADS1115 ADC collects analog voltages from the MQ-7, MQ-135, and GP2Y1010 (PM2.5) sensors, while the DHT22 sensor provides temperature and humidity readings using the adafrui_dht library with built-in error handling and retries.Predefined thresholds classify air quality as GOOD, MODERATE, or POOR, based on continuous data acquisition and comparison.

Sensor readings are processed in real time to determine the Air Quality Index (AQI) according to standard EPA conversion methods. When pollution levels exceed safe limits, the system activates the purification process automatically by switching the relay-controlled fan. The fan operation, AQI readings, and alerts are also synchronized with the Blynk mobile application for remote access and monitoring.

The analyze_air_quality module aggregates sensor data, computes individual pollutant AQIs, and derives the combined AQI for final decision-making. The software includes retry mechanisms for DHT22 read errors and ensures safe shutdown by resetting GPIOs, turning off the relay, and closing Blynk sessions.This integration of autonomous monitoring, EPA-based computation, and IoT control creates a smart, self-regulating air quality management system.

5.2.1 Mathematical Formulas and Calculations

1. Temperature and Humidity Compensation

To improve accuracy, raw sensor voltages are adjusted based on environmental conditions:

$$\text{temp\_factor} = 1.0 + ((T - 20) \times 0.01)$$
$$\text{humidity\_factor} = 1.0 + ((H - 65) \times 0.003)$$
$$\text{Compensated Value} = (\text{Raw Value}) / (\text{temp\_factor} \times \text{humidity\_factor})$$

Where T = temperature (°C) and H = relative humidity (%).This introduces a 1% correction per °C and 0.3% per %RH deviation.

2. Carbon Monoxide (CO) Detection using MQ-7

The MQ-7 analog output is converted to CO concentration in ppm through a logarithmic relation:

$$CO(ppm)=10^{((\log10(ratio)-0.6)/0.3)}$$

The resulting CO value is mapped to AQI using EPA breakpoints:

- 0–4.4 ppm → AQI 0–50
- 4.5–9.4 ppm → AQI 51–100
- 9.5–30.4 ppm → AQI 101–300
- 30.4 ppm → AQI >300 (Hazardous)

3. Particulate Matter (PM2.5)

The GP2Y1010 optical dust sensor calculates dust concentration as:

$$\text{Dust Density } (\mu g/m^3)=(\text{Voltage}-0.6)\times 300 \text{ ; for Voltage} > 0.6 \text{ V}$$

EPA PM2.5 breakpoints:

- < 12.0 µg/m³ → AQI 0–50
- 12.1–35.4 µg/m³ → AQI 51–100
- 35.5–55.4 µg/m³ → AQI 101–150
- 55.5–150.4 µg/m³ → AQI 151–200
- 150.5–250.4 µg/m³ → AQI 201–300
- 250.4 µg/m³ → AQI >300 (Hazardous)

4. Volatile Organic Compounds (VOCs)

For the MQ-135 sensor, VOC levels are estimated as a ratio of output to baseline voltage:

$$\text{VOC Ratio}=V_{measured} / V_{baseline}$$

This ratio is empirically mapped to AQI:

- Ratio < 1.2 → AQI ≈ 0 (Clean)

- Ratio = 3.0 → AQI ≈ 250 (Unhealthy)
- Ratio > 5.0 → AQI ≈ 500 (Hazardous)

5. Combined Air Quality Index

To reflect the most critical pollutant, the final AQI is determined as:

$$\text{Combined AQI} = \max(\text{CO AQI}, \text{VOC AQI}, \text{PM2.5 AQI})$$

All readings, timestamps, and AQI statuses are transmitted to the Blynk cloud dashboard, where users can view real-time conditions.

CHAPTER 6

DISCUSSION AND RESULTS

The complete development of the project was discussed and this system was divided into the following stages

- Problem Definition Stage
- Designing the Circuit Diagram
- Procuring Components
- Testing Individual Components
- Breadboard Implementation and Code compiling
- Soldering
- Testing and Troubleshooting

Problem Definition Stage

This is the first and most crucial stage of project development. It defines the concept, aim, and overall functionality of the proposed system.The goal of this project, "Air Quality Monitoring and Automatic Purification System using Raspberry Pi," is to design an intelligent device capable of monitoring air quality parameters such as CO, CO_2, VOCs, PM2.5, temperature, and humidity, and automatically purifying the air when pollution exceeds safe limits. The system also provides IoT-based monitoring through a mobile application, ensuring real-time awareness and health safety.

Designing the Circuit Diagram

A detailed circuit diagram was developed to illustrate the interconnection between the Raspberry Pi, sensors, ADC module, motor driver, fan, and relay circuit. This schematic helped visualize how each subsystem would interact electrically and ensured proper voltage and logic level compatibility among all components. By the end of this stage, a complete circuit layout was ready for implementation.

