

IoT-Based Adaptive Lighting System and Fire Detection in Rural Areas

K. Jithendra Reddy¹, Lingutla Nithin Kumar², Madan B³, Prasad Kumar K⁴
Department of Artificial Intelligence and Machine Learning
Ballari Institute of Technology and Management, Ballari, Karnataka, India

Guide: Asha Jyothi P
Year of Submission: 2025

Abstract—The existing rural street lighting system usually relies on manual switching, which is the cause of unnecessary power consumption and late response in case of fire incidents. Integration of IoT and sensor-based automation is an effective solution to these problems with intelligent control and real-time monitoring. This work aims to design and develop an IoT-based low-cost adaptive street lighting and fire detection system for enhancing energy efficiency and safety in rural and semi-urban areas. The system integrates an LDR for detecting ambient light, an IR sensor for detecting motion, and a flame sensor for fire detection. Sensor data processing will be made by Arduino Uno, while NodeMCU-ESP8266 will send real-time notifications through the Blynk IoT platform. The lighting system will operate in two modes: motion-based ON/OFF and dim-bright adaptive control with PWM. Fire detection will trigger a local buzzer and also generate a remote alert. Experimental testing of the prototype demonstrated accurate day/night detection, reliable motion-triggered lighting, and fast fire response in seconds. IoT alerts were also delivered almost instantly to the user's smartphone. Adaptive lighting modes resulted in a significant reduction in power consumption over conventional always-on lighting. An effective and low-cost solution has been proposed for the integrated adaptive lighting and real-time fire detection in rural infrastructure. It enhances the level of safety with improved energy efficiency and provides immediate emergency alerts through IoT connectivity.

Keywords—IoT, Adaptive Lighting, Fire Detection, Arduino, NodeMCU (ESP8266), Rural Infrastructure, Energy Efficiency.

I. INTRODUCTION

The project aims to build a smart lighting setup using IoT technology that also detects fire without human intervention. Smart lighting systems have been widely explored for improving energy efficiency and automation in homes, streets, and public spaces [1], [2]. In towns and villages alike, lights often get switched on or off by hand, which increases power consumption and maintenance costs [1], [11]. IoT-based automation offers a reliable way to operate devices remotely and improve response time during emergencies [1], [10].

Fire incidents in rural and semi-urban areas frequently go unnoticed until they grow serious due to lack of continuous monitoring. Recent research has focused on IoT-based fire detection using flame, smoke, and temperature sensors to enable early alerts [4]–[6]. With improved sensors and low-cost microcontrollers, automated systems can now monitor surroundings continuously and respond almost instantly.

This work uses an Arduino Uno connected to flame, light (LDR), and infrared sensors to manage adaptive street lighting and local fire alerts. Similar sensor combinations have been used in fire and lighting systems in prior studies [5], [9], [12]. Once processed, the data is forwarded to a NodeMCU module, enabling real-time cloud-based alerts via the Blynk platform [7], [8]. These signals help manage lighting, detect fire early, and issue instant warnings to users.

II. LITERATURE REVIEW

IoT-based automation systems have gained significant attention for improving public infrastructure, particularly in lighting and safety applications. Several researchers have explored smart lighting solutions using LDR, IR, and microcontroller-based control mechanisms. Vishnupriya and Balaji [1] demonstrated an LDR-driven smart lighting model capable of reducing unnecessary power consumption, while Sharma and Sharma [2] implemented PWM-based brightness control for more efficient indoor illumination. Singh and Kapoor [3] developed an IoT-enabled streetlight system that adapts lighting based on motion and environmental sensing, confirming the benefits of sensor-based street illumination.

Fire detection has also evolved through IoT integration. Gunawan et al. [4] designed an Arduino-based flame-sensing system for rapid emergency alerting. Khan et al. [5] reviewed modern fire-detection sensors, emphasizing improved accuracy using optical, thermal, and flame sensors. Islam and Akter [6] proposed a complete IoT-enabled fire alerting model capable of delivering real-time notifications to remote users. These works support the effectiveness of sensor fusion in improving early fire response times.

