

**Design Task 04**  
**Intelligent Air Quality Monitoring System**  
**EC 8010 – Design Proficiency**  
**Group 29**



**SUBMITTED BY:**

DILSHAN W.N.M.A.M. (2020/E/033)

DILUSHANTH S.(2020/E/034)

DINESH V. (2020/E/035)

**FACULTY OF ENGINEERING**

**UNIVERSITY OF JAFFNA**

**OCTOBER 2025**

## 1. Abstract

This project covers the design and implementation of an intelligent air quality monitoring system based on a microcontroller capable of real-time measurement and control of indoor air quality. Integrating environmental sensors and occupancy detectors provides the basis for measuring air quality and automatic adjustment of ventilation via a Pulse Width Modulation controlled fan. Data is recorded with timestamps onto an SD card for analysis later, while current readings are displayed on an LCD. With this design, efficient indoor ventilation management is attained with a scalable basis for future IoT-based environmental control systems.

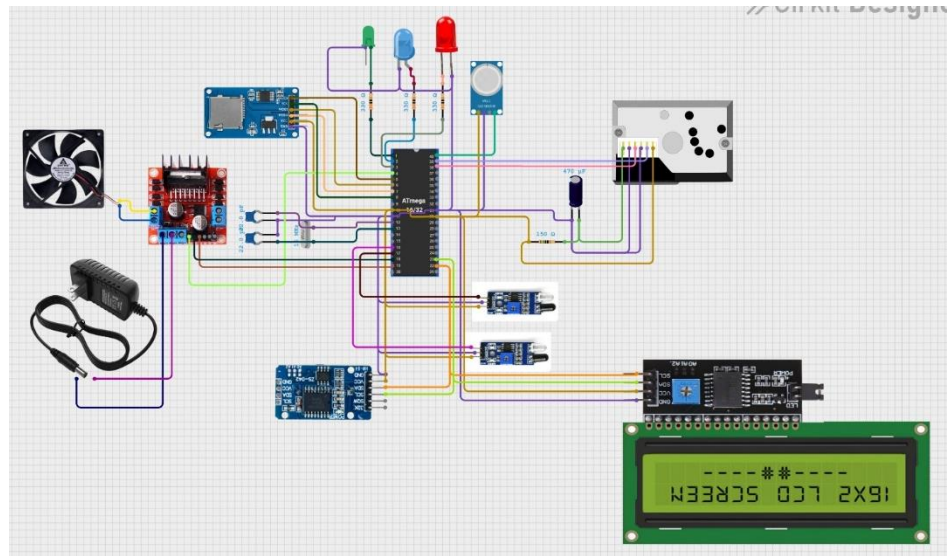
## 2. Introduction

Indoor air quality significantly influences human health and comfort. It is well known that fatigue, poor concentration, and respiratory problems may be caused by prolonged exposure to high concentrations of carbon dioxide ( $\text{CO}_2$ ), volatile organic compounds (VOCs), or particulate matter (PM2.5/PM10). Most buildings have manual control of their ventilation systems, which is usually inefficient and a waste of energy. This project proposes the design and implementation of an intelligent air quality monitoring system based on a microcontroller to automatically adjust the ventilation fan speed depending on the pollution level and room occupancy. The design focuses on real-time data acquisition, embedded signal processing, and power-efficient control using Pulse Width Modulation.

## 3. Objectives

- Acquire air-quality and occupancy data in real-time using multiple sensors.
- Dynamically regulate fan speed using PWM proportional to pollution and occupancy level.
- Provide visual status through an LCD display.
- Log sensor data with timestamps onto an SD card for later analysis.
- Operate as a standalone embedded system without using Arduino, ESP32, or Raspberry Pi platforms.

## 4. System Design



*Figure 1:Simulation Design*

## 4.1. Overall Design

The system is designed to continuously monitor indoor air quality and automatically control ventilation using sensor feedback. It consists of five major subsystems Sensing, Processing, Actuation, Indication, and Data Logging/Communication.

The ATmega32A microcontroller acts as the central processing unit, reading real-time data from all sensors, processing the information, and controlling the fan speed according to the detected air quality and occupancy.

### Sensing Subsystem

The sensing part includes several modules that provide environmental and occupancy data:

#### **MQ135 Gas Sensor (connected to ADC0):**

This sensor measures the concentration of gases such as CO<sub>2</sub>, ammonia, benzene, and smoke. Its analog voltage output changes according to the presence of pollutants. It gives a general indication of air quality in parts per million (PPM).

#### **DHT11 Temperature and Humidity Sensor (connected to PD2):**

The DHT11 measures ambient temperature and relative humidity. These parameters affect air quality and human comfort, so they are used to adjust the final air quality index. The sensor transmits a digital signal containing both temperature and humidity data.

#### **IR Sensors (connected to PD3 and PD4):**

Two infrared sensors are placed near the room entrance to detect people entering or leaving. The number of people directly influences the air quality due to CO<sub>2</sub> emissions from breathing. The two-sensor arrangement allows detection of direction — whether a person enters or exits.

#### **DS3231 Real-Time Clock (RTC) Module (I<sup>2</sup>C interface):**

This highly accurate RTC module provides real-time date and time information, which is used for time-stamped data logging on the SD card. Unlike the DS1302, the DS3231 has temperature-compensated crystal oscillators, ensuring better accuracy and lower drift.

### Processing Unit

The ATmega32A microcontroller is the heart of the system. It performs the following key operations:

- Reads analog and digital sensor data at regular intervals (every 50 milliseconds).
- Converts the analog sensor readings into numerical values (PPM, °C, %, etc.).
- Calculates the total air quality “risk” percentage by analyzing gas, humidity, and occupancy data.
- Generates PWM signals for controlling the fan motor.
- Manages the display, data logging, and Wi-Fi communication simultaneously.

The ATmega32A was chosen because it provides all necessary features 10-bit ADCs for analog sensors, hardware PWM for smooth fan control, I<sup>2</sup>C for RTC, SPI for SD card, and UART for Wi-Fi module. It runs at 16 MHz, providing enough processing speed to handle all real-time tasks efficiently.

## **Actuation System**

The microcontroller adjusts the fan speed according to the air quality level.

Example of fan control logic:

- Risk < 20%: Fan OFF (air quality good)
- Risk 20–50%: Fan speed medium (PWM  $\approx$  150)
- Risk > 50%: Fan speed maximum (PWM = 255)

PWM (Pulse Width Modulation) is generated using Timer1 on ATmega32A. PWM allows smooth variation of motor speed by changing the width of voltage pulses rather than turning the fan fully ON or OFF.

## **Indication and Display System**

The indication and display system provides real-time visual and text-based feedback about the air quality and system status. Three LEDs are used to show the air condition clearly. The green LED indicates good air quality when the total risk level is below 20%, the yellow LED represents moderate air quality between 20% and 50%, and the red LED turns on when the risk level exceeds 50%, showing poor air quality that requires ventilation.

A 16x2 I<sup>2</sup>C LCD display is used to show detailed information such as gas concentration (in PPM), temperature, humidity, number of people detected, total risk percentage, fan status (OFF / MEDIUM / HIGH), and the current date and time from the RTC module. The display automatically switches between two screens every two seconds—one showing sensor readings and the other showing the time and fan operation. This allows users to quickly understand environmental conditions and system responses at a glance.

## **Data Logging and Communication System**

The data logging and communication unit records sensor data and sends it for remote monitoring and analysis. The SD card module, connected through the SPI interface, saves readings such as date, time, number of people, gas concentration, temperature, humidity, risk percentage, and fan speed into a CSV file at one-minute intervals. This helps to maintain long-term data without damaging the SD card due to frequent writes.

# **5. Detailed Design**

## **5.1. Gas Sensor Interface (MQ135)**

The MQ135 sensor detects harmful gases like CO<sub>2</sub>, NH<sub>3</sub>, and smoke in the air. It works by changing its internal resistance when exposed to gases. The sensor is powered with 5V and connected to an analog pin of the ATmega32A. Using the MQ135 library, the analog reading is automatically converted to gas concentration in PPM (parts per million). These readings help measure the overall air quality level.

## **5.2. Temperature and Humidity Sensor (DHT11)**

The DHT11 sensor measures both temperature and humidity in the environment. It sends digital signals directly to the microcontroller using a single data pin. The temperature and humidity values are displayed on the LCD and also stored on the SD card. These parameters help in identifying heat or moisture buildup inside a room that can affect comfort and air quality.

## **5.3. People Counting System (IR Sensors)**

Two infrared sensors are placed at the entrance to count people entering and leaving. When a person crosses the first IR beam and then the second, it is recorded as entry, while the reverse order counts as exit. This helps the system understand how crowded the room is. To avoid wrong counts, a small delay and timeout are used so the system resets if someone stops halfway.

#### 5.4. Real-Time Clock (RTC DS1302)

The RTC module provides accurate date and time data even when the main power is off. It uses a small backup battery to keep time running continuously. The time data is used for stamping logs in the SD card and for displaying current time on the LCD, making all records organized and easy to track.

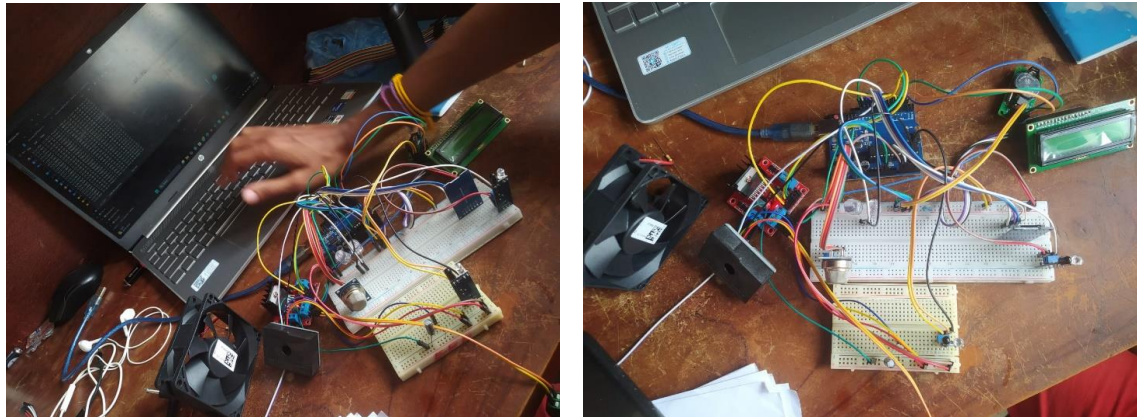
#### 5.5. Data Logging System (SD Card Module)

Every time the main loop executes (approximately every 50 milliseconds), the system writes a log entry to the SD card. Each entry includes:

- Date and time from the RTC
- People count
- Gas concentration in PPM
- Dust density in  $\mu\text{g}/\text{m}^3$
- Calculated total risk percentage

The SD card file is opened, data is written, and the file is immediately closed. Closing the file after each write ensures the data is saved even if power is suddenly lost. However, logging 20 times per second creates a significant issue.

SD cards have limited write endurance - typically 100,000 writes per sector. At 1,400 writes per minute. The assignment specification calls for 1-minute logging intervals, which would extend the SD card life to over 10 years. This is an important modification that should be implemented.



*Figure 2: Implementation Design*

#### 5.6. Communication Module

The Wi-Fi module allows the system to send data wirelessly to an IoT dashboard or cloud server. This helps in remote monitoring of air quality and occupancy through a mobile or computer. The Wi-Fi link can also be used to send simple control commands, like turning the fan on or off from the web interface.

#### 5.7. Microcontroller (ATmega32A)

The ATmega32A is the main controller that connects and manages all sensors and modules. It reads data, performs risk calculations, controls the display and LEDs, logs data, and communicates through Wi-Fi. It acts as the brain of the entire system, ensuring all components work together smoothly.

## 6. Cost Analysis

Component	Cost(LKR)
ATMega32	970
MQ 135	360
GP2Y	1500
Wifi module	205
RTC	220
PIR sensor	230
Oscillator	40
Display	680
100nF	9
2pF	8
15pF	4
18pF	8
20pF	8
potential meter	30
DC jack	30
12V fan	250
L298N	200
IRF540N	70
LED	40
Jumper wire	170
Resistor+usbisp+ courior	100
Total	5132

## 7. Limitations, Challenges, And Future Work

### 7.1 Limitations

- Sensor readings (MQ-135, dust sensor) are not highly accurate and affected by temperature and humidity.
- IR-based people counting may miss detections when multiple people pass simultaneously.
- Single-point sensing cannot represent large room air quality variations.
- SD card logging too frequent, reducing card lifespan.
- Code contains blocking delays affecting real-time response.

### 7.2 Challenges Faced

- Designing the wiring diagram: Managing multiple sensors, power lines, and modules on a single board caused connection errors and noise interference.
- Component selection: Finding compatible, locally available, and affordable components that meet project requirements was difficult.
- Communication protocol selection: Choosing between I2C and SPI for multiple modules required testing to avoid data conflicts.
- Power management: Ensuring stable 5V and 12V supply for sensors and fan without voltage drops.

- Code integration: Combining several libraries (LCD, RTC, SD, sensors) led to memory and timing issues.

### **7.3 Future Work**

- Replace delay functions with non-blocking timing (millis).
- Add DHT22 for temperature and humidity monitoring.
- Reduce SD card logging rate to once per minute.
- Upgrade to NDIR CO<sub>2</sub> and laser dust sensors for higher accuracy.
- Improve occupancy detection using time-of-flight or camera sensors.
- Expand to multi-room systems with wireless communication and centralized control.

## **8. Conclusion**

The intelligent air quality monitoring system was successfully designed and implemented using the ATmega328A microcontroller. It effectively measures gas concentration, dust levels, and room occupancy to control ventilation automatically and maintain healthy air quality. The system operates at a low cost while demonstrating key embedded design concepts such as sensor interfacing, PWM control, data logging, and real-time decision-making.

Although minor issues like sensor accuracy and timing delays exist, the design provides a solid foundation for future improvements such as IoT integration, advanced sensors, and multi-room monitoring. Overall, the project achieves its goal of creating a practical, intelligent, and affordable air quality management system.