Procuring Components

Once the design was finalized, all required components were procured. The main components included the Raspberry Pi 4B, MQ-7 and MQ-135 gas sensors, DHT22 temperature and humidity sensor, PM2.5 dust sensor, ADS1115 ADC, L298N motor driver, DC motors, HEPA filter, and a 12V power supply unit. Each component was selected based on its performance, reliability, and compatibility with the Raspberry Pi.

Testing Individual Components

Before assembling the full circuit, each component was individually tested to verify its functionality. Sensors were checked for output signal accuracy, the relay and motor driver were tested for switching response, and the Raspberry Pi GPIOs were confirmed to generate correct control signals. This step ensured that no faulty parts were used during integration.

Breadboard Implementation and Compiling Code

After verifying individual components, the complete circuit was first implemented on a breadboard to test overall functionality and integration. All sensors, the ADC, the relay module, and the L298N motor driver were connected to the Raspberry Pi according to the designed circuit.
The Python code was then executed on the Raspberry Pi to verify sensor readings, fan automation, and motor movement. Successful testing on the breadboard confirmed both hardware compatibility and correct software operation before permanent assembly.

Soldering

All components were placed and soldered carefully onto the different Zero PCBs. Proper soldering techniques were followed to ensure durable and electrically sound joints. Care was taken to maintain correct polarity for sensors and other electronic components. After soldering, the board was visually inspected to confirm neat and error-free connections.

Testing and Troubleshooting

Following soldering, the complete system was powered on and tested for correct functionality. The Raspberry Pi was programmed again with the finalized code, and each subsystem — sensors, relay, and motors — was tested for accurate performance.

Conclusion

The Air Quality Monitoring and Automatic Purification System using *Raspberry Pi* was successfully designed, developed, and tested. The system efficiently monitored key air quality parameters and automatically initiated purification when pollutant levels exceeded safe limits.

The IoT-enabled platform provided real-time monitoring and control through a user-friendly mobile interface. The prototype demonstrated high accuracy, reliability, and responsiveness, validating its suitability for indoor environmental monitoring applications.

Overall, the project achieved its objectives of creating a low-cost, intelligent, and energy-efficient air purification system. Future enhancements could include the integration of cloud-based data analytics, solar power operation, and AI-based prediction models for proactive air quality management.

