

AUTOMATIC ROOM LIGHTS CONTROL BY USING ARDUINO DESIGN

Submitted in partial fulfillment of the requirements for the award of
Master of Engineering degree in Computer Science and Engineering.

By

MOIRANGTHEM SANATHOI SINGH (Reg.No- 42310006)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

CATEGORY -1 UNIVERSITY BY UGC

**Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

APRIL - 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Category - I University by UGC

Accredited "A++" by NAAC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **MOIRANGTHEM SANATHOI SINGH (42310006)** who carried out the project entitled "**AUTOMATIC ROOM LIGHTS CONTROL BY USING ARDUINO DESIGN**" under my supervision from November 2023 to April 2024.

Internal Guide

Dr.L.LAKSHMANAN, M.E., Ph.D.

Head of the Department
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Jeppiyaar Nagar, Rajiv Gandhi Salai,
Chennai - 600 119.

Submitted for Viva voce Examination held on 20-04-2024

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

DECLARATION

I am pleased to acknowledge the favour of the Head of Institution, Tamil Nadu State Institute of Science and Technology, for their kind encouragement and for the privilege of carrying out this project.

I, MOIRANGTHEM SANATHOI SINGH (42310006), hereby declare that the Project Report entitled "AUTOMATIC ROOM LIGHTS CONTROL BY USING ARDUINO DESIGN" done by me under the guidance Dr.L.LAKSHMANAN, M.E., Ph.D. is submitted in partial fulfillment of the requirements for the award of Master of Engineering degree in Computer Science and Engineering.

DATE: 20/04/24

PLACE: Chennai

M. Sanathoi Singh

SIGNATURE OF THE CANDIDATE/S

M. Sanathoi Singh

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to The Board of Management of **Sathyabama Institute of Science and Technology** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr.T. Sasikala M.E., Ph.D**, Dean School of Computing, and **Dr.L.Lakshmanan M.E., PhD**, Head of the Department, Dept. of Computer Science and Engineering for providing me necessary support during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.L.Lakshmanan M.E., Ph.D.** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

By recognising the presence of a human, an automatic light control system that uses an Arduino and a PIR (passive infrared) sensor may switch on and off the lighting in a home. This system may be utilised in places like toilets, garages, classrooms, stairwells, and other places where people aren't always present but still require light. Additionally, there is no need to worry about power costs since, in the absence of a person, the lights go out and one must pay the bills according to usage. This research proposes an automatic light control system that uses a PIR sensor and an Arduino to control the lights in the room automatically. This system's primary parts are an Arduino Uno, a PIR sensor, and a relay module. The PIR sensor, which aids in the detection of human presence, is the component that powers the system the most out of all of them. The suggested technique can save at least 50% of electricity, according to experimental data.

Keywords: Automatic room light, LDR sensor, PIR sensor, and Arduino

TABLE OF CONTENTS

CHAPTER	DESCRIPTION	PAGE NO
	ABSTRACT	v
	LIST OF FIGURES	viii
	LIST OF ABBREVIATIONS	ix
1	INTRODUCTION	1
2	LITERATURE SURVEY	2
	2.1 System Study	4
	2.1 Existing System	4
	2.2 Proposed System	5
	2.3 System Specification	6
3	HARDWARE SPECIFICATION	7
4	SOFTWARE DEVELOPMENT	10
	4.1 METHODOLOGY	10
	4.2 MODULES DESCRIPTION	11
5	SYSTEM DESIGN AND ANALYSIS	14
	5.1 Architecture Diagram	14
	5.2 Data Flow Diagram	15

CHAPTER	DESCRIPTION	PAGE NO
6	SYSTEM TESTING AND IMPLEMENTATION	16
	6.1 System Testing	16
	6.1.1 Unit Testing	16
	6.1.2 Integration Testing	17
	6.1.3 Functional Testing	17
	6.1.4 Performance Testing	17
	6.1.5 White Box Testing	18
	6.1.6 Black Box Testing	18
	6.1.7 Acceptance Testing	18
	6.2 Input Design and Output Design	19
7	CONCLUSION & FUTURE WORK	21
	REFERENCES	22
	APPENDIