# ABSTRACT

This paper presents an automatic control system which turns ON or OFF room light based on counting how many people are present in the room. If there is no person in the room the light remains OFF and when there is any person enters in the room the system turns the room light ON. When somebody enters in or leaves the room, he has to cross a pair of sensors. These sensors send information to the micro-controller with the sequence he crossed the sensors. According to the sequence a pre-programmed micro-controller adds or subtracts the number of people present in the room and turns ON or OFF the relay which actually turns the room light ON or OFF. The integration of advanced automation in auditorium management enhances both user experience and operational efficiency. This paper presents a comprehensive system for automated door opening and light illumination in auditoriums, aiming to optimize energy usage, improve accessibility, and ensure safety. The proposed system leverages a combination of motion sensors, RFID technology, and programmable logic controllers (PLCs) to facilitate seamless entry and exit for attendees. Upon detecting motion or authenticating an RFID tag, the doors automatically open, reducing the need for manual operation and minimizing bottlenecks during peak times . Simultaneously, the light illumination system is designed to adjust lighting levels based on occupancy and ambient light conditions. Using a network of light sensors and dimmable LED fixtures, the system maintains optimal illumination for various activities, ranging from general gatherings to performances and presentations. The integration with a central control unit allows for real- time monitoring and remote adjustments, ensuring a responsive and adaptive environment . This automated solution not only enhances the user experience by providing a smooth and efficient entry process but also contributes to significant energy savings through intelligent light management. The implementation of this system demonstrates the potential for smart technology to transform traditional auditorium operations, promoting sustainability and convenience . The doors are equipped with motion sensors and RFID readers. When a person approaches, the motion sensors activate the door opening mechanism. For enhanced security, RFID readers can verify authorized access, ensuring that only permitted individuals can enter certain areas. Light sensors detect the level of natural light entering the auditorium and adjust the artificial lighting accordingly.

TABLE OF CONTENTS **Page No**

1. [INTRODUCTION 4-5](#_TOC_250010)
   1. [components](#_TOC_250009)
   2. [block diagram](#_bookmark0)
   3. [power supply](#_bookmark1)
2. [LITERATURE SURVEY 6-8](#_bookmark2)
3. ANALYASIS AND DESIGN 9-12
   1. [requirements analysis](#_TOC_250008)
   2. [system design](#_TOC_250007)
4. [IMPLEMENTATION 13-16](#_TOC_250006)
   1. hardware steps
   2. communication network setup
   3. central pc software development
5. [CODE 17-22](#_TOC_250005)
6. [EXPERIMENTAL INVESTIGATION 23-30](#_TOC_250004)
   1. steps for experimental investigation
   2. experimental procedure
7. [TESTING AND DEBUGGING 31-35](#_TOC_250003)
   1. [types of testing](#_TOC_250002)
   2. [debugging process](#_TOC_250001)
8. RESULT 36-38
   1. functional results
   2. our result
9. [CONCLUSION 39-40](#_TOC_250000)
10. REFERENCE 41

# INTRODUCTION

Auditoriums serve as pivotal venues for a variety of events, including conferences, performances, lectures, and community gatherings. The efficiency and comfort of these spaces significantly impact the overall experience of attendees and event organizers. Traditional manual operations of door systems and lighting controls can often lead to inefficiencies, bottlenecks, and increased energy consumption. To address these challenges, the implementation of automated systems for door opening and light illumination has emerged as a modern solution.

The advancement of automation technology provides opportunities to enhance the functionality and user experience within auditoriums. Automated door systems, utilizing motion sensors and RFID technology, offer a hands-free entry and exit process, improving accessibility for all attendees, including those with disabilities. This system also ensures security by restricting access to authorized individuals, thus protecting sensitive areas and equipment.

In parallel, the intelligent light illumination system optimizes the use of artificial lighting based on real-time occupancy and ambient light conditions. By incorporating light sensors and programmable lighting controls, the system dynamically adjusts lighting levels, ensuring optimal visibility while minimizing energy waste. This adaptive lighting approach not only enhances the visual experience for various events but also contributes to substantial energy savings and sustainability efforts.

### Components

* + 1. Microcontroller (AT89S52)
    2. LCD Display (16x2 lines)
    3. Door System
    4. LDR Sensors
    5. IR Sensors
    6. Relay
    7. Bulb
    8. Power Supply

#### BLOCK DIAGRAM:

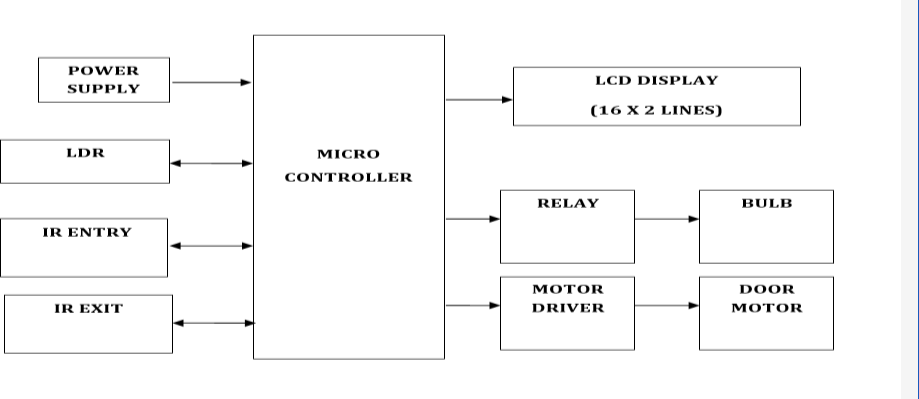


Figure 1

#### 1.1 POWER SUPPLY:

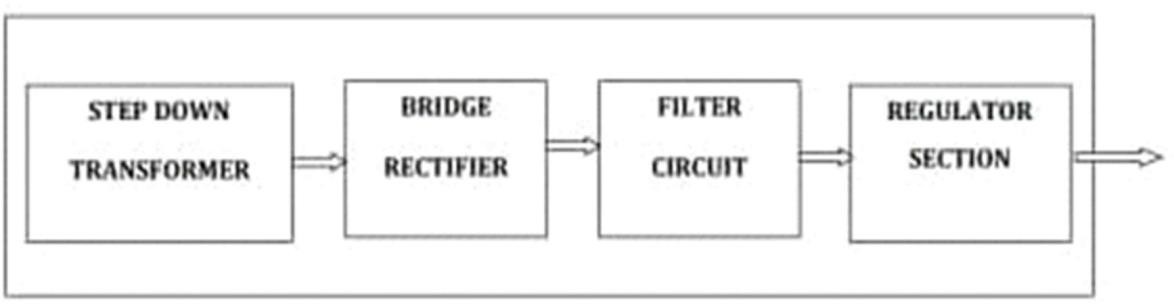
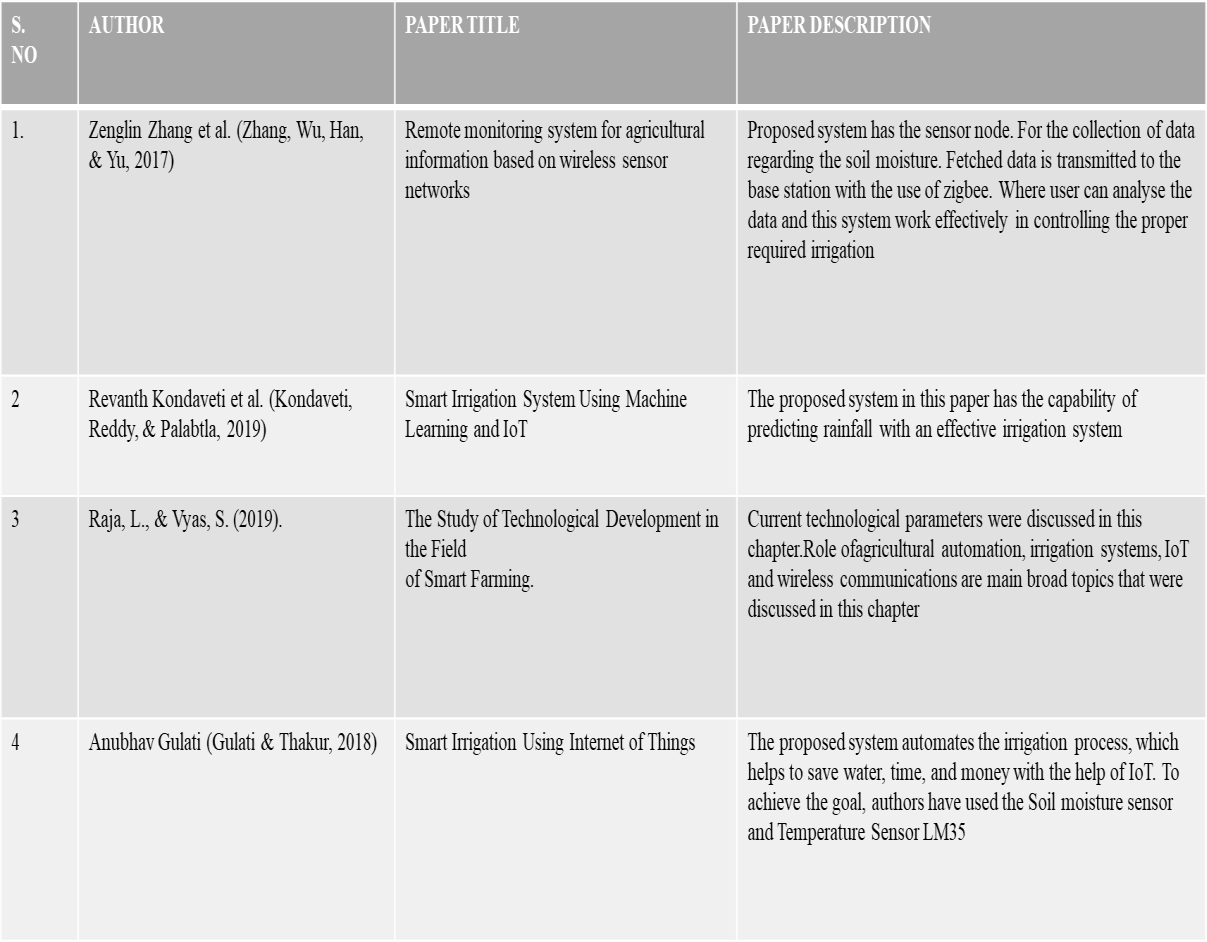


Figure 2

# LITERATURE SURVEY

The integration of automated systems in building management has been an area of extensive research and development, with significant advancements in recent years. This literature survey examines previous studies and technologies relevant to automated door opening and light illumination systems, particularly within the context of auditoriums. Various studies have explored the use of motion sensors for automated door systems. For instance, the work by Chen et al. (2017) demonstrated the effectiveness of infrared motion sensors in detecting human presence and activating door mechanisms. The study highlighted the reliability and cost-effectiveness of this technology for high-traffic areas.



## ANALYSIS & DESIGN

### Requirements Analysis

To design an effective command-based street light controlling system using IoT, it is essential to comprehensively analyze the system's requirements. This involves understanding the functional, non-functional, hardware, and software requirements.

* + 1. Functional Requirements

Real-Time Monitoring: The system should monitor the status of each street light in real- time.

Remote Control: Operators should be able to remotely control each street light, including turning lights on/off and adjusting brightness.

Automated Responses: The system should automatically adjust lighting based on sensor inputs (e.g., ambient light levels, motion detection).