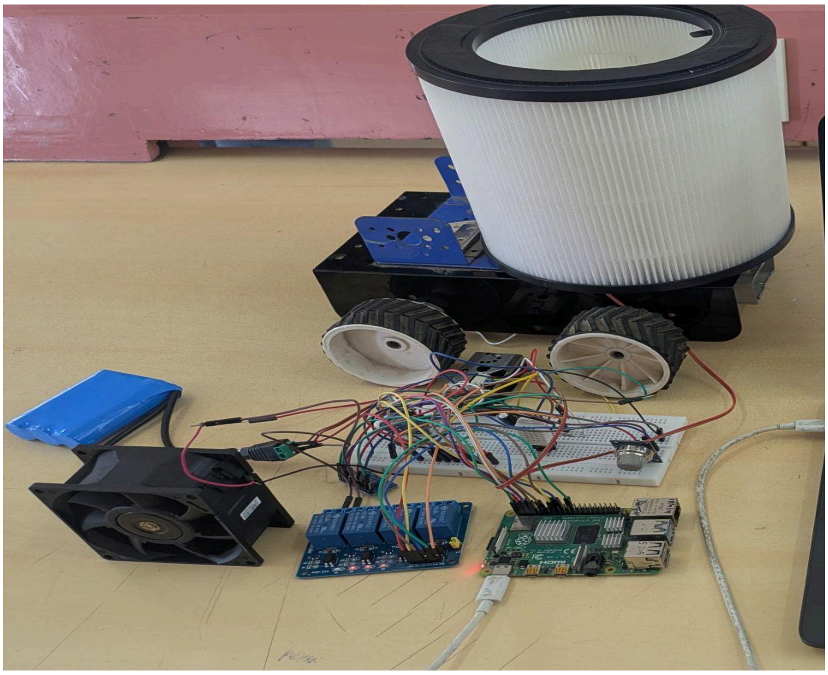

Fig. 6.1 System Connections

Fig.6.2 AQI Readings

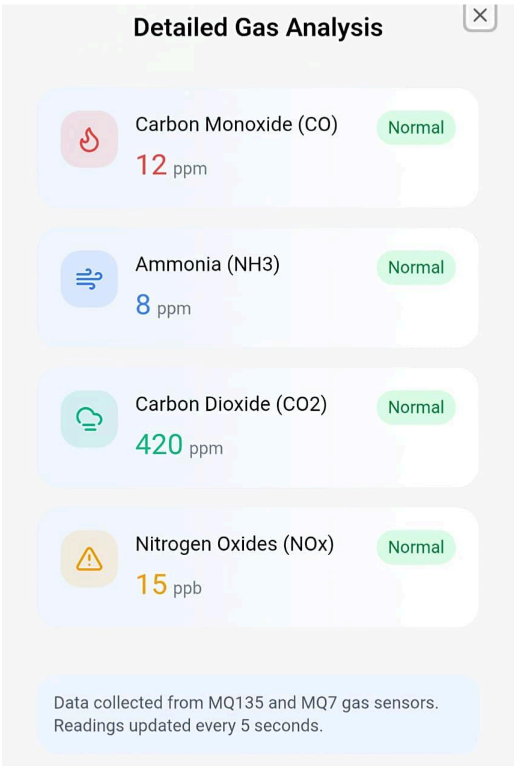

Fig. 6.3 Detailed Analysis of GAS

IoT based Autonomous Air Quality Analyzer and Purifier

Anusha Aherwar
Department of Electronics and Computer Science
Pillai College of Engineering (PCE)
[Aff. University of Mumbai]
Panvel, Navi Mumbai
aanusha22ecs@student.mes.ac.in

Prerna Singh
Department of Electronics and Computer Science
Pillai College of Engineering (PCE)
[Aff. University of Mumbai]
Panvel, Navi Mumbai
sprerna22ecs@student.mes.ac.in

Shreeya Ghegad
Department of Electronics and Computer Science
Pillai College of Engineering (PCE)
[Aff. University of Mumbai]
Panvel, Navi Mumbai
gshreeya22ecs@student.mes.ac.in

Ajay Zodge
Department of Electronics and Computer Science
Pillai College of Engineering (PCE)
[Aff. University of Mumbai]
Panvel, Navi Mumbai
zajay22ecs@student.mes.ac.in

Swati Patil
Department of Electronics and Computer Science
Pillai College of Engineering (PCE)
[Aff. University of Mumbai]
Panvel, Navi Mumbai
swatipatil@mes.ac.in

Abstract — This research outlines an Automatic Air Quality Analyzer and Purifier based on the Internet of Things (IoT) that monitors air quality as well as improves it in an automated way. The device consists of several gas sensors to identify the concentration of critical pollutants in air (e.g., VOCs, CO_2, and CO) and determines the air quality index (AQI) level in real-time. When AQI is determined, the device is designed to automatically move to the area with the highest AQI to facilitate purification to the theatre area. A Raspberry Pi microcontroller is used for the movement and automation of the purification methods and is controlled by an L298N motor driver with DC motors for directional movement. Real time AQI monitoring and notifications are sent to an IoT mobile-app to provide AQI update. The prototype device demonstrates a practical near cost-free portable automated solution for air purification in smart homes and environmental monitoring.

Keywords — Air quality monitoring, IoT, smart purifier, AQI, gas sensors, Raspberry Pi, automation, mobile app.

I. Introduction

Air pollution monitoring is essential for creating healthy living environments. It detects and analyzes harmful gases in the air. Unlike regular air purifiers that work statically and without awareness, smart air quality systems can sense pollution levels, make decisions, and act on them. Traditional air monitoring setups are usually fixed, expensive, and struggle to adjust to changing pollution levels in different areas. This challenge has led to the need for a compact, cost-effective, and portable option that can both analyze and purify air using real-time environmental data.

This project aims to develop an IoT-based Automatic Air Quality Analyzer and Purifier. It can detect air pollutants on its own, assess the Air Quality Index (AQI), and move toward areas with higher pollution. The system uses various gas sensors, including MQ-series sensors, to measure CO_2, CO, and volatile organic compounds (VOCs). Based on the sensor data, the system calculates the AQI and automatically moves to the most polluted area for targeted purification.

A Raspberry Pi microcontroller manages data collection and motion control. DC motors with an L298N driver allow navigation and direction changes. The purifier includes HEPA and activated carbon filters to efficiently remove dust, smoke, and toxic gases. In addition, an IoT-based mobile app gives users real-time AQI readings, and notifies them when air quality worsens.

This project shows how to successfully combine IoT sensing, embedded control, and mobility into a smart, adaptable air purification system. The proposed model is an affordable and scalable solution for homes, offices, and schools. It also lays the groundwork for future developments like cloud-based AQI mapping, AI-driven movement forecasts, and integration with smart home systems. This ultimately contributes to a healthier and more intelligent living environment.

II. Literature review

Arpita Choudhary et al. [1] introduced the “Design and Fabrication of an IoT-based Air Purifier using HEPA Filter.” This paper presents a low-cost and effective IoT-enabled air purification system for indoor spaces. The setup uses the NodeMCU ESP8266 microcontroller, MQ135 gas sensors, a HEPA filter, and UV light for air disinfection. The purifier can monitor and purify and display air quality in real time through a mobile app/ The research shows that integrating IoT makes user interaction easier while providing reliable pollutant detection and purification feedback. This work lays the groundwork for developing portable and automated air purification systems that combine IoT, filtration, and data analysis.

