IJARSCT

International Journal of Advanced Research in Science, Communication and Technology

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

ISSN: 2581-9429

Volume 5, Issue 4, May 2025

Impact Factor: 7.67

Automated Environmental Control for Energy Efficiency in Malls

Dr. N. Sree Divya[1], Bhavana Deyyam[2], Ch. Laxmi Pravalika[3]

Assistant Professor, Mahatma Gandhi Institute of Technology, Hyderabad, India[1]

UG Student, Mahatma Gandhi Institute of Technology, Hyderabad, India[2,3]

nsreedivya_it@mgit.ac.in, bhavana20030810@gmail.com, pravalika.chinnolla.92@gmail.com

Abstract: *An intelligent and scalable environmental control system has been developed to optimize energy consumption in malls using real-time sensor data and automated control. The prototype leverages an Arduino microcontroller integrated with IR sensors for people counting, a DHT11 sensor for temperature and humidity, and an LDR for light intensity detection. Lighting is represented using LEDs, which activate based on the number of occupants, while fan speed is adjusted according to ambient temperature. The ESP8266 module hosts a wireless local dashboard, enabling real-time monitoring of occupancy, environmental conditions, and device states. By dynamically adjusting resources based on human activity and environmental inputs, the system enhances energy efficiency and promotes sustainability in commercial spaces.*

Keywords: Energy optimization, IoT, Arduino, occupancy-based automation, ESP8266, environmental control, mall energy management, smart systems

I. INTRODUCTION

Commercial buildings such as shopping malls are among the highest consumers of electricity due to constant demand for lighting, air conditioning, and ventilation. Much of this energy is often wasted in underutilized spaces, leading to inefficiencies and increased operational costs. With the growing emphasis on sustainability and smart infrastructure, there is a pressing need for intelligent systems that adapt resource usage based on real-time environmental and human factors.

To address this, an automated environmental control system has been designed that utilizes Internet of Things (IoT) technology to monitor occupancy and ambient conditions, and dynamically control lighting and ventilation accordingly. The system relies on a combination of sensors—infrared (IR) sensors for people counting, DHT11 for temperature and humidity sensing, and an LDR for light detection—to gather real-time data. An Arduino Uno microcontroller processes this input and controls outputs such as LED lights and a fan, while the ESP8266 module provides a web-based interface for real-time status monitoring [1].

The aim is to optimize energy consumption by increasing resource usage in occupied areas and reducing it in vacant zones. This not only improves operational efficiency but also contributes to sustainable development by minimizing power wastage in large commercial spaces like malls.

II. EASE OF USE

A. Deployment Simplicity

The system is designed to be easily deployable in both prototype and real-world scenarios. Sensor modules, including IR sensors, DHT11, and LDR, are simple to install and require minimal calibration. All components connect to an Arduino Uno using jumper wires and a breadboard, which allows easy replacement and rearrangement. The circuit design is modular, enabling non-technical staff to understand and replicate the setup with basic instructions.

ISSN: 2581-9429 Volume 5, Issue 4, May 2025 Impact Factor: 7.67

B. Dashboard Accessibility

A key feature enhancing usability is the web-based dashboard hosted locally via the ESP8266 module. Any user connected to the same Wi-Fi network can access the dashboard using the module's IP address. The dashboard provides real-time monitoring of:

- Current occupancy count
- Temperature and humidity readings
- Light intensity levels
- Status of the fan and lighting system

This eliminates the need for external displays or software, making it intuitive for facility managers to monitor room conditions using a phone, tablet, or PC.

C. Maintenance and Expansion

Component replacement is straightforward due to the plug-and-play nature of the setup. Additionally, the system logic coded into the Arduino can be modified using the Arduino IDE, making it adaptable for future expansions such as adding new sensors or integrating with cloud services. This ensures that the system remains usable and relevant over time without requiring a complete redesign.

III. SYSTEM DESIGN

A. Functional Overview

The proposed system operates as an intelligent environmental controller that adjusts lighting and fan operations based on real-time inputs. It utilizes an Arduino Uno microcontroller to process data received from multiple sensors, including infrared (IR) sensors for people counting, a DHT11 sensor for monitoring temperature and humidity, and an LDR for measuring ambient light. Based on the sensor data, the system activates LEDs (representing mall lights) and a DC fan, simulating actual energy-consuming appliances. An ESP8266 Wi-Fi module hosts a local dashboard that displays the current system status, allowing users to monitor changes wirelessly [2].

B. Block Diagram

The system is composed of multiple input and output components, connected through a central Arduino Uno. The block diagram below illustrates the interaction between modules:

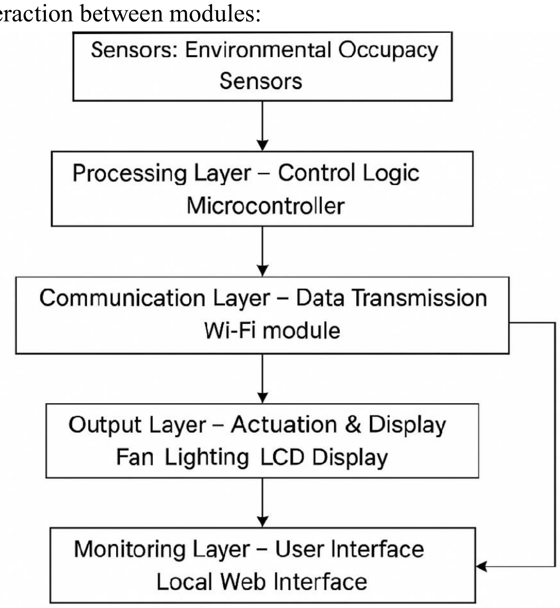

Fig. 1. Block diagram of the automated environmental control system.

DOI: 10.48175/IJARSCT-26471

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 4, May 2025 Impact Factor: 7.67

C. Operational Flow

- **IR Sensors** detect entry and exit to maintain a live people count.
- **LDR** continuously checks ambient light to determine natural lighting.
- **DHT11** monitors room temperature and humidity levels.

Arduino Uno processes the input:

- Controls the number of active LEDs based on occupancy:
- 1 person → 1 LED, 2 → 2 LEDs, up to 4 LEDs.
- Controls the fan based on temperature with a programmed delay.

ESP8266 transmits processed data to a local dashboard for visualization.

D. Dashboard Visualization

The ESP8266 module hosts a web server accessible via its local IP address. The dashboard displays:

- Number of occupants
- Current temperature and humidity
- Light intensity
- LED and fan statuses

IV. HARDWARE COMPONENTS USED

A. Component List and Description

The system utilizes low-cost and readily available electronic components to achieve intelligent environmental control. Each component plays a critical role in sensing, processing, or actuating specific functions of the system. The table below summarizes the components used and their respective purposes.

Table 1: List of Hardware Components

Component	Quantity	Purpose
Arduino Uno	1	Central microcontroller for processing sensor data and controlling outputs.
ESP8266 Wi-Fi	1	Transmits data to a local web dashboard via Wi-Fi.
IR Sensors	2	Detects entry and exit to count people.
DHT11 Sensor	1	Measures temperature and humidity.
LDR (Light Sensor)	1	Detects ambient light intensity.
LEDs	4	Simulate light activation based on occupancy.
DC Fan	1	Operates based on temperature thresholds.
Relay Module	2	Switches fan and LED circuit safely.
Jumper Wires	Multiple	Connects components electrically.
Power Supply (5V)	1	Powers all modules and sensors.