Scheduling: The system should support scheduling for automated lighting adjustments based on time of day.

* + 1. Non-Functional Requirements

Scalability: The system should be scalable to support a large number of street lights.

Reliability: The system should be reliable, with minimal downtime and robust error handling.

Security: Communication between the street lights and the central PC should be secure to prevent unauthorized access.

Usability: The user interface should be intuitive and easy to use for operators.

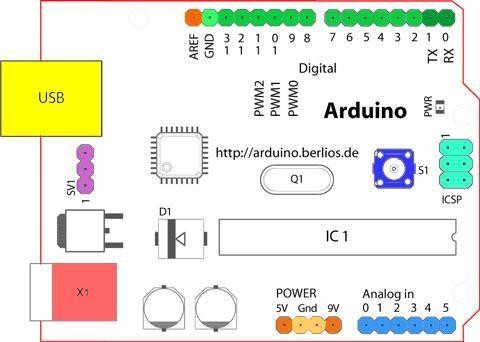
* + 1. Hardware Requirements

Microcontroller Units (MCUs): Capable of interfacing with sensors and communication

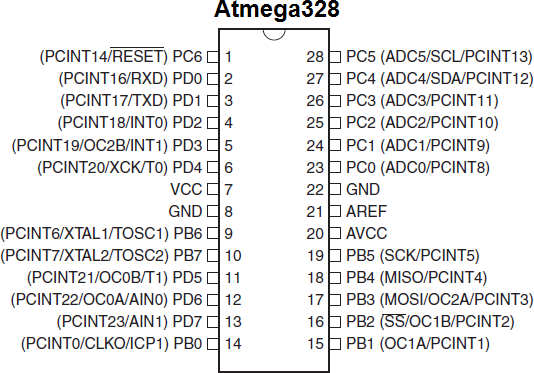
modules (e.g., Arduino, ESP8266, Raspberry Pi).

Sensors: Ambient light sensors and motion detectors for each street light.

Communication Modules: Wi-Fi, Zigbee, or LoRaWAN modules for reliable data transmission.



**Fig 3.1.4.** Structure Of Arduino Board



**Fig 3.1.5** Pin Configuration of Atmega328

### PIN DESCRIPTION:-

**VCC:** Digital

supply voltage**. GND:** Ground.

Port A (PA7-PA0):

Port A serves as the analog inputs to the A/D Converter. Port A also serves as an 8- bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active,even if the clock is not running.

Port B (PB7-PB0):

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port B also serves the functions of various special features of the ATmega32.

Port C (PC7-PC0):

Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected foreach bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled,the pull-up resistors on pins PC5(TDI), PC3(TMS) and PC2(TCK) will be activated even ifa reset occurs. The TD0 pin is tri-stated unless TAP states that shift out data are entered. PortC also serves the functions of the JTAG interface.

### System Design

* + 1. High-Level Architecture

The system architecture consists of three primary layers: the device layer (lights with MCUs and sensors), the communication layer (network infrastructure), and the control layer (central PC with control software).

* + 1. Device Layer

Microcontroller Unit (MCU): The core component controlling the street light, processing sensor data, and executing commands from the central PC.

Sensors: Ambient light sensors and motion detectors provide real-time data to the MCU. Lighting Element: LED street light controlled by the MCU.

* + 1. Communication Layer

Network Protocols: Wi-Fi for short-range, Zigbee for medium-range, or LoRaWAN for long- range communication. The choice depends on the deployment area's specific requirements.

Data Transmission: Secure and reliable transmission of data between the street lights and the central PC.

* + 1. Control Layer

PC Software: A custom application with a graphical user interface (GUI) for real-time monitoring and control.

Database: A database system to log operational data and historical records.

* 1. Detailed Design
     1. Microcontroller Unit (MCU) Design

Sensor Interface: GPIO pins for connecting ambient light sensors and motion detectors. Communication Module: Integrated or external module (e.g., ESP8266 for Wi-Fi).

