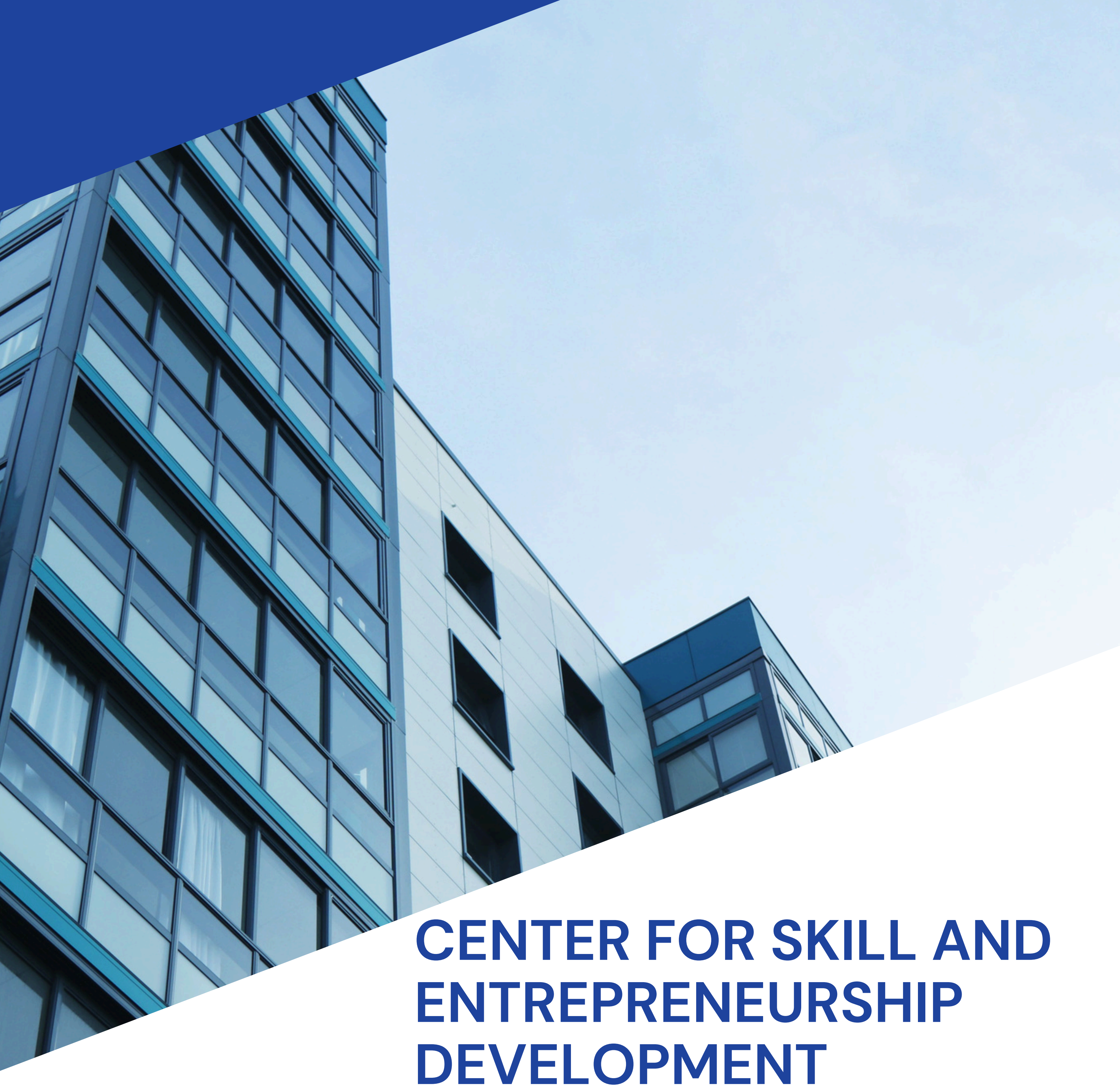
Project Report

Smart Air Quality Monitoring System

Power To Design The Desired



CENTER FOR SKILL AND ENTREPRENEURSHIP DEVELOPMENT (CSED)

INDUSTRIAL INTERNET OF THINGS

PROGRAM CODE: IIOT-1

PROGRAM NAME: FUNDAMENTALS OF IOT

PRJECT NAME :SMART AIR QUALITY MONITORING SYSTEM

DCS®

Power To Design The Desired

Team

Aditya Patel (Team Lead)
Ankush Singh
Dev Saraswat
Nitesh Kumar Gangwar
Shruti Yadav
Shreya Rathore
Satvik
Akshay
Divyanshi
Anurag Bharti
Lokesh
Ashpreet
Shaurya
Prateek Pal
Pranjal

Mentor

Mr.Rahul Chaple

Table of Contents

Table of Contents

Introduction

Keeping an eye on air quality is crucial these days due to increasing pollution and its effects on our health. This project uses an ESP32 microcontroller, along with a DHT11 and SGP30 sensors, to measure temperature, humidity, and CO_2 levels.
The ESP32 microcontroller is like the brain of this system. It handles all the processing and has built-in Wi-Fi, so you can easily check the air quality data remotely. The DHT11 sensor helps measure the temperature and humidity around you. The SGP30 sensor focuses on detecting CO_2 levels, which is important for understanding air quality.
Putting these pieces together, this system continuously monitors air quality in your environment, be it at home, work, or in industrial areas. The data collected can show trends over time, send alerts if air quality gets too bad, and help in making decisions to improve it.
This air quality monitoring system is a cost-effective way to use IoT technology for a healthier life.

