VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnanasangama, Macche, Santibastwada Road
Belagavi-590018, Karnataka

A
BE MINI-PROJECT (19EC6DCMPR) REPORT
ON

"IoT-based Smart Lighting System using Motion and Noise detecting techniques"

Submitted in partial fulfillment of the requirement for the degree of

Bachelor of Engineering
in
Electronics & Communications Engineering - ECE
by

USN: 1DS20EC140 **Name of the student: Pratiksha Sahoo**
USN: 1DS20EC150 **Name of the student: Rishav Jaiswal**
USN: 1DS20EC177 **Name of the student: Shivanshu Singh**
USN: 1DS20EC185 **Name of the student: Shubhang Sharan Srivastava**

Under the guidance
of
Prof. Bindu H. M.
Assistant Professor
ECE Dept., DSCE, Bengaluru

Department of Electronics & Communication Engineering
(An Autonomous College affiliated to VTU Belgaum, accredited by NBA & NAAC, Ranked by NIRF)
Shavige Malleshwara Hills, Kumaraswamy Layout,
Bengaluru-560111, Karnataka, India

2022-23

Certificate

Certified that the mini-project work **(19EC6DCMPR)** entitled "**IoT-based Smart Lighting System using motion and sound detecting techniques**" carried out by **Pratiksha Sahoo** (1DS20EC140), **Rishav Jaiswal** (1DS20EC150), **Shivanshu Singh** (1DS20EC177), **Shubhang Sharan Srivastava** (1DS20EC185) are bonafide students of the ECE Dept. of Dayananda Sagar College of Engineering, Bangalore, Karnataka, India in partial fulfillment for the award of Bachelor of Engineering in Electronics & Communication Engineering of the Visvesvaraya Technological University, Belagavi, Karnataka for the VI Semester course during the academic year 2023. It is certified that all corrections/suggestions indicated for the mini-project work have been incorporated in the mini-report submitted to the ECE department. This Mini-Project report has been approved as it satisfies the academic requirement in respect of mini-project work prescribed for the said degree.

Mini-Project Guide Sign: ____________________

Name: **Prof. Bindu H. M.**

Mini-Project Section Coordinator Sign: ____________________

Name: **Prof. Lavanya Krishnamurthy**

Mini-Project Convener & Chief Coordinator Sign: ____________________

Name: **Dr. Shashi Raj K.**

Dr. T. C. Manjunath: ____________________

HOD, ECE, DSCE

Dr. B. G. Prasad: ____________________

Principal, DSCE

External Mini-Project Viva-Voce (SEE)

Name of the mini-project examiners (int & ext) with date:

1: Signature: ____________________

2: Signature: ____________________

Declaration

Certified that the mini-project work entitled, "**IoT-based Smart Lighting System using motion and sound detecting techniques**" with the course code **19EC6DCMPR** is a bonafide work that was carried out by ourselves in partial fulfillment for the award of degree of Bachelor of Engineering in Electronics & Communication Engineering of the Visvesvaraya Technological University, Belagavi, Karnataka during the academic year 2022-23 for the VI Semester Autonomous Course. We, the students of the mini-project batch no. C1 do hereby declare that the entire mini-project has been done on our own & we have not copied or duplicated any other's work. The results embedded in this mini-project report has not been submitted elsewhere for the award of any type of degree.

Student Name-1: Ms. Pratiksha Sahoo
USN: 1DS20EC140

Sign: ____________________

Student Name-2: Mr. Rishav Jaiswal
USN: 1DS20EC150

Sign: ____________________

Student Name-3: Mr. Shivanshu Singh
USN: 1DS20EC177

Sign: ____________________

Student Name-4: Mr. Shubhang Sharan Srivastava
USN: 1DS20EC185

Sign: ____________________

Date: 08/07/2023

Place: Bengaluru

Acknowledgement

We thankfully acknowledge our deep sense of gratitude to our Chairman, Dr. Hemachandra Sagar; Vice Chairman, Dr. Premachandra Sagar; Secretary, Galiswamy; Joint Secretary, Tintisha Sagar for their support while carrying out the mini-project work.

We also thankfully acknowledge our deep sense of gratitude to the principal of DSCE, Dr. B. G. Prasad and HOD of the Electronics and Communication Engineering department, Dr. T. C. Manjunath for their encouragement and support throughout the mini-project tenure.

We thankfully acknowledge our mini-project guide, Prof. Bindu H. M. for her valuable guidance and encouragement throughout the mini-project work.

We thankfully acknowledge the section coordinators, Prof. Trupti Tagare, Prof. Kavita Guddad, Prof. Lavanya Krishnamurthy, Prof. R Santhosh Kumar for their valuable support and encouragement during the mini-project phase.
We thankfully acknowledge the mini-project convener & the chief coordinator, Dr. Shashi Raj K for his constant support and encouragement during the mini-project phase.

We too thankfully acknowledge the teaching and non-teaching staffs of the institution, and our friends who have directly or indirectly supported and inspired us during the mini-project work.

Last, but not the least, we would like to thank our family members and relatives for their blessings and best wishes without which the mini-project work would not have come to an end.

Finally, we thank the Almighty for their blessings without which we would not have succeeded in carrying through the mini-project work.

Abstract

The rapid advancement of Internet of Things (IoT) technology has led to the development of innovative solutions for various domains. In this paper, we propose an IoT-based smart lighting system designed specifically for outdoor areas, assimilating motion and noise detecting techniques with the microcontroller.