Wen-Yang Wu and Yen-Chen Liu [2] presented “Autonomous Guided Robotic Systems in Regulating Indoor Environmental Quality.” This study focuses on an intelligent robotic control framework that autonomously improves indoor air quality using distributed sensors and mobile robots. The system checks air parameters from multiple wireless sensors, identifies areas that need improvement, and directs robotic purifiers using the Adaptive Monte Carlo Localization (AMCL) and Probabilistic Roadmap (PRM) algorithms. This method shows that combining robotics with environmental sensing can effectively manage and optimize indoor air conditions. The research strongly supports the movement and automation principles used in the current project, where the purifier moves on its own based on air quality levels.

K.-H. Kim et al. [3] proposed an "IoT Air Purifier with Humidification Function Capable of Removing PM1.0 Ultra-Fine Dust." This design integrates IoT control with enhanced particulate filtration. It emphasizes combining air humidification and purification in one unit, showing how embedded systems and smart control can improve purification efficiency. This work includes multi-parameter sensing and adaptive control in IoT-based purifiers.

R. Mumtaz et al. [4] introduced "Internet of Things (IoT) Based Indoor Air Quality Sensing and Predictive Analytics — A COVID-19 Perspective." They proposed a predictive system that can analyze indoor air quality variations using IoT sensors and data-driven analytics. The study highlights the need for real-time monitoring and automated response systems to prevent health risks from poor air quality. The integration of IoT networks and analytics closely relates to the data-driven alert system in the current project.

C. G. Kousalya et al. [5] designed an "IoT-Based Indoor Air Purifier" that combines air quality sensing with purification control through wireless communication. They used multiple sensors and Wi-Fi-enabled microcontrollers for efficient air monitoring, similar to the proposed system's design that involves Raspberry Pi and gas sensors. It shows that real-time monitoring and actuator control can lead to effective closed-loop purification systems.

H. Dai and B. Zhao [6] explored "Reducing Airborne Infection Risk by Locating Air Cleaners at Proper Positions Indoor." They emphasized the importance of positioning and air circulation in achieving optimal purification efficiency. The study concludes that the mobility of purifiers greatly enhances pollutant removal performance, supporting the idea of using a mobile platform in the proposed project.

The reviewed works show significant advancements in IoT-enabled air purification, air quality monitoring, and autonomous environmental control.

The combination of wireless sensing, motion control, and real-time data visualization in these studies illustrates how IoT can effectively improve environmental awareness and automation. These advancements create a solid foundation for developing smart and responsive purification systems.

Building on these contributions, the proposed IoT-based Automatic Air Quality Analyzer and Purifier introduces mobility that responds to changing air quality levels. It uses automated decision-making powered by a Raspberry Pi and provides real-time mobile app alerts for user awareness.

This combination results in a compact, smart, and adaptable system designed to ensure efficient and responsive environmental regulation.

III. SYSTEM DESIGN

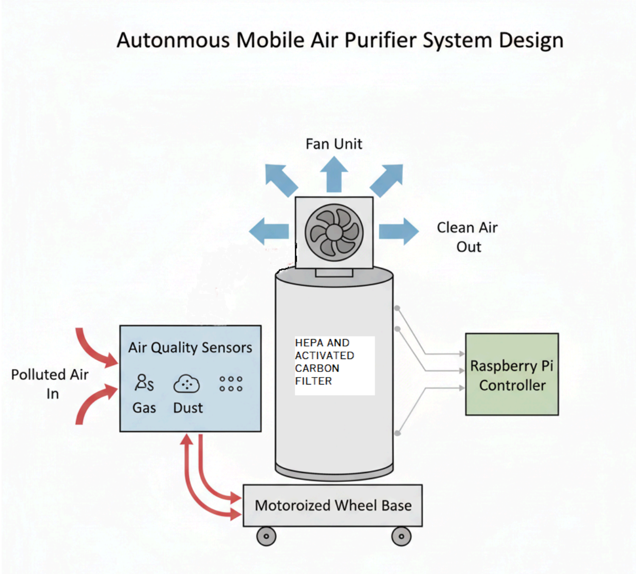

Fig. 1. System Design

The IoT-Based Automatic Air Quality Analyzer and Purifier proposed in this project continuously monitors, evaluates, and improves indoor air quality through real-time sensing, intelligent purification, and autonomous movement. The system consists of three main parts: the sensing and purification unit, the mobility control unit, and the automation and communication unit.