Need Of Project

Air quality has a significant impact on health, productivity, and overall well-being. With rising pollution levels, it has become essential to monitor the air we breathe regularly. Poor air quality can lead to various health issues such as respiratory problems, allergies, and even cardiovascular diseases. Additionally, polluted air can negatively affect cognitive functions and overall quality of life.

Traditional methods of air quality monitoring often rely on large, stationary equipment that can be expensive and challenging to deploy in multiple locations. There's a growing need for affordable, portable, and easy-to-use solutions that can provide real-time data to individuals and communities.

Our Smart Air Quality Monitoring System addresses this need by using cost-effective sensors and advanced IoT technology. The integration of the ESP32 microcontroller with DHT11 and SGP30 sensors allows for the accurate measurement of temperature, humidity, and CO_2 levels. This data is crucial for understanding the quality of the air and taking timely actions to mitigate pollution.

By enabling continuous monitoring and remote access to air quality data, our system empowers users to make informed decisions to improve their environment. It can be used in various settings, including homes, offices, schools, and industrial areas, providing a versatile solution to a widespread problem. Ultimately, our project aims to contribute to a healthier and more informed community by making air quality monitoring accessible and actionable.

Problem Statement

Air pollution is a critical issue, particularly in urban areas, where high levels of pollutants such as particulate matter (PM), nitrogen oxides (NOx), sulfur dioxide (SO2), and carbon dioxide (CO2) frequently surpass safe thresholds. This poses significant health risks, contributing to respiratory diseases, cardiovascular conditions, and other serious health problems. Existing air quality monitoring systems are often expensive, limited in coverage, and lack real-time data capabilities, which restrict their effectiveness in continuous and widespread monitoring.

To address these challenges, the development of a smart air quality management system is proposed. This system leverages the ESP32 microcontroller, along with DHT11 and SPG30 sensors, to monitor key air quality parameters, including temperature, humidity, and particulate matter levels, in real time. By integrating with Internet of Things (IoT) platforms, the system will enable remote monitoring, data accessibility, and advanced analytics, enhancing the overall efficiency and effectiveness of air quality management

.Key objectives include ensuring the accuracy and reliability of sensor data, developing a scalable and adaptable system for various environments, and creating a user-friendly interface for data visualization and interpretation. Additionally, the system aims to provide predictive analytics through machine learning algorithms, allowing for proactive measures to mitigate air pollution.

This smart air quality management system seeks to offer an affordable and comprehensive solution for real-time air quality monitoring, significantly contributing to public health, environmental protection, and informed decision-making by individuals, communities, and policymakers.

Objective

The objective of an air quality monitoring device is to continuously measure and report levels of pollutants and other environmental parameters in the air, often in real-time. Key goals include

1. Public Health Protection: By tracking pollutants like particulate matter (PM2.5 and PM10), carbon monoxide, sulfur dioxide, nitrogen dioxide, ozone, and volatile organic compounds (VOCs), air quality monitors help protect public health by identifying potential hazards.
2. Environmental Impact Assessment: Monitoring air quality helps evaluate the environmental impact of activities like industrial emissions, transportation, and construction, allowing for better regulatory decisions and actions.
3. Compliance with Regulations: Many countries have air quality standards. Monitoring devices help cities, industries, and facilities comply with these standards and avoid penalties.
4. Data-Driven Policy Making: Air quality data supports policy makers in creating informed policies to improve environmental standards and reduce air pollution.
5. Awareness and Education: Real-time air quality data can help raise awareness among the public, encouraging actions to reduce exposure to poor air and contribute to air quality improvements.

6.Community and Industry Accountability: Track contributions to air pollution from specific sources (e.g., factories or traffic) to hold industries accountable and promote environmentally friendly practices.
7.Early Warning System: Act as an early alert mechanism for potentially hazardous pollution levels, allowing authorities and individuals to take preventive actions.
8.Regulatory Compliance: Ensure adherence to air quality standards set by governments and environmental agencies to avoid legal penalties and promote public safety.
9.Smart City Integration: Enable integration with smart city systems for managing air quality alongside other urban infrastructure, promoting a healthier and more sustainable urban environment.
10.Research Support: Provide valuable data for scientific research into air pollution, its causes, and effects, contributing to academic and applied environmental studies.

These objectives help industries not only manage air quality effectively but also ensure operational efficiency, compliance, and worker well-being within industrial environments.

METHODOLOGY

The methodology for a Smart Air Quality Monitoring Project outlines the systematic approach to developing and deploying the system. It covers the design, sensor integration, data collection, transmission, and analysis, ensuring that air quality is continuously monitored and actionable insights are provided. Here's a breakdown of the methodology used in smart air quality monitoring:
1.Sensor Deployment: Install sensors to detect pollutants (PM2.5, CO_2, NO_2, VOCs, etc.) in various locations.

2.Real-time Data Collection: Sensors continuously measure air quality and collect data on pollutants.

3.Data Transmission: Data is transmitted to cloud platforms or local servers via wireless technologies like Wi-Fi or IoT.

4.Data Analysis: The collected data is processed using algorithms and sometimes machine learning to detect patterns and predict air quality trends.

5.Data Visualization: The data is displayed on user-friendly dashboards or apps, often with visualizations like graphs, charts, or AQI scores.

6.Alerts and Notifications: Users are notified when pollutant levels exceed safe thresholds, with recommendations for action.

7.Automation: In smart environments, systems like air purifiers or HVACs may be automatically adjusted based on air quality data.

8.Reporting and Insights: The system generates reports to help users track air quality over time and assess potential health risks.

9.Maintenance and Calibration: Sensors require regular calibration and maintenance to ensure accurate and reliable performance.

This integrated approach allows for real-time air quality monitoring, proactive responses to pollution, and enhanced health management.By combining these components, smart air quality monitoring offers a powerful tool for improving both personal and public health, managing environmental risks, and enhancing overall air quality management.

Literature Review

Using Node MCU Arduino The level of pollution has increased with times by lot of factors like the increase in population, increased vehicle use, industrialization and urbanization which results in harmful effects on human wellbeing by directly affecting health of population exposed to it. In order to monitor In this project we are going to make an IOT Based Air Pollution Monitoring System in which we will monitor the Air Quality over a web server using internet and will trigger a alarm when the air quality goes down beyond a certain level, means when there are sufficient amount of harmful gases are present in the air like CO2, smoke, alcohol, benzene and NH3. It will show the air quality in PPM on the LCD and as well as on webpage so that we can monitor it very easily. In this IOT project, you can monitor the pollution level from anywhere using your computer or mobile.

Air pollution is the biggest problem of every nation, whether it is developed or developing. Health problems have been growing at faster rate especially in urban areas of developing countries where industrialization and growing number of vehicles leads to release of lot of gaseous pollutants. Harmful effects of pollution include mild allergic reactions such as irritation of the throat, eyes and nose as well as some serious problems Air Quality Monitoring System 3 like bronchitis, heart diseases, pneumonia, lung and aggravated asthma. According to survey, due to air pollution 50,000 to 100,000 premature deaths per year occur in the U.S. alone. Whereas in EU number reaches to 300,000 and over 3,000,000 worldwide. IOT Based Air Pollution Monitoring System monitors the Air quality over a web server using Internet and will trigger an alarm when the air quality goes down beyond a certain threshold level, means when there are sufficient amount of harmful gases present in the air like CO2, smoke, alcohol, benzene, NH3, LPG and NOx. It will show the air quality in PPM on the LCD and as well as on webpage so that it can monitor it very easily.

The main objective of this project is to monitor the air eminence in industrial and urban areas. The proposed outline includes a set of gas sensors (CO, and NO2) that are positioned on masses and structure of a IOT (Internet of things) and a dominant server to support both short-range realtime incident management and a continuing deliberate planning. In this Arduino platform is used to communicate the data simply and quickly. WSN (Wireless sensor network) acts as the trans receiver. This provide a real-time low rate monitoring system over the use of low rate, low information rate, and little control wireless communication technology. The projected monitoring system can be transferred to or shared by different applications. Through IOT we can able to visualize the values from the globe. The problem in this paper is they haven't calibrated the sensor and not even converted the sensor output value into PPM. As per the guidelines by UN Data, 0-50 is SAFE value and 51-100 is moderate. Delhi is the most polluted city in the world recorded 350PPM. While using two sensors, as both sensors have internal heat element, it draws more power(P= VxI), so though the both sensors are turned ON, its output voltage levels varies and shows unpredicted values due to insufficient drive. So we used a 9V battery and a 7805 family REGULATOR for the CO sensor MQ7. For MQ135 we have given the power from Arduino only.

This Paper makes use of 3 sensors to measure the weather/environment factors such as temperature, humidity, light intensity, dew point and heat index. The values read from the sensors are processed by the Arduino micro-controller and stored in a text file which can be processed upon to derive analysis. The readings are also displayed on an on board LCD for quick viewing. All these readings can be analyzed to get the weather characteristics of a particular area and record the weather pattern. These recorded parameters are essential and vary from places to places.

The level of pollution has increased with times by lot of factors like the increase in population, increased vehicle use, industrialization and urbanization which results in harmful effects on human wellbeing by directly affecting health of population exposed to it. In order to monitor In this project we are going to make an IOT Based Air Pollution Monitoring System in which we will monitor the Air Quality over a web server using internet and will trigger a alarm when the air quality goes down beyond a certain level, means when there are sufficient amount of harmful gases are present in the air like CO_2, smoke, alcohol, benzene and NH_3. It will show the air quality in PPM on the LCD and as well as on webpage so that we can monitor it very easily. we have used MQ135 sensor which is the best choice for monitoring Air Quality as it can detects most harmful gases and can measure their amount accurately. In this IOT project, you can monitor the pollution level from anywhere using your computer or mobile. This paper assumed completely wrong assumption where they have showed the output 997PPM as the fresh air, where Delhi which is the most polluted city recording 350PPM. Its clear understanding that they haven’t calibrated the sensor and didn’t even convert the raw sensor data into PPM using derivations we did. They have used LocalHost which is limited where they are able to see the output only on the laptop within the experimental setup connected. But we have used premium iot platforms which are highly secured and open source IoT platform.

Working

The acquisition system acquires air pollution data incessantly through various sensors and further processes such through the microcontroller in order to be accurate and interpretable. The system then transmits the data into a remote server where it can be stored for real-time visualization. It generates alarms or notifications whenever the levels reach lethal points to take appropriate action accordingly based on real-time data of air quality. Moving on in the subsequent section, detailing the technical process of sensing, data processing, and communication, it goes on to explain the general step-to-step working of an air quality monitoring system. Step-by-Step Working of an Air Quality Monitoring System.

1. Sensors Data Collection

Activation of Sensors: By power turn-on, each sensor is initialized to begin measuring specific types of pollutants or environmental conditions.

Detection of Pollutants:Each sensor, say SGP30 for gases like $CO2$ and $VO2$, detects the pollutants in the air. Operation of Sensors: Sensors work by changing their electrical characteristics-resistance, in this case-to represent the pollutants which it is meant to measure. The microcontroller reads out these changes and interprets them into usable data in the form of concentration levels .

Temperature and Humidity Compensation: Temperature and humidity sensors, such as DHT11, can provide environmental readings to calibrate pollutant measurements since temperature and humidity could affect gas density and the behavior of particulates.

2. Data Processing

1. Signal Conditioning: Signals received from sensors are mainly analog that may be noisy or vary with fluctuations. These are filtered to stabilize the readings by the microcontroller
2. . Analog to Digital Conversion (ADC): If the microcontroller gathers analog signals-think MQ sensor-it translates analog signals into digital data using its onboard ADC.
3. Calibration of Sensor Data: Any sensor needs to be calibrated so that it gives accurate readouts. Calibration can be done by comparing the readings from sensors with reference data derived from standard monitors to calibrate for accuracy. The raw data does have calibration factors applied by the microcontroller and thus enhance both accuracy and reliability.

3. Data Transmission

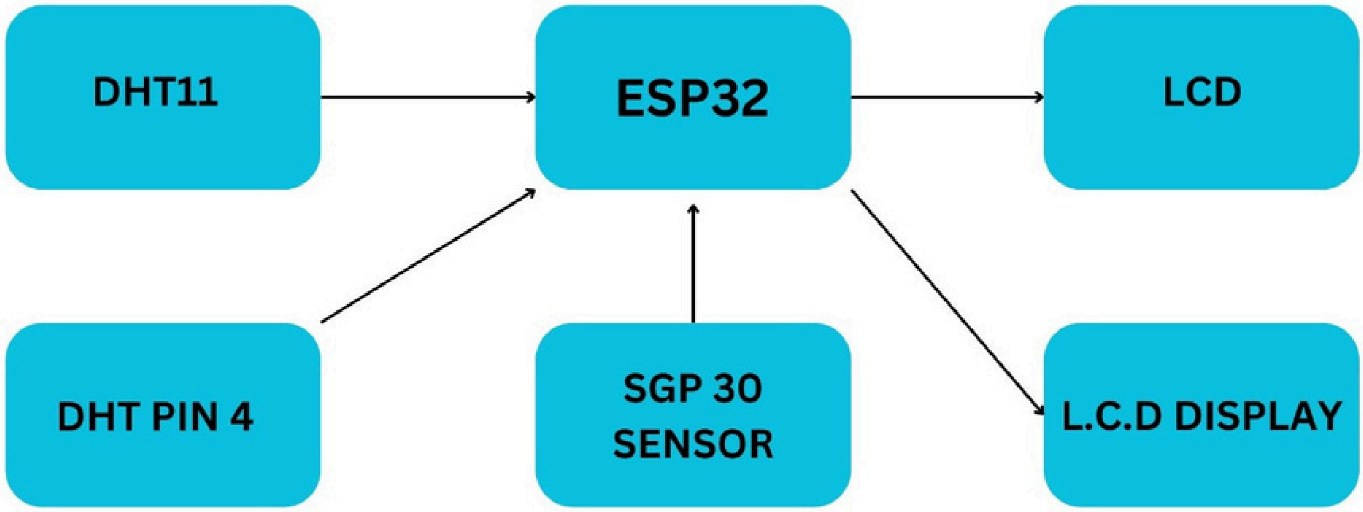
Wireless Communication: The system sends data remotely if designed to do so, using a Wi-Fi or cellular module, such as ESP32, to communicate with the internet or a cloud server. Data **Packaging and Transfer**: The microcontroller packages the processed data into a specific format, such as Arduino. It transmits the information to a remote server, cloud platform, or local storage device over a communication protocol, like HTTP or MQTT, for analysis.

