A PROJECT REPORT ON

SMART EXHAUST FAN FOR AIR QUALITY MONITORING USING IOT

Mini project submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

(2020-2024)

BY

A. LOUKYA SREE	**20241A1201**
U. SUHITHA	**20241A1252**
V. HARIKA	**20241A1255**
B. SATHVIKA	**21245A1204**

Under the Esteemed guidance

of

N. Mounika

Assistant Prof

Dept of IT.

DEPARTMENT OF INFORMATION TECHNOLOGY GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY (AUTONOMOUS), HYDERABAD.

CERTIFICATE

This is to certify that it is a bonafide record of Mini Project work entitled "**SMART EXHAUST FAN FOR AIR QUALITY MONITORING USING IOT"** done by **A.LOUKYA SREE (20241A1201), U.SUHITHA (20241A1252), V.HARIKA (20241A1255), B.SATVIKA (21245A204)** of **B.Tech (IT)** in the Department of Information Technology, **Gokaraju Rangaraju Institute of Engineering and Technology** during the period 2020-2024 in the partial fulfilment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY IN INFORMATION TECHNOLOGY** from GRIET, Hyderabad.

N.MOUNIKA,	**Dr.N.V.Ganapathi Raju,**
Assistant Prof.	(Head of the Department)
(Internal Project Guide)	

(Project External)

ACKNOWLEDGEMENT

We take immense pleasure in expressing gratitude to our internal guide, **N.Mounika, Assistant Professor, Dept. of IT, GRIET.** We express our sincere thanks for her encouragement, suggestions, and support, which provided the impetus and paved the way for the successful completion of the project work.

We wish to express our gratitude to **Dr.N.V.Ganapathi Raju,** HOD IT**,** and our Project Coordinator, **K.Archana** for their constant support during the project.

We sincerely thank **Dr.Jandhyala N Murthy**, Director, GRIET, and **Dr. J. Praveen**, Principal, GRIET, for providing us a conductive environment for easily carrying through our academic schedules and project.

We also take this opportunity to convey our sincere thanks to the teaching and non-teaching staff of GRIET College, Hyderabad.

Name: Akula Loukya Sree
Email: loukyaakula7@gmail.com
Contact No: 9052920038
Address: Sangareddy, Telangana.

Name: Suhitha Ulavalapudi
Email: suhitha.rinky@gmail.com
Contact No: 9966326569
Address: Malkajgiri, Hyderabad.

Name: Vanguru Harika Reddy
Email: harika282003@gamil.com
Contact No: 7396401503
Address: Jeedimetla, Hyderabad.

Name: Bathula Sathvika
Email: sathvikabathula0@gmail.com
Contact No: 8885755404
Address: Lingampally, Telangana.

DECLARATION

This is to certify that the project entitled **"SMART EXHAUST FAN FOR AIR QUALITY MONITORING USING IOT"** is a bonafide work done by us in partial fulfilment of the requirements for the award of the degree **BACHELOR OF TECHNOLOGY IN INFORMATION TECHNOLOGY** from Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad.

We also declare that this project is a result of our own effort and has not been copied or imitated from any source. Citations from any websites, books and paper publications are mentioned in the Bibliography.

This work was not submitted earlier at any other University or Institute for the award of any degree.

A.LOUKYA SREE	**20241A1201**
U.SUHITHA	**20241A1252**
V.HARIKA	**20241A1255**
B.SATHVIKA	**21245A1204**

TABLE OF CONTENTS

ABSTRACT

Bad Air quality results in a serious problem in industries, households, and other areas where harmful gases are detected, therefore the proposed Smart Exhaust Fan to control air quality is developed using IoT. Harmful gases cause damage to living tissues, impairs the central nervous system, causes severe illness or, in extreme cases even causes death when inhaled or absorbed. Industrial gases like Hydrogen Sulphide, Carbon Monoxide, Nitrogen Oxides and household gases like gas fuels, Ammonia, Chlorine, fire gases, etc. are extremely toxic and hazardous for the living. Short-term effects of harmful gases include illnesses such as pneumonia, bronchitis, other discomforts like irritation to the nose, eyes and skin, dizziness and nausea. Long-term effects include heart diseases, lung cancer, and respiratory diseases. People with weaker immune systems are often more sensitive to such polluted air.

Thus, the Exhaust Fan which is used here can help in extracting the impure air and bring in fresh air thereby improving the air quality. The application uses sensors like MQ-135 to monitor the air quality and warns us about the raised air quality index using serial monitor. In addition to this, an app interface is used to send the respective user a notification or message to warn them about the air quality levels (if increased beyond the required threshold) and provide us with a Smart Exhaust Fan control option. This project helps society to specifically monitor the air quality and also helps users to automatically take required precautions to avoid any accidents or leakages using the IOT technique.

Keywords: MQ-135 gas sensor, 12V DC Exhaust Fan.

Domain: Internet of Things.

1.INTRODUCTION

1.1 Introduction to the project

Electrical fan is one of the commonly used electronic appliance devices for air control, especially in the home environment.

They get rid of excess of moisture and unwanted odours in rooms, kitchens, washrooms, etc. They suck out moist or hot air, replacing it with fresh and cool air. Ensuring proper ventilation in such places is of prime importance.

Some of the existing works have limited functionality in the form of the standalone system. As such, this work aims to improve the functionality by design and implementation of remote fan control on a wireless sensor network (WSN) in the smart home system.

It means that the fan device is connected with the others devices such as Air Quality sensor, humidity and temperature sensor, curtain, and smart plug.

The control for user will be conducted using smartphone based application that provides slider-based or three buttons-based speed controls.

A small fan-driven using DC motor is used as a representation/model for an exhaust fan commonly found on kitchen and bathroom walls. The DC motor works at 12V DC and 0.14A.

The Exhaust Fan takes into consideration the ppm and temperature range of air and acts accordingly i.e. if the measured air quality is greater than the threshold value specified, the exhaust fan is switched on manually(physical switch). If the air quality is below the threshold value, the exhaust fan remains switched off.

Throughout the process, the air quality value will be continuously measured, monitored and stored in the Database.

The stored data can be used for data analysis and produce reports.

1.2 Existing Systems

Traditional gas detectors: A gas detector can sound an alarm to operators in the area where the leak is occurring, giving them the opportunity to leave. This type of device is important because there are many gases that can be harmful to organic life, such as humans or animals. But these were only something that was connected by hardware and as a result, only people who were closer to the alarm were notified.

Exhaust Fan: An exhaust fan is a fan that is employed to control the interior environment through venting out not wanted odours, moisture, particulates, smoke, and other contaminants that might be present in the air. General locations for exhaust fans involve bathrooms and kitchens. The exhaust fans are as well termed fresh air fans. The major parts like motor, capacitor, cord are similar as that of table fans. They mostly work based on the hardware controls.

Disadvantages of existing systems:

- No cloud storage.
- No application interface.

1.3 Proposed System:

We developed a Smart Exhaust Fan to monitor air quality and temperature using IoT with cloud which can be used to monitor the harmful gas levels in the air and used to control the exhaust fan as required.

If the temperature and air quality levels go beyond the threshold, the micro controller uses the mobile application BLYNK to notify the person about the air quality levels and additional cloud connection adds more importance of it.

Advantages of Proposed System:

- Implementation of Cloud storage.
- Application Interface.

2.REQUIREMENT ENGINEERING

2.1 SOFTWARE REQUIREMENTS

• Operating System:- Windows 11	• IDE – Arduino Software
• BLYNK Mobile App	• Programming Language – Sketch

2.2 HARDWARE REQUIREMENTS

• ESP32 Wi-Fi Module	• 12V DC Exhaust fan
• MQ-135, DHT-11 Sensor	• Bread board

3.LITERATURE SURVEY

[1]A prototype of control system (with hard- and soft-control method) for electrical fan is designed for smart home system. The system utilizes STM32L100 microcontroller in order to support low power operation. Based on the testing conducted, the fan can be configured for various speed, and it requires 43.1 mA (517.2 mW) in an idle condition and 145.1 mA (1,741.1 mW) in a processing condition.

[2] proposed system defines that detection and monitoring of the LPG gas is sensed using MQ – 5 sensors. In the system, when the leakage of gas is detected, the buzzer will be on, along with that the alert message will be displayed in the LCD. And while monitoring of gas based on the cylinder weight, which is measured using the load sensor, it will send the message to the owner of the application or system.

[3] This device is used as the detect gas is already present in the market which is commonly used in many places such as industries where there are many chances of detonation which may lead to great destruction and the loss of manpower; in homes, where the LPG gas is used most commonly in our daily lives inevitably where it can detect the leakage of LPG gas; in cars, where most vehicles carry the cylinder and many more places. In 1910, Dr. Walter Snelling was the first to commercialise LPG gas. It combines propane and commercial propane. It's extremely volcanic, and many accidents happen as a result of LPG leaks. As a result, it is critical to connect and prevent gas leakage. Gas detectors can be classified in a number of ways. They are classified according to the type of gas they detect, the automation that powers the sensor's output, and the components that influence the sensor's power (semiconductors, oxidation, catalytic, photoionization, infrared, etc.). In our daily lives, we use a variety of devices for various purposes, and the majority of them can emit any type of gas or chemical when in use in the air. It is difficult for a human to keep track of the levels of application of the leaked gas or liquid in any scheme. If there is a gas leak when no one is around, it may cause a detonation if there is even a spark, or the surrounding area will have the dangerous gas, which may cause suffocation and fitness issues in breathing. There are many applications for observing and monitoring gas leakage, but researchers will still attempt to build the advanced application where the value of the application will be lower.

[4] The traditional Gas Leakage Detector Systems though have great precision, fail to acknowledge a few factors in the field of alerting the people about the leakage. Therefore we have used the IoT technology to make a Gas Leakage Detector for society which having Smart Alerting techniques involving sending text message to the concerned authority and an ability performing data analytics on sensor readings. Our main aim is to proposing the gas leakage system for society where each flat have gas leakage detector hardware. This will detect the harmful gases in environment and alerting to the society member through alarm and sending notification.