B. Integration Method

All components are mounted on a breadboard to simplify assembly and allow easy modifications. Power is supplied via USB or external 5V supply [3]. Inputs from sensors are connected to the digital and analog pins of the Arduino Uno, while outputs (LEDs and fan) are controlled via relays. The ESP8266 communicates with the Arduino using serial communication for data transmission to the dashboard.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, May 2025
Impact Factor: 7.67

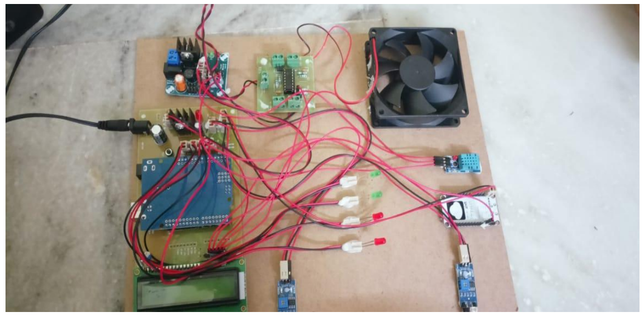

Fig. 2. Physical setup of the system using Arduino and sensors.

V. SOFTWARE TOOLS USED

A. Arduino IDE

The Arduino Integrated Development Environment (IDE) is used for writing, compiling, and uploading the embedded C/C++ code to the Arduino Uno board. It provides an intuitive interface and includes built-in libraries for handling input/output operations with sensors and modules such as IR sensors, DHT11, and LDR. The serial monitor in the IDE is also utilized for real-time debugging and data verification during development [4].

B. ESP8266 Firmware and Web Interface

The ESP8266 module is programmed using the Arduino IDE with the ESP8266 board support package installed. A lightweight HTML/CSS web page is embedded into the ESP8266 code to serve a local dashboard via Wi-Fi. This web interface displays real-time data such as occupancy count, temperature, humidity, light level, and the operational status of the fan and LEDs [5].

C. Serial Communication

Communication between the Arduino Uno and ESP8266 is established through serial protocols (using the SoftwareSerial library), allowing sensor data to be transmitted to the ESP8266 and reflected on the web dashboard [6].

Additional Libraries Used

The following libraries were included in the code to support sensor functionality:

- DHT.h for DHT11 temperature and humidity sensor
- SoftwareSerial.h for ESP8266 communication
- ESP8266WiFi.h and ESP8266WebServer.h for dashboard hosting
- Adafruit_Sensor.h for sensor calibration and integration [7].

VI. EXECUTION

A. Sensor Calibration and Data Acquisition

The system begins by initializing all sensors connected to the Arduino Uno. IR sensors are placed at both the entrance and exit of the simulated mall room. Every time a person is detected entering or leaving, the count is incremented or decremented accordingly. The DHT11 sensor collects temperature and humidity data at regular intervals, while the LDR measures light intensity to detect ambient lighting conditions.

B. Logic for Lighting Control

The number of LEDs turned on is directly proportional to the number of occupants present:

- 1 person → 1 LED
- 2 persons → 2 LEDs
- 3 persons → 3 LEDs
- 4 or more persons → 4 LEDs

DOI: 10.48175/IJARSCT-26471

This logic simulates dynamic lighting control in response to real-time footfall, ensuring energy is not wasted in unoccupied areas.

C. Logic for Fan Control with Delay

Fan activation is based on temperature thresholds received from the DHT11 sensor. The logic includes a delay condition to prevent rapid on/off cycling:

- If temperature > 30°C → Fan ON (after 5-second delay)
- If temperature < 28°C → Fan OFF (with delay)

This ensures smooth operation and protects the hardware from frequent toggling.

D. Dashboard Display via ESP8266

The ESP8266 is programmed to act as a web server. It receives serial data from the Arduino and hosts a local dashboard accessible through any web browser on the same Wi-Fi network. The dashboard displays:

- Current number of people in the room
- Real-time temperature and humidity
- Light intensity value
- Current status of LEDs and fan

E. Real-Time Testing

The system was tested under varying conditions to ensure responsiveness:

- People entering/exiting simulated using hand gestures over IR sensors.
- Room temperature adjusted manually for testing fan logic.
- Dashboard observed to verify real-time data accuracy and device behavior.