The proposed smart lighting system utilizes a network of interconnected IoT devices, including motion sensors, noise sensors, and smart LED lights. The motion sensors detect human presence and movement within the designated area, while the noise sensors capture ambient sound levels. By integrating these sensors with the IoT architecture, the system can collect real-time data and make intelligent decisions based on the detected inputs.

To enable seamless communication and control, the system leverages IoT protocols and cloud-based platforms. This allows for remote monitoring, control, and configuration of the smart lighting system, providing administrators with real-time insights and the ability to make informed decisions for efficient resource allocation and maintenance.

In conclusion, the IoT-based smart lighting system proposed in this paper offers a comprehensive solution for outdoor areas, incorporating motion and noise detecting techniques. By leveraging the capabilities of IoT devices and cloud-based platforms, the system provides intelligent lighting control, improving energy efficiency, safety, and overall user experience in outdoor environments.

Keywords: *IoT, Lighting System, Microcontroller, Motion, Noise, Sensors.*

Table of Contents

List of Figures

List of Tables

Nomenclature and Acronyms

Abbreviations (Alphabetical Order):

AI	Artificial Intelligence
ADDR	Address Line
A0	Analog Input
DSCE	Dayananda Sagar College of Engineering
D0	Digital Output
ECE	Electronics and Communication Engineering
GND	Ground
GPIO	General Purpose Input/Output
LED	Light Emitting Diode
ML	Machine Learning
IDE	Integrated Development Environment
IoT	Internet of Things
RX2	Receiver-2
SCL	Serial Clock Line
SDL	Serial Data Line
VCC	Voltage Common Collector
Wi-Fi	Wireless Fidelity

Symbols (Alphabetical Order):

dB	Decibels
lux	Illuminance
m	Meter
V	Volts

Chapter 1: Introduction

1.1. Overview

The building sector is the most energy-consuming sector, accounting for more than half of worldwide electricity consumption and more than one-third of total energy consumption. The International Energy Agency projects that if specific steps to improve building energy efficiency are not implemented, building energy demand will grow by 50% by 2050. Lighting systems account for a significant amount of energy usage in buildings [1]. The IoT-based smart light system proposed combines the ultrasonic distance sensor, sound sensor, light intensity sensor, ESP32 Wi-Fi/Bluetooth microcontroller module and the BLYNK IoT Cloud platform to create an intelligent lighting solution.

1.2. Literature Review

- During the period 2000-2010, researchers and technology enthusiasts began exploring the concept of connecting everyday objects to the internet. Prototypes of smart lighting systems started emerging, demonstrating the potential of integrating lighting fixtures with sensors, wireless connectivity, and control systems. These early systems focused on basic functionalities like remote control and automation [3].
- Between the years, 2010-2015, the emergence of energy-efficient LED lighting technology played a significant role in the development of smart lighting systems. LED lights offered numerous advantages, including longer lifespan, lower energy consumption, and controllability. This period also witnessed the development and standardization of wireless communication protocols, such as Zigbee and Z-Wave, which played a crucial role in the growth of IoT-based smart lighting. The wireless nature of these protocols made installation and configuration of smart lighting systems easier and more flexible [4].

- In the recent years, with the rise of smartphones, smart home devices, and voice assistants, IoT-based smart lighting systems started gaining widespread adoption. Integration with popular smart home platforms, like Apple HomeKit, Google Home, and Amazon Alexa, made it easier for users to control their lighting through voice commands and smartphone apps. Smart lighting systems expanded beyond basic functionality, incorporating features such as motion sensing, daylight harvesting, colour tuning, and energy monitoring. The introduction of cloud-based platforms allowed for remote management, data analytics, and the ability to create personalized lighting experiences [5].
- Future advancements in IoT-based smart lighting systems may see AI algorithms to analyse the sensor data, user preferences, and environmental factors to optimize lighting conditions automatically. ML algorithms can be implemented by learning from the user's behaviour patterns to anticipate lighting needs and make intelligent adjustments [6].

1.3. Objectives

The main aim of this project is to create a low-cost lighting control system that allows the user to control the operation of the lights via a cloud-based system. The cross-platform programs display excess energy use, management, and even monitoring capabilities to reduce efforts to keep up with rising energy costs. The scope of this article includes device principles, the device development process, software and hardware workflow, and installation. The research further seeks to streamline and easily adapt smart lighting systems for any purpose.

1.4. Methodology

1. System Design and Sensor Integration:

- Identify the requirements and specifications of the smart lighting system based on the project objectives.

- Determine the hardware components needed, such as the BH-1750 light sensor, HC-SR04 ultrasonic distance sensor, KY-038 sound sensor, and integrate the sensors with the microcontroller.
- Configure the sensors, test its integration and verify the data accuracy.

2. Data Acquisition and Processing:

- Develop the code or programming logic to acquire data from the sensors through the microcontroller.
- Implement algorithms for processing and analyzing the acquired sensor data.
- Determine the thresholds or rules for decision-making based on the sensor inputs.

3. Decision-Making and Control:

- Develop algorithms or rules to make decisions regarding lighting actions based on the processed sensor data.
- Determine the control mechanisms for the lighting system, such as turning the lights on/off, dimming, or changing the color temperature.
- Program the microcontroller to send control signals to the lighting fixtures based on the decisions made.