[5] IoT based Fan speed control system has been designed and tested using IoT concepts. A computer program was developed to control the speed of the fan by using smart phone and is successfully implemented. The fan motor is operated and tested at different speed input value from the smart phone and the experimental results are noted. These results show that the user can switch on the fan, regulate its speed by using the smart phone. The user can operate this fan even he/she is in remote location. By using temperature and humidity sensors the speed control of fan becomes very easy so that it automatically changes its speed when the temperature is changed according to the user's comfort zone. All the values related to temperature, humidity and speed of the motor are also stored in the cloud storage, and the software can display these data for operators to check. The main advantage of this method of control is that the user can control the Computer Science & Information Technology (CS & IT) 45 speed of the fan by giving the set point value via the smart phone. Therefore, the fan motor can be efficiently controlled by using the Wi-Fi directly from anywhere and anytime in this world with proper internet connection.

[6] IoT Based Smart Exhaust Fan is well-designed, easy and affordable for home user. This project is an IoT based exhaust fan which can control and monitor the performance of the exhaust fan. This device can help the user to monitor, control and can reduce power consumption because it works automatically. The first objective have been achieved as a profile was design and build an IoT based smart exhaust fan which can turn ON and OFF automatically when temperature and air quality exceeded more than required level in a space. Next, the second objective is to measure and record data for data collection of temperature and air quality in a space also have been successfully done. Lastly, through this project, the objective of to optimize IoT implementation for the prototype is succeeded. The electronic

system to monitor and control the temperature and air quality of the exhaust fan have been obtained by the usage of various components such as Arduino Mega, DHT22, MQ-135, ESP8266-01, and fan. The acrylic house block was successfully built to set up the device or prototype into it so that to make experiment or testing process easier for result purpose. The electronic system was successfully integrated with the exhaust fan inside the acrylic house block which satisfied the objective of this project.

4.TECHNOLOGY

4.1 ABOUT IOT

The Internet of Things, or popularly referred to as IoT. IoT Technology on a wide spectrum can easily be described as the global integration of various physical devices digitally. These devices are connected with the internet, sensors and other hardware that can be controlled from anywhere.

4.1.1 Working of IoT

In a nutshell, the Internet of Things operates as follows:

- Sensors, for example, are part of the hardware that collects data about devices.
- The data collected by the sensors is then shared and combined with software via the cloud
- After that, the software analyses the data and sends it to users via an app or a website.
- Finally, it’s the user’s call upon how they would like to utilize the analyzed data.

4.1.2 Applications of IoT

There are enormous applications of IoT technologies, this is due to the fact that it can be adjusted in any technology that has the ability to provide information without the support of human intervention.

A few applications of IoT technologies include:

- Mobile Technologies such as Wearables & Fitness Trackers
- Health Care Industry
- Home Automation
- Smart Grids
- Self-Driving Transportation
- Remote Traffic Monitoring
- Smart Cities
- And many more.

4.2 ABOUT EXHAUST FAN

A smart exhaust fan, also known as a smart ventilation fan or smart bathroom fan, is a type of ventilation fan that incorporates smart technology for enhanced functionality and control. These fans are designed to improve indoor air quality by efficiently removing moisture, odours, and pollutants from bathrooms, kitchens, or other spaces.

4.2.1 Working of Exhaust Fan

A smart exhaust fan operates using advanced technology and connectivity to provide efficient ventilation and automated control. Here's a general overview of how a smart exhaust fan works:

Connectivity: Smart exhaust fans are equipped with wireless connectivity options such as Wi-Fi or Bluetooth, allowing them to connect to a home automation system or a dedicated mobile app.

Sensors and Controls: Smart exhaust fans are integrated with various sensors to monitor the air quality, humidity levels, and occupancy in the room. These sensors can include temperature sensors, humidity sensors, motion sensors, and even air quality sensors.

Automation and Control: The smart exhaust fan can be programmed or configured to operate automatically based on predefined conditions. For example, it can be set to turn on when the humidity level exceeds a certain threshold or when the presence of occupants is detected in the room.

Remote Control: With the help of a mobile app or home automation system, users can remotely control and monitor the smart exhaust fan. This allows them to turn it on or off, adjust the fan speed, or change settings from anywhere within the network range.

Integration with Smart Home Systems: Smart exhaust fans can integrate with other smart home devices and systems. For instance, they can work in conjunction with smart thermostats, allowing them to adjust their operation based on the temperature settings in the house.

Energy Efficiency: Smart exhaust fans are designed to optimize energy consumption. They can adjust the fan speed based on the detected air quality or humidity levels, ensuring that the fan operates only when necessary. This helps to reduce energy waste and lower electricity bills.

Notifications and Alerts: Smart exhaust fans can send notifications and alerts to the user's smartphone or connected devices. For instance, they can alert the user when the filters need to be cleaned or replaced, or when the fan malfunctions.

By utilizing these features, smart exhaust fans provide improved ventilation, energy efficiency, and convenience compared to traditional exhaust fans. They offer automated control, remote access, and integration with other smart devices to enhance the overall comfort and air quality in a room or building.

4.2.2 Applications of Exhaust Fan

Smart exhaust fans find applications in various settings where ventilation is crucial. Here are some common applications of smart exhaust fans:

Residential Buildings: Smart exhaust fans are commonly used in homes and apartments to improve indoor air quality and regulate humidity levels. They are particularly beneficial in areas like kitchens, bathrooms, and laundry rooms, where moisture and odors can accumulate. The smart features allow homeowners to automate the fan operation and control it remotely.

Commercial Buildings: Offices, restaurants, hotels, and other commercial establishments benefit from smart exhaust fans. These fans help remove odors, pollutants, and excess moisture, creating a comfortable and healthy environment for employees and customers. The ability to integrate with building management systems allows for centralized control and monitoring of multiple fans across the facility.

Industrial Settings: Industrial facilities, such as manufacturing plants, warehouses, and workshops, often require powerful ventilation systems to remove heat, fumes, and airborne particles. Smart exhaust fans equipped with sensors can automatically adjust the fan speed and operation based on air quality and temperature conditions, ensuring optimal ventilation and worker safety.

Greenhouses and Agricultural Settings: Smart exhaust fans play a vital role in greenhouse ventilation. They help regulate temperature, humidity, and airflow to create an optimal growing environment for plants. By connecting to weather sensors and climate control systems, smart exhaust fans can adjust their operation based on environmental conditions to maintain the desired growing conditions.

Server Rooms and Data Centers: Cooling and ventilation are crucial in server rooms and data centers to prevent overheating and ensure optimal performance of the equipment. Smart exhaust fans with temperature and humidity sensors can automatically activate when the temperature exceeds a threshold, expelling hot air and maintaining a stable environment for the servers.

Public Facilities: Smart exhaust fans can be used in public spaces such as restrooms, parking garages, fitness centers, and shopping malls. These fans help remove unpleasant odors, control humidity, and provide fresh air circulation, enhancing the overall comfort and cleanliness of the facilities.

The applications of smart exhaust fans are diverse, and their ability to provide efficient ventilation, automated control, and integration with other systems makes them valuable in various settings where air quality, moisture control, and temperature regulation are essential.

4.3 ARDUINO UNO

The Arduino UNO is a programmable microcontroller board that is open-source and can be utilized in a wide range of electronic projects. It is powered by the Atmega328P microprocessor. This board is capable of communication with other Arduino and Raspberry Pi boards, as well as controlling motors, LEDs, servos, displays, sensors, and various other Arduino extensions.

4.3.1 FEATURES OF ARDUINO UNO:

• Microcontroller: The Arduino UNO utilizes the Atmega328P microcontroller.

• Operating Voltage: The board operates at 5V.

• Input Voltage (recommended): The recommended input voltage ranges from 7V to 12V.

• Input Voltage (limits): The input voltage limits range from 6V to 20V.

• Digital I/O Pins: It offers a total of 14 digital I/O pins, of which 6 provide PWM (Pulse Width Modulation) output.

• Analog Input Pins: The Arduino UNO has 6 analog input pins.

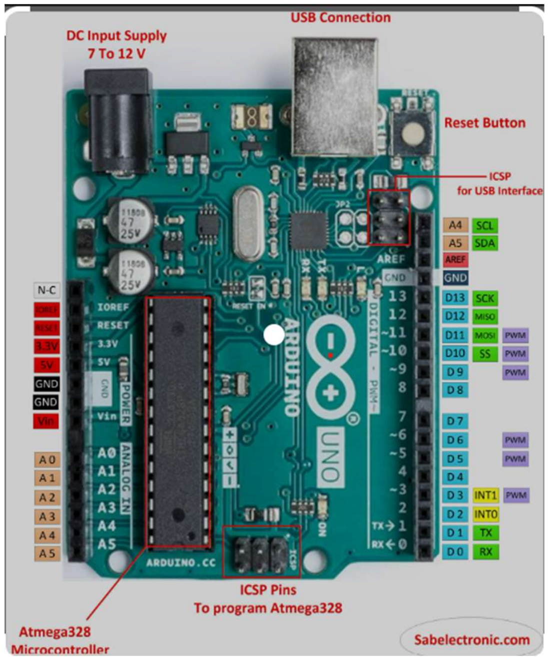
• DC Current per I/O Pin: Each I/O pin can handle a direct current of up to 20 mA.

• DC Current for 3.3V Pin: The 3.3V pin supports a direct current of up to 50 mA.

• Flash Memory: It has 32 KB of flash memory, with 0.5 KB being used by the bootloader.

• SRAM: The Arduino UNO is equipped with 2 KB of SRAM (Static Random-Access Memory).

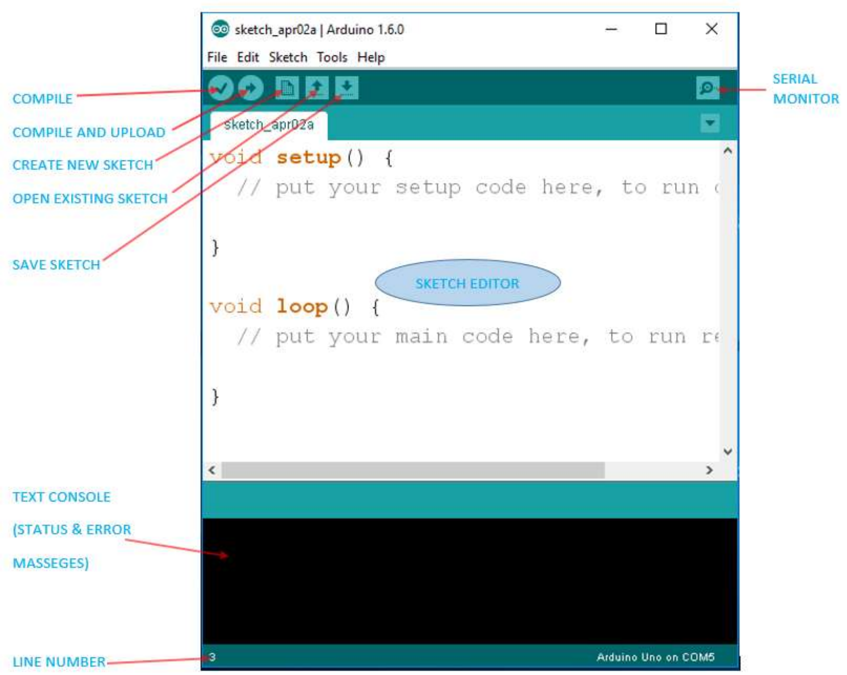
• EEPROM: The EEPROM (Electrically Erasable Programmable Read-Only Memory) has a Capacity of 1 KB on the Atmega328P microcontroller.



