Drone IOT Data Acquision System

Omkar Bhintade
Electronics and Telecommunication Engineering Department
Atharva College of Engineering,
Mumbai University India
omkarbhintade0103@gmail.com

Mayuresh Bane
Electronics and Telecommunication Engineering Department
Atharva College of Engineering,
Mumbai University India
mayurbane2003@gmail.com

Yugant Kawle
Electronics and Telecommunication Engineering Department
Atharva College of Engineering,
Mumbai University India
yugantkawle@gmail.com

Dhanashree Pannase
Electronics and Telecommunication Engineering Department
Atharva College of Engineering,
Mumbai University India
dhanashreeannase@atharvacoe.ac.in

Abstract
In an era of rapid technological advancement, the integration of Unmanned Aerial Vehicles (UAVs) with Internet of Things (IoT) technology has revolutionized the field of environmental surveillance. This paper presents an innovative approach to real-time environmental monitoring by deploying a UAV drone equipped with IoT-enabled sensors, all controlled by the ESP8266 microcontroller. The system captures vital atmospheric data including temperature, humidity, air quality, atmospheric pressure, distance, and noise levels during flight operations. These data are then wirelessly transmitted to a cloud-based MongoDB database, which serves as a central repository for storage and retrieval. A Flask-based web application provides a real-time interactive dashboard for remote monitoring. This system presents a scalable, affordable, and efficient solution for real-time aerial environmental surveillance, particularly beneficial in applications such as smart city planning, disaster management, industrial site inspections, and agricultural analysis.

Keywords
UAV, Drone, IoT, ESP8266, Environmental Monitoring, MongoDB, Flask Dashboard, Real-Time Surveillance, Sensors, Data Acquisition.

1. INTRODUCTION

The increasing frequency of environmental hazards, air pollution, and climate variability has created an urgent need for advanced environmental monitoring systems. Conventional ground-based sensor stations provide limited coverage, especially in hazardous or hard-to-reach regions. UAV drones, when integrated with IoT infrastructure, offer a dynamic, mobile, and intelligent platform for gathering environmental data in real time.

This project leverages the mobility of a UAV platform with the compact and efficient capabilities of the ESP8266 microcontroller and various environmental sensors to create a fully functional aerial surveillance system. The drone collects real-time atmospheric and environmental metrics and relays them through Wi-Fi to a cloud storage system. A user-friendly dashboard provides visualization and remote access to the data, ensuring quick decision-making and alerts in critical situations.

This research is particularly impactful in fields like disaster zone assessment, air quality studies in congested cities, agricultural yield analysis, and remote environmental inspections in industrial zones.

2. PROPOSED SYSTEM

The proposed system aims to create a compact yet powerful drone-mounted IoT module capable of real-time environmental data collection and transmission. The key design components include:

- **ESP8266NodeMCU microcontroller** with built-in Wi-Fi for data acquisition and transmission.
- **DHT11 Sensor** for temperature and humidity.
- **MQ135 Sensor** for air quality and gas detection.
- **BMP180 Sensor** for barometric pressure and altitude.
- **VL53L0X Sensor** for distance measurement.
- **Sound Sensor** for monitoring noise pollution levels.
- **MongoDB Cloud Database** for real-time data storage.
- **Flask Web Application** for live visualization and remote monitoring.

The entire module is mounted on a quadcopter drone, which allows the system to be deployed in various outdoor and urban environments.

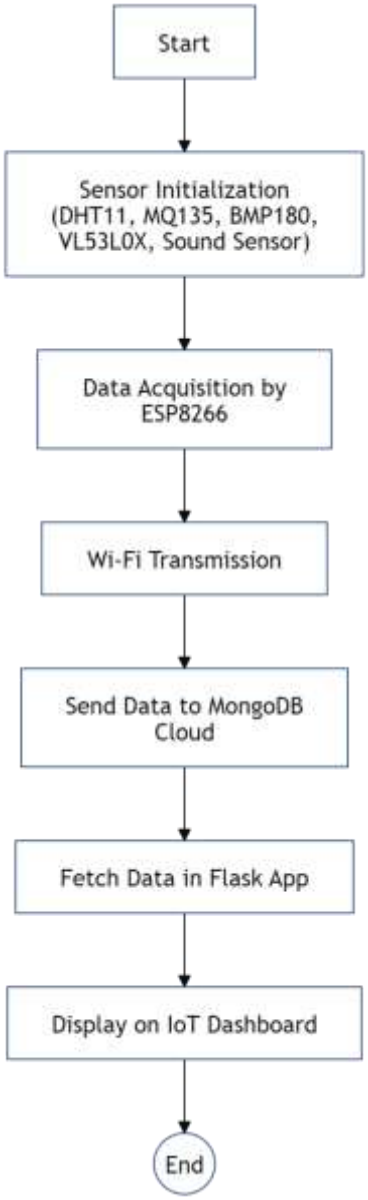

Fig 1: Flowchart

3. IMPLEMENTATION

The implementation consists of both hardware and software components carefully designed for lightweight integration on a drone. The ESP8266 handles real-time sensor readings and transmits them using built-in Wi-Fi.

Hardware Components:

- ESP8266 NodeMCU
- DHT11: Measures temperature and humidity.

- MQ135: Monitors air quality and pollution.
- BMP180: Captures barometric pressure and altitude.
- VL53L0X: Laser-based distance sensor.
- Sound Sensor: Measures ambient noise level.
- Drone Platform with battery.

Software Components:

- Arduino IDE for ESP8266 firmware.
- Python with Flask for dashboard development.
- MongoDB for cloud-based data storage.
- REST APIs and MQTT for communication.

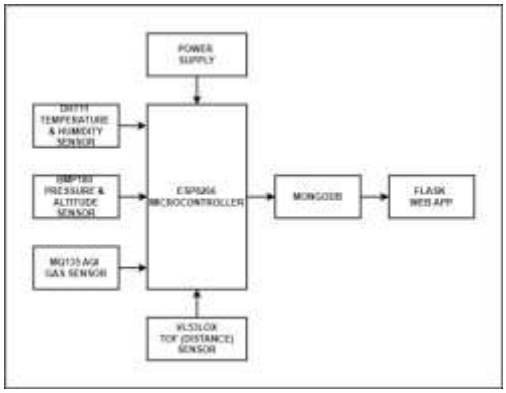

Fig 2: Block Diagram

4. Design, Working, and Process Details

This section provides an in-depth look at the system's internal structure and behaviour. The IoT-based aerial surveillance system mounted on a UAV drone is designed for real-time environmental monitoring, and its functionality is broken down into stages—data collection, transmission, storage, and visualization. To simplify and represent this process, four essential diagrams have been created: the Context Level Diagram (Level 0 DFD), Data Flow Diagram (Level 1 DFD), Sequence Diagram, and Control Flow Diagram. These diagrams not only visualize system flow and interaction but also highlight the logical processes behind data movement and decision-making within the system.

4.1 Context Level Diagram

The Context Level Diagram presents a top-level overview of the drone-based surveillance system as a single process interacting with external entities. The system takes input from real-time environmental sensors mounted on the drone, processes the data using the ESP8266 microcontroller, and transmits it via Wi-Fi to a cloud-based MongoDB database. This central process is then accessed by an external user through a web-based dashboard powered by Flask.

The user interacts with the system by requesting environmental data via the dashboard interface. In response, the system fetches the latest records from the cloud database and returns them to the user in a user-friendly format such as graphs, values, or alerts. This context diagram highlights the flow of information into and out of the system, focusing on its relationship with the user and the external cloud database.

Fig 3: Context Level Diagram

4.2 Data Flow Diagram

The Level 1 Data Flow Diagram breaks down the overall system process into smaller, more detailed sub-processes that explain how sensor data is handled internally. The process begins with the environmental sensors—DHT11, MQ135, BMP180, VL53L0X, and the sound sensor—collecting data at regular intervals. These raw values are received by the ESP8266 microcontroller, which acts as the central processing unit for the drone's IoT system.

The ESP8266 formats the sensor readings into structured data packets, usually in JSON format,

and establishes a Wi-Fi connection to transmit this data to the cloud-based MongoDB server. The database acts as a central data repository, storing all incoming records for later analysis. The Flask web application regularly queries this database and updates the IoT dashboard in real time. The user can then visualize live data such as temperature, air quality, and altitude through this dashboard. The DFD ensures a better understanding of how data flows between each component of the system.

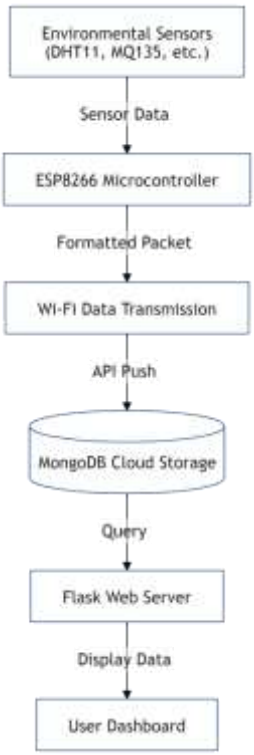

Fig 4: DFD Diagram

4.3 Sequence Diagram

The Sequence Diagram illustrates the chronological order in which operations take place during one cycle of data monitoring, starting from sensor activation to data visualization on the dashboard. The diagram focuses on the interaction between the key objects and entities: the user, the onboard sensors, the ESP8266 microcontroller, the MongoDB cloud database, and the Flask-based web application.

The sequence begins when the drone is powered on, triggering the ESP8266 to initialize and request data from each sensor. The sensors respond with real-time values, which are captured and formatted by the microcontroller. Once the Wi-Fi connection is verified, the ESP8266 sends the sensor data to MongoDB. The Flask server, which continuously polls the database, retrieves the new data and refreshes the dashboard. Finally, the user views the data through the web interface. This sequence ensures that users receive real-time updates and can respond immediately to abnormal environmental conditions.

Fig 5: Sequence Diagram

4.4 Control Flow Diagram

The Control Flow Diagram (CFD) shows the logical decision-making process and the structured flow of control within the system. It visually represents how the program logic moves from one stage to another, ensuring that each sensor reading is handled properly, and decisions such as uploading data or retrying connections are carried out as per defined conditions.

At startup, the drone powers up the ESP8266 and all connected sensors. Each sensor is initialized and begins to collect data. The ESP8266 checks whether it is connected to Wi-Fi. If connected, it formats the data and pushes it to the MongoDB database. If not, the program enters a loop to reattempt Wi-Fi connection until successful. Once the data is successfully transmitted, the Flask application fetches the latest values and displays them. This diagram helps in understanding how control is maintained throughout the system, especially in real-time situations where network failures or sensor errors could occur.

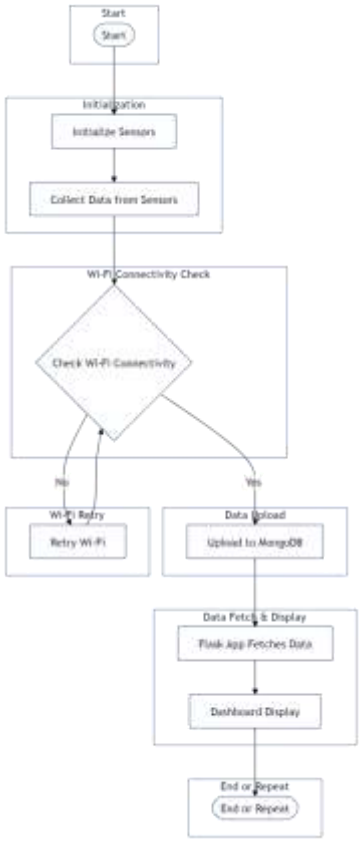

Fig 6: Control Flow Diagram

5. FUTURE SCOPE

There is significant room for enhancement in this project:

- **GPS Integration:** For mapping sensor data with location.
- **Camera Module:** Add visual surveillance with live feed.
- **LoRa/NB-IoT Communication:** Extend range for rural and remote deployment.
- **Mobile App:** Real-time mobile notifications and data access.
- **AI Integration:** Use machine learning for pattern recognition, forecasting, or anomaly detection.
- **Weatherproof Housing:** Allow drone to operate under rain and varying temperatures.
- **Swarm UAV Network:** Use multiple drones to cover larger regions and share data in mesh format.

With these improvements, the UAV system can evolve into a complete smart surveillance and response platform.

6. CONCLUSION

The proposed aerial surveillance system combines the mobility of UAVs with the power of IoT to offer a flexible, real-time environmental monitoring solution. By integrating lightweight sensors with the ESP8266 and cloud-based dashboard visualization, the system ensures accurate and accessible environmental surveillance. This low-cost, scalable, and adaptable design makes it suitable for a variety of fields including disaster management, agriculture, industry, and urban development.

Its modularity and expandability ensure long-term usability and practical real-world application potential.

7. REFERENCES

- Chen, S., Yu, Z., Li, X., & Wang, Z. (2018). An Agricultural Weather Drone Monitoring System Based on IoT Technology. *IEEE Access*, 6, 6651–6659.
- Li, Y., Geng, H., Qiao, X., & Feng, J. (2017). An IoT-Based Autonomous Weather Monitoring System. *IEEE Internet of Things Journal*, 4(2), 369–377.
- Park, Y., Kim, Y., & Park, K. (2021). Development of an Autonomous Weather Data Monitoring System Using Drone-Based IoT. *Sensors*, 21(2), 647.
- Rizvi, S. S. S., Shah, M. A., & Khan, M. U. K. (2019). Internet of Things-Based Weather Monitoring System using Drones. *ICECTA*, IEEE.