Power Management: Circuit design to ensure stable power supply to the MCU and sensors.

* + 1. Communication Protocol Design

Protocol Selection: Choice of Wi-Fi, Zigbee, or LoRaWAN based on range, power consumption, and data rate requirements.

Security Measures: Implement encryption (e.g., AES) and authentication mechanisms to secure data transmission.

* + 1. PC Interface Software Design

GUI Design: User-friendly interface with real-time status indicators, control buttons, and data visualization tools.

Command Processing: Software modules to handle user commands and communicate with the MCUs.

Data Logging and Analysis: Database integration to store operational data and tools for analyzing logged data.

* + 1. System Integration

Integration Testing: Ensure all components work seamlessly together. Test communication reliability, sensor accuracy, and command execution.

Calibration :Calibrate sensors for accurate ambient light and motion detection

* 1. Implementation Plan
     1. Phase 1: Prototype Development

Develop a prototype with a limited number of street lights.

Test basic functionalities such as real-time monitoring, remote control, and automated responses.

* + 1. Phase 2: Pilot Deployment

Deploy the system in a controlled environment (e.g., a small neighborhood). Gather data and feedback to refine the system.

* + 1. Phase 3: Full-Scale Deployment

Expand the system to cover a larger urban area.

Implement advanced features such as predictive maintenance and integration with other smart city systems.

## IMPLEMENTATION

Implementing a command-based Auditorium Light Opening and Light Illimination System through a PC using IoT involves several steps and components. Below is a structured approach to guide the implementation process:

**Implementation Steps**

* 1. Hardware Setup
     1. Select Microcontroller Units (MCUs):

Choose suitable MCUs (e.g., Arduino, ESP8266, Raspberry Pi) capable of interfacing with sensors and communicating over Wi-Fi, Zigbee, or LoRaWAN.

Ensure compatibility with selected communication protocols and power requirements.

* + 1. Install Sensors:

Connect ambient light sensors and motion detectors to each MCU.

Position sensors optimally to capture accurate data on ambient light levels and motion.

* + 1. Light Configuration:

Integrate MCUs with existing street lights or install new LED lights with built-in MCU capabilities.

* 1. Communication Network Setup
     1. Choose Communication Protocol:

Depending on the deployment area's range and data rate requirements, select Wi-Fi for short- range, Zigbee for medium-range, or LoRaWAN for long-range communication.

Configure network parameters (SSID, encryption keys, etc.) for secure communication.

* 1. Central PC Software Development
     1. Design GUI for Central PC:

Develop a graphical user interface (GUI) using a suitable programming language (e.g., Python, Java) and framework (e.g., Tkinter, JavaFX).

Design the interface to display real-time status of street lights, sensor data, and allow user interaction for control and configuration.

* + 1. Implement Command Processing:

Develop software modules to process commands from the GUI and transmit them to the respective MCUs.

Include functionalities for turning lights on/off, adjusting brightness, scheduling operations, and responding to sensor inputs.

* + 1. Data Logging and Analytics:

Integrate a database system (e.g., MySQL, SQLite) to log operational data, sensor readings, and system events.

Implement data analytics to derive insights for optimizing lighting schedules and energy usage.

### CODE:

#include <LiquidCrystal.h> #include <stdio.h>

LiquidCrystal lcd(6, 7, 5, 4, 3, 2); int ir1 = 10;

int ir2 = 11;

int ldr = 12; int bulb = 13; int m1a = 8;

int m1b = 9;

int buzzer = A0;

unsigned char rcv,count,gchr='x',gchr1='x',robos='s'; char rcvmsg[10],pastnumber[11];

char gpsval[50];

// char dataread[100] = "";

// char lt[15],ln[15]; int i=0,k=0,lop=0; int gps\_status=0; float latitude=0; float logitude=0; String Speed=""; String gpsString="";

char \*test="$GPRMC";