4. User Interface Development:

- Design and develop a user interface, such as a mobile app or web portal, to allow users to interact with the smart lighting system.
- Implement features for customization, manual control, and monitoring of the lighting settings.

5. Performance Evaluation:

- Evaluate the performance of the smart lighting system based on predefined metrics, such as energy efficiency, accuracy of light control, and user satisfaction.

- Gather data and analyze the system's performance to identify any limitations or areas for improvement.

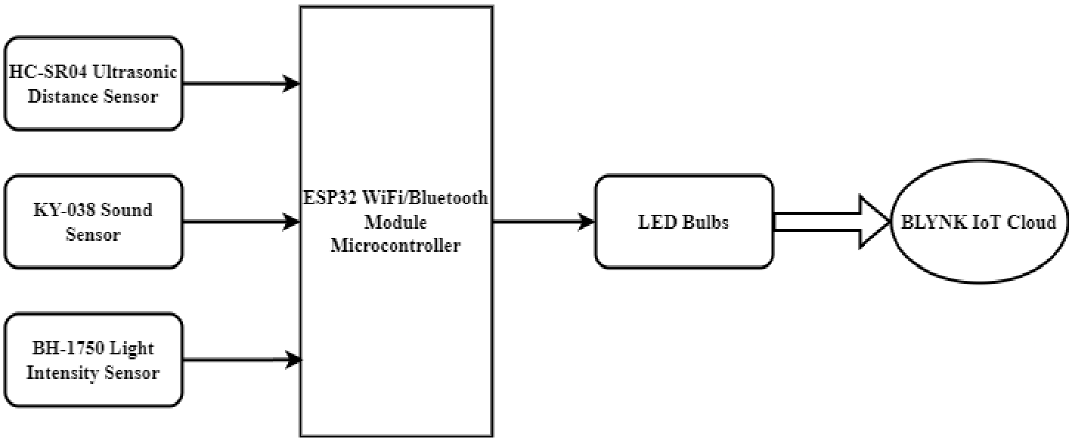
1.5. Organization of the mini-project work

- Chapter-1 gives a brief introduction to the mini-project work that is done during the given period. It includes a brief overview of the project, literature survey gathered from different sources, objectives and methodology involved in the project.
- Chapter-2 includes block diagram, flowchart, circuit diagram and the working principle of the project.
- Chapter-3 gives the description of hardware and software tools used in the project, and their working.
- Chapter-4 includes the results and discussions gathered from this project.
- Chapter-5 includes the applications, advantages, outcomes and limitations with regards to the project.
- Chapter-6 gives the conclusion to the project and mentions their future works to be done.

Chapter 2: Block Diagram, Flowchart, Circuit Diagram and Working Principle

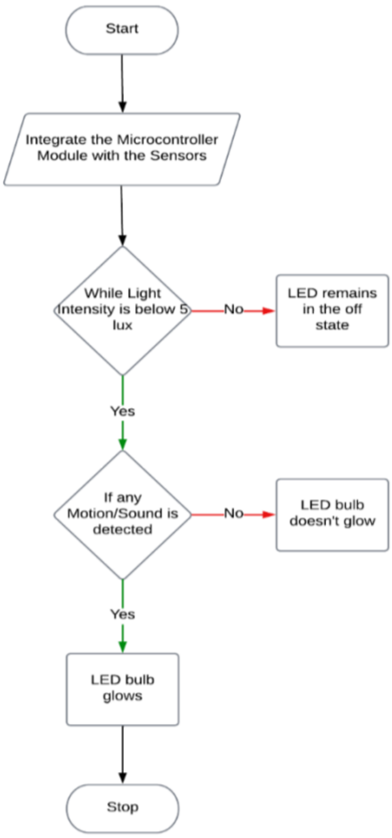
Block Diagram

Fig.1:



Flowchart

Fig.2:



Circuit Diagram

Table-1:

Sl.No:	Components used:	Pin connections to ESP32 module:
1)	BH-1750 Light Intensity Sensor	• VCC is connected to 3.3V. • GND is connected to GND. • SCL is connected to D22. • SDL is connected to D21. • ADDR has no connection.
2)	HC-SR04 Ultrasonic Distance Sensor	• VCC is connected to 3.3V. • Trigger is connected to D27. • Echo is connected to D14. • GND is connected to GND.
3)	KY-038 Sound Sensor	• A0 is connected to D35. • GND is connected to GND. • VCC is connected to 3.3V. • D0 has no connection.
4)	LED bulbs	• Positive terminal is connected to RX2, i.e, GPIO-16. • Negative terminal is connected to GND.

Working Principle

The IoT-based smart light system utilizes a combination of sensors and the ESP-32 microcontroller to create an intelligent lighting solution. The ultrasonic sensor detects the presence of motion, the sound sensor detects ambient noise levels, and the light intensity sensor measures the brightness of the surroundings. These sensor inputs are processed by the ESP-32 microcontroller, which communicates with the BLYNK IoT Cloud Web dashboard and Mobile App.

