Arduino: Build an I.O.T. Environment Monitor

A synopsis report submission for the first-year project of

Master of Technology in Computer Science and Engineering

By

Sameer Dixit(Roll No.: 230216008)

UNDER THE SUPERVISION OF

Dr. Rashi Agarwal

Associate Professor

SUBMITTED TO:

Dr. B B Sagar (Associate Professor)

Dr. Shashwati Banerjea (Associate Professor)

December 13, 2024

Figure 1: DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING Harcourt Butler Technical University, Kanpur

Project Report: Department of Computer Science and Engineering

Arduino: Build an IoT Environment Monitor

Introduction

In today's rapidly evolving world, environmental monitoring plays a crucial role in ensuring the well-being of our surroundings. This project focuses on building an Internet of Things (IoT) environment monitor using Arduino. The device collects data from various sensors and transmits it to the cloud for real-time tracking. This project aims to monitor key environmental parameters such as temperature, humidity, air quality, and light intensity. The scope of this project covers the use of Arduino as the hardware platform, interfacing it with sensors, and leveraging IoT technology for data transmission and visualization. IoT environmental monitors play a significant role in real-time tracking of conditions. On the other hand, the cloud computing space is filled with numerous cloud service providers (CSPs) that offer varying levels of performance, pricing, and reliability that we will be discussing later in our report. Machine learning models provide a powerful tool to rank CSPs based on user requirements.

Motivation

IoT-based environmental monitoring systems have been explored in various applications, from smart homes to industrial systems. Existing solutions primarily focus on temperature and humidity monitoring, but recent advancements have incorporated air quality and light sensors as well. This project builds on these technologies, offering a cost-effective solution with real-time monitoring capabilities.
Similarly, cloud service providers play a critical role in delivering infrastructure, platform, and software services. A large body of research has focused on comparing CSPs based on key performance indicators like cost, reliability, and scalability. Machine learning has emerged as a key approach in evaluating CSPs due to its ability to process large datasets and

identify patterns.
Similarly, cloud service providers play a critical role in delivering infrastructure, platform, and software services. A large body of research has focused on comparing CSPs based on key performance indicators like cost, reliability, and scalability. Machine learning has emerged as a key approach in evaluating CSPs due to its ability to process large datasets and identify patterns.

Problem Statement

Cloud Service Providers There are numerous CSPs across the globe with different services, pricing models, and infrastructures.
The list of major CSPs across the globe. As per Gartner's Magic Quadrant for Cloud Infrastructure and Platform Services published in 2021Get this report: Gartner's Magic Quadrant for Cloud Infrastructure and Platform Services.

1. Amazon Web Services (AWS),
2. Microsoft Azure,
3. Google Cloud Platform (GCP),
4. IBM Cloud.

Once the top four vendors are identified, the primary goal is to choose the most appropriate one among them for user's software requirement specification.

Components and Tools: The components and tools used in this project are categorized into hardware and software:

Hardware

- Arduino Uno
- DHT11/DHT22 (Temperature and Humidity Sensor)
- MQ135 (Air Quality Sensor)

- Light Sensor (LDR)
- Wi-Fi Module (ESP8266/ESP32)
- Jumper Wires and Breadboard

Software

- Arduino IDE for coding
- ThingSpeak/Blynk for IoT platform
- APIs for data visualization

Abstract

The system consists of sensors connected to the Arduino board, which processes the data and sends it to an IoT platform via the Wi-Fi module.

Block Diagram

The block diagram includes sensors (DHT11, MQ135, LDR), Arduino, Wi-Fi module, and the cloud for data storage and visualization.

