Visvesvaraya Technological University
Belagavi, Karnataka-590 018

A MINI PROJECT REPORT
On

'Air Quality monitoring System'
Submitted
In partial fulfillment requirements for the award of the Degree

of
BACHELOR OF ENGINEERING
IN
INFORMATION SCIENCE AND ENGINEERING

by

Varun S Amin 4NM21IS204
Vinay Kumar U 4NM21IS207
Vinyas S 4NM21IS210

Under the Guidance of

Ms. Vanishree B S
Assistant Professor

NITTE EDUCATION TRUST | N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India
(ISO 9001:2015 Certified)
Accredited with 'A' Grade by NAAC

Department of Information Science and Engineering

NITTE EDUCATION TRUST | N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India
(ISO 9001:2015 Certified)
Accredited with 'A' Grade by NAAC

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that Mr. **Varun S Amin (**4NM21IS204), Mr. **Vinay Kumar U (**4NM21IS207) and Mr. **Vinyas S (**4NM21IS210) has satisfactorily completed the Internet of Things Mini Project work entitled "**Air Quality monitoring System**" of Third Year Bachelor of Engineering in Information Science and Engineering at NMAMIT, Nitte in the academic year 2023 - 24.

Project Guide	**Head, Dept. of ISE**
Ms. Vanishree B S	**Dr. Ashwini B**
Assistant Professor	**Associate Professor**

ABSTRACT

The goal of this project is to create an IoT-based air pollution monitoring system. Monitoring the air quality from anywhere using a computer or mobile device surroundings and environment. There are numerous approaches and devices available for the measuring and monitoring of air quality. The IoT-based air pollution monitoring system would not only assist us in monitoring the air quality, but would also be capable of sending alert signals whenever the air quality deteriorated and goes down beyond a certain level. The proposed system has various parameters such as Air Quality, Temperature and Humidity sensors with ESP32 micro-controller which collects and upload data into the cloud using ESP32 Wi-Fi module. The data is transmitted to the cloud platform using Blynk and alerts the user through an application.

TABLE OF CONTENTS **Page no.**

1. Introduction

Air pollution is one of the most serious challenges to the modern environment. Everyone is affected by air pollution on a daily basis, including humans, animals, crops, cities, forests, and aquatic environments. Furthermore, it should be kept under control at a particular level in order to avoid the rate of increase from growing Global warming. The goal of this project is to create an IoT-based air pollution monitoring system. Monitoring the air quality from anywhere using a computer or mobile device surroundings and environment. There are numerous approaches and devices available for the measuring and monitoring of air quality. You can use your computer or mobile device to monitor the pollution level from anywhere. In recent years, automobile emissions, factory chemicals, and There is smoke and dust everywhere. That is why air is now so polluted. Air pollution has a negative impact on human health, particularly in areas where The air in our bodies is used to breathe.In our lungs may cause some diseases, such as asthma, cough, lung disorders The air pollution cannot be detected by human feelings. The air pollution may contain a lot of dangerous substances, such as LPG gas, carbon monoxide, and methane. Substances in the polluted air are very dangerous. For example, if the carbon monoxide is above 100ppm, it makes human feel dizzy, nauseous, and within minutes they could die. The IoT-based air pollution monitoring system would not only assist us in monitoring the air quality, but would also be capable of sending alert signals whenever the air quality deteriorated and goes down beyond a certain level The proposed system alert users, notify data of polluted area that has been checked for the immediate surroundings.

The proposed system has various parameters such as Air Quality, Temperature and Humidity sensors with ESP32 micro-controller which collects and upload data into the cloud using ESP32 Wi-Fi module. The data is transmitted to the cloud platform using Blynk and alerts the user through an application. The proposed system has various applications like in industry where the pollution levels check of dangerous gasses is paramount. Further pollution information is used to alert the users about the air quality in their surroundings are not good for health.

.

2. Components and Modules

In this section, various components and modules being used for IoT based Visitor Counter project development is discussed:

2.1 WIFI module-ESP32

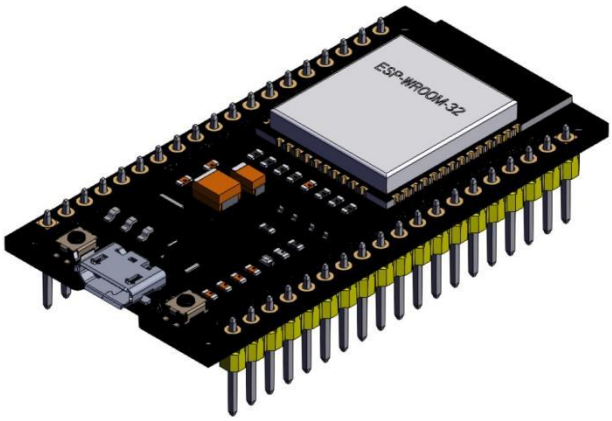

Figure 1: WIFI module-ESP32

The ESP-32 is a versatile and powerful microcontroller noted for its IoT and embedded system capabilities. It has a dual-core CPU, which provides more processing capability for data collecting and analysis. The ESP-32, which has built-in Wi-Fi and Bluetooth capabilities, can quickly connect to the internet and other devices. The modular design of the ESP-32, as well as its support for numerous communication protocols, make it a popular choice for a wide range of IoT applications.Its low cost, small size, and comprehensive library support all contribute to its popularity among developers and makers.

2.2 DHT11 Sensor

This DHT11 Digital Relative Humidity and Temperature Sensor Module is pre-calibrated with resistive sense technology coupled with NTC thermistor, for the precise reading of the relative Humidity and surrounding temperature DHT 11 break-out board is a very popular, low-cost sensor from Aosong, the breakout provides easy installation of the DHT11 sensor module.The board is also equipped with high- performance 8-Bit microcontroller which is connected to the DHT11 sensor module. The output of the DHT11 is in the form of a digital signal on a single data pin. The sensing update frequency is to be measured at every 2sec (0.5Hz).

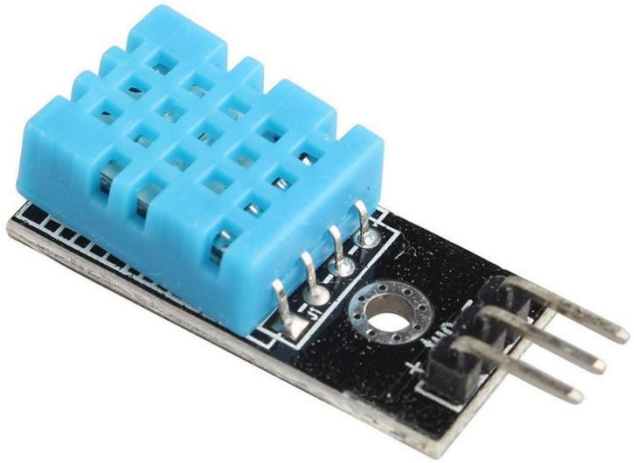

Figure 2: DHT11 Sensor

2.3 Gas Sensor (MQ-135)

The MQ-135 gas sensor is widely used for detecting air pollutants such as carbon dioxide (CO2), carbon monoxide (CO), ammonia (NH3), and different volatile organic compounds (VOCs). MQ-135 is commonly used to detect and measure the concentration of hazardous gases in the environment in air quality monitoring systems, air purifiers, and safety equipment.

Figure 3: MQ-135 Gas Sensor

3. Circuit description & working principle

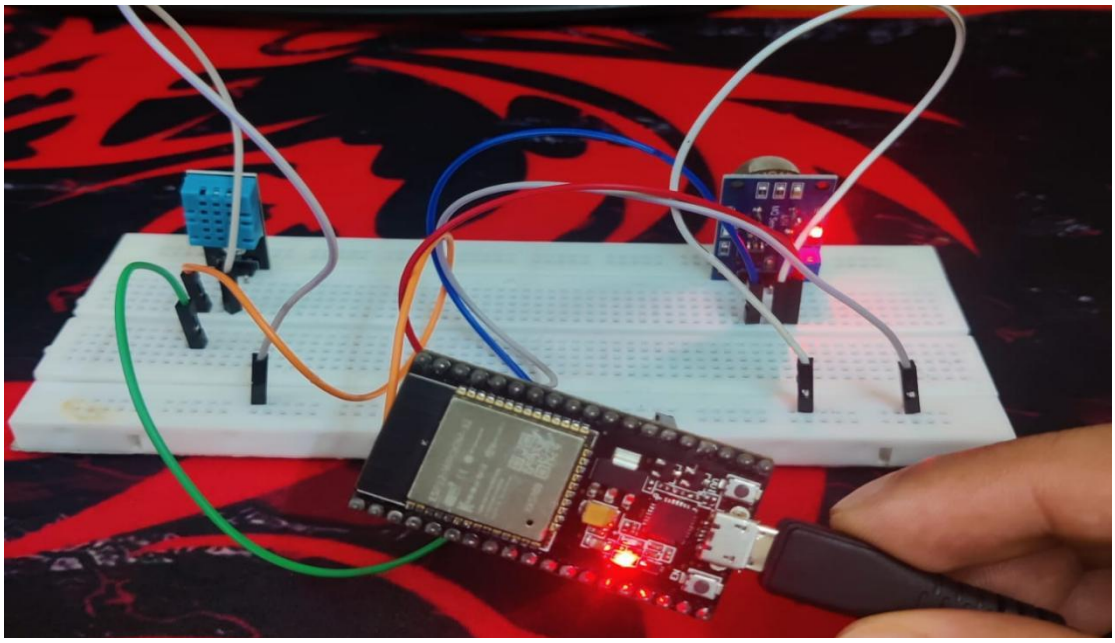

Figure 4: Project design

The core of this system is an ESP32 development module, a versatile microcontroller equipped with built-in Wi-Fi capabilities. This module is responsible for processing data from the connected sensors and can communicate with your phone if needed.

The DHT11 sensor is connected to the ESP32 module, with the sensor's data pin (DHT_PIN_DATA) linked to a specific digital pin on the ESP32. The DHT11 sensor accurately measures temperature and humidity, providing reliable and stable readings. It utilizes a capacitive humidity sensor and a thermistor to capture environmental data and transmits it digitally to the ESP32.

Additionally, the MQ135 sensor is also connected to the ESP32 module, with the sensor's analog output (AOUT) pin (MQ135_PIN_AOUT) connected to another digital pin on the ESP32. This sensor is primarily used for detecting various gases, such as NH3, NOx, alcohol, benzene, smoke, and CO2. It boasts high sensitivity and a rapid response time, making it suitable for air quality monitoring.

The ESP32 is programmed to continuously read data from these sensors. When the temperature reading from the DHT11 sensor or the gas concentration reading from the MQ135 sensor exceeds predefined thresholds (temperatureThreshold and gasThreshold), the ESP32 triggers specific actions like generating local alerts or performing specific tasks.

4.Algorithms & flowchart

4.1 Algorithm :

The algorithm of overall process goes like this,

Step 1: Initialize the ESP32 module and the DHT11 and MQ135 sensors. Establish a connection to the Wi-Fi network.

Step 2: Collect data from the DHT11 sensor (temperature and humidity). Collect data from the MQ135 sensor (gas value).

Step 3:Process the collected data. This could involve converting raw sensor readings into meaningful units.

Step4: Check if the collected data exceeds predefined threshold values.

Step 5: If any value exceeds its threshold, generate an alert message. The message should contain information about which value(s) exceeded the threshold and the current readings using Blynk.

Step 6: Send the alert message to the predefined phone number with the help of Blynk App.

Step 7: Put the ESP32 into sleep mode for a predefined interval to save power.

Step7: Wake up the ESP32 and go back to step 2.

4.2 Flowchart :

The provided code initializes the ESP32 microcontroller and connects it to a Wi-Fi network using authentication credentials. It reads data from a DHT11 temperature and humidity sensor and an MQ135 gas sensor. The data is then sent to the Blynk platform for visualization. If the temperature surpasses a predefined threshold, a high-temperature alert is sent to Blynk. Similarly, if the gas level exceeds another threshold, a pollution alert is logged. The system repeats these actions at regular intervals. The setup includes configuring the sensors and Blynk, while the main loop handles data reading, threshold checks, and data transmission. A brief delay is introduced before the loop continues, maintaining continuous monitoring and reporting.

The Flowchart:

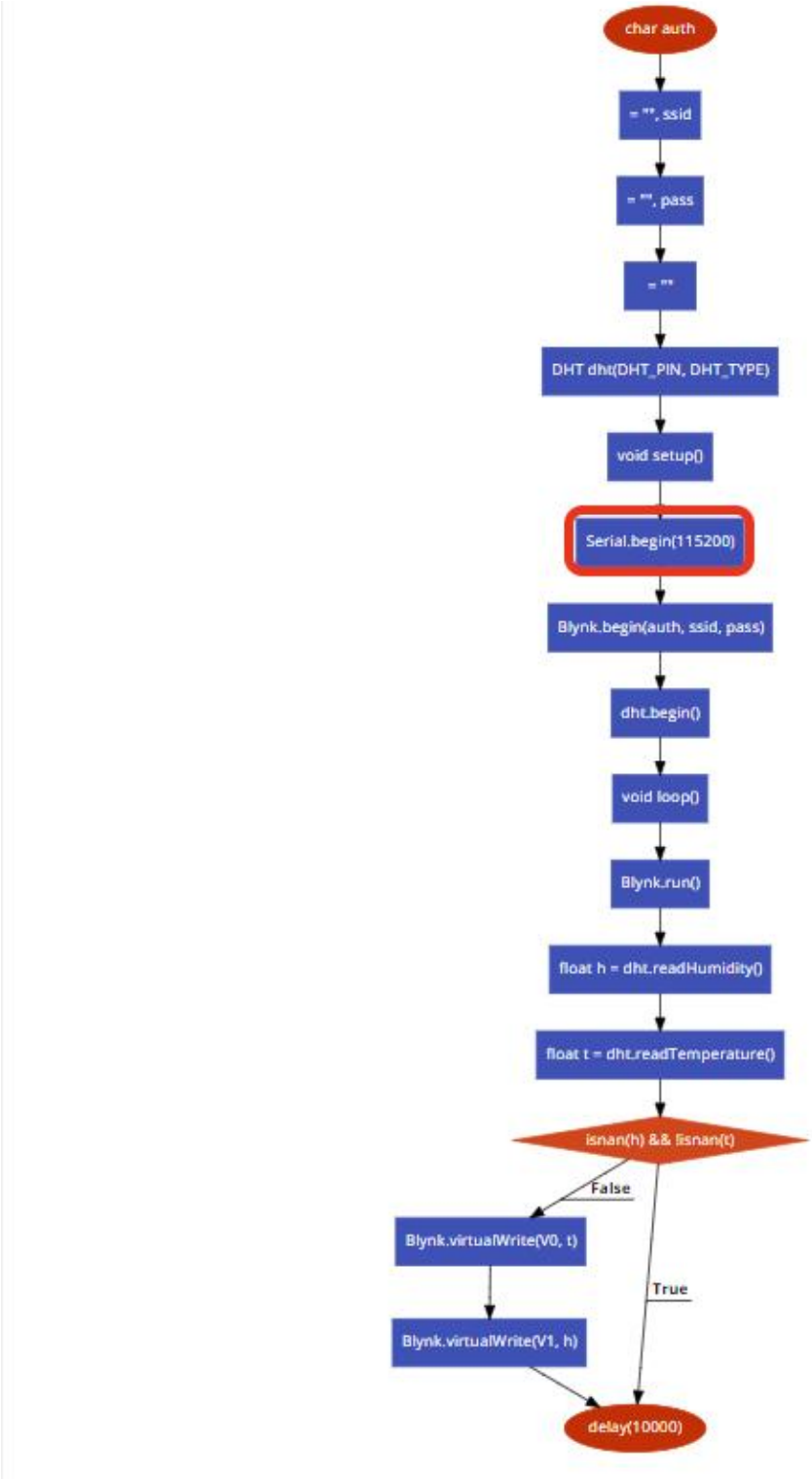

Figure 5: Flowchart of overall process

5. Implementation

Code:

```
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
#include <DHT.h>

char auth[] = "";  // Your Blynk authentication token
char ssid[] = "";  // Your Wi-Fi SSID
char pass[] = "";  // Your Wi-Fi password

BlynkTimer timer;
#define DHT_PIN_DATA 0  // Digital pin for DHT11
#define DHT_TYPE DHT11
#define MQ135_PIN_AOUT 12  // Analog pin for MQ135

DHT dht(DHT_PIN_DATA, DHT_TYPE);
int gasThreshold = 300;
int temperatureThreshold = 25;

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

  // Read gas sensor data from MQ135
  int analogSensor = analogRead(MQ135_PIN_AOUT);

  // Send sensor data to Blynk app
  Blynk.virtualWrite(V0, t);
  Blynk.virtualWrite(V1, h);
  Blynk.virtualWrite(V2, analogSensor);

  // Check for temperature threshold
  if (t > temperatureThreshold) {
    Blynk.notify("High Temperature Alert! Temperature is " + String(t) + "°C");
    Blynk.email("email@email.com", "High Temperature Alert",+ String(t) + "°C");
  }

  // Check for gas threshold
  if (analogSensor > gasThreshold) {
    Blynk.notify("pollution_alert", "Bad Air");
    Blynk.email("email@email.com", "pollution_alert", "Bad Air");

  }
}

void setup() {
  // Initialize serial communication for debugging
  Serial.begin(115200);
```

```
  // Initialize Blynk with auth, Wi-Fi credentials, and server details
  Blynk.begin(auth, ssid, pass, IPAddress(117, 236, 190, 213), 8080);

 // Initialize DHT11 sensor
  dht.begin();

  // Set up a timer to call sendSensorData every 30 seconds
  timer.setInterval(30000L, sendSensorData);
}

void loop() {
  // Run Blynk framework and timer
  Blynk.run();
  timer.run();
}
```