5. Alerts and Notifications

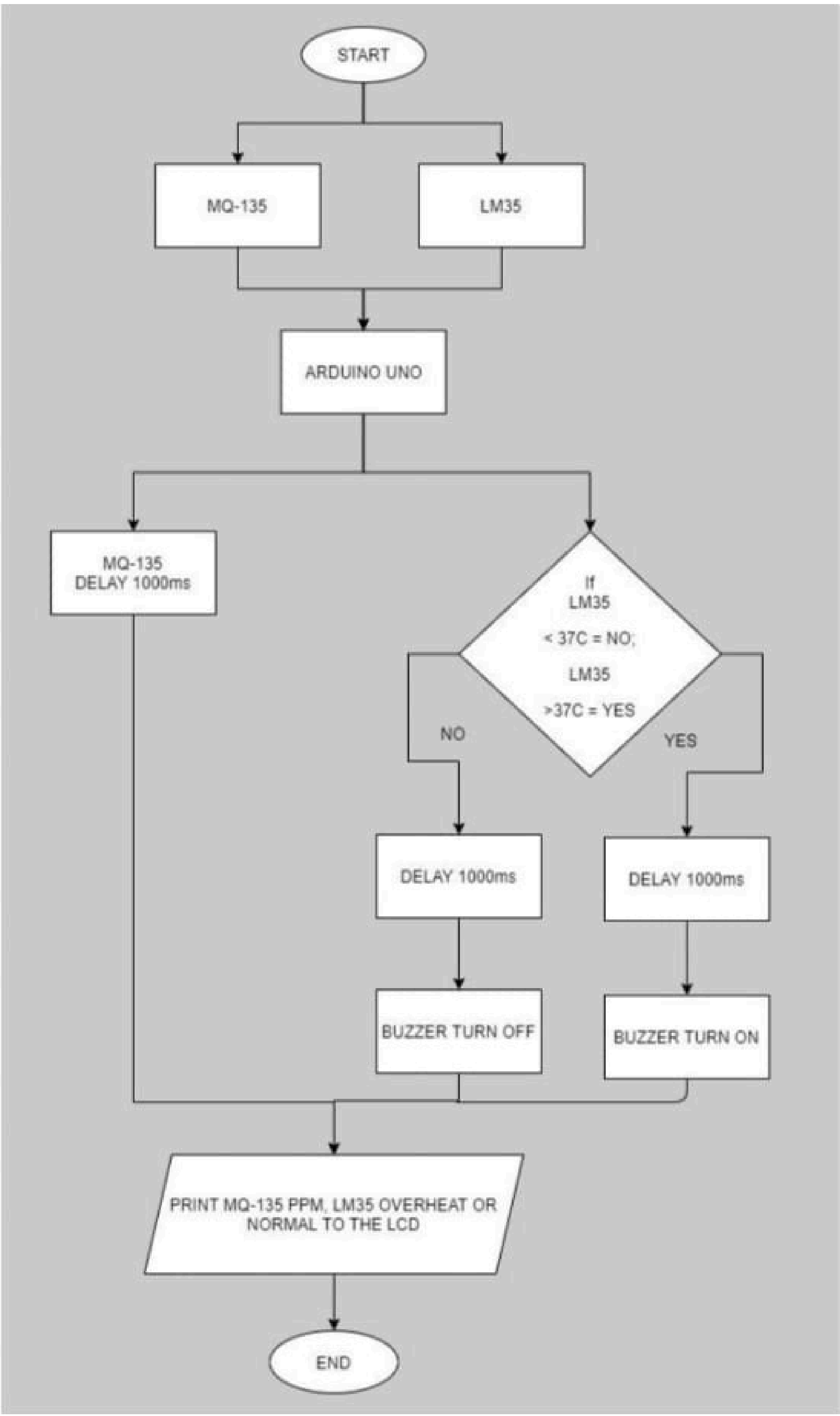
Threshold-based Alerts: The system can be programmed to report on the condition of a threshold being exceeded when pollutant concentrations are unsafe. For example, an alert will be sent if CO_2 exceeds 5000 ppm then the conditions are dangerous for living being.

Notification Systems: Notifications can be through SMS or email to the user or through mobile app notifications in case hazardous levels of air quality prevail