Chapter 3: Hardware and Software tools used, with their descriptions, and working

Hardware tools used and their description

Fig.3:

ESP-32 WIFI/Bluetooth Module Microcontroller

KY-038 Sound Sensor

HC-SR04 Ultrasonic Distance Sensor

BH-1750 Light Sensor

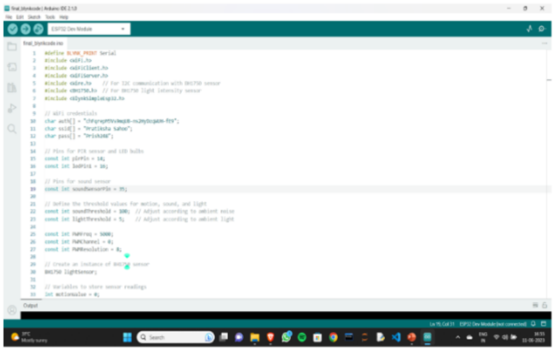
Table-2:

Sl.No:	Hardware tools used:	Features:
1)	ESP32 WiFi/Bluetooth Microcontroller module	• Low cost and low power system. • Supports dual Bluetooth and a Wi-Fi network.
2)	HC-SR04 Ultrasonic distance Sensor	• Gives a digital output of 3.3V. • Ranges between 2cm-4m.
3)	KY-038 Sound Sensor	• Gives a digital output of 5V. • Ranges between 3dB-100dB.

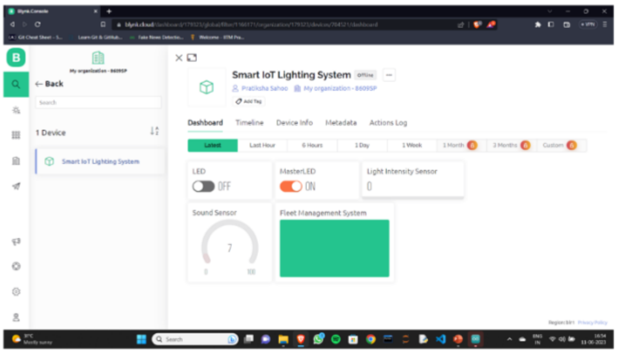
4)	BH-1750 Light Intensity Sensor	• Gives a digital output of 5V. • Ranges between 0 lux-1 lakh lux.

Software tools used and their description

Fig.4:



Arduino IDE Platform



BLYNK IoT Cloud Platform

Table-3:

Sl.No:	Software tools used:	Features:
1)	Arduino IDE platform	• Uses C++ as the programming language. • The library manager makes it easy to find, download, and manage libraries.
2)	BLYNK IoT Cloud	• Includes widgets like buttons, sliders, gauges, graphs, and other elements. • Makes it user-friendly to manage IoT gadgets through a smartphone app.

Working

- Hardware Setup:
    - Connect the ultrasonic distance sensor, sound sensor, and light intensity sensor to the ESP-32 microcontroller according to their respective pin configurations.
    - Ensure that the ESP-32 is properly connected to the internet and has access to the BLYNK IoT Cloud platform.
- Sensor Data Acquisition:
    - The ultrasonic distance sensor detects human presence by sensing its distance from the reference point. It generates a signal when motion is detected.
    - The sound sensor detects sound intensity and produces an electrical signal proportional to the detected sound level.
    - The light intensity sensor measures the amount of ambient light and generates an electrical signal representing the light level.
- Microcontroller Integration:
    - The ESP-32 microcontroller reads the sensor data from the ultrasonic distance sensor, sound sensor, and light intensity sensor.
    - It uses the appropriate digital or analog pins to interface with these sensors and collects their data.
- Data Processing:
    - The microcontroller processes the sensor data to determine the appropriate actions to be taken based on predefined conditions.
    - If any motion is detected by the ultrasonic sensor, the microcontroller can decide to turn on the lights.
- Control and Communication:
    - The ESP-32 microcontroller communicates with the BLYNK IoT Cloud platform using Wi-Fi or other available network connectivity options.

    - It establishes a connection with the BLYNK IoT Cloud and sends the processed sensor data.
- BLYNK IoT Cloud Integration:
    - The BLYNK IoT Cloud receives the sensor data from the microcontroller.
    - The BLYNK platform provides a web dashboard and a mobile app interface to monitor and control connected devices.
    - The received sensor data can be displayed on the BLYNK dashboard, allowing users to monitor the status of the sensors.
- Dashboard Actions:
    - Based on the received sensor data, the BLYNK IoT Cloud platform can trigger predefined actions.
    - If any motion is detected, the platform sends a command to the microcontroller to turn the lights on.
    - It can also perform actions based on sound intensity or light level, such as adjusting the brightness of the lights.
- User Interaction:
    - Users can interact with the smart light system through the BLYNK mobile app or web dashboard.
    - They can monitor sensor readings, control the lights manually, set automation rules, and receive notifications.
- Feedback and Monitoring:
    - The BLYNK IoT Cloud platform provides feedback on the status of the lights, sensor readings, and any other relevant information.
    - Users can continuously monitor the system's behavior and make adjustments as needed.