Arduino UNO Pinout

4.4 ARDUINO IDE

The Arduino IDE is an open-source software, which is used to write and upload code to the Arduino boards. The program or code written in the Arduino Ide is called "Sketching", the file is saved with the extension '.ino'.



Arduino IDE Sketch Blueprint

Every Arduino sketch program has two common functions setup() and loop(), which are called automatically in the background. The code to be executed is written inside the curly braces within these functions. The 'void setup()' includes the initial part of the code, which is executed only once. It is called as the preparation block. The 'void loop()' includes the statements, which are executed repeatedly. It is called the execution block.

4.5 ESP32 MODULE

The ESP32 is a versatile microcontroller module developed by Espressif Systems. It offers a powerful combination of features and capabilities, making it a popular choice for a wide range of applications.

4.5.1 FEATURES OF ESP32 Module:

• Microcontroller: The ESP32 module is equipped with a dual-core Xtensa LX6 microcontroller.

• Wireless Connectivity: It supports Wi-Fi 802.11 b/g/n/e/I, enabling wireless connection to local networks and the internet. Additionally, it also supports Bluetooth Classic and Bluetooth Low Energy (BLE), making it suitable for various IoT applications.

• GPIO Pins: With 34 physical GPIO pins, each pin can function as a general-purpose I/O or be linked to an internal peripheral signal.

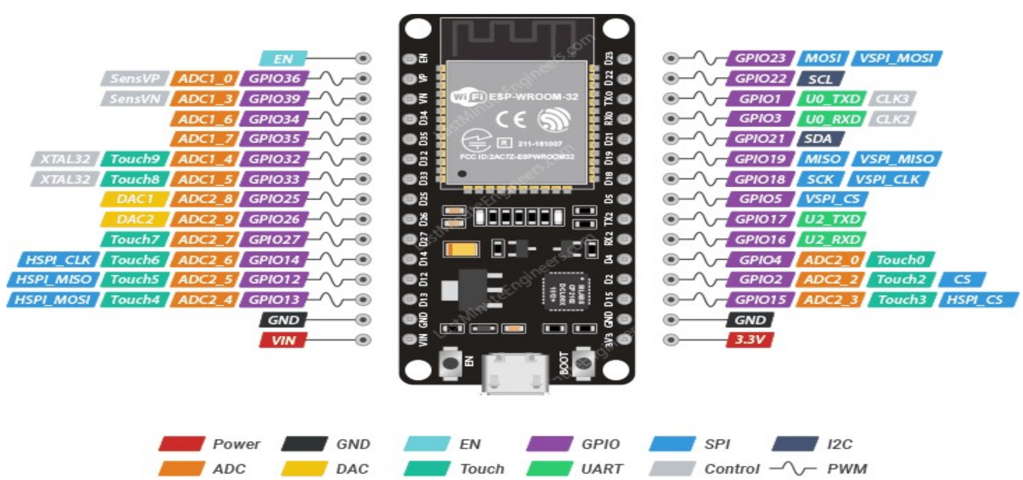
• Analog Inputs: These inputs allow the reading of analog signals from sensors like temperature and light sensors. Voltage levels ranging between 0V and 3.3V can be measured and assigned a value between 0 and 4095, with 0V corresponding to 0 and 3.3V corresponding to 4095.

• Memory: The ESP32 module is equipped with a substantial amount of onboard memory, encompassing both RAM and Flash memory. This facilitates code execution, data storage during runtime, and overall system performance.

• Additional Features: Alongside the mentioned features, the ESP32 module provides PWM (Pulse Width Modulation) for controlling analog outputs, I2C and SPI interfaces for device communication, UART for serial communication, and more.

• Power Management: The module encompasses power management capabilities, including various power modes and sleep modes. This allows for efficient power consumption and extends battery life in applications powered by batteries.

ESP32 Wi-Fi Module



ESP32 PIN DIAGRAM

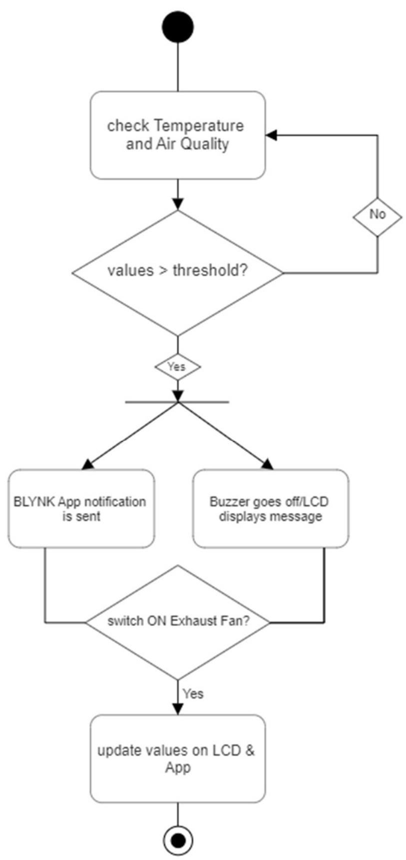
5.DESIGN REQUIREMENT ENGINEERING

5.1 UML DIAGRAMS

UML, which stands for Unified Modelling Language, is a widely recognized language used for specifying, visualizing, constructing, and documenting software system artifacts. Unlike popular programming languages such as C++, Java, and COBOL, UML serves as a diagramming language specifically designed for creating software blueprints. The primary objective of UML is to establish a versatile 16 modelling language that can be easily understood and applied by all modelers.

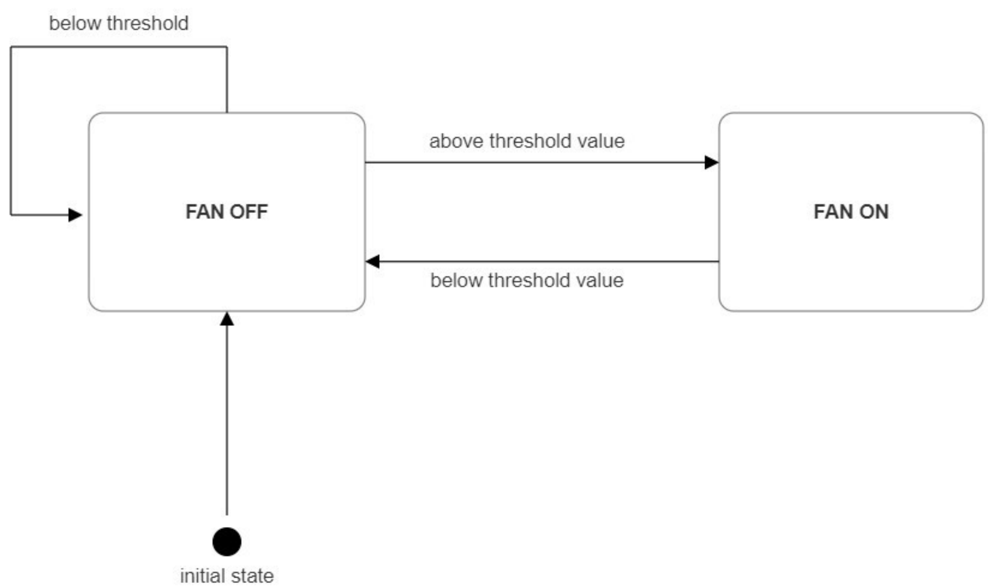
5.1.1 Activity Diagram

An activity diagram is a significant behavioural diagram utilized within UML diagrams to illustrate the dynamic characteristics of a system. It serves as a more intricate version of a flowchart, providing a visual representation of the information flow between various activities. Activity diagrams effectively portray the dynamic behaviour of a system.



5.1.2 State-Chart Diagram

The State-Chart diagram represents the various states that an object can undergo throughout its lifespan. It is essential for the described system to have a limited number of states, and the control is transferred between these states. The diagram also showcases the flow of data from one state to another, revealing how the object's state evolves over time. Additionally, it serves as a logical representation of the model or project's functionality, incorporating pathways, loops, conditions, and other elements.



For instance, the 12V DC Exhaust Fan has two states:

1. FAN OFF
2. FAN ON.

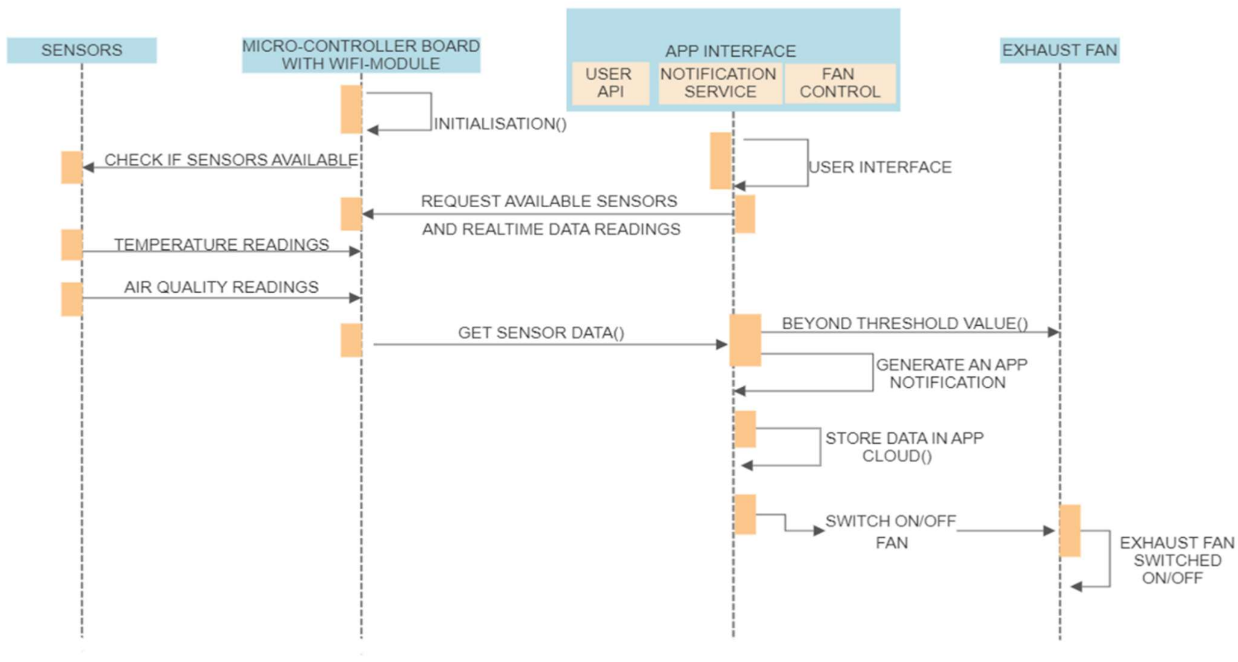
FAN OFF State: This is the Exhaust Fans' initial state. In other words, the fan remains switched off when the air quality is below threshold value.