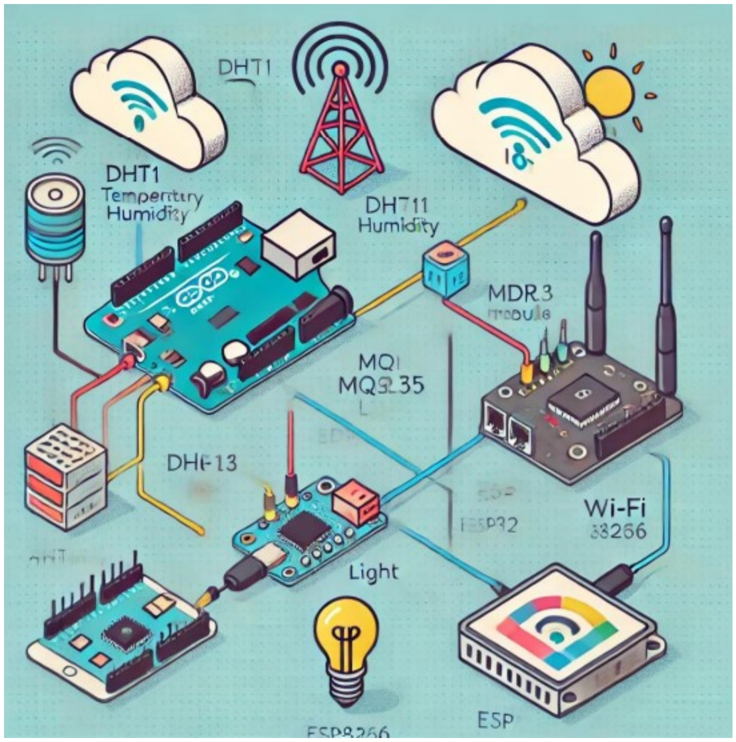

Figure 2: Block Diagram

Flowchart

1. **Sensors collect environmental data**
2. **Data is processed by Arduino**
3. **Wi-Fi module sends data to the cloud**
4. **The user accesses the data via an IoT dashboard**

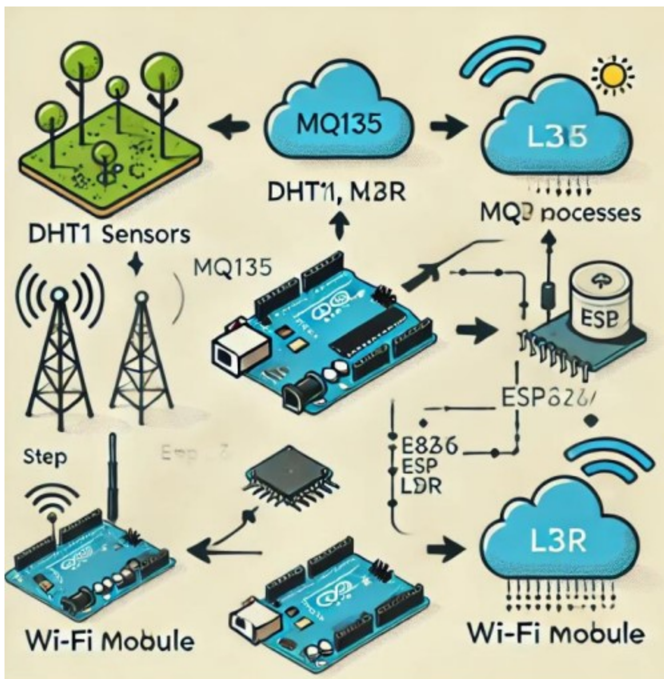

Figure 3: Sensors collect environmental data

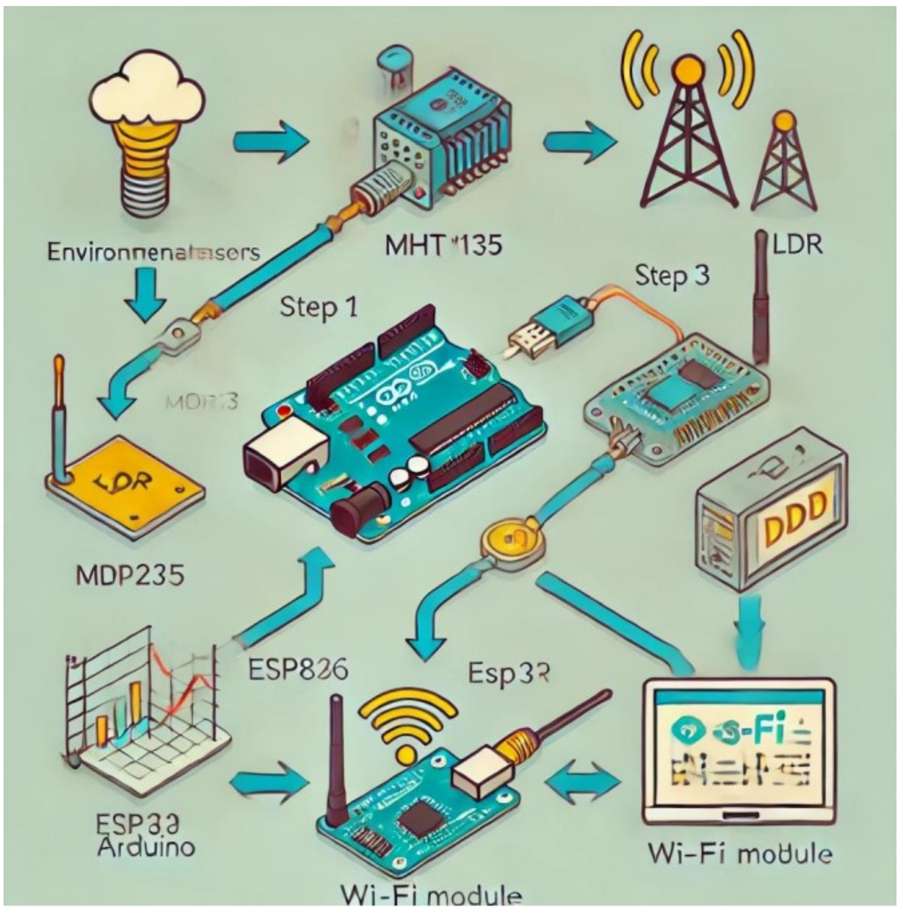

Figure 4: Data is processed by Arduino

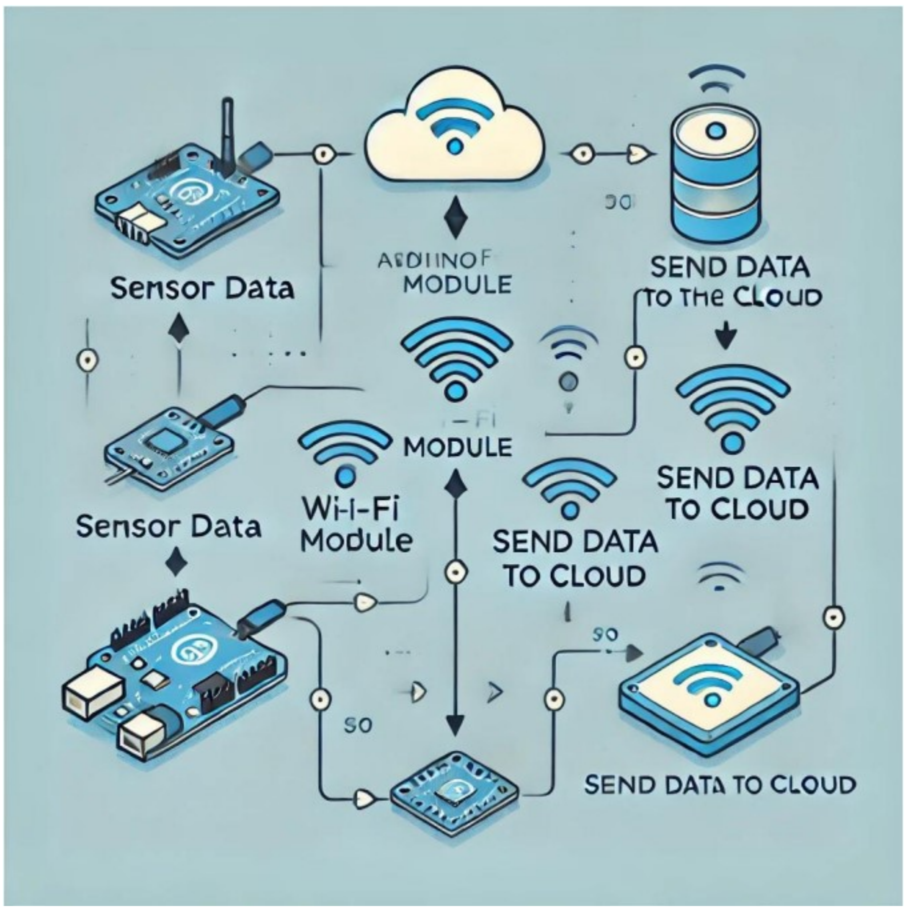

Figure 5: Wi-Fi module sends data to the cloud

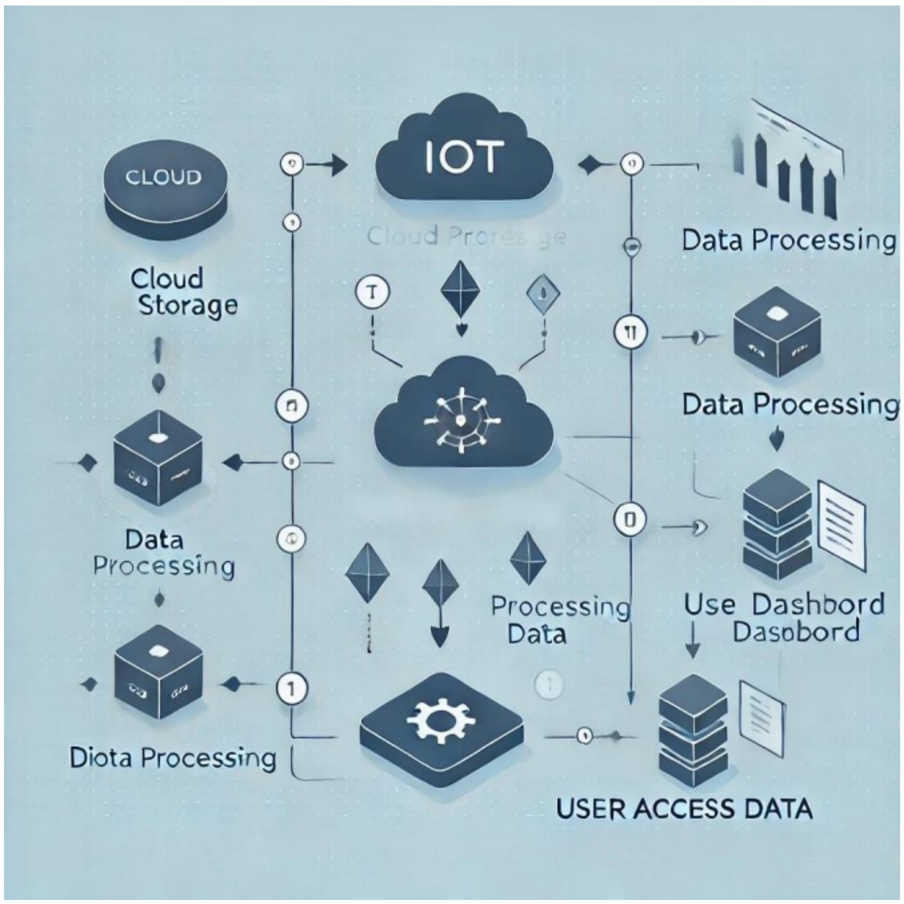

Figure 6: Wi-Fi module sends data to the cloud

0.1 Goals and Objectives with Circuit Design and Assembly

The circuit involves connecting the sensors to the respective pins of the Arduino board, and setting up the Wi-Fi module for internet connectivity.

Power is supplied through a USB connection or a dedicated power source, and appropriate jumper wires are used to establish connections between the components.

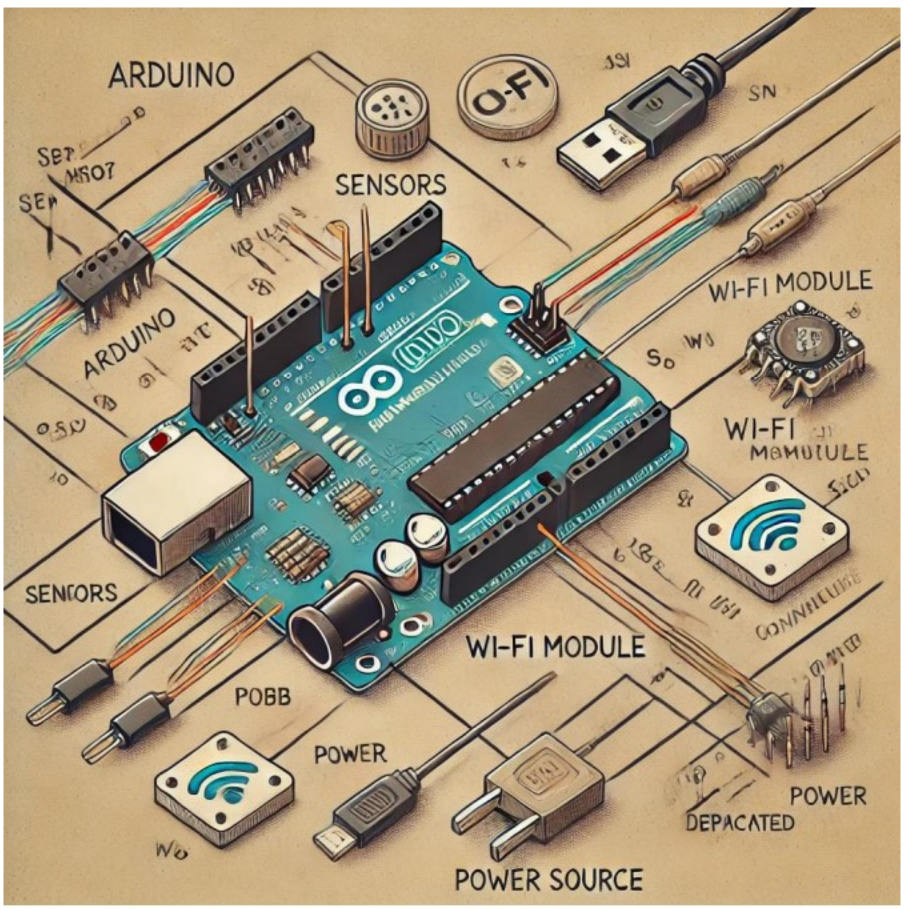

Figure 7: Circuit Design and Assembly

0.1.1 Software Implementation

The Arduino code initializes the sensors, collects data, and sends it to the IoT platform via the Wi-Fi module. Below is a brief code snippet to illustrate how the temperature and humidity sensor (DHT11) is initialized.

Listing 1: DHT11 Code

```
void setup() {
        Serial.begin(9600);
        dht.begin();
}

void loop() {
        float humidity = dht.readHumidity();
        float temperature = dht.readTemperature();
}
// Send data to cloud
}
```

The data collected from the sensors is sent to an IoT platform like ThingSpeak or Blynk. This platform provides real-time monitoring and historical data visualization.

Steps to Set Up ThingSpeak/Blynk

1. Create an account on the platform.
2. Set up a new channel for data logging.
3. Configure the Arduino code to send data to the platform using API keys.

0.2 Literature Survey and Research Methodology

- **Wide Detection Range**: It can detect a range of harmful gases, making it suitable for both industrial and domestic applications.

- **Sensitivity**: The sensor provides an analog output corresponding to the concentration of gases, allowing for precise monitoring.
- **Low Cost**: MQ135 is one of the more affordable gas sensors, making it popular for DIY projects, IoT systems, and academic purposes.
- **Durability**: It has a long lifespan and works reliably in various environmental conditions.
- **Easy to Interface**: It can easily be integrated with micro controllers like Arduino, Raspberry Pi, or other IoT platforms via its analog and digital pins.

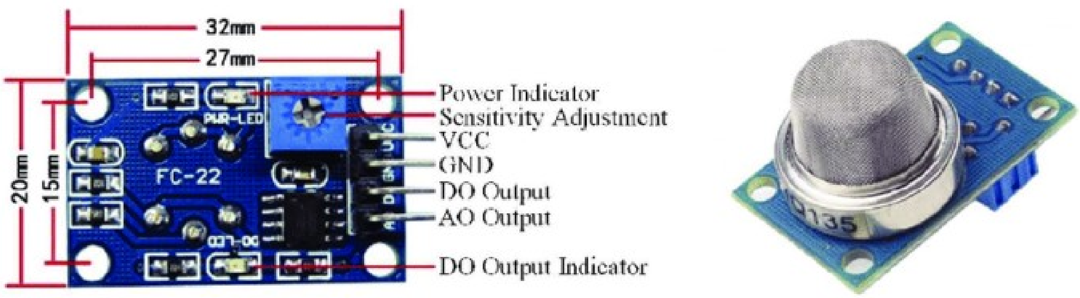

Figure 8: Key Features and Advantages of the MQ135 Gas Sensor

Applications

- **Air Quality Monitoring**: Commonly used in IoT-based systems to monitor indoor air quality, especially in homes, offices, and schools.
- **Industrial Safety Systems**: Used to detect the presence of harmful gases in factories or workplaces to ensure safety standards.
- **Smart Homes**: Integrated with home automation systems for real-time air quality checks and triggering alarms or ventilation systems if unsafe gas levels are detected.
- **Environmental Monitoring**: Applied in environmental stations to detect pollution levels in urban areas or near industrial zones.

IoT Cloud Integration

The data collected from the sensors is sent to an IoT platform like ThingSpeak or Blynk. This platform provides real-time monitoring and historical data visualization.

Steps to set up ThingSpeak/Blynk:

1. Create an account on the platform.
2. Set up a new channel for data logging.
3. Configure the Arduino code to send data to the platform using API keys.

Testing and Results

The environment monitor was tested in different conditions to measure temperature, humidity, air quality, and light levels. The following graphs show the data collected during the testing phase, highlighting variations in environmental factors over time.

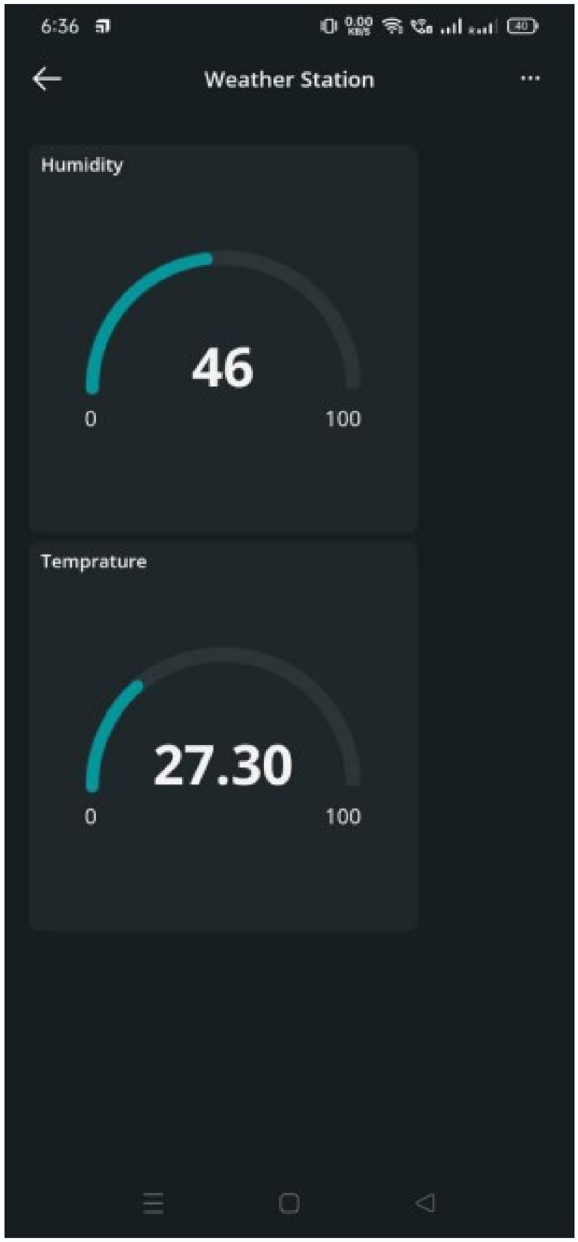

Figure 9: Environmental Data Analysis

Applications and Future Scope

This project has applications in smart homes, industries, agriculture, and environmental research. Future work may include integrating additional sensors (e.g., CO_2 sensors) and leveraging machine learning for predictive environmental analysis.

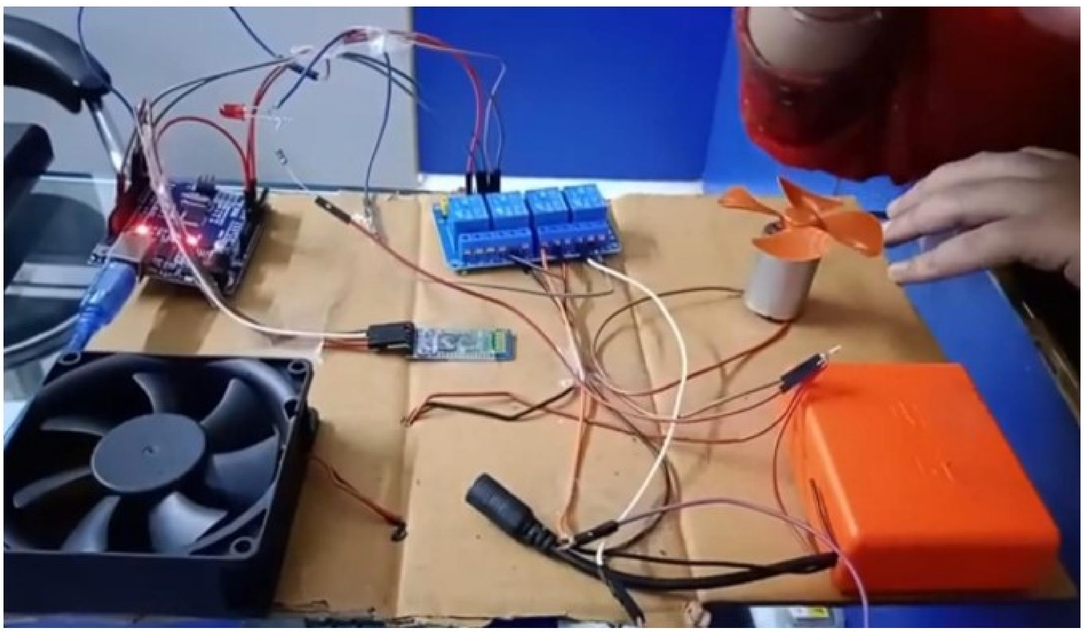

Figure 10: Smart solutions for environmental research with future sensor and AI enhancements.

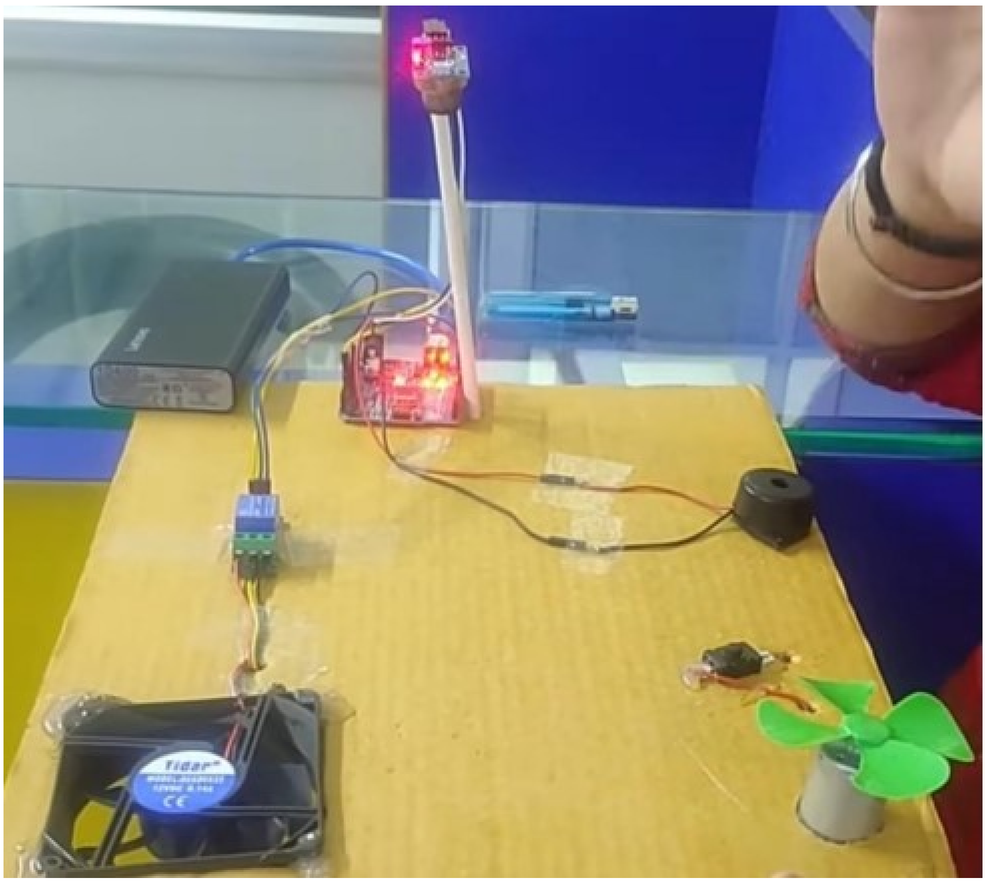

Figure 11: Environmental monitoring with potential for sensor upgrades and predictive analysis.

Conclusion

In conclusion, this project successfully demonstrates the integration of Arduino with IoT for real-time environmental monitoring. It offers a cost-effective solution for tracking various environmental parameters and provides significant opportunities for future improvements.