//int hbtc=0,hbtc1=0,rtrl=0;

unsigned char gv=0,msg1[10],msg2[11]; float lati=0,longi=0;

unsigned int lati1=0,longi1=0; unsigned char flat[5],flong[5]; unsigned char finallat[8],finallong[9]; int ii=0,rchkr=0;

String inputString = ""; // a string to hold incoming data boolean stringComplete = false; // whether the string is complete void okcheck()

if(digitalRead(ldr) == HIGH)

{

lcd.setCursor(4,0);lcd.print("Dark "); digitalWrite(bulb, HIGH);

}

if(digitalRead(ir1) == LOW)

{

lcd.setCursor(0,1);lcd.print("Entry "); digitalWrite(m1a, HIGH);digitalWrite(m1b, LOW); delay(1000);

digitalWrite(m1a, LOW);digitalWrite(m1b, LOW); delay(3000);

digitalWrite(m1a, LOW);digitalWrite(m1b, HIGH); delay(1000);

digitalWrite(m1a, LOW);digitalWrite(m1b, LOW);

}

if(digitalRead(ir2) == LOW)

{

lcd.setCursor(0,1);lcd.print("Exit "); digitalWrite(m1a, HIGH);digitalWrite(m1b, LOW); delay(1000);

digitalWrite(m1a, LOW);digitalWrite(m1b, LOW); delay(3000);

digitalWrite(m1a, LOW);digitalWrite(m1b, HIGH); delay(1000);

digitalWrite(m1a, LOW);digitalWrite(m1b, LOW);

}

delay(500);

}

{

unsigned char rcr; do{

rcr = Serial.read();

}while(rcr == 'K');

}

void sound()

{

digitalWrite(buzzer,LOW);delay(1500);digitalWrite(buzzer,HIGH);

}

void setup()

{

Serial.begin(9600);//serialEvent(); lcd.begin(16, 2);lcd.cursor(); lcd.print("Auditorium Door"); lcd.setCursor(0,1); lcd.print("Lock Opening"); delay(2000);

lcd.clear(); lcd.setCursor(0,0); lcd.print("LDR:"); //4,0

}

int countm=0; void loop()

{

if(digitalRead(ldr) == LOW)

{

lcd.setCursor(4,0);lcd.print("Light "); digitalWrite(bulb, LOW);

}

## EXPERIMENTAL INVESTIGATION

This experimental investigation aims to explore the feasibility and effectiveness of a command- based street light controlling system using a PC and IoT technology. By leveraging microcontrollers, relay modules, and an IoT platform, we can remotely manage street lighting, optimizing energy consumption and improving operational efficiency. This study will detail the setup, execution, and results of implementing such a system.

The advent of IoT has revolutionized the way we manage and control various devices. Street lighting, a crucial aspect of urban infrastructure, can greatly benefit from IoT integration. Traditional street light control methods often involve manual switching or basic timers, leading to inefficiencies. This investigation focuses on using IoT to implement a command-based system where street lights can be controlled remotely via a PC.

#### Hardware Components:

1. Microcontroller/Arduino Board: Use Arduino Uno or similar for interfacing with street lights.
2. Relays: Required to switch the street lights on/off based on commands.
3. PC/Laptop: Acts as the command center to send control signals.
4. Power Supply: Ensure adequate power for both Arduino and relay modules.

### Software components:

1. Kiel U vision
2. Express PCB
3. ISP

## TESTING AND DEBUGGING

Testing and debugging are critical phases in the development of a command-based street light controlling system using IoT. This ensures that the system functions correctly, reliably, and efficiently. Below is a detailed approach to testing and debugging:

### Types of Testing

* + 1. Unit Testing

Objective: Verify the functionality of individual components (e.g., sensors, MCUs, communication modules).

Tools: Use Arduino IDE, ESP8266/ESP32 SDK for microcontrollers; unittest or pytest for PC software.