VII. RESULTS

A. Lighting Control Results

The LED-based lighting system responded accurately to changes in occupancy detected by the IR sensors. As more individuals were detected inside the room, additional LEDs were activated accordingly. The system successfully simulated dynamic lighting control based on real-time foot traffic.

Table 2: LED Activation Based on People Count

People Count	Number of LEDs ON
0	0
1	1
2	2
3	3
4 or more	4

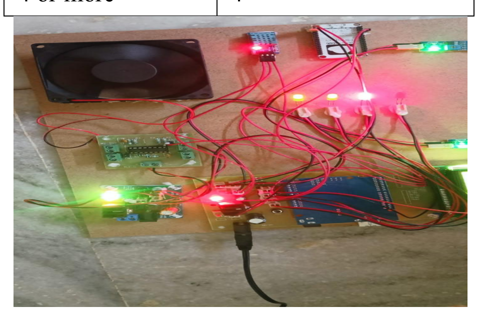

Fig. 3. Hardware implementation of the automated environmental control system.

DOI: 10.48175/IJARSCT-26471

B. Fan Speed Control Based on Temperature and Occupancy

Unlike a simple ON/OFF setup, the fan in this system operates at four discrete speed levels depending on both the **temperature** and **people count**. The logic is designed to increase airflow in crowded and warmer environments, ensuring both energy efficiency and comfort.

Table 3: Fan Speed Logic

People Count	Temperature (˚C)	Fan Speed (%)
1-2	< 28	30
2-3	28-30	60
3-4	30-32	80
>=4	> 32	99

This multi-level fan control was tested in real-time using simulated room heat, and the output confirmed smooth and accurate switching between speed levels based on the combined conditions of occupancy and ambient temperature.

C. Dashboard Monitoring Accuracy

The ESP8266-based web dashboard provided real-time updates on:

- People count
- Temperature and humidity
- Light level
- Fan speed percentage
- LED status

D. Performance Evaluation

- **System Latency:** Less than 1 second for sensor-to-output response.
- **Sensor Accuracy:** IR counting worked with ~95% accuracy under controlled testing.
- **Fan Response:** Smooth transitions between speed levels without jitter.
- **Dashboard Performance:** Live data updates observed with minimal delay on local network.

VIII. CHALLENGES FACED

IR Sensor Accuracy and Alignment

One of the primary challenges encountered during development was the **inconsistent people counting** due to misalignment of IR sensors. Slight variations in sensor positioning or fast consecutive movements caused inaccurate readings, leading to either overcounting or undercounting. Proper alignment and testing under different lighting and motion conditions were required to improve detection accuracy.

Fan Speed Control Tuning

Implementing **precise multi-level fan speed control** using PWM (Pulse Width Modulation) introduced challenges related to calibration and timing. Ensuring smooth transition between speed levels (30%, 60%, 80%, 99%) without abrupt surges required careful coding and appropriate use of delay functions. Fine-tuning the logic to respond to both temperature and people count simultaneously also added complexity.

ESP8266 Connectivity

The **ESP8266 module** occasionally faced connectivity drops during early stages of testing, especially when switching between serial communication and dashboard updates. Managing the serial communication with Arduino while hosting a responsive dashboard required optimization of both software logic and baud rate configurations.

Power Supply Stability

Providing stable and sufficient power to the Arduino, sensors, and ESP8266 simultaneously was a challenge, especially during Wi-Fi transmission and relay switching. In some cases, the system would reset unexpectedly due to voltage drops. This was resolved by using a regulated 5V external power supply and isolating the ESP8266's power line.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, May 2025
Impact Factor: 7.67

Real-Time Dashboard Lag

When multiple sensor inputs changed rapidly, the **ESP8266-hosted dashboard** sometimes experienced data lag or stale readings. To mitigate this, minimal HTML/CSS was used and sensor values were refreshed periodically using simple JavaScript functions for better efficiency.