FAN ON State: This is the Exhaust Fans' acquired state. In other words, when the air quality value exceeds the threshold, the fan turns on and continues to operate until the air quality value decreases.

5.1.3 Sequence Diagram

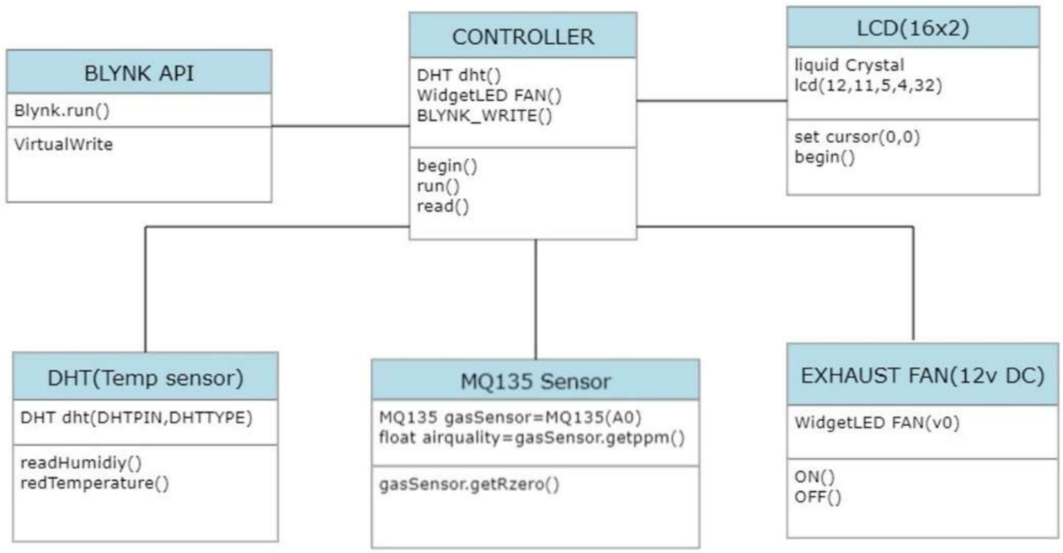
A sequence diagram, categorized as an interaction diagram in UML, illustrates the sequence and order of interactions among system processes.

It utilizes a message sequence chart to depict these interactions. Sequence diagrams are also known as event diagrams or timing diagrams.



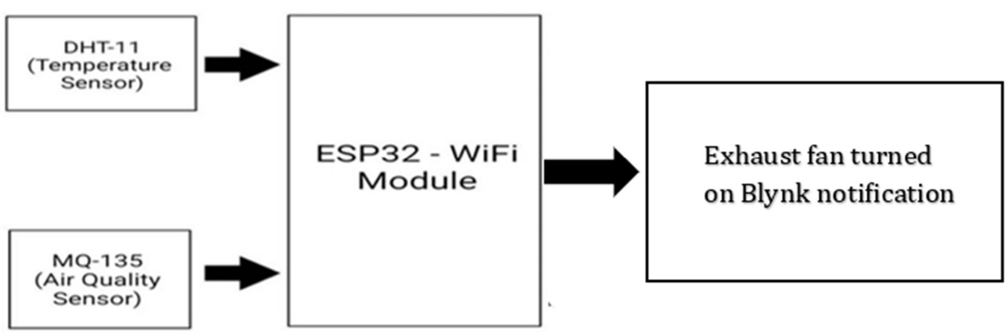
5.1.4 Class Diagram

The class diagram is a static diagram that provides a static view of an application. It serves multiple purposes, including visualization, description, and documentation of various system components. Moreover, it aids in the creation of executable software programs. The class diagram showcases the characteristics and behaviors of classes, as well as the constraints imposed on the system. Due to their direct translation to object-oriented languages, class diagrams are widely used in the design of object-oriented systems. They depict a collection of classes, interfaces, relationships, collaborations, and constraints, and are often referred to as structural diagrams. Class diagrams play a critical role in creating component and deployment diagrams, as well as in supporting forward and backward engineering processes.



5.1.5 Architecture

An architecture diagram, based on UML, is employed by system designers and developers to illustrate the high-level structure of a system or application. It serves to verify that the system meets user requirements and can also describe design patterns present in the system. The architecture diagram provides a means to abstract the overall framework of a software system, defining boundaries, relationships, and constraints between components. It offers a comprehensive view of the physical deployment and evolution plan of the software system. Developers and designers find this diagram highly useful in their work.



6.IMPLEMENTATION

The implementation of the smart exhaust fan with air quality monitoring system involves integrating various components and technologies to create a cohesive and efficient system. The system architecture typically consists of the following components:

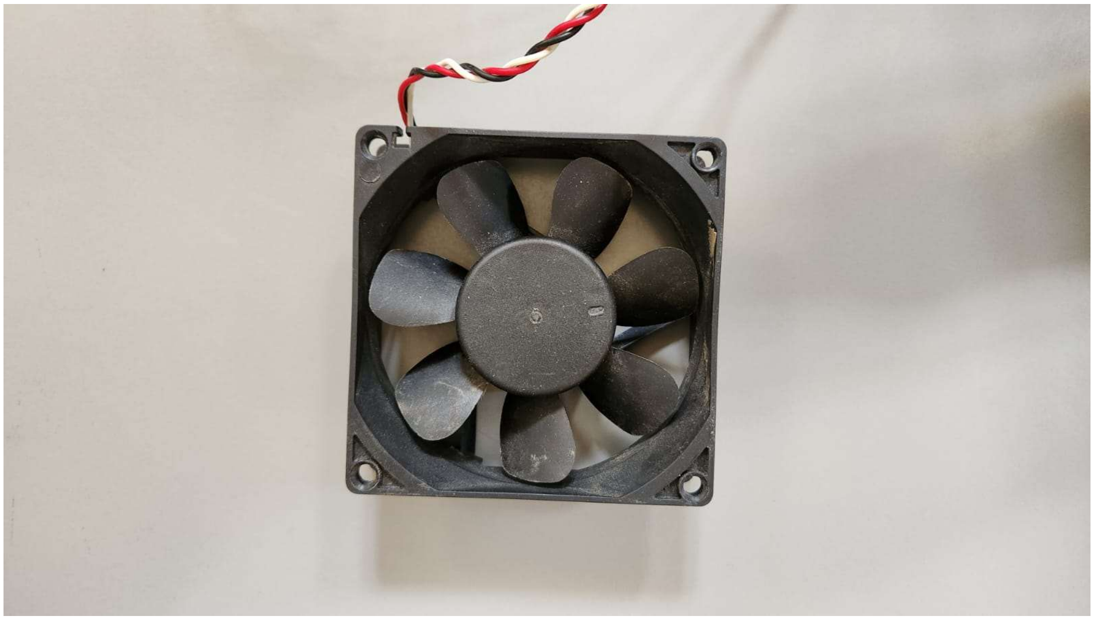
1. **Smart Exhaust Fan**: This is the core component of the system and is responsible for expelling stale or polluted air from the indoor space. The smart exhaust fan is equipped with connectivity options, sensors, and automation capabilities to enable intelligent and efficient operation.

2. **Air Quality Sensors**: These sensors are used to collect real-time data on various parameters related to indoor air quality. Common sensors include temperature sensors, humidity sensors, CO2 sensors, VOC sensors, and particulate matter sensors. The data collected by these sensors provides insights into the current air quality conditions.

3. **IoT Platform**: The IoT platform serves as the central hub for data transmission, storage, analysis, and control. It enables seamless communication between the smart exhaust fan, air quality sensors, and user interface. The IoT platform processes the data received from the sensors, applies predefined rules or algorithms, and controls the operation of the smart exhaust fan based on the analyzed data.