6. Results

Figure 6 : Sensor readings

Figure 6 shows basic UI of the Blynk Interface with the initial sensor readings, showing us the temperature, humidity and gas value.

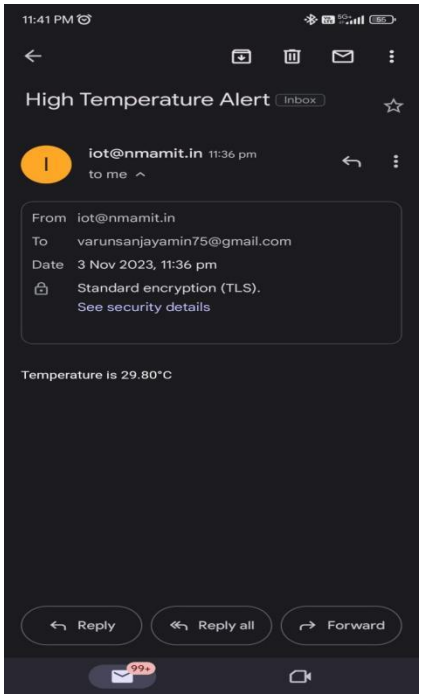

Figure 7 : Email Notification for High temperature alert

Figure 7 shows the email notification where the temperture is raises up to the threshold value.

7. Conclusion

Furthermore, the air quality monitoring system showcases the potential for IoT (Internet of Things) applications in enhancing our daily lives. By leveraging the power of connected devices and cloud platforms, it empowers users to take proactive measures in response to changing environmental conditions.utilizes an ESP32 microcontroller in conjunction with DHT11 and MQ135 sensors to monitor environmental conditions. It offers real-time data collection and alerts through the Blynk platform, enabling users to track temperature, humidity, and gas levels, with automatic notifications for high-temperature and pollution events. The system's simplicity and flexibility make it an accessible and scalable solution for a variety of settings, from homes to offices and industrial environments. As concerns about air quality and its impact on health and well-being continue to grow, systems like these are pivotal in addressing those concerns and creating a more informed and responsive society.

8.Future Work

Add More Sensors: Incorporate additional sensors to monitor other environmental parameters. For example, a particulate matter sensor like the PMS5003 can be used to measure PM2.5 and PM10 levels in the air. A CO2 sensor could be added to monitor carbon dioxide levels.

Machine Learning: Use machine learning algorithms to predict future air quality based on past data. This could help in forecasting potential air quality issues and take preventive measures.

Power Optimization: If your system is battery-powered, you could work on power optimization strategies to increase the battery life. This could involve putting the ESP32 into deep sleep mode between readings.

Enclosure Design: Design a suitable enclosure for your system. The enclosure should protect the components from the elements while allowing air to flow over the sensors.

Calibration: Regular calibration of sensors is crucial for accurate readings. Implement a method for easy calibration of sensors.

Connectivity: Consider adding support for different connectivity options like Wi-Fi or cellular networks to ensure reliable data transmission even in remote areas.

These future directions can enhance the capabilities and usability of the air quality monitoring system, making it a valuable tool for individuals, businesses, and communities seeking to improve and maintain air quality for better health and well-being.

References

- Poonam Paul, Ritik Gupta, Sanjana Tiwari, Ashutosh Sharma, “IoT based Air Pollution Monitoring System with Arduino”, IJART, May 2005.
- SaiKumar, M. Reji, P.C. KishoreRaja ”AirQuality Index in India”, IEEE conference Chennai, August 2014.
- Mohan Joshi, ”Research Paper on IoT based Air and Sound Pollution monitoring system”, IETS Journal, pp. 11-17, September 2015.
- Parmar, G., Lakhani, S., & Chattopadhyay, M. K. (2017, October). An IoT based low cost air pollution monitoring system.