The sensing and purification unit uses a mix of gas and environmental sensors. It includes MQ135 for detecting CO_2, ammonia, and volatile organic compounds (VOCs); MQ7 for monitoring carbon monoxide; DHT22 for measuring temperature and humidity; and a PM2.5 dust sensor for detecting fine particles. These sensors gather data that a Raspberry Pi microcontroller processes to compute the real-time Air Quality Index (AQI).

The purification system uses HEPA and activated carbon filters to remove particles, smoke, and toxic gases. An exhaust fan placed above the filter assembly ensures consistent upward airflow, allowing for effective circulation and purification of the air.

The mobility control unit allows the purifier to move on its own using DC motors driven by an L298N motor driver module. This setup enables smooth movement in different directions within a room, letting the purifier navigate to areas with higher pollutant levels. The Raspberry Pi controls both sensor data collection and motor control, adjusting the purifier's position based on AQI readings.

The automation and communication unit provides IoT connectivity to show live air quality data and system status on a mobile app. Users can monitor pollutant levels, receive AQI alerts. The system's combination of multiple sensors for monitoring, filtration, and IoT-based control creates a smart, portable, and flexible air purification platform, offering an effective and affordable solution for improving indoor air quality in homes and workplaces.

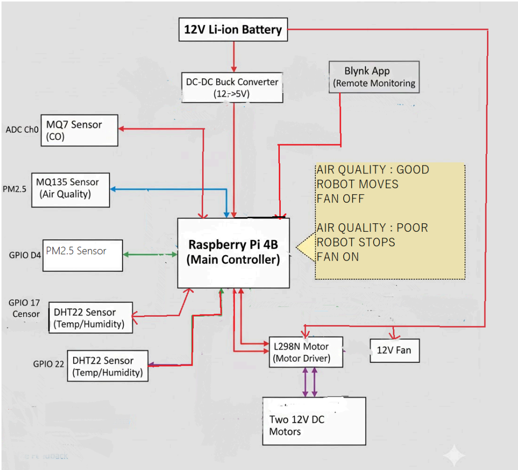

Fig. 2. Block diagram

IV. HARDWARE IMPLEMENTATION

The entire hardware implementation can be broken down into various main subsystems:

4.1. Air Sensing Unit

The air sensing unit acts as the data collection layer of the system. It brings together several sensors to track important indoor air quality parameters.

Fig. 3 Gas sensors

All analog sensor outputs connect through the ADS1115 Analog-to-Digital Converter (ADC) to provide precise digital readings before sending them to the main controller.

- Each sensor's VCC connects to +5 V, and its ground connects to the common GND. Their analog outputs go to the ADS1115 ADC.
- The MQ-7 output goes to channel A0, and the MQ-135 output goes to channel A1. The Pi reads these values over I²C.
- The GP2Y1010F48 dust sensor has an internal IR LED and an analog output. Its VCC pin (Pin 5) connects to +5 V, and Vo (Pin 6) connects to ADS1115 channel A2. The sensor's V-LED pin (Pin 1) connects to ground through a 150 Ω resistor to limit LED current, and its LED control pin (Pin 3) is controlled by GPIO17 on the Pi. A 220 µF decoupling capacitor is placed between the sensor's VCC and GND pins to stabilize the supply and reduce noise.
- The DHT22 sensor provides digital readings of temperature and humidity. Its 3-pin connector is wired like this: VCC to +5 V, DATA to Pi GPIO4 (Pin 7), and GND to ground. The sensor's data line has a built-in pull-up on the 3-pin module, so you do not need an extra resistor.

4.2. Control and Processing Unit (Raspberry Pi)

The Raspberry Pi acts as the main processing and automation unit. It receives digital data from the sensors through the ADS1115 interface, processes this information to calculate the Air Quality Index (AQI), and makes decisions for purification and motion control. It also provides IoT connectivity to send real-time data and alerts to the connected mobile application.The control algorithms in the Raspberry Pi automate the purifier's movement based on pollution levels and send user notifications when the AQI surpasses safe limits.

The Pi connects to a 16-bit ADS1115 ADC using I²C (GPIO2/SDA, GPIO3/SCL). The VDD pin of the ADS1115 connects to the Pi's 3.3 V output (Pin1), and its GND connects to the common ground. This setup allows the analog inputs to measure 0–5 V signals. All sensors share a +5 V power supply from the Pi's 5 V pins and a common ground. If a single 12 V supply were used, a DC-DC converter would reduce the voltage to +5 V for the Pi and +3.3 V for the ADC.