Chapter 4: Results and Discussions

Results

Fig. 5:

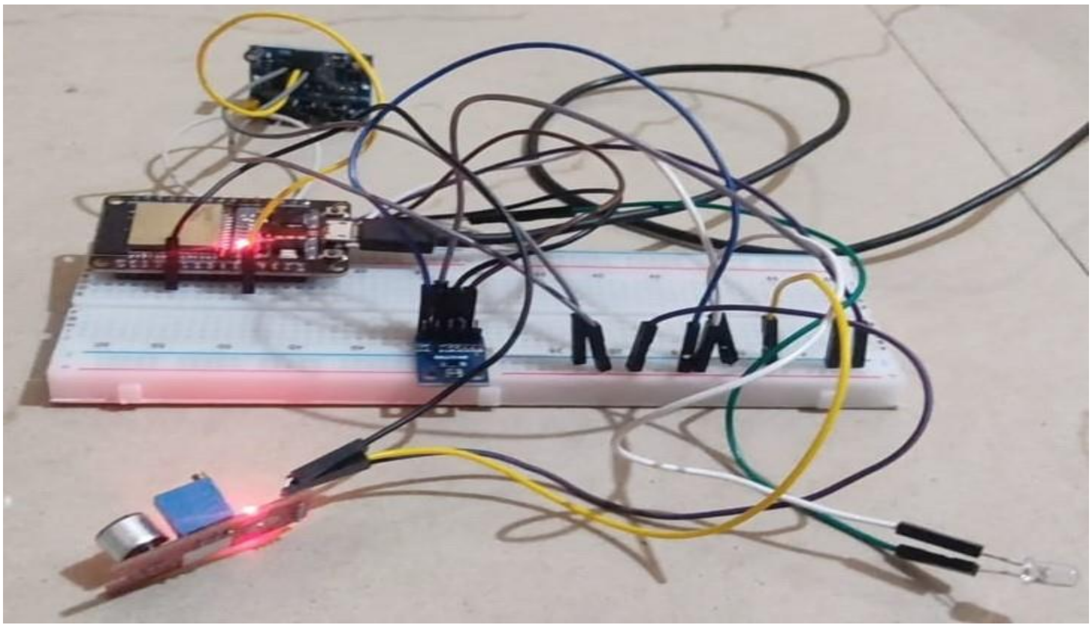


Fig. 6:

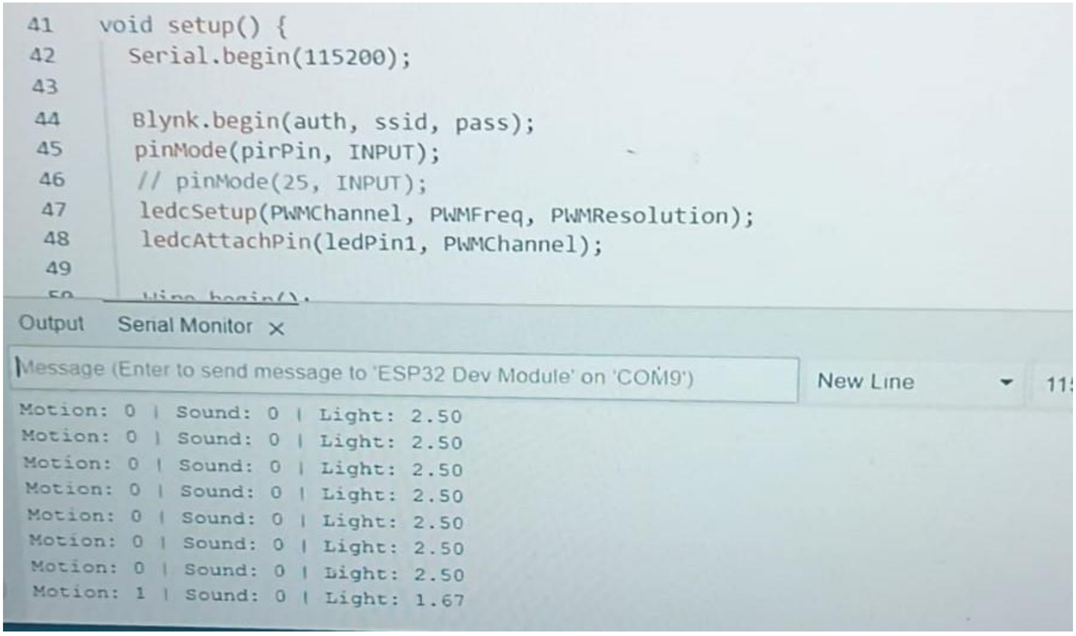


Fig. 7:

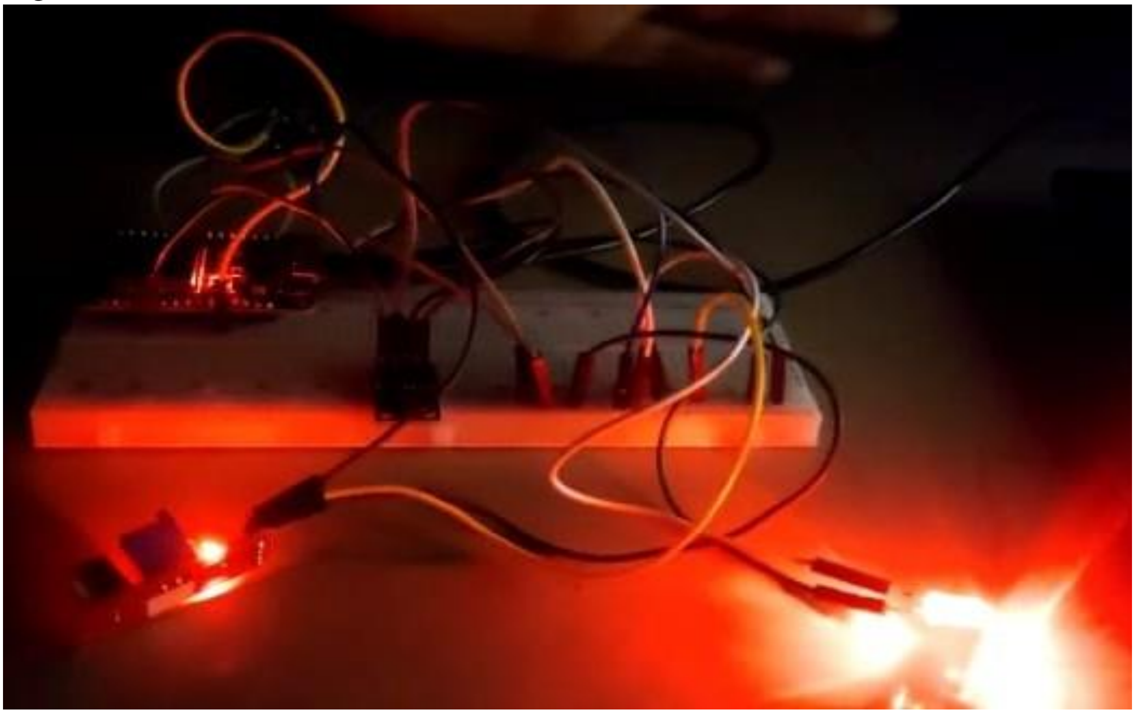


Fig. 8:

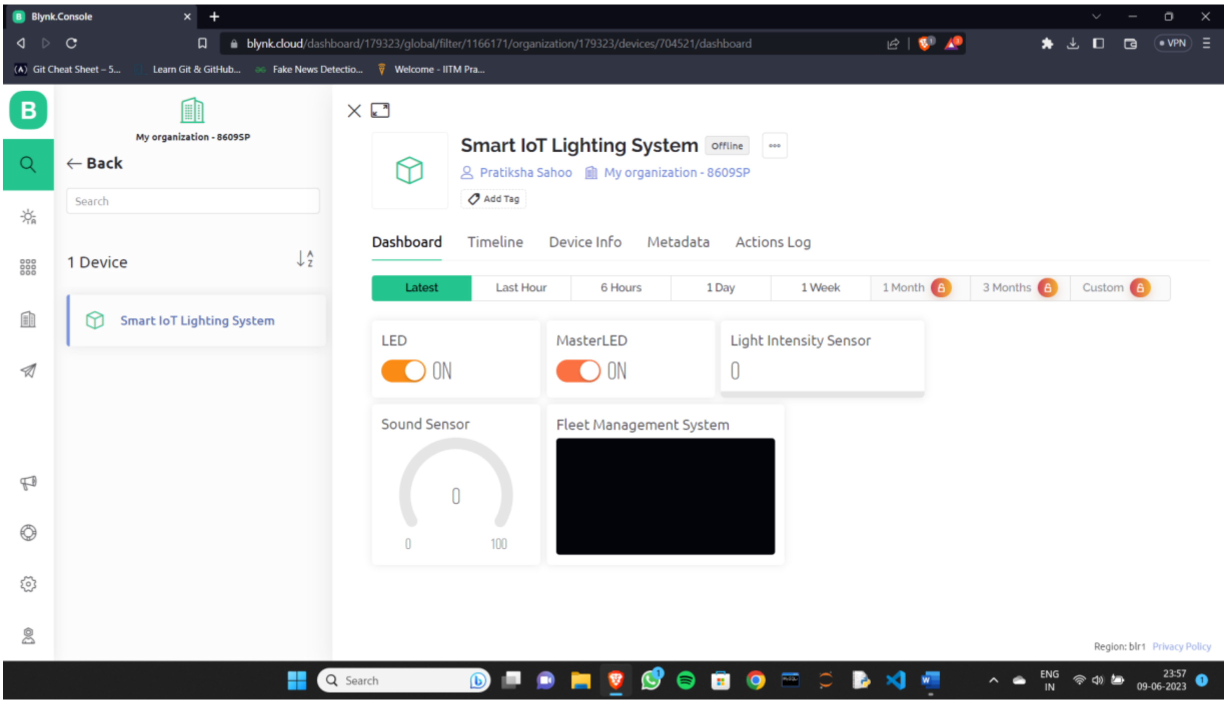
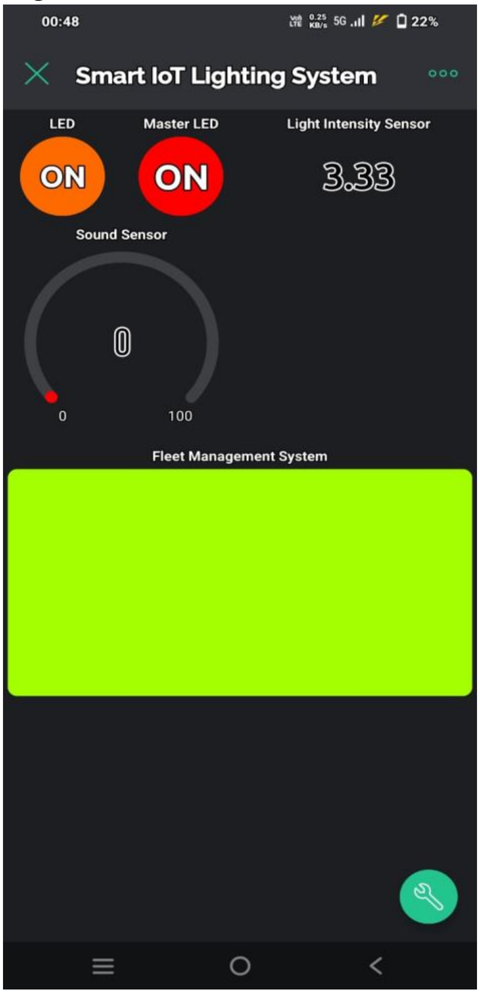