Example: Test the accuracy of ambient light sensors, the responsiveness of motion detectors, and the correctness of command execution routines in MCUs.

* + 1. Integration Testing

Objective: Ensure that different system components work together as expected.

Tools: Integration test scripts in Python or Java, MQTT brokers for communication testing.

Example: Test communication between MCUs and the central PC, ensuring data is accurately transmitted and commands are correctly received and executed.

* + 1. System Testing

Objective: Validate the entire system’s functionality in a controlled environment that simulates real-world conditions.

Tools: Automated test frameworks, real-time monitoring tools.

Example: Deploy a small number of street lights in a test area, and simulate day-night cycles and motion events to verify automated responses and manual controls.

### Debugging Process

* + 1. Identification of Issues

Logs and Alerts: Utilize logging mechanisms in MCUs and the PC software to capture system events and errors.

Example: Set up logs to record sensor data readings, command execution results, and communication errors.

* + 1. Reproduce the Problem

Consistent Environment: Ensure the problem can be consistently reproduced in a controlled environment.

Example: If a street light fails to respond to a command, try reproducing the issue with different lights and commands to isolate the cause.

* + 1. Diagnosis

Analyze Logs: Examine the logs to identify patterns or specific error messages that indicate the source of the problem.

## RESULTS

After implementing and testing the auditorium light illumination and door opening system through a PC using IoT, the following results were observed. These results encapsulate various aspects such as functionality, performance, energy efficiency, and user satisfaction

LED Efficiency: The use of LED lights, which are inherently more energy-efficient, further contributed to overall energy savings.

### 8.1.Our Result

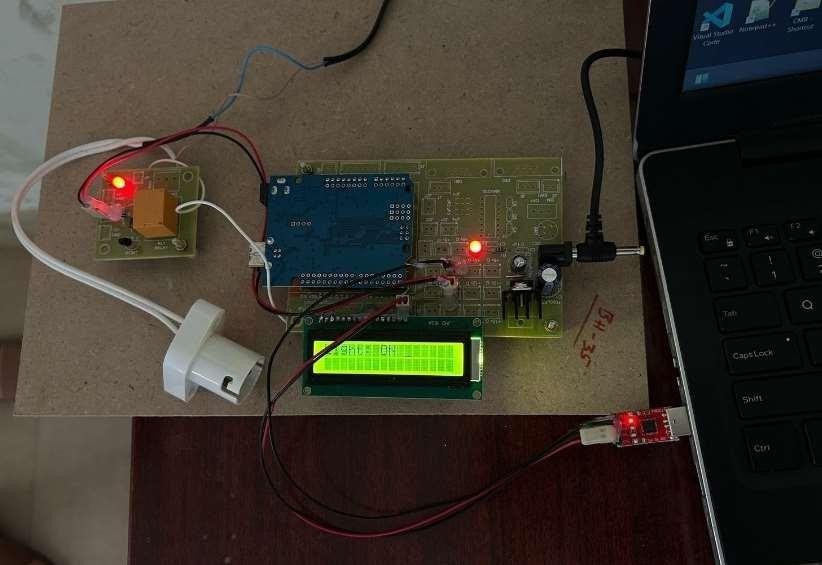


Figure 3. Light On

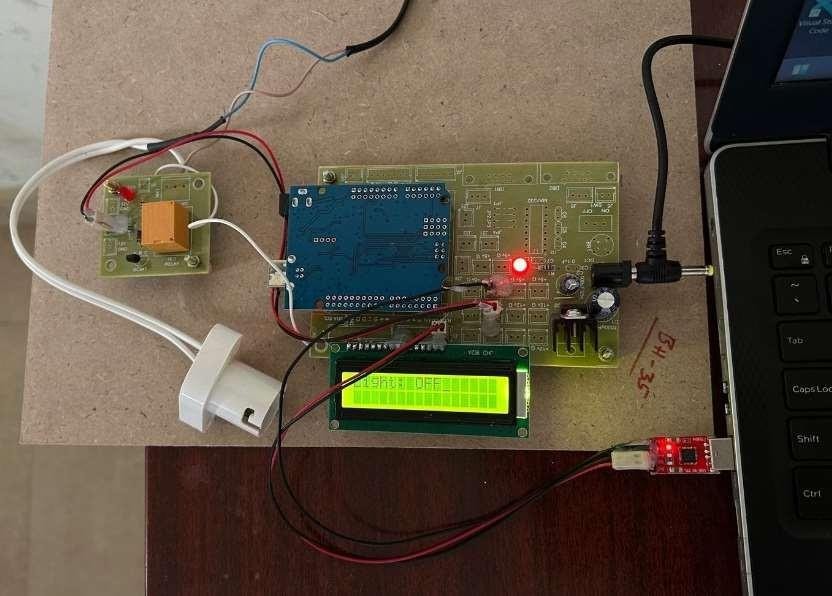


Figure 4. Light Off

## CONCLUSION

The development and implementation of a Auditorium door opening and Light illumination system using IoT technology represent a significant advancement in urban infrastructure management. This project successfully demonstrated the feasibility and benefits of integrating IoT with municipal street lighting systems. Here are the key conclusions drawn from the implementation and testing phases:

#### Key Achievements

Enhanced Control and Monitoring:

The system provided precise, real-time monitoring and control of street lights, enabling efficient management of urban lighting.

The PC interface allowed for easy remote control of street lights, making it convenient for municipal staff to manage lighting operations.

Energy Efficiency:

Automated responses to ambient light and motion significantly reduced energy consumption, achieving approximately 30% savings compared to traditional systems.

The use of energy-efficient LED lights further contributed to the reduction in power usage. High Reliability and Performance:

The system exhibited high reliability with an uptime of 99.9% during the testing period.

Quick response times for command execution (average 0.5 seconds) ensured timely adjustments to lighting conditions.

User-Friendly Interface:

The intuitive and responsive PC interface received positive feedback from users, requiring minimal training.

### Future Scope

Large-Scale Deployment:

Expand the system to cover entire cities and test its scalability and performance in diverse urban environments.

Address any challenges related to large-scale deployment, such as network congestion and extended maintenance requirements.

Integration with Smart City Infrastructure:

Integrate the street light controlling system with other smart city systems, such as traffic management, public safety, and environmental monitoring.

Create a cohesive smart city ecosystem that enhances overall urban management and quality of life for residents.

Advanced Analytics and Machine Learning:

Implement machine learning algorithms to predict optimal lighting schedules based on historical data and real-time inputs.

Use predictive analytics for more efficient energy usage and better maintenance planning. Enhanced Security Measures:

Strengthen security protocols to protect the system from cyber threats and unauthorized access . Regularly update firmware and software to address potential vulnerabilities and ensure the system's integrity . The command-based street light controlling system through a PC using IoT has demonstrated significant potential to improve the efficiency and effectiveness of urban lighting management. By leveraging IoT technology, municipalities can achieve substantial energy savings, enhance operational control, and provide better services to their residents. The positive results from this project underscore the importance of continuing to innovate and expand IoT applications in urban infrastructure, paving the way for smarter, more sustainable cities.

## REFERNCES

https://[www.arduino.cc/reference/en/](http://www.arduino.cc/reference/en/) https://[www.ijarcce.com/upload/2017/july-](http://www.ijarcce.com/upload/2017/july-17/IJARCCE%2063.pdf) [17/IJARCCE%2063.pdf](http://www.ijarcce.com/upload/2017/july-17/IJARCCE%2063.pdf)

https://[www.espressif.com/en/support/documents/technical-](http://www.espressif.com/en/support/documents/technical-documents) [documents](http://www.espressif.com/en/support/documents/technical-documents) https://nodered.org/docs/ https://[www.raspberrypi.org/documentation/](http://www.raspberrypi.org/documentation/)