IoT platforms such as NodeMCU and Blynk Cloud have become widely adopted for wireless communication and monitoring. Prasad and Chandavarkar [7] demonstrated seamless NodeMCU-based home automation, while Singh and Kaur [8] validated Blynk as a reliable mobile-based monitoring tool for IoT projects. For rural infrastructure, Kumar and Sahu [11] highlighted the need for low-cost IoT deployments, and Naik et al. [10] evaluated IoT emergency systems suitable for resource-limited regions.

Although many studies address either smart lighting or fire detection individually, very few integrate both systems into

a unified low-cost solution. This motivates the present work, which combines adaptive lighting and IoT-based fire alerts into a single deployable system using affordable sensors and microcontrollers tailored for rural and small-town environments.

III. PROPOSED SYSTEM

The setup offers a low-cost solution that links adaptive lighting with fire detection, designed mainly for rural and small-town locations [10], [11]. IoT-based lighting and fire-alert systems have shown potential for improving safety and reducing power waste in public infrastructure [1], [3], [5]. The proposed system uses an LDR, IR sensor, and flame sensor connected to an Arduino Uno for decision-making, while the NodeMCU (ESP8266) handles Wi-Fi connectivity and pushes alerts to the Blynk platform [7], [8].

A. Architecture of the System

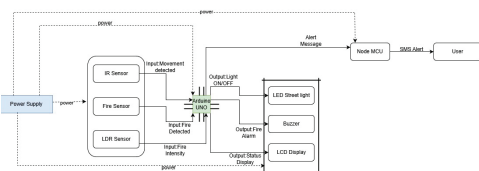


Fig. 1. System Architecture

Explanation: The architecture diagram shows how the LDR, IR sensor, and flame sensor connect to the Arduino. The NodeMCU module provides IoT connectivity, sending alerts through the Blynk app. LEDs, buzzer, and other components respond based on sensor readings. This design follows patterns seen in prior IoT lighting and fire-alert systems [1], [3], [5].

B. Dual Lighting Operation Modes

To improve energy efficiency, the system operates in two intelligent lighting modes. Similar sensor-based and PWM-driven lighting systems have been proposed for homes and street lighting [1], [2], [12].

1) *Mode 1: Motion-Based Lighting:* The LED remains OFF during nighttime until the IR sensor detects motion. When motion is detected, the Arduino powers the LED briefly, then turns it off after a set interval. This reduces wasted power by only lighting when needed, as also reported in IR-based smart lighting systems [2], [12].

2) *Mode 2: Dim-Bright Adaptive Lighting:* The LED stays DIM during idle periods at night. When motion is detected, the brightness increases to BRIGHT using PWM control. After a timeout with no motion, brightness returns smoothly to DIM. This maintains visibility while conserving energy [1], [2]. PWM-based dimming is commonly used in modern smart lighting designs [2], [12].

C. Fire Detection and IoT Alert Mechanism

The flame sensor continuously monitors for heat or flame signatures. When fire is detected:

- The buzzer is activated to warn nearby individuals.

- A real-time notification is sent to the user via the Blynk mobile application using NodeMCU.

This IoT-based alert ensures incidents are reported instantly, even when nobody is physically present [4]–[6].

D. Advantages of the Proposed System

- Reduces electricity consumption through sensor-based adaptive lighting [1]–[3].
- Provides fast fire alerts using IoT notifications [5], [6].
- Uses inexpensive, easy-to-maintain components suitable for rural areas [10], [11].
- Scalable to multiple streetlights or larger community networks.

IV. HARDWARE AND SOFTWARE REQUIREMENTS

The proposed smart lighting and fire alert setup runs on sensors and microcontrollers that gather data, process it, and send updates in real time. Similar hardware and software stacks are widely used in IoT-based lighting and fire-detection systems [1], [2], [4], [5]. The main parts and software tools are listed below.