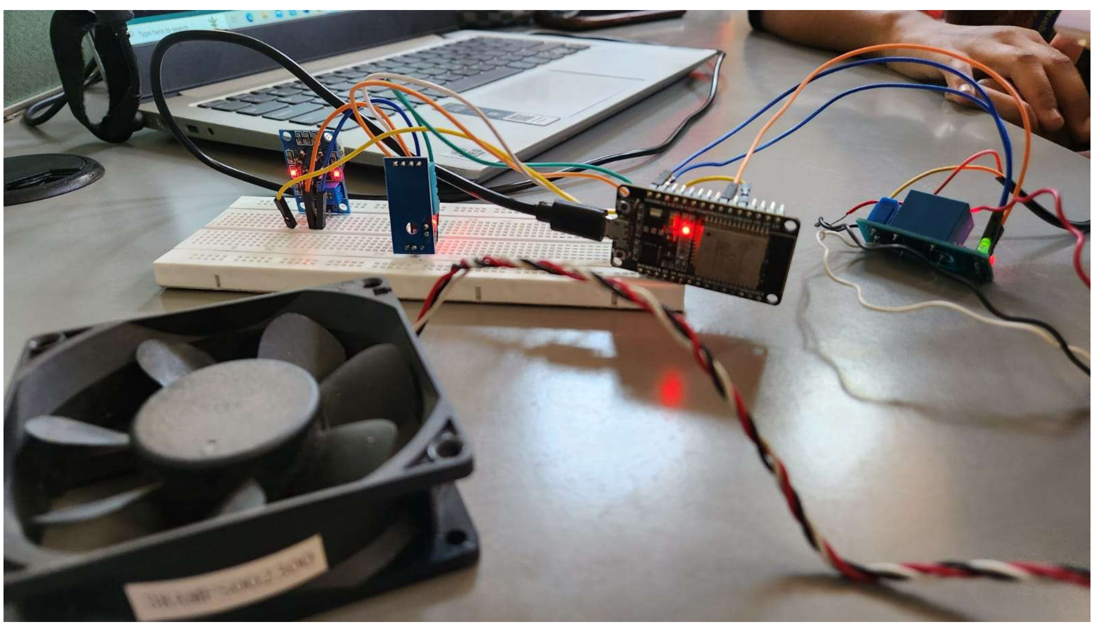
4. **User Interface**: The user interface allows users to interact with the system, monitor the air quality status, and control the smart exhaust fan. The user interface can be in the form of a mobile application, a web-based dashboard, or a physical control panel. It provides real-time information about the air quality parameters, alerts users when air quality exceeds predefined thresholds, and allows for manual control of the smart exhaust fan.

5. **Connectivity**: Wireless connectivity technologies such as Wi-Fi, Bluetooth, or Zigbee are used to establish communication between the smart exhaust fan, air quality sensors, IoT platform, and user interface. This enables remote access, control, and monitoring of the system.



Exhaust Fan OFF

6.1 Circuit of the system:



6.2 Circuit Pin-Mapping:

5v RELAY MODULE	ESP32 PIN
signal pin	33
5v vcc	5v
gnd	gnd

5v Relay Module	External Connections
NO	positive exhaust fan
CC	positive 12v power source
NC	high

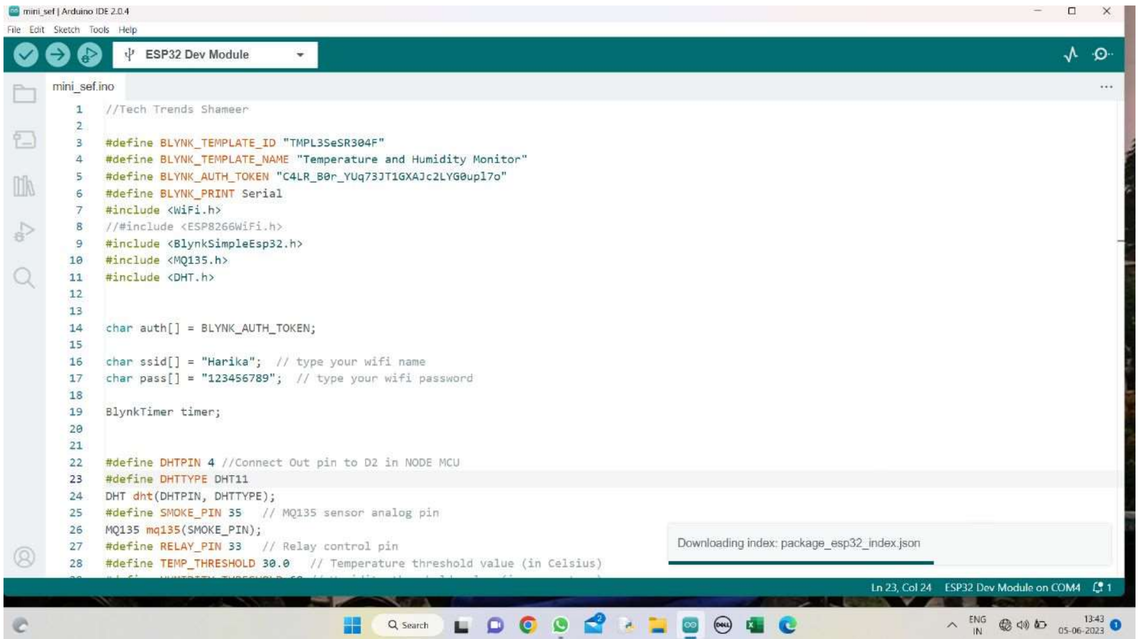
Dht 11	Esp 32 pin
vcc	3.3v
gnd	gnd
data pin	4

MQ135	ESP32 pin
vcc	5v
gnd	gnd
data pin	35

6.3 Installing the Required Libraries:

In this project we have used the following libraries:

- WiFi.h
- BlynkSimpleESP32.h
- MQ135.h
- DHT.h



Library – Wifi.h:

The WiFi.h library allows Arduino boards to connect to Wi-Fi networks and perform various operations such as connecting to a network, checking the connection status, configuring network settings, and sending/receiving data over Wi-Fi.

Library – BlynkSimpleESP32.h:

The BlynkSimpleESP32.h library is a popular library used in Arduino programming for connecting ESP32 boards to the Blynk IoT platform. It provides a set of functions and classes that simplify the process of integrating ESP32 devices with the Blynk cloud infrastructure.

Library – MQ135.h:

The MQ135.h library is designed to facilitate the usage of the MQ135 gas sensor module with Arduino boards. This library provides functions and methods to read values from the MQ135 sensor and interpret them to measure the air quality in terms of the presence of hazardous gases like ammonia, carbon dioxide, benzene, and others.

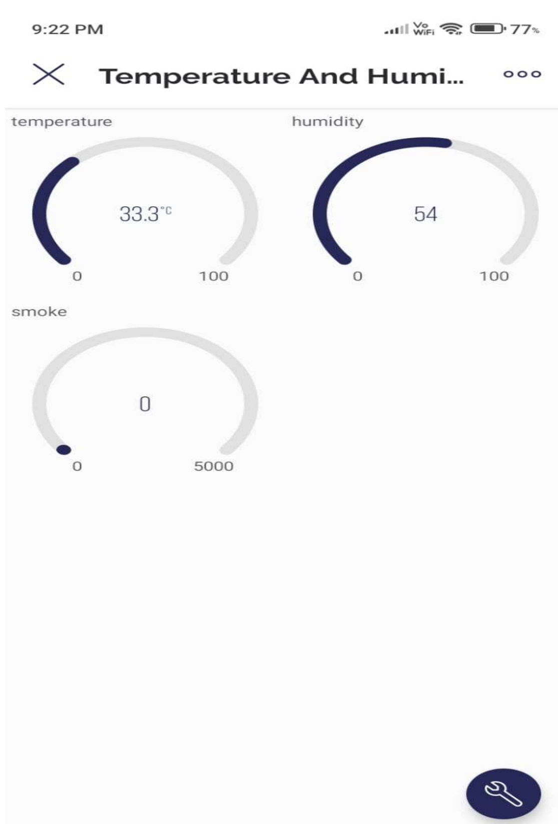
Library – DHT.h:

The DHT.h library is a widely used library in Arduino projects for interfacing with DHT series temperature and humidity sensors. This library provides functions to read temperature and humidity values from the DHT sensor and offers convenient methods for data retrieval and manipulation.

6.4 Blynk Mobile Application:

Overview of Blynk IoT:

Blynk is a popular IoT platform that provides a comprehensive set of tools and resources to develop IoT projects quickly and efficiently. It offers a drag-and-drop mobile app builder, a cloud-based backend infrastructure, and support for a wide range of hardware platforms. Blynk simplifies the process of connecting and controlling devices over the internet, enabling seamless communication between our smart exhaust fan and the connected mobile application.



Features and Capabilities:

Blynk offers several key features that make it an ideal choice for our smart exhaust fan project:

a. **Mobile Application**: Blynk provides a customizable mobile application that serves as the user interface for controlling and monitoring our smart exhaust fan. We can design the app's layout, add buttons, sliders, graphs, and other interactive elements to create an intuitive and visually appealing control panel.

b. **Drag-and-Drop Interface**: Blynk offers a user-friendly drag-and-drop interface, allowing us to easily create a customized user interface in the mobile application. We can choose from a wide range of widgets, such as buttons, switches, sliders, and gauges, and configure their behavior to control the fan's operations.

c**. Data Visualization**: Blynk provides built-in widgets for visualizing sensor data and system parameters. We can display real-time data such as fan speed, temperature, humidity, and air quality readings in the form of graphs, gauges, or charts, providing users with valuable insights into the fan's performance.

d. **Advanced Automation**: With Blynk, we can implement advanced automation and logic-based controls for our smart exhaust fan. We can define rules and triggers based on various conditions, such as time of day, sensor readings, or user inputs, to automate fan operations. This includes features like scheduled operation, threshold-based control, and conditional logic.

Integration Steps:

To integrate Blynk IoT with our smart exhaust fan system, the following steps need to be followed:

a**. Setting up a Blynk Account**: Begin by creating an account on the Blynk platform through their website or mobile application. This account will serve as the central hub for managing our project.

b**. Creating a Blynk Project**: Once the account is created, we can create a new Blynk project. This involves selecting the appropriate hardware platform and configuring the desired widgets and controls for our smart exhaust fan.

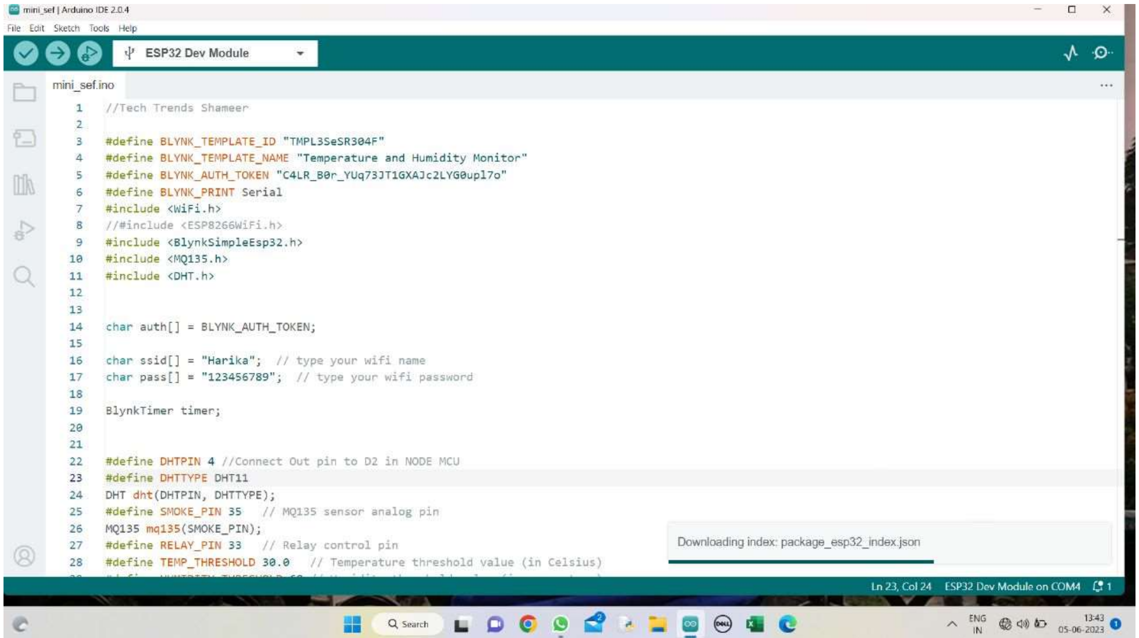
c. **Obtaining Authentication Token**: To establish communication between our hardware device and the Blynk platform, we need to obtain an authentication token. This token acts as a unique identifier for our device and ensures secure data transfer.