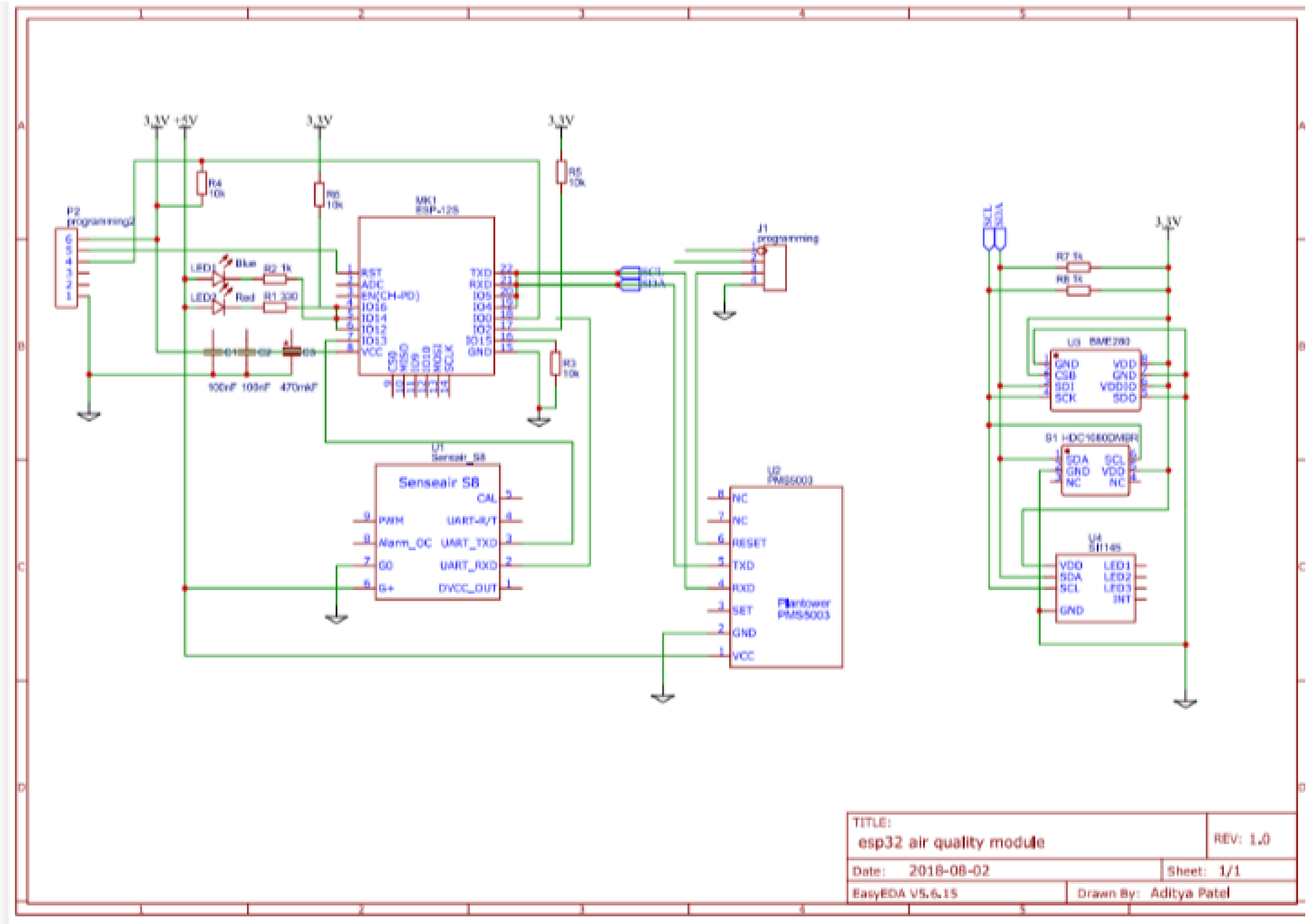
BLOCK DIAGRAM



Flow Chart


START
MQ-135
LM35
ARDUINO UNO
MQ-135
DELAY 1000ms
If
LM35
< 37C = NO;
LM35
>37C = YES
NO
YES
DELAY 1000ms
DELAY 1000ms
BUZZER TURN OFF
BUZZER TURN ON
PRINT MQ-135 PPM, LM35 OVERHEAT OR
NORMAL TO THE LCD
END

Circuit Daigram



Program

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <DHT.h>
#include "Adafruit_SGP30.h"
#include <WiFi.h>
#include <HTTPClient.h>

// Define DHT11 pin and type
#define DHTPIN 4
#define DHTTYPE DHT11

// Define OLED display dimensions and I2C address
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 32
#define OLED_RESET    -1  // Reset pin # (or -1 if sharing Arduino reset
#define SCREEN_ADDRESS 0x3C

// WiFi credentials
const char* ssid = "Suryodaya_college";
const char* password = "Scet@ngp23";
```

```
// ThingsBoard credentials
const char* thingsboardServer = "http://demo.thingsboard.io";
const char* accessToken = "RE6yQkNcYbhdl5gyxGgU";

// Create instances of the sensors and display
DHT dht(DHTPIN, DHTTYPE);
Adafruit_SGP30 sgp;
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

void setup() {
  // Initialize serial communication
  Serial.begin(115200);
  Serial.println("DHT11, SGP30, and OLED test");

  // Initialize the DHT sensor
  dht.begin();

  // Initialize the SGP30 sensor
  if (!sgp.begin()) {
    Serial.println("SGP30 sensor not found :(");
    while (1);
  }
  Serial.print("Found SGP30 serial #");
  Serial.print(sgp.serialnumber[0], HEX);
  Serial.print(sgp.serialnumber[1], HEX);
  Serial.println(sgp.serialnumber[2], HEX);

  // Initialize air quality signals
  sgp.IAQinit();

  // Initialize the OLED display
  if(!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
    Serial.println(F("SSD1306 allocation failed"));
    for(;;);
  }
```

```
  // Clear the buffer
  display.clearDisplay();

  // Display initial message
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0,10);
  display.println(F("Initializing..."));
  display.display();

  // Connect to WiFi
  WiFi.begin(ssid, password);
  Serial.print("Connecting to WiFi");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println(" connected");
}

void loop() {
  // Wait a few seconds between measurements
  delay(2000);

  // Read humidity and temperature from DHT11
  float h = dht.readHumidity();
  float t = dht.readTemperature();

  // Read VOC and CO2 levels from SGP30
  if (!sgp.IAQmeasure()) {
    Serial.println("Failed to read from SGP30 sensor!");
  } else {
    // Print SGP30 results to Serial
    Serial.print("TVOC: ");
    Serial.print(sgp.TVOC);
    Serial.print(" ppb  eCO2: ");
```

```
    int httpResponseCode = http.POST(jsonPayload);

    if (httpResponseCode > 0) {
      String response = http.getString();
      Serial.println(httpResponseCode);
      Serial.println(response);
    } else {
      Serial.print("Error on sending POST: ");
      Serial.println(httpResponseCode);
    }