Blynk.Console
blynk.cloud/dashboard/179323/global/filter/1166171/organization/179323/devices/704521/dashboard
My organization - 8609SP
Back
Search
1 Device
Smart IoT Lighting System
Smart IoT Lighting System
Offline
Pratiksha Sahoo
My organization - 8609SP
Add Tag
Dashboard
Timeline
Device Info
Metadata
Actions Log
Latest
Last Hour
6 Hours
1 Day
1 Week
1 Month
3 Months
Custom
LED
ON
MasterLED
ON
Light Intensity Sensor
0
Sound Sensor
0
0
100
Fleet Management System
Region: blr1
Privacy Policy

Fig. 9:



Discussions

- The system provides efficient energy by automatically dimming the lights to a certain intensity when no one is around the space. This results in significant energy savings over the time.
- The system provides convenient and safety by automatically turning on the lights to its full intensity when someone enters the space, eliminating the need to manually switch them on. This can also help improve the safety by ensuring that the space is well-lit when someone enters.
- It provides easy maintenance by reducing enough cost while maintaining the prototype than the usual traditional lighting system.

Chapter 5: Applications, Advantages and Limitations

Applications

- **Internet of Things (IoT) Integration:** This project shows how to combine lighting systems with IoT technologies. Using a smartphone app or online interface, anyone can remotely monitor and manage the lighting system thanks to the ESP32 microcontroller, sensors, and Blynk [2].
- **Energy-efficient buildings:** The system can be installed in office or commercial buildings to optimize lighting use, lower energy use, and boost overall energy efficiency. It helps achieve sustainability objectives and cut back on operating expenses [4] [6].
- **Public areas:** By supplying sufficient lighting when motion is detected, this system can assist parks, parking lots, and other public areas. It guarantees pedestrian security and raises visibility in these locations generally [1].

Advantages

- **Energy Efficiency:** By automatically altering lighting based on occupancy and environmental circumstances, the system greatly lowers energy use. Over time, it ensures that lights are only switched on as needed, resulting in significant energy savings [4].
- **Convenience and Automation:** The system intelligently reacts to motion and noise, providing users with a simple and hands-free lighting experience. It delivers smooth automation and gets rid of the necessity for manual light switching [2].
- **Enhanced Safety and Security:** It improves safety in a variety of scenarios by incorporating motion sensing into the lighting system. A sense of security and a deterrent to potential attackers can be created by programming lights to turn on when motion is sensed [6].
- **Remote Monitoring and Control:** The system supports remote monitoring and control of the lighting system by integrating IoT technology and

platforms like Blynk. Through a web interface or mobile app, users may log in to the system, check the status, make changes, and receive notifications remotely [3].

Limitations

- **Sensor Accuracy:** The accuracy of the sensors used for motion detection, noise detection, and light intensity measurement are crucial to the system's performance. Inefficient lighting controls and the range of the sensors brought on by faulty sensor readings can be inconvenient or wasteful of energy.
- **Sensitive to Environmental conditions:** The system's performance may be impacted by outside interferences, changes in ambient light levels, and changes in noise levels, among other environmental conditions. These elements may affect the sensors' accuracy and dependability, resulting in inaccurate lighting control or false triggers.

Chapter 6: Conclusions and Future Work

Conclusions

In conclusion, the IoT-based smart light system utilizing ultrasonic distance sensor, sound sensor, and light intensity sensor integrated with the ESP-32 microcontroller and BLYNK IoT Cloud proves to be a highly efficient and intelligent solution. The system effortlessly detects human presence, ambient noise, and lighting conditions to automatically control the illumination, ensuring optimal energy consumption and enhanced convenience. The seamless integration with the BLYNK IoT Cloud Web dashboard and Mobile App enables users to remotely monitor and control the smart lights with ease. This combination of advanced sensors, a powerful microcontroller, and the user-friendly BLYNK IoT Cloud platform offers a comprehensive and convenient solution for modern lighting automation needs.

Future Work

In the future, IoT-based smart light systems will continue to evolve and offer advanced features and capabilities. These systems will become even more integrated with other smart devices and home automation platforms, allowing seamless control and customization of lighting settings. They will incorporate advanced sensors, and AI and ML algorithms to automatically adjust lighting based on occupancy, ambient light levels, and user preferences, optimizing energy efficiency. Enhanced connectivity and security protocols will ensure reliable and secure communication between devices, making IoT-based smart light systems an integral part of intelligent homes and buildings.

References

[1] Yusi Cheng, Chen Fang, Jingfeng Yuan and Lei Zhu, "Design and Application of a Smart Lighting System Based on Distributed Wireless Sensor Networks", *Applied Sciences, MDPI*, Nanjing, China, 2020, DOI: 10.3390/app10238545.