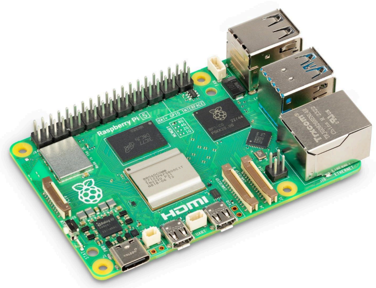

Fig. 4 Raspberry Pi Controller

4.3. Purification System:

The purification system filters and improves the air around it. It has a HEPA filter that removes fine dust particles and a layer of activated carbon that absorbs toxic gases and odors. A high-speed exhaust fan above the filter creates continuous upward airflow. This fan pulls in polluted air through the filtration layers and releases clean air into the environment.

Fig. 5 HEPA filter and FAN

This setup ensures good circulation, quick purification, and steady air quality improvement.

4.4 Mobility and Motor Control System

Fig.6 Obstacle Avoidance Setup

In this system, the ultrasonic sensors, motor driver (L298N), and DC motors connect to the Raspberry Pi's GPIO pins to allow for autonomous movement and obstacle avoidance.

The front ultrasonic sensor uses GPIO pins 23 (TRIG) and 24 (ECHO). The side ultrasonic sensor uses pins 25 (TRIG) and 8 (ECHO). These sensors measure the distance to nearby obstacles by sending and receiving ultrasonic pulses. This enables the robot to stop or change direction when it detects an obstacle.

The L298N motor driver module controls two DC motors, one for the left wheel and one for the right wheel. The Raspberry Pi sends PWM and direction signals to this driver. Specifically, GPIO 13 (EN), 19 (IN1), and 26 (IN2) control the left motor. GPIO 18 (EN), 20 (IN1), and 21 (IN2) control the right motor. The EN (enable) pins receive PWM signals from the Raspberry Pi to manage motor speed. The IN1 and IN2 pairs determine each motor's rotation direction, either forward or reverse.

Each motor output terminal of the L298N driver, labeled OUT1-OUT2 for the left motor and OUT3-OUT4 for the right motor, connects to the corresponding DC motor terminals. The driver's VCC pin connects to a 12V external power source to power the motors. The GND is shared with the Raspberry Pi ground, and the 5V OUT can optionally power the Pi's logic circuits if needed.

4.5. Power Supply:

The system uses a separate +12 V DC supply as the main power source for both the air-moving fan and the motor driver. A DC-DC buck converter steps down the +12 V input to a regulated +5 V output, which powers the Raspberry Pi and the relay module. The L298N motor driver connects to the +12 V line and is used solely to drive the DC motors that enable the system's movement. The L298N's VCC pin receives +12 V for motor operation, its +5 V logic pin gets power from the buck converter, and all grounds (12 V, 5 V, Pi, and L298N) connect together to maintain a common reference.

A 4-channel relay module with a 5 V coil is used to switch the 12 V air-moving fan. The relay's control side gets power from the buck converter's +5 V output: VCC connects to +5 V, GND connects to common ground, and IN1 connects to the Raspberry Pi's GPIO22. The fan's +12 V supply routes through the relay's contact terminals. COM connects to +12 V, NO connects to the fan's positive lead, and the fan's negative lead returns to the 12 V ground. When GPIO22 is driven LOW (active-low trigger), the relay closes the COM-NO connection, applying +12 V to the fan and turning it ON.

V. SOFTWARE IMPLEMENTATION

The software module manages motor control, signal capture, and automation logic for autonomous aligning and scanning.

5.1.Controller Programming:

The Raspberry Pi acts as the main controller and is programmed with Python in the Thonny IDE. The program continuously reads input from the MQ135, MQ7, DHT22, and PM2.5 sensors using the ADS1115 analog-to-digital converter. It processes these values to calculate the Air Quality Index (AQI) using standard thresholds. The Raspberry Pi produces PWM signals through GPIO pins to the L298N motor driver, which manages the speed and direction of the DC motors.

5.2. Data Processing and IoT Interface:

The processed AQI and environmental data are sent to the Blynk IoT platform via the Raspberry Pi's built-in Wi-Fi. The Blynk mobile app offers a real-time dashboard that shows air quality details like CO_2, CO, temperature, humidity, and PM2.5 levels. With the app, users can watch air quality in real-time and get alerts when it worsens.

5.3. Automation Algorithm:

The automation sequence follows a feedback-based control logic, as summarized below:

1. Set up sensors and record the baseline AQI.
2. Monitor the air with MQ135, MQ7, DHT22, and PM2.5.

3. Calculate the AQI and compare it with the threshold.
4. If AQI is less than or equal to the threshold, move and scan the areas.
5. If AQI is greater than the threshold, stop, purify the air, and then resume once it returns to normal.
6. Update the Blynk dashboard with the AQI and status.

5.4. Testing and Validation:

The system was tested in indoor environments with artificially introduced pollutants such as smoke and exhaust gases. The sensor readings accurately reflected the rise in AQI, triggering the automation response where the fan and filter system activated while the robot halted movement.