IX. CONCLUSION

An intelligent, sensor-driven system was successfully designed and implemented to automate environmental controls in a simulated mall environment. The solution dynamically adjusted lighting and fan speed based on real-time occupancy and temperature conditions, using an Arduino microcontroller integrated with IR, DHT11, and LDR sensors. The inclusion of multi-level fan speed control based on combined sensor inputs further enhanced the system's adaptability and energy efficiency. A major strength of the design was the real-time dashboard hosted via the ESP8266 module, which allowed users to monitor system status wirelessly without the need for additional hardware or cloud infrastructure. The overall results demonstrated that the system is both scalable and effective for reducing energy consumption in high-traffic environments such as shopping malls. It responds quickly to changing environmental conditions and requires minimal user intervention once deployed. This project lays the foundation for smarter commercial infrastructure by integrating basic IoT technologies with automation logic to create cost-effective, sustainable energy management systems.

X. FUTURE SCOPE

Multi-Zone Expansion

The current system is designed to simulate a single room or zone within a mall. In the future, the project can be expanded to cover multiple rooms or store zones, with each zone independently monitored and controlled. A centralized dashboard can be developed to give a unified view of the entire mall's energy usage.

Integration with Real Electrical Fixtures

Instead of using LEDs and a small DC fan, the system can be scaled to control real-world appliances such as tube lights, ceiling lights, air conditioners, and industrial fans using relay modules and smart switches. This would make the system directly applicable in commercial installations.

Cloud Integration and Data Logging

Adding cloud-based storage and dashboards would allow for remote monitoring and historical data analysis. This can support advanced features like predictive energy usage, monthly reporting, and fault detection.

Mobile App Development

A dedicated mobile application can be developed to give mall managers real-time access to system status and control options, enabling remote adjustments and alerts for unusual conditions.

Machine Learning for Predictive Control

Incorporating machine learning algorithms can enhance the system by predicting crowd behavior or environmental changes. This would enable preemptive control actions, further improving energy efficiency and user comfort.

ACKNOWLEDGMENT

This Project Survey would be incomplete without the introduction of the people who made it possible and whose guidance, encouragement crowns all efforts with success.

We are thankful to our honorable Prof. G. Chandramohan Reddy Sir, Principal of MGIT and Dr. D. Vijaya Lakshmi, Professor and HOD, Department of IT, MGIT, for providing excellent infrastructure and a conductive atmosphere for completing the project successfully.

We are deeply indebted to our project guide, Dr. N. Sree Divya, Assistant Professor, Department of IT, MGIT, for her constant support, valuable suggestions, immense patience, and expertise throughout the course of our project.

We are profoundly grateful to our project coordinator, Dr. N Sree Divya, Assistant Professor, Department of IT, MGIT, for her unwavering support, and valuable suggestions throughout the course of our project.

REFERENCES

[1]. R. K. Rajput, Basic Electrical and Electronics Engineering, Laxmi Publications, 2015.
[2]. M. Ali, "Internet of Things (IoT): Architecture and Applications," IEEE Explore, 2020.
[3]. Arduino, "Arduino Uno Board," [Online]. Available: https://www.arduino.cc/en/Main/ArduinoBoardUno.
[4]. ESP8266 Community, "ESP8266WiFi and Web Server Library," [Online]. Available: https://arduino-esp8266.readthedocs.io/.
[5]. Adafruit, "DHT11 Sensor Library," [Online]. Available: https://github.com/adafruit/DHT-sensor-library.
[6]. Random Nerd Tutorials, "ESP8266 Web Server with HTML," [Online]. Available: https://randomnerdtutorials.com/esp8266-web-server/.
[7]. N. Sree Divya, Bhavana Deyyam, and CH. Laxmi Pravalika, "Automated Environmental Control for Energy Efficiency in Malls," International Journal of Research Publication and Reviews, vol. 6, no. 4, pp. 4074–4086, Apr. 2025. DOI: 10.55248/gengpi.6.0425.1460.