[2] Francis Jesmar P. Montalbo and Erwin L. Enriquez, "An IoT Smart Lighting System for University Classrooms", *International Symposium on Educational Technology (ISET)*, Batangas City, Philippines, 2020, DOI: 10.1109/ISET49818.2020.00011.

[3] Timothy Malche and Priti Maheshwary, "Internet of Things (IoT) for building smart home system", *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, India, 2017, DOI: 10.1109/I-SMAC.2017.8058258.

[4] Xu Qin, Zhengtian Wu, Baochuan Fu and Shuxian Li, "An Energy-saving Design of Smart Light Controller Based on the Position of Person", *2018 13th World Congress on Intelligent Control and Automation (WCICA)*, Changsha, China, 2018, DOI: 10.1109/WCICA.2018.8630735.

[5] Reda Jabeur, Najat Ouaaline and Abderrazak Lakrim, "A Fuzzy Logic Controller Controls a Smart Lighting System for Energy Savings", *2021 9th International Renewable and Sustainable Energy Conference (IRSEC)*, Morocco, Russia, 2021, DOI: 10.1109/IRSEC53969.2021.9741196.

[6] S. Dhanasekaran, P. Gomathi, A. Renaldo Maximus, Thiyagarajan Krishnan and B. Kannan, "Solar Tree based Smart City Street Light Control System using IoT BLYNK Platform", *2022 International Conference on Automation, Computing and Renewable Systems (ICACRS)*, Pudukkottai, India, 2022, DOI: 10.1109/ICACRS55517.2022.10029107.

Appendix

```
#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <WiFiServer.h>
#include <Wire.h>    // For I2C communication with BH1750 sensor
#include <BH1750.h> // For BH1750 light intensity sensor
#include <BlynkSimpleEsp32.h>
#define SOUND_SPEED 0.034
// WiFi credentials
char auth[] = "chFqrepM5Vs9mqUB-ns2HyDzqWUH-fE9";
char ssid[] = "Pratiksha Sahoo";
char pass[] = "Prish248";
// Pins declaration
const int trigPin = 27;
const int echoPin = 14;
const int ledPin1 = 16;
const int soundSensorPin = 35;
// Define the threshold values for motion, sound, and light
long duration;
float distanceCm;
const int soundThreshold = 150; // Adjust according to ambient noise
const int lightThreshold = 5; // Adjust according to ambient light
const int PWMFreq = 5000;
const int PWMChannel = 0;
const int PWMResolution = 8;
BH1750 lightSensor;
// Variables to store sensor readings
int motionValue = 0;
int soundValue;
float lightValue = 0.0;
// Variable to track the state of light from Blynk app
static bool masterControl = true;
```

```
static bool lightOn = false;
void setup() {
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass);
  ledcSetup(PWMChannel, PWMFreq, PWMResolution);
  ledcAttachPin(ledPin1, PWMChannel);
  Wire.begin();
  lightSensor.begin();
  Serial.print("Initial Light State: ");
  Serial.println(lightOn);
  pinMode(soundSensorPin, INPUT);
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
}
void loop() {
  Blynk.run();
  soundValue = digitalRead(soundSensorPin);
  lightValue = lightSensor.readLightLevel();
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  Serial.println(duration);
  distanceCm = duration * SOUND_SPEED/2;
  Serial.print("Distance (in cm): ");
  Serial.println(distanceCm);
  Serial.print(" | Sound: ");
  Serial.print(!soundValue);
  Serial.print(" | Light: ");
  Serial.println(lightValue);
  if (masterControl)
    return;
```

```
  if (lightValue < lightThreshold) {
    if (distanceCm < 20 || soundValue < 1) {
      ledcWrite(PWMChannel, 255);
      delay(2000);
    } else {
      ledcWrite(PWMChannel, 10);
    }
  } else {
    ledcWrite(PWMChannel, 0);
  }
  delay(30);
   double soundLevel = map(soundValue, 0, 1023, 0, 100);  // Map the analog value to a
range of 0 to 100
   Blynk.virtualWrite(V2, soundLevel);
   Blynk.virtualWrite(V3, lightValue);
   if (motionValue == HIGH)
  {
    Blynk.virtualWrite(V1, "Motion is detected!");
  }
  else
  {
    Blynk.virtualWrite(V1, "No motion is detected!");
  }
}
BLYNK_CONNECTED()
{
  Blynk.syncVirtual(V0);
  Blynk.syncVirtual(V4);
  Blynk.syncVirtual(V1);
  Blynk.syncVirtual(V2);
  Blynk.syncVirtual(V3);
}
BLYNK_WRITE(V0) // Button widget on V0
{
```

```
  int value = param.asInt();  // Get value from the button
  BLYNK_LOG("Got value in V0 %d ",value);
  lightOn = (value == HIGH);
  Serial.print("Light Status: ");
  Serial.println(lightOn);
  if (!masterControl)
    return;
  if (lightOn) {
    ledcWrite(PWMChannel, 255);
  } else {
    ledcWrite(PWMChannel, 0);
  }
}
BLYNK_WRITE(V4) // Button widget on V4
{
  int value = param.asInt();  // Get value from the button
  BLYNK_LOG("Got value in V4 %d ",value);
  masterControl = (value == HIGH);
  Serial.print("Master Control Status: ");
  Serial.println(masterControl);
  if (!masterControl)
    return;
  if (lightOn) {
    ledcWrite(PWMChannel, 255);
  } else {
    ledcWrite(PWMChannel, 0);
  }
}
```