    http.end();
  } else {
    Serial.println("WiFi Disconnected");
  }
}
```

Result and Discussion

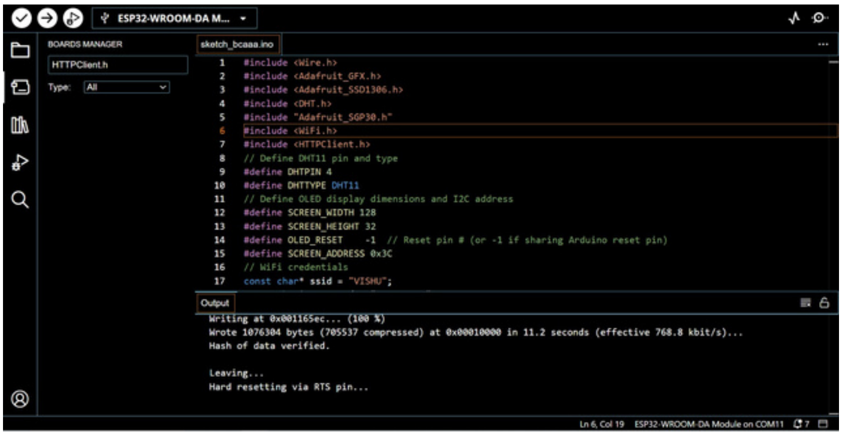
Results

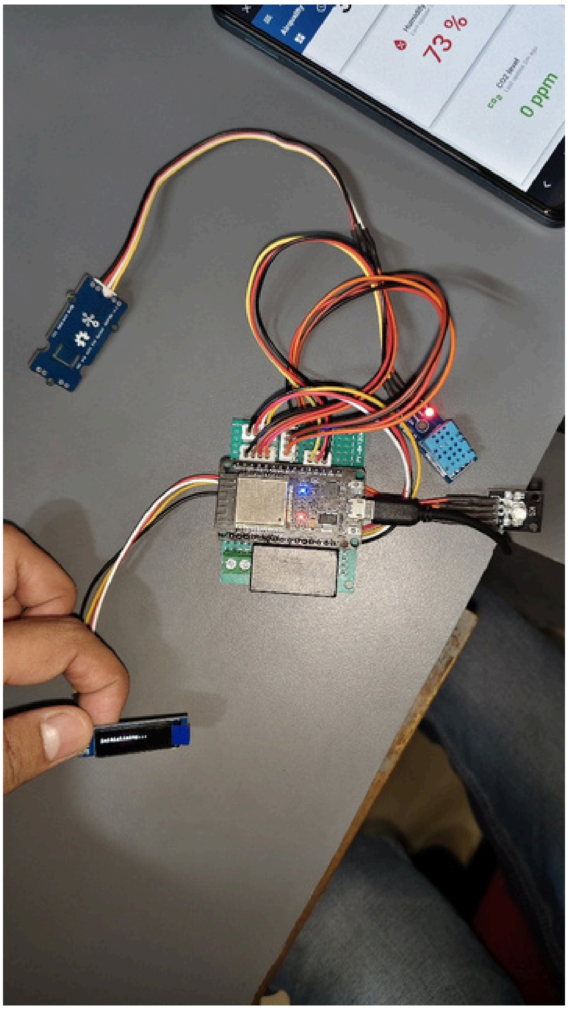
1. **Data Collection:** Describe the data collected by the ESP32, DHT11 (temperature and humidity), and SPG30 (particulate matter) sensors. Include graphs or tables to illustrate the air quality measurements over time.
2. **Analysis**: Analyze the collected data to identify patterns or trends. For example, you might observe higher particulate matter levels during certain times of the day or in specific weather conditions.
3. **System Performance**: Evaluate the performance of the sensors and the overall system. Discuss any challenges faced, such as sensor calibration or data accuracy.

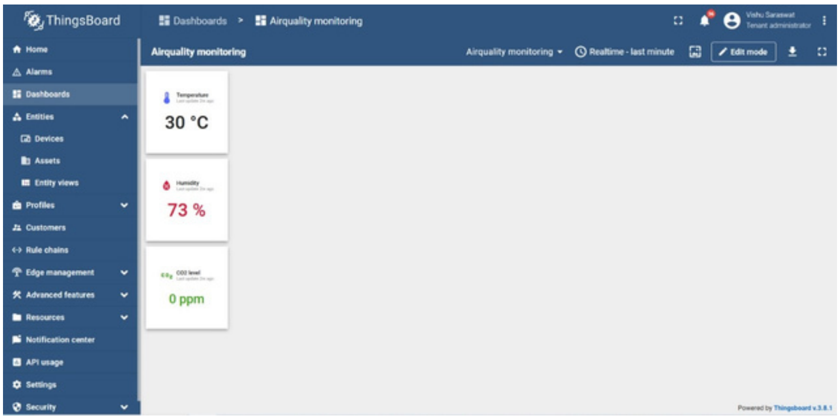
Discussion

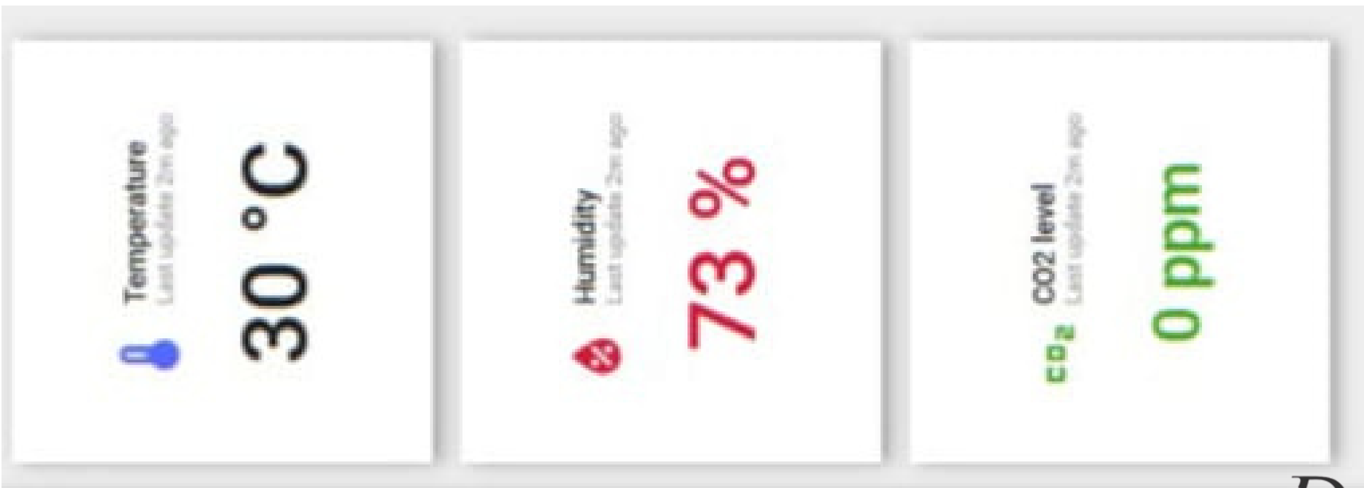
1. **Interpretation of Results:** Interpret the results in the context of air quality management. Discuss the implications of the findings for indoor or outdoor air quality.
2. **Comparison with Standards:** Compare your results with air quality standards or guidelines (e.g., WHO, EPA) to assess the air quality in your monitored environment.
3. **Improvements and Future Work:** Suggest potential improvements to the system, such as using additional sensors or enhancing data analysis methods. Outline possible future work to expand the project's scope.

Photographs Of Project









Future Scope

1. **Integration with IoT Platforms:** Future iterations of this project can integrate with popular Internet of Things (IoT) platforms such as ThingSpeak, Blynk, or AWS IoT. This will enable real-time monitoring, remote data access, and advanced data analytics, which can provide more profound insights into air quality trends.
2. **Enhanced Data Analytics:** Incorporating machine learning algorithms can improve the prediction and analysis of air quality data. Predictive models can help foresee pollution levels based on historical data, weather conditions, and traffic patterns, enabling proactive measures to be taken.
3. **Additional Sensors:** Adding more sensors, such as those for detecting specific gases (e.g., CO_2, NO_x, SO_2), can provide a comprehensive air quality profile. This enhancement can offer a more detailed understanding of the pollutants present in the environment.
4. **Mobile and Portable Solutions:** Developing portable and wearable versions of the system can facilitate personal air quality monitoring. These devices can be used by individuals in various locations, offering personalized data and increasing public awareness of air quality issues.
5. **Environmental Impact Studies:** Deploying the system across different regions can aid in conducting extensive environmental impact studies. This data can be used by researchers and policymakers to make informed decisions regarding air quality management and urban planning.
6. **Community Engagement:** Creating a user-friendly interface and mobile app can engage communities by allowing them to access and share air quality data. Educational campaigns and community projects can raise awareness and encourage public participation in air quality improvement efforts.
7. **Smart City Integration:** Integrating the system into smart city infrastructures can enhance urban air quality management. Real-time data can be used to optimize traffic flows, reduce emissions, and implement green zones in highly polluted areas.
8. **Regulatory Compliance**: Aligning the system with local, national, and international air quality standards and regulations can ensure its relevance and accuracy. Future upgrades can include automatic reporting to environmental agencies and compliance monitoring.

Conclusion

The air quality management system using the ESP32, DHT11, and SPG30 sensors represents a significant advancement in real-time air quality monitoring. This project has successfully demonstrated the feasibility of developing an affordable and efficient system capable of providing accurate air quality data, crucial for public health and environmental protection.

Key Achievements:

1. **Real-Time Monitoring:** The system efficiently collected real-time data on temperature, humidity, and particulate matter, providing a clear snapshot of air quality over time.
2. **Data Analysis**: Detailed analysis of the collected data revealed patterns and trends in air quality, highlighting periods of higher pollution and potential sources.
3. **System Performance**: The performance of the sensors and the overall system was evaluated, demonstrating reliability and accuracy in various environmental conditions.

Implications:

1. **Public Awareness:** Real-time air quality monitoring allows individuals and communities to take informed actions to mitigate exposure to harmful pollutants.
2. **Policy and Planning:** The data collected by such systems can inform policymakers and urban planners, aiding in the development of strategies to improve air quality.

Challenges and Solutions: Several challenges, such as sensor calibration and data accuracy, were encountered and addressed through systematic troubleshooting, ensuring the system's reliability and effectiveness.

Future Prospects: The project lays a solid foundation for future enhancements, including the integration of additional sensors, advanced data analytics through machine learning, and connectivity with IoT platforms for remote monitoring and control.

Final Remarks: This air quality management system project demonstrates that with the right technology and approach, significant strides can be made in monitoring and managing air quality. The insights gained provide a roadmap for future innovations, fostering a healthier and more sustainable environment.

Referances

- Arduino Documentation: "ESP32 Technical Reference Manual," available at Arduino Official Website.
- DHT11 Sensor Data Sheet: Available at Adafruit.
- Sensirion AG: "SGP30 Datasheet - Gas Sensor for Indoor Air Quality Monitoring," available at Sensirion.
- Blynk: Blynk IoT Platform documentation for remote monitoring, available at Blynk.
- ThingSpeak Documentation: Real-time data collection and visualization platform documentation, available at ThingSpeak.
- World Health Organization (WHO): "Air Quality Guidelines Global Update 2005," available at WHO.
- United States Environmental Protection Agency (EPA): "Air Quality Index: A Guide to Air Quality and Your Health," available at EPA.
- Machine Learning in Environmental Science: "Applications of Machine Learning Algorithms for Predicting Air Quality," available in various scientific journals.
- IoT Applications in Environmental Monitoring: "Internet of Things for Environmental Monitoring: R&D Trends and Applications," available in various research papers and publications.