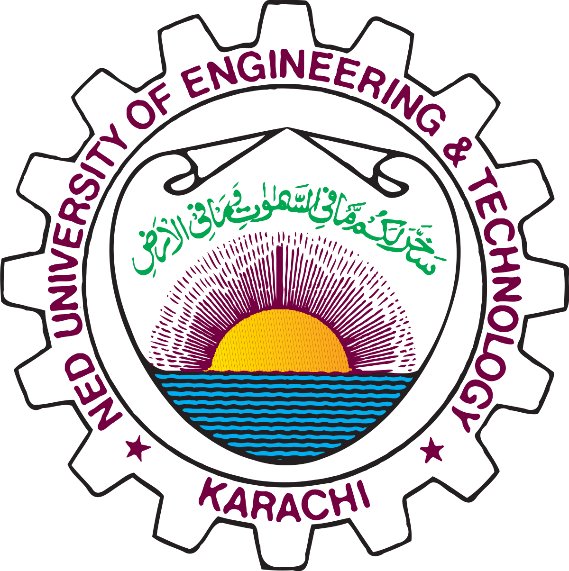
Mini Project Report

Microprocessor Based System Design CS-301

Third Year-Computer and Information Systems Engineering

Batch: 2022



Group Members:

Ali Mehdi ……………………………………………………. (CS-22140)

Hashir Badar …………………………………………………. (CS-22150)

Submitted to: Dr. Muhammad Khurram

Submission Date: 19/11/2024



# **Abstract**

This project implements a smart air quality monitoring system using an ESP32 microcontroller with DHT11 sensors to measure temperature and humidity. The system features a Flask-based web interface that displays real-time environmental data and enables Over-The-Air (OTA) firmware updates. Key components include a SQLite database for data logging, a responsive web dashboard with alert thresholds, and secure device authentication. The solution provides remote monitoring through a web interface with interactive data visualization, while the OTA update functionality allows for seamless firmware deployment without physical access to devices. The system demonstrates effective integration of IoT hardware with web technologies, implementing security features like password authentication and encrypted sessions. Designed for both home and industrial environments, AirSync offers reliable environmental monitoring with maintenance-free operation through its self-updating capability, representing a complete IoT solution from sensor to user interface.

# **Introduction**

## **Context and Motivation**

With increasing concerns about environmental conditions and indoor air quality, this project implements an IoT-based monitoring system that collects, transmits, and visualizes temperature and humidity data in real-time. Air quality directly impacts health, productivity, and equipment performance, making continuous monitoring essential for residential, industrial, and commercial environments. Traditional monitoring systems often lack remote accessibility and real-time analytics, which this project addresses by integrating embedded sensors, wireless communication, and a cloud-based dashboard.

## **Objectives**

The primary goals of this project are:

1. **Hardware Integration:**

* Interface DHT11 sensor with ESP32 microcontroller for accurate temperature/humidity readings.
* Ensure robust Wi-Fi connectivity for data transmission.

1. **Software Integration:**

* Design a Flask-based backend to process sensor data.
* Develop an interactive frontend dashboard (HTML/CSS/JS) for real-time visualization.

1. **Data Management:**

* Store readings in SQLite database for historical analysis.
* Display the latest 10 records dynamically on the web interface.

1. **Security and Maintenance:**

* Implement a login system to prevent unauthorized access.
* Enable OTA (Over-the-Air) firmware updates for seamless maintenance.

## **Scope of the Project**

The system covers the following functionalities:

* Real-time sensor data acquisition (temperature, humidity).
* Wireless data transmission via HTTP/MQTT protocols.
* Web-based dashboard with authentication and responsive design.
* Local database storage for audit and analysis.
* OTA updates to enhance system longevity.

Limitations:

* The DHT11 sensor has limited precision (±2°C, ±5% RH).
* The system currently monitors only two parameters (expandable to CO₂, PM2.5, etc.).
* No mobile app (web-only access).

## **Significance of the Work**

The project demostrates:

* Cost-Effectiveness: Uses affordable components (~$5 ESP32 + DHT11).
* Scalability: Modular design supports additional sensors (e.g., gas detectors).
* User-Centric Design: Intuitive dashboard for non-technical users.
* Research Potential: Provides a foundation for smart city or industrial IoT applications.

By bridging the gap between physical sensors and cloud analytics, this system offers a template for future IoT environmental monitors.