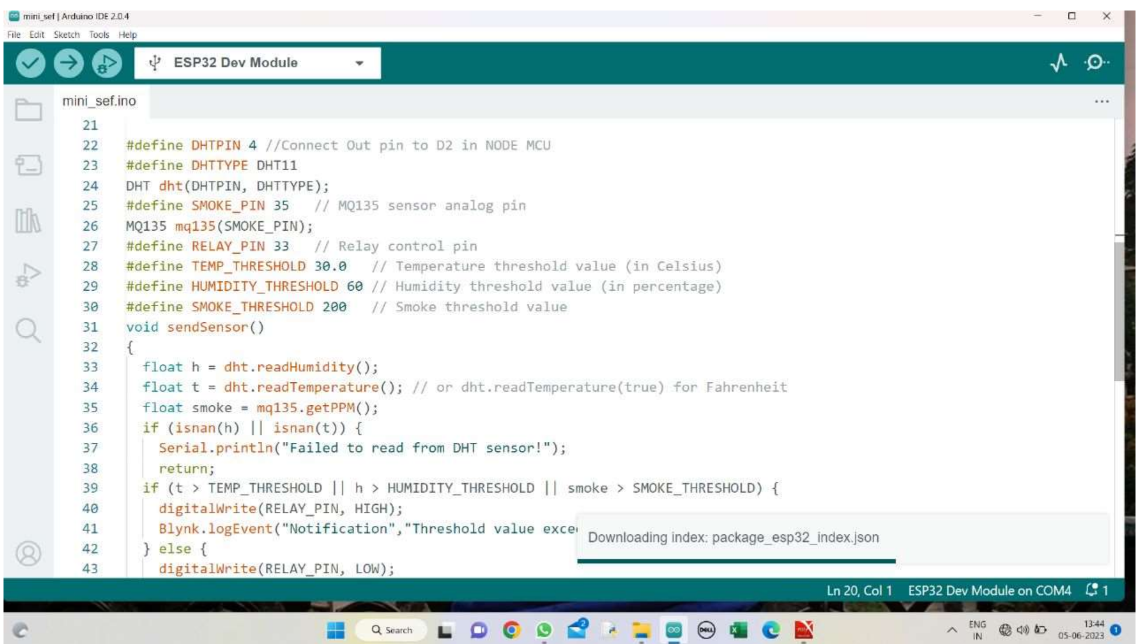
d. **Hardware Configuration**: Configure our smart exhaust fan device to connect to the Blynk platform. This typically involves programming the device with the Blynk library or utilizing Blynk-compatible development boards, such as Arduino or Raspberry Pi.

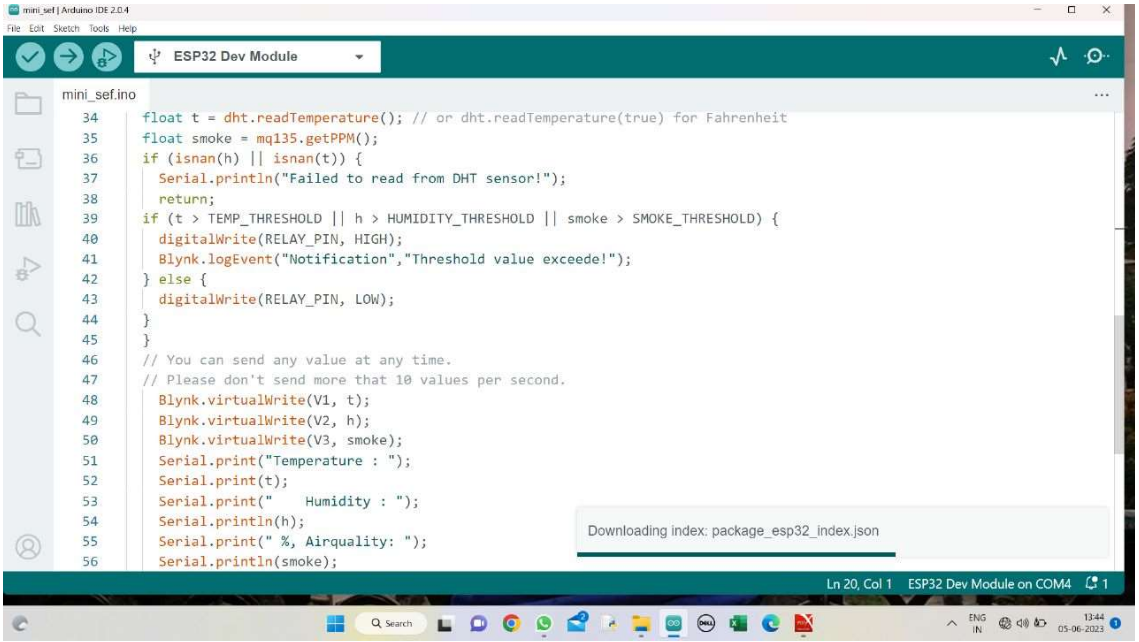
e. **Mobile Application Setup**: Install the Blynk mobile application on a smartphone or tablet. Use the authentication token obtained earlier to link the application to our project. This enables the mobile app to communicate with the smart exhaust fan device.

f. **Widget Configuration**: Configure the widgets

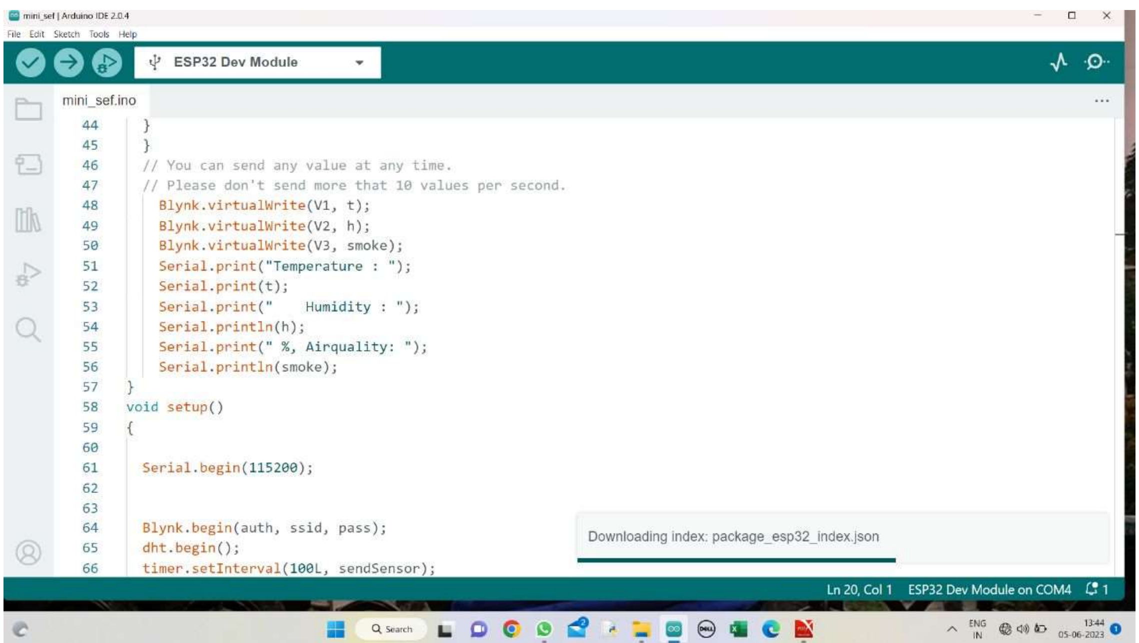
6.5 SAMPLE CODE:







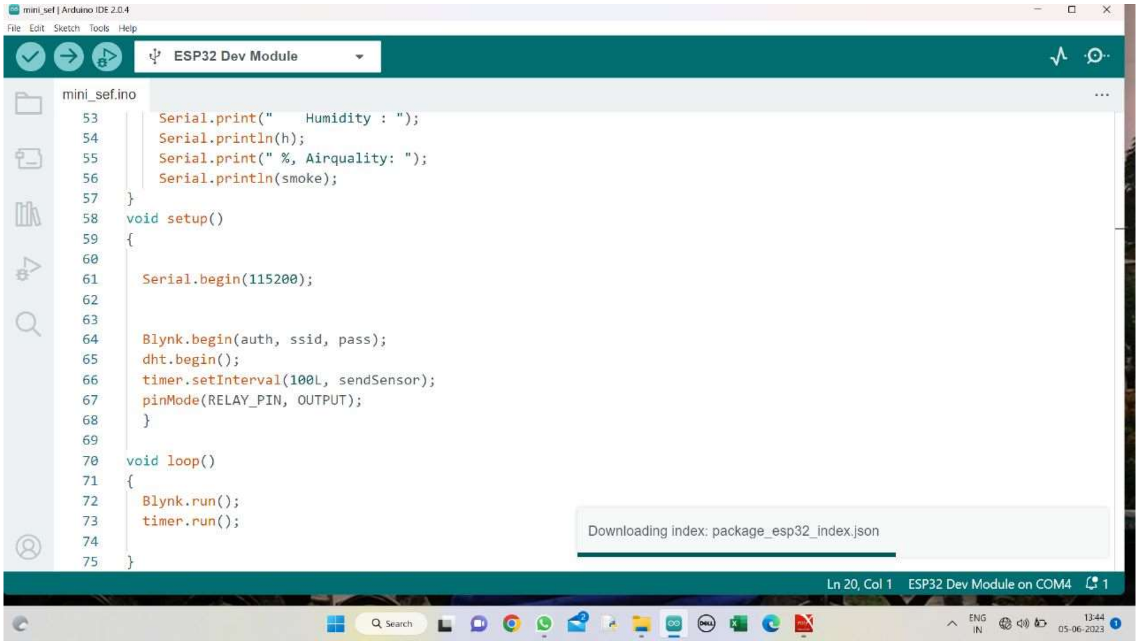
```
  float t = dht.readTemperature(); // or dht.readTemperature(true) for Fahrenheit
  float smoke = mq135.getPPM();
  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  if (t > TEMP_THRESHOLD || h > HUMIDITY_THRESHOLD || smoke > SMOKE_THRESHOLD) {
    digitalWrite(RELAY_PIN, HIGH);
    Blynk.logEvent("Notification","Threshold value exceede!");
  } else {
    digitalWrite(RELAY_PIN, LOW);
  }
  }
  // You can send any value at any time.
  // Please don't send more that 10 values per second.
    Blynk.virtualWrite(V1, t);
    Blynk.virtualWrite(V2, h);
    Blynk.virtualWrite(V3, smoke);
    Serial.print("Temperature : ");
    Serial.print(t);
    Serial.print("    Humidity : ");
    Serial.println(h);
    Serial.print(" %, Airquality: ");
    Serial.println(smoke);
```



```
  }
  }
  // You can send any value at any time.
  // Please don't send more that 10 values per second.
    Blynk.virtualWrite(V1, t);
    Blynk.virtualWrite(V2, h);
    Blynk.virtualWrite(V3, smoke);
    Serial.print("Temperature : ");
    Serial.print(t);
    Serial.print("    Humidity : ");
    Serial.println(h);
    Serial.print(" %, Airquality: ");
    Serial.println(smoke);
}
void setup()
{

  Serial.begin(115200);


  Blynk.begin(auth, ssid, pass);
  dht.begin();
  timer.setInterval(100L, sendSensor);
```


mini_sef | Arduino IDE 2.0.4
File Edit Sketch Tools Help
ESP32 Dev Module
mini_sef.ino
  Serial.print("   Humidity : ");
  Serial.println(h);
  Serial.print(" %, Airquality: ");
  Serial.println(smoke);
}
void setup()
{

 Serial.begin(115200);

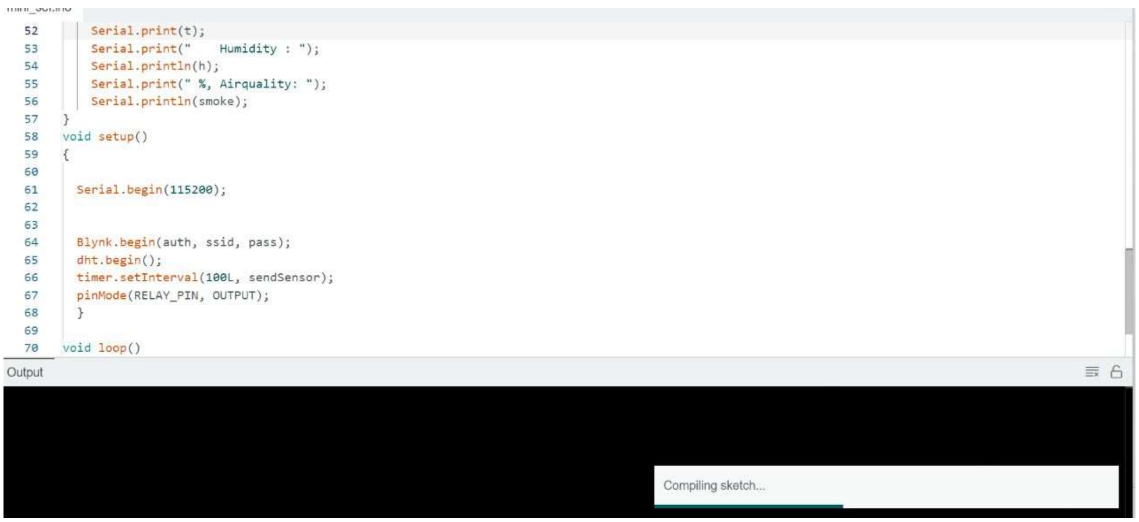
 Blynk.begin(auth, ssid, pass);
 dht.begin();
 timer.setInterval(100L, sendSensor);
 pinMode(RELAY_PIN, OUTPUT);
 }

void loop()
{
 Blynk.run();
 timer.run();

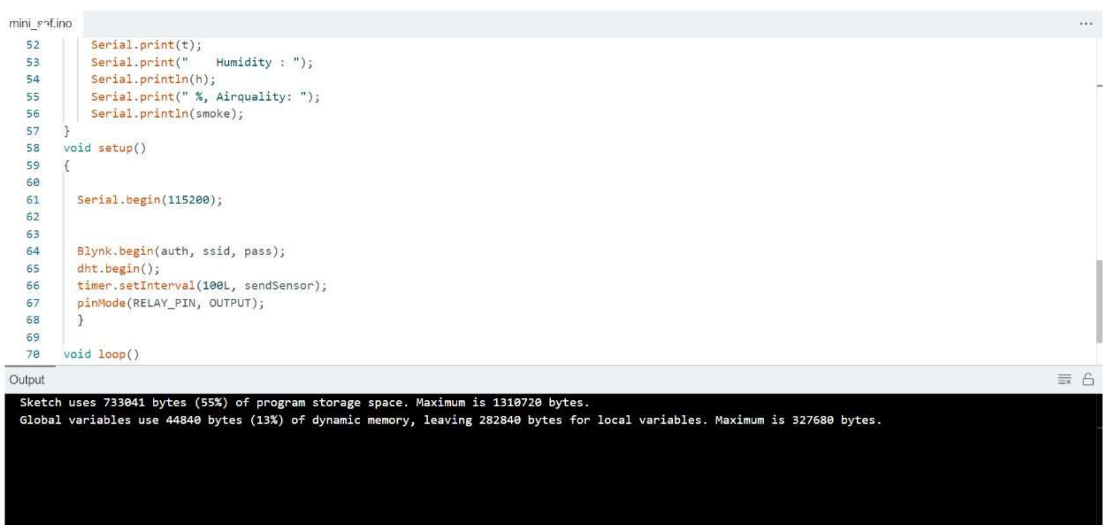
}
Downloading index: package_esp32_index.json
Ln 20, Col 1 ESP32 Dev Module on COM4

7.RESULTS

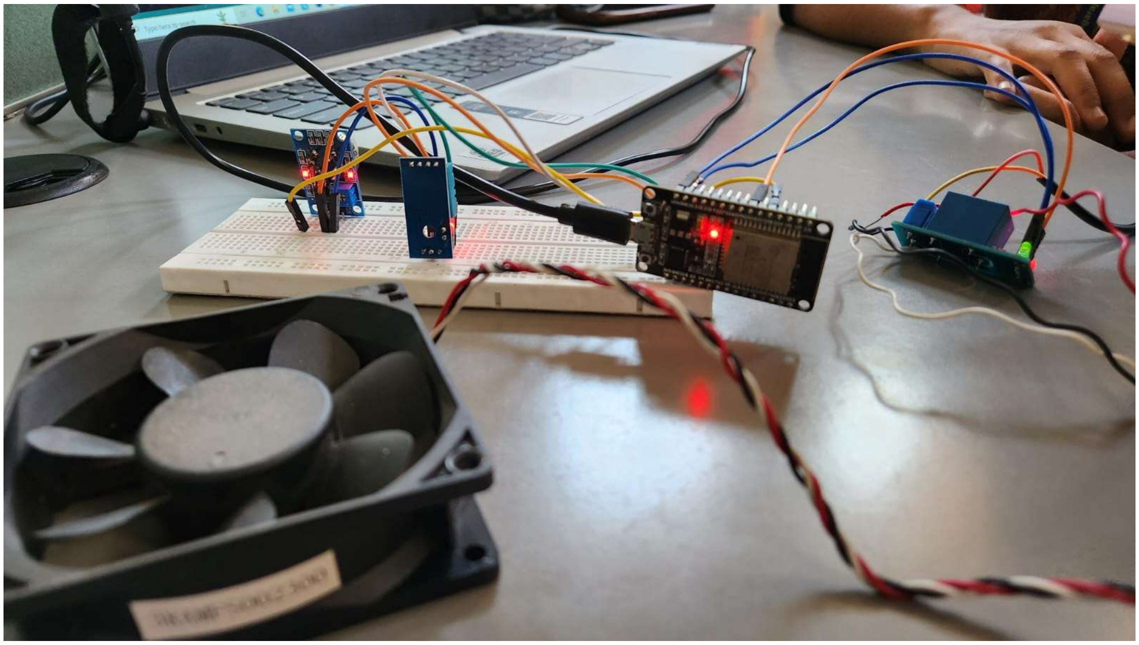
Code Compiled:



Code Uploaded:



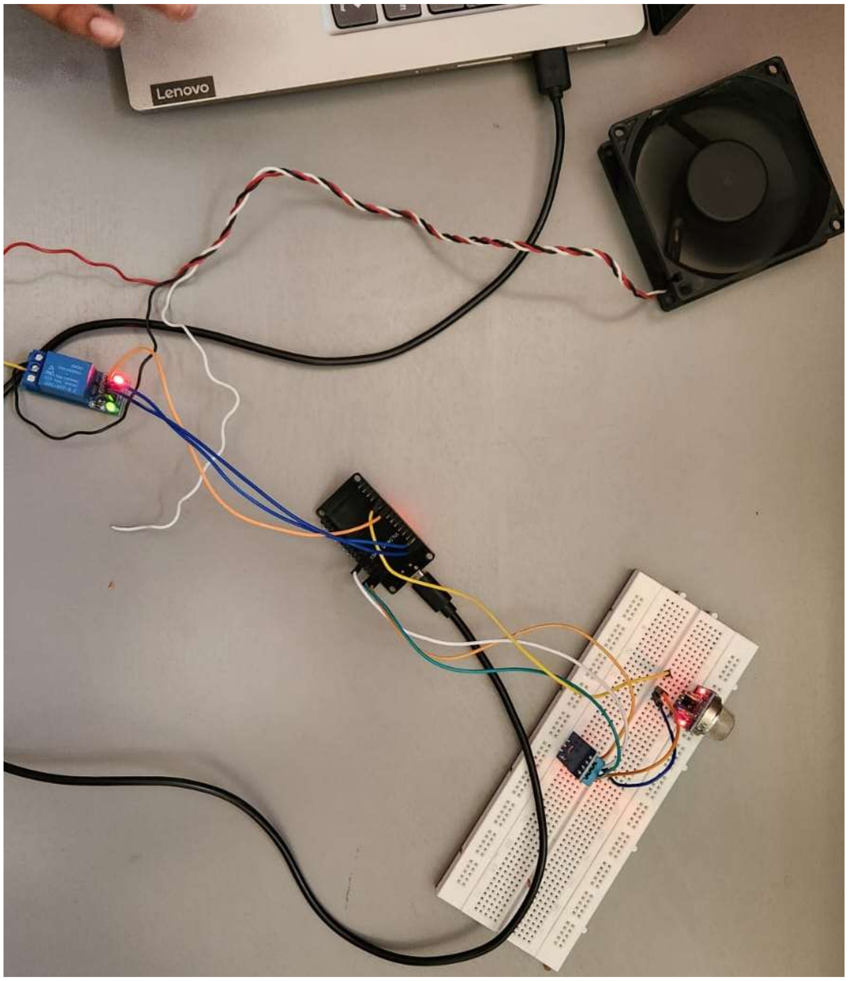
Response to low air quality value:



Exhaust Fan shows no response.

Here, although power supply is provided to the relay, the fan is not operating as neither of the temperature, humidity or air quality values have exceeded the threshold values.

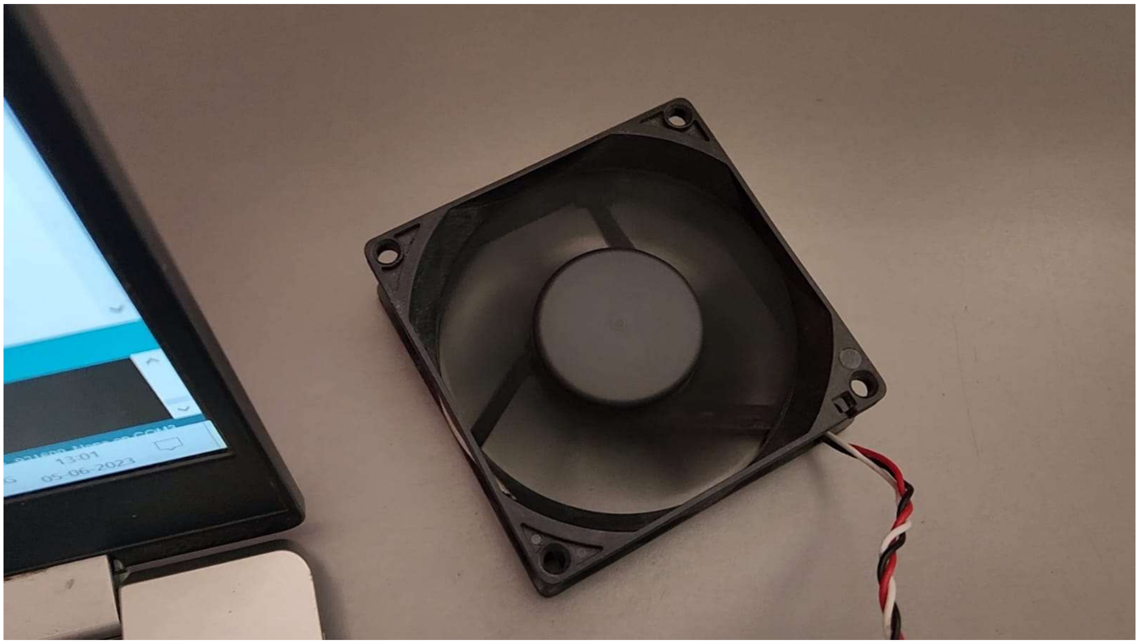
Response to exceeded air quality value:



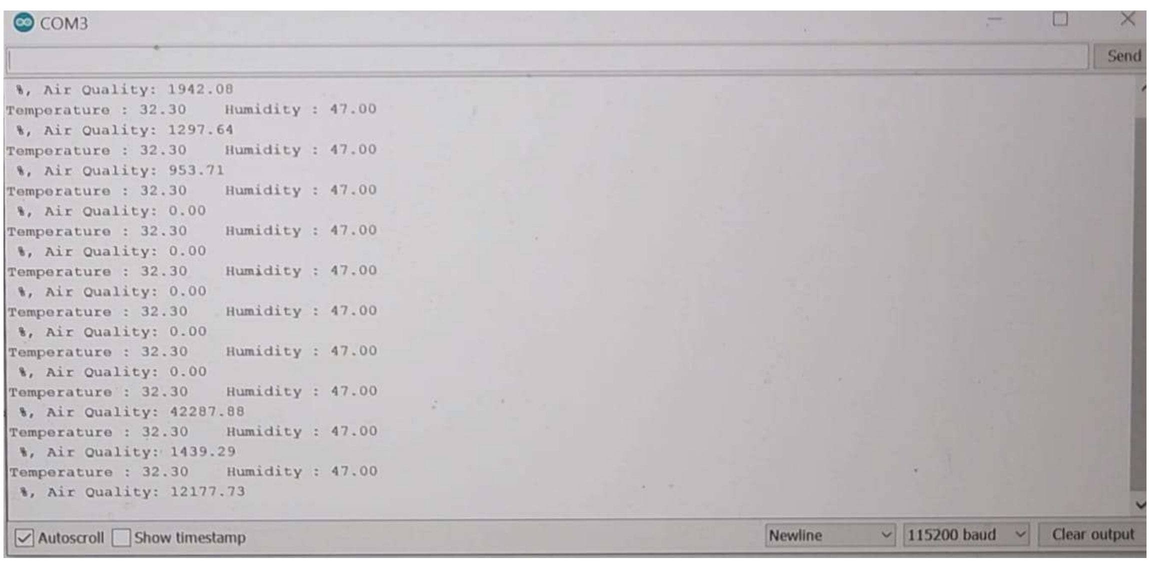
Exhaust Fan shows positive response.

Here, the power is supplied to the relay and the exhaust fan begins to operate with increase in temperature/humidity/air quality value.

Operating Exhaust Fan:



Serial Monitor values:



8.CONCLUSION

This Smart Exhaust Fan system is a very cheap and affordable design that plays a crucial role in keeping track of health. This design is compact and can be implemented and installed anywhere, especially in household areas. Here, the relay supplies the exhaust fan with power supply, after reading the respective temperature, humidity and air quality values, if either of the values exceed certain threshold value, only then will the exhaust fan will be switched on. It will operate until the value decreases or is balanced. It will provide better and healthier environment to stay in.

9.FUTURE ENHANCEMENTS

Smart Scheduling: Allow users to set schedules for the fan to turn on/off at specific times or under certain conditions. This can help optimize energy consumption and ensure the fan operates when needed.

Voice Control: Integrate voice assistant technologies like Amazon Alexa or Google Assistant to enable users to control the fan using voice commands. This can provide a hands-free and convenient user experience.

Energy Efficiency Optimization: Implement algorithms or machine learning techniques to analyze the data collected by the fan and provide recommendations to optimize energy usage. This could include suggestions on when to run the fan at lower speeds or turn it off based on usage patterns.

Mobile Alerts and Notifications: Set up push notifications or alerts in the mobile application to inform users about air quality changes, maintenance reminders, or filter replacement notifications. This ensures users stay informed about the performance and maintenance requirements of the fan.

Integration with Smart Home Systems: Enable integration with popular smart home platforms like Apple HomeKit, Google Home, or Samsung SmartThings. This allows users to control the fan alongside other smart devices in their home ecosystem.

Data Analytics and Insights: Provide users with visualizations and analytics based on the data collected by the fan. This can include historical trends of air quality, energy consumption patterns, or the impact of fan usage on indoor environment conditions.

Geolocation-based Automation: Use geolocation data from the mobile application to automatically adjust the fan's settings based on the user's location. For example, the fan could turn on when the user is approaching home or turn off when the user leaves.

Integration with Air Purifiers: If applicable, integrate the smart exhaust fan with air purifiers or filtration systems. This allows for coordinated control and enhances the overall air quality management capabilities of the system.

10.BIBLIOGRAPHY

[1] Syifaul Fuada, Maulana Yusuf Fathany, Trio Adiono,
Khilda Afifah, "Controlling mini exhaust fan through an android based smartphone for IoT based smartphone system", TEM – August 2021.https://doi.org/10.18421/TEM103-37

[2] Raeesa, Navashree, Relin Jane Mascarenhas, "Gas detection system using Arduino", International Journal of Engineering Research & Technology (IJERT)-2021.https://www.ijert.org/gas-detection-system-using-arduino

[3] "Gas Leakage Detection and Prevention using IOT",
Authors- Sanjoy Das, Sahana S, Soujanya K, Swathi M C, International Journal of Scientific Research & Engineering Trends. – 2020.

[4] Shital Imade, Priyanka Rajmanes, Aishwarya Gavali, "Gas Leakage Detection and Smart Alerting Systen using IOT",
https://www.google.com/url?sa=t&source=web&rct=j&url=https://www.pramanaresearch.org/gallery/22.%2520feb%2520ijirs%2520-%2520d539.pdf&ved=2ahUKEwjS8syA967_AhU-cmwGHcoVAdMQFnoECA0QAQ&usg=AOvVaw3LGTzW44-Kb5UDGM19zwK-
International Journal of Innovative Research & Studies – 2018.

[5] Mehran Ektesabi,Saman Asghari Gorji Suchart Yammen, "IoT-based Home Appliances System (Smart Fan)",
Doi-10.5121/csit.2018.81604, 8th International Conference on Artificial Intelligence, Soft Computing and Applications- 2018.

[6] Navindran A/L Alagari1, Jais Lias, "IoT Based Smart Exhaust Fan" ,
http://penerbit-uthm.edu.my/periodicals/index.php/eee/article/view/4190
Evolution in Electrical and Electronics Engineering.