A. Hardware Components

TABLE I
 HARDWARE REQUIREMENTS

S.No	Component	Description
1	Arduino Uno	Main controller for reading sensor inputs and controlling outputs such as LEDs, buzzer, and LCD display. Built around the ATmega328P microcontroller.
2	NodeMCU (ESP8266)	Provides Wi-Fi connectivity and enables IoT-based real-time alerts and remote monitoring through the Blynk platform.
3	IR Sensor	Detects human or vehicular movement and sends digital signals to the Arduino for motion-based lighting control.
4	LDR Sensor	Senses ambient light intensity and helps distinguish between day and night conditions.
5	Flame/Fire Sensor	Detects the presence of heat or flame and triggers immediate fire alerts in the system.
6	LCD Display (16x2 I2C)	Displays system status messages such as “Day Mode”, “Motion Detected”, and “Fire Alert”.
7	Buzzer	Generates an audible alarm in the event of fire detection.
8	LED / LED Array	Acts as the streetlight source; brightness is controlled using PWM based on sensor inputs.

B. Software Requirements

The software tools and environments used in this project enable coding, debugging, IoT communication, and system monitoring. Similar toolchains are reported in IoT lighting and fire-detection literature [1], [2], [5], [8]. The major software resources include:

- **Arduino IDE:** Used to write, compile, and upload programs to both the Arduino Uno and NodeMCU.
- **Embedded C/C++:** Programming languages used to implement sensor logic, control algorithms, and communication routines.
- **Blynk IoT Platform:** Provides mobile-based real-time monitoring and notifications through Wi-Fi-enabled NodeMCU [8].
- **Required Libraries:** BlynkSimpleEsp8266.h, LiquidCrystal_I2C.h, ESP8266WiFi.h, and standard Arduino core libraries.
- **Simulation Tools (optional):** Tinkercad or Proteus for virtual testing and circuit validation.

V. SYSTEM IMPLEMENTATION

The implementation connects sensors, outputs, and communication modules to an Arduino Uno and NodeMCU (ESP8266). The Arduino reads sensor values and controls outputs (LEDs, buzzer, LCD). NodeMCU handles Wi-Fi and pushes notifications to Blynk, following architectures similar to other IoT automation systems [7], [8].

A. Sensor Input Processing

The system uses three primary sensors:

- **LDR:** Measures ambient light to determine day/night operation.
- **IR sensor:** Detects motion for adaptive lighting.
- **Flame sensor:** Detects heat/flame signatures for fire alerts.

Sensor readings are sampled at regular intervals. Arduino logic applies thresholds to determine actions: change LED state, trigger buzzer, or send IoT notification. Similar multi-sensor processing pipelines are discussed in earlier studies [4], [5], [9].

B. Lighting Control Algorithm

Two modes:

- **Motion-Based Mode:** LED OFF until motion is detected; then LED ON for a fixed duration.
- **Dim-Bright Mode:** LED stays DIM; on motion detection, PWM raises brightness to BRIGHT, and dims back after a timeout.

PWM-based lighting control and motion-triggered behavior are similar to earlier smart lighting implementations [1], [2], [12].

C. Fire Detection and IoT Alert Workflow

When the flame sensor detects a value above a threshold:

- 1) Arduino activates the buzzer.
- 2) NodeMCU sends a Blynk notification with the fire alert.
- 3) LCD displays “Fire Alert”.

Such IoT-based fire-alert workflows are in line with modern fire-detection systems [4]–[6], [9].

D. System Workflow

The system loops continuously: LDR check → if night, enable motion monitoring → IR detects motion → adjust LED → flame sensor monitors in parallel → if fire detected, alarm and IoT alert.

E. Code Upload and Testing

Code is written and uploaded via Arduino IDE. Each component is tested individually (unit testing) then integrated. Tests include controlled motion events, varying ambient light, and controlled small fire sources to verify detection and notification timings, as suggested in prior sensor evaluation studies [4], [5].

VI. SYSTEM DESIGN DIAGRAMS

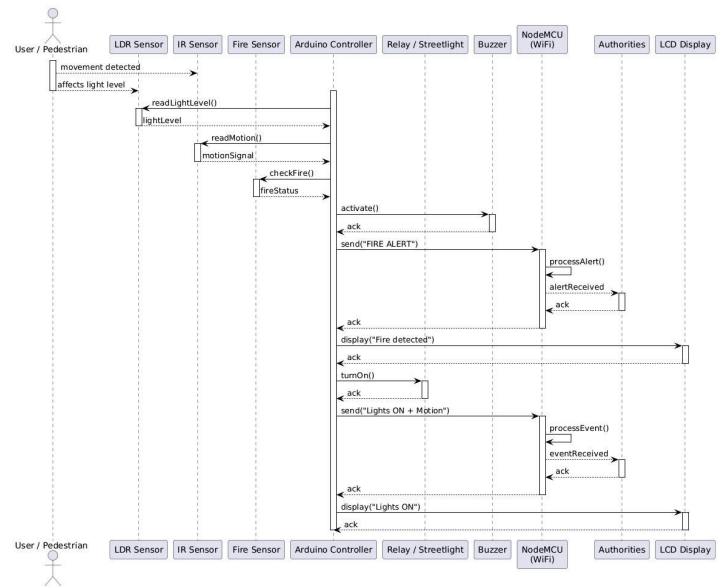


Fig. 2. Sequence Diagram

Explanation: The sequence diagram shows the real-time interaction between sensors, the Arduino controller, and output devices. The IR sensor detects movement and sends a signal to the Arduino, which then switches the LED streetlight to BRIGHT mode. If no motion is present, the Arduino reduces brightness or turns the LED OFF. Meanwhile, the flame sensor continuously monitors for fire and alerts the Arduino upon detection. The Arduino activates the buzzer and sends an IoT alert via NodeMCU.

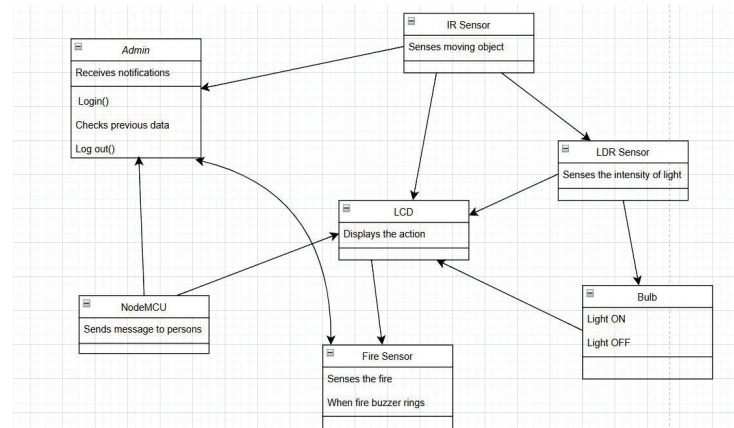


Fig. 3. Class Diagram

Explanation: The class diagram represents the software structure. The Sensor class is the base for IRSensor, LDRSensor, and FlameSensor. ArduinoController processes sensor inputs, controls LEDLight brightness via PWM, and activates the Buzzer. The NodeMCU module handles IoT communication. Relationships show aggregation between controller and peripherals.

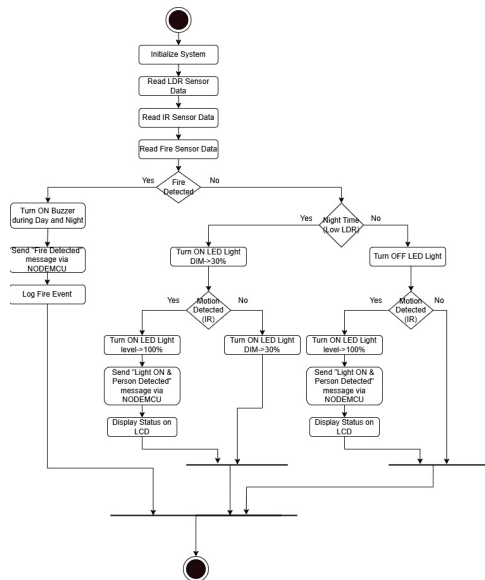


Fig. 4. Activity Diagram

Explanation: The activity diagram outlines the workflow: system initialization, continuous monitoring, motion detection branch (DIM/BRIGHT control), fire detection branch (alarm + IoT alert), and loop back for continuous operation. Lighting control and fire monitoring run effectively in parallel.

VII. RESULTS AND PERFORMANCE ANALYSIS

The IoT-powered smart street lights with fire sensing were built, set up, and tested indoors and outdoors. Testing assessed reaction time, fire detection capability, and power savings. Each component was verified independently before integration, similar to test procedures described in previous IoT-based fire and lighting systems [1], [4], [5].

A. Adaptive Lighting Performance

LDR reliably distinguished day/night. IR detection triggered LED activation with near-instant response. Motion-Based Mode showed significant energy savings since LEDs remain off until needed. Dim-Bright Mode provided continuous low-level illumination and raised brightness on demand via PWM with smooth transitions and no flicker [1], [2], [12].

B. Fire Detection and IoT Alerting Performance

The flame sensor detected controlled fire sources quickly, and the system triggered the buzzer and sent Blynk notifications within seconds. This combination ensures timely alerts even when the area is unattended [4]–[6].

C. Power Efficiency

Motion-triggered operation and dim-bright mode significantly reduce energy consumption versus always-on lighting—especially valuable in rural areas with limited power supplies [1], [10], [11].

D. System Reliability

During tests the system remained stable, sensors provided consistent readings, and NodeMCU maintained Wi-Fi connectivity for notifications. Modular hardware makes maintenance straightforward.

E. Prototype Output Images

The proposed IoT-based adaptive lighting and fire detection system was tested under various lighting and environmental conditions. The results demonstrate the correct operation of day/night detection, dim-bright lighting control, motion sensing, and real-time fire alerting. The following images show the actual working prototype during testing.

1) *Fire Detection Test (Daytime):* When a flame source was introduced near the flame sensor, the system immediately detected it and triggered the buzzer. At the same time, the LCD displayed a warning message indicating a fire alert.

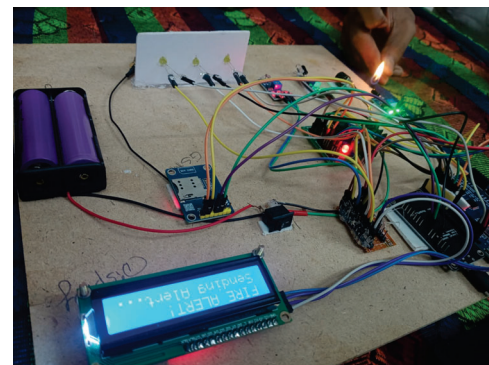


Fig. 5. Fire detection event showing LCD alert and flame sensor activation.

2) *Day Mode Operation:* During daytime, the LDR sensed high ambient light levels and the system automatically turned OFF the LED streetlight. The LCD displayed the system status as “Day Mode LED OFF”.

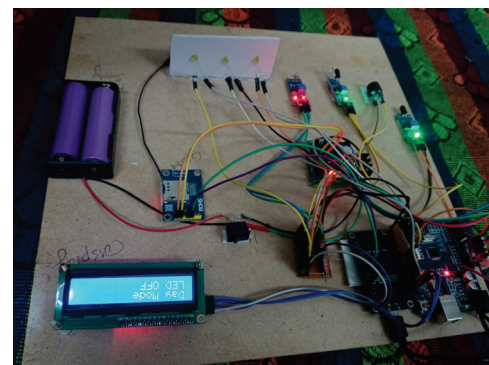


Fig. 6. Day Mode operation with LED OFF and LCD displaying system status.

3) *Night Mode – DIM Operation:* In low light conditions, the LDR triggered Night Mode. With no motion detected by the IR sensor, the LED glowed at minimal brightness (DIM mode).



Fig. 7. Night Mode (DIM) operation with LEDs glowing at low intensity.

4) *Night Mode – Fire Alert Condition:* When the flame sensor detected fire during night mode, the system displayed a fire alert on the LCD and activated the buzzer.

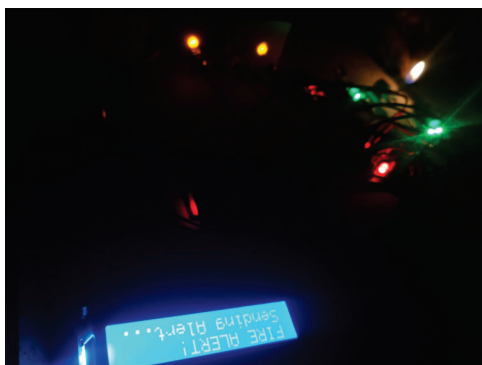


Fig. 8. Night Mode operation with fire alert shown on LCD.

5) *Night Mode – Motion-Based BRIGHT Operation:* When motion was detected, the IR sensor triggered the LED to switch from DIM to BRIGHT mode for improved visibility. The LCD displayed the motion detection value.



Fig. 9. Night Mode BRIGHT operation after IR motion detection.

F. Performance Summary

Table II summarizes the system’s reaction time and accuracy during testing.

TABLE II
 SYSTEM PERFORMANCE SUMMARY

Test Case	Output	Response Time
Day/Night Detection	Success	< 1 sec
Motion-Based Brightening	Success	0.5–1.0 sec
Dim–Bright PWM Control	Smooth/Stable	Instant
Fire Detection	Success	1–3 sec
IoT Alert Delivery (Blynk)	Success	2–6 sec

The results confirm that the system performs reliably in all tested scenarios. It responds quickly to motion and fire, maintains stable lighting control using PWM, and successfully delivers IoT alerts within a few seconds [1], [2], [5].

VIII. CONCLUSION

This project demonstrates a practical, low-cost IoT-based solution combining adaptive street lighting and fire detection suitable for rural deployment. The system conserves energy through intelligent lighting modes and improves safety via early fire alerts delivered remotely using NodeMCU and Blynk [1], [5], [7], [8]. Testing confirmed responsiveness, reliability, and energy savings, making the solution viable for wider adoption in resource-constrained environments [10], [11].

IX. FUTURE SCOPE

Possible enhancements:

- **Solar Power Integration:** Add solar panels and battery storage to make the system self-sustaining in off-grid areas.
- **Advanced Wireless Communication:** Use LoRaWAN or NB-IoT for long-range, low-power communication beyond Wi-Fi coverage.
- **Cloud-Based Data Logging:** Store sensor data for analytics, fault detection, and predictive maintenance.
- **Dedicated Mobile Application:** Develop a custom mobile app for richer controls, historical data visualization, and multi-site management.
- **AI/ML Integration:** Use models to predict failures or detect anomalies indicating fire risk or abnormal power consumption [5].

REFERENCES

- [1] R. Vishnupriya and K. Balaji, “IoT Based Smart Lighting Control System,” 2023.
- [2] K. Sharma and S. Sharma, “Smart Light for Home with Direction and Intensity Adjustment,” 2020.
- [3] V. Singh and R. Kapoor, “IoT-Based Smart Street Lighting System,” in *Proc. IEEE Int. Conf. on Computing, Communication and Automation*, 2020.
- [4] T. S. Gunawan, M. Kartiwi, and N. Ismail, “Prototype Design of Fire Alarm System Using Arduino and Flame Sensor,” in *Proc. Int. Conf. on Computing and Informatics*, 2018.
- [5] F. Khan, Z. Xu, J. Sun, F. Khan, A. Ahmed, and Y. Zhao, “Recent Advances in Sensors for Fire Detection,” 2022.

- [6] M. S. Islam and M. Akter, "Smart Fire Detection System Using IoT," in *Proc. IEEE Int. Conf. on Robotics, Electrical and Signal Processing Techniques*, 2019.
- [7] A. S. Prasad and B. R. Chandavarkar, "Home Automation Using ESP8266 NodeMCU Module," *Int. Journal of Emerging Technologies*, 2020.
- [8] D. Singh and M. Kaur, "IoT-Based Monitoring Using Blynk Cloud," *Journal of Electronics and Communication Engineering*, 2021.
- [9] G. Kuznetsov, N. Kopylov, E. Sushkina, and A. Zhdanova, "Adaptation of Fire-Fighting Systems to Localization of Fires," 2022.
- [10] B. M. Naik, R. S. Patil, and K. Hegde, "IoT-Based Emergency Response Systems," *Int. Journal of Sensor Networks*, 2021.
- [11] P. Kumar and A. Sahu, "IoT Solutions for Smart Rural Infrastructure," in *Proc. IEEE Int. Conf. on ICT for Rural Development*, 2020.
- [12] A. A. Dhumal and S. D. Patil, "Arduino-Based Intelligent Lighting System Using IR Sensor," *Int. Journal of Engineering Research*, 2018.