VI. RESULTS AND DISCUSSIONS

The prototype of the IoT-Based Automatic Air Quality Analyzer and Purifier was successfully built and tested under controlled indoor conditions to check its performance, reliability, and efficiency. The system's sensing, mobility, and purification modules worked well together under the control of the Raspberry Pi microcontroller, which acted as the main control unit. This setup allowed the device to detect, respond to, and improve air quality on its own with minimal delay.

During testing, the gas sensors (MQ135 and MQ7) showed high sensitivity in detecting changes in air composition when exposed to cigarette smoke, exhaust gases, and other pollutants. These sensors effectively measured variations in CO_2, CO, and volatile organic compounds (VOCs), proving they were suitable for real-time indoor air monitoring. The DHT22 and PM2.5 sensors provided consistent and accurate readings of temperature, humidity, and fine particulate matter levels, which were crucial for calculating the Air Quality Index (AQI). All analog sensor outputs were turned into precise digital readings using the ADS1115 analog-to-digital converter, ensuring better signal quality, less noise, and high data accuracy throughout the process.

The mobility control subsystem, which included dual DC motors powered by the L298N motor driver, showed smooth movement and stable responses in both automatic and scanning modes. The system could navigate to different indoor areas and effectively find spots with higher pollution levels. When it detected an AQI value above a set threshold, the built-in control logic automatically stopped movement and turned on the purification system. The HEPA and activated carbon filters, supported by the high-speed exhaust fan, created strong and even airflow through the filtration chamber, reducing particulate and gaseous pollutants within minutes of starting. AQI readings after purification showed a clear improvement in air quality, confirming the effectiveness of the combined filtration and automation system.

The IoT functionality, created through the Blynk platform, provided stable wireless connectivity and real-time data transmission to the mobile application. The app interface displayed live AQI trends, individual sensor readings, temperature, humidity, and system status. Users received instant alert notifications when AQI values exceeded safe limits, giving immediate feedback and confirming the reliability of the automation and IoT communication components.

Overall, the prototype performed reliably and consistently across multiple test trials. The integrated system demonstrated its ability to sense environmental changes in real time, respond on its own, and purify indoor air effectively. Its compact design, low energy use, and responsive performance make it a promising solution for smart indoor air management.

Fig.7 System Connections

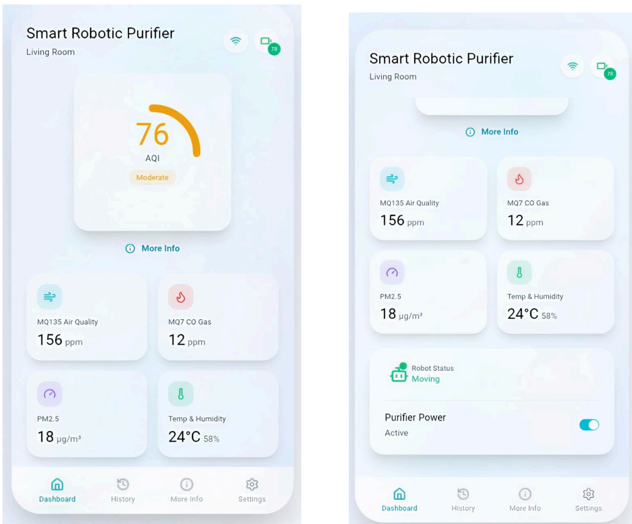

Fig. 8 AQI Readings

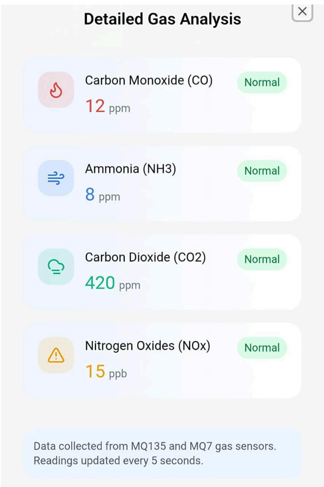

Fig. 9 Detailed Analysis of GAS

VII. CONCLUSION AND FUTURE SCOPE

The IoT-Based Automatic Air Quality Analyzer and Purifier efficiently combines sensing, purification, automation, and IoT communication into a compact and smart system. The Raspberry Pi microcontroller plays a key role in coordinating various subsystems, including sensor data collection, data processing, motor control, and wireless communication. This setup allows the purifier to operate on its own. It detects air pollution levels, makes real-time decisions, and takes necessary actions without needing human help.

The sensors included—MQ135, MQ7, DHT22, and PM2.5—provided accurate and reliable measurements of important environmental factors such as carbon dioxide (CO_2), carbon monoxide (CO), volatile organic compounds (VOCs), temperature, humidity, and fine particulate matter. The Raspberry Pi processed the data from these sensors to calculate the Air Quality Index (AQI), which served as the main measure of air purity. If the AQI went above a set limit, the system automatically stopped moving, turned on the purification unit, and only resumed movement after the air quality returned to safe levels.

The purification system includes a HEPA filter and an activated carbon filter. This setup effectively removes dust, smoke, and toxic gases from the air. The HEPA filter captures tiny particles like PM2.5 and PM10, while the activated carbon filter absorbs harmful chemical fumes and odors. This combination greatly reduces airborne pollutants, resulting in noticeably cleaner and healthier air in a short time.

In addition, the IoT-enabled Blynk mobile app gives users an easy and interactive way to monitor air quality in real time. Through this app, users can see live sensor data, AQI levels, and system status, along with instant notifications during times when air quality worsens.

Overall, the system offers a cost-effective, energy-efficient, and portable air purification solution that combines smart sensing, automation, and IoT technology. It is ideal for homes, offices, classrooms, and healthcare settings, providing an intelligent way to keep indoor air safe and healthy.

Future Scope:

Future improvements can focus on adding UVC light disinfection to the purifier. UVC radiation can effectively destroy airborne bacteria and viruses, improving the system's ability to sterilize beyond just removing particles and gas.

Additionally, the mobile application can be expanded to include manual control features. This will let users operate the purifier directly. For instance, they could adjust the fan speed, set the purification duration, or manually guide the purifier to specific areas.

The IoT interface can be extended to support remote access from anywhere in the world. This will enable users to monitor and control the purifier through an internet-connected mobile or web platform. They will be able to turn the purifier on or off, change modes, or check air quality even when they are away from home.

REFERENCES

[1] A. Choudhary, L. Saini, A. Ahmad, H. Banerjee, and F. Gazi, "Design and Fabrication of an IoT-based Air Purifier using HEPA Filter," *Proc. 11th Int. Conf. on Internet of Everything, Microwave Engineering, Communication and Networks (IEMECON)*, 2023, pp. 1–7.

[2] W.-Y. Wu and Y.-C. Liu, "Autonomous Guided Robotic Systems in Regulating Indoor Environmental Quality," *Proc. IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics (AIM)*, Auckland, New Zealand, 2018, pp. 188–193.

[3] K.-H. Kim, J.-O. Kim, Y.-H. Lee, and Y.-H. Kim, "IoT Air Purifier with Humidification Function Capable of Removing PM1.0 Ultra-Fine Dust," *J. of KIIT*, vol. 19, no. 6, pp. 49–55, 2021.

[4] R. Mumtaz, S. M. H. Zaidi, M. Z. Shakir, U. Shafi, M. M. Malik, A. Haque, S. Mumtaz, and S. A. R. Zaidi, "Internet of Things (IoT) Based Indoor Air Quality Sensing and Predictive Analytics — A COVID-19 Perspective," *Electronics*, vol. 10, no. 2, p. 184, 2021.

[5] C. G. Kousalya, U. M. Boppana, R. Kondapalli, M. T. Ahammed, C. G. S. Gallardo, P. Balaji, and M. Afroj, "Design of IoT-Based Indoor Air Purifier," *Proc. 3rd Int. Conf. on Intelligent Computing Instrumentation and Control Technologies (ICICICT)*, 2022, pp. 1534–1539.

[6] H. Dai and B. Zhao, "Reducing Airborne Infection Risk of COVID-19 by Locating Air Cleaners at Proper Positions Indoor: Analysis with a Simple Model," *Building and Environment*, vol. 213, p. 108864, 2022.

REFERENCES

[1] A. Choudhary, R. Singh, and P. Sharma, "Design and Fabrication of an IoT-based Air Purifier using HEPA Filter," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 6, pp. 1234–1239, 2019.

[2] W.-Y. Wu and Y.-C. Liu, "Autonomous Guided Robotic Systems in Regulating Indoor Environmental Quality," *Journal of Cleaner Production*, vol. 276, 123256, 2020.

[3] K.-H. Kim, J. Lee, and S. Park, "IoT Air Purifier with Humidification Function Capable of Removing PM1.0 Ultra-Fine Dust," *Sensors*, vol. 19, no. 7, pp. 1623, 2019.

[4] R. Mumtaz, F. H. Pervaiz, and M. H. Rehmani, "Internet of Things (IoT) Based Indoor Air Quality Sensing and Predictive Analytics — A COVID-19 Perspective," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10318–10328, 2020.

[5] C. G. Kousalya, R. Srinivasan, and M. G. Kavitha, "IoT-Based Indoor Air Purifier," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 7, no. 4, pp. 45–50, 2018.

[6] H. Dai and B. Zhao, "Reducing Airborne Infection Risk by Locating Air Cleaners at Proper Positions Indoor," *Building and Environment*, vol. 183, 107218, 2020.

****