# **System Architecture**

## **Overall System Design**

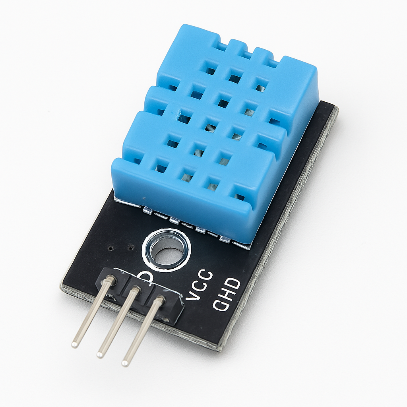
The system follows a three-layer architecture

## **Hardware Components**

1. **Hardware Integration:**

* Wi-Fi 802.11 b/g/n + Bluetooth 4.2.
* Dual-core 32-bit LX6 CPU (240 MHz).
* 512 KB SRAM, 4MB Flash.
* GPIO pins for sensor interfacing.

1. **DHT11 Sensor:**

* Temperature Range: 0–50°C (±2°C accuracy).
* Humidity Range: 20–90% RH (±5% accuracy).
* Sampling Rate: 1Hz (1 reading per second).

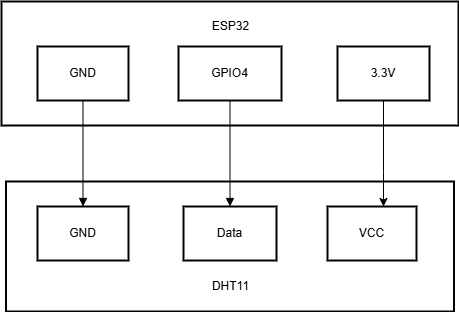
## **Software Components**

1. **Frontend:** HTML, CSS, JavaScript
2. **Backend** Pyhton Flask framework
3. **Database:** SQLite
4. **Firmware:** Arduino IDE (C++)

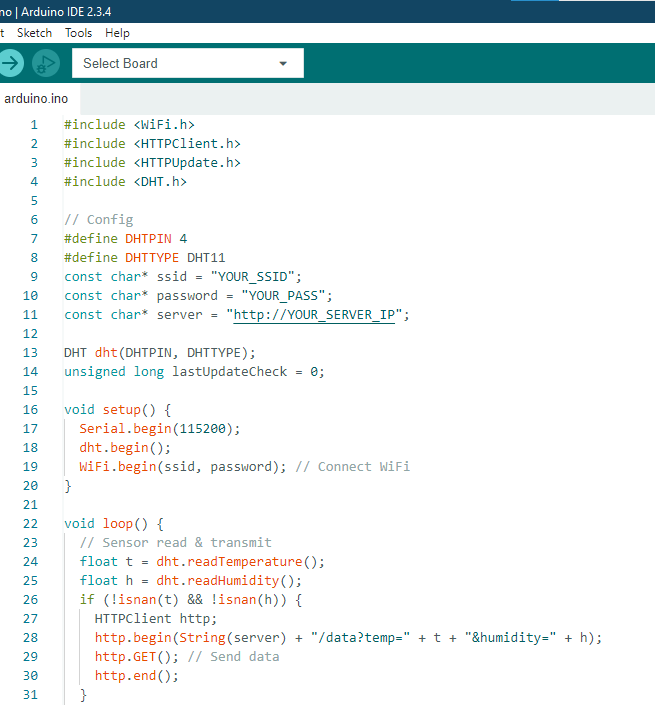
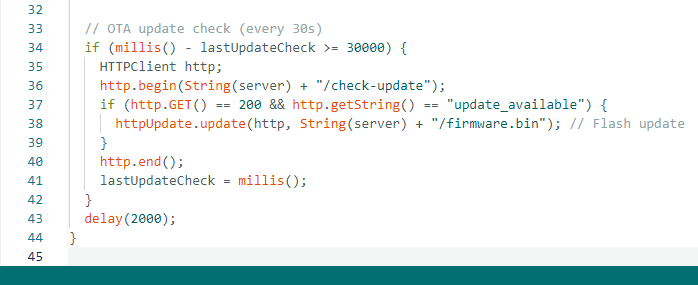
# **System Architecture**

## **Hardware Implementation**

ESP32 (3.3V logic) connected to DHT11 (5V powered).



## **Firmware Development**

Code Structure:

­­

**Key features:**

* Wi-Fi Setup: Auto-connects in setup()
* Sensor Handling: DHT11 reads with NaN validation
* Data Transmission: HTTP GET with URL parameters
* OTA Updates: Server-triggered firmware flashing
* Timing: Non-blocking 30s update checks

Full code in GitHub mentioned at the end of the report.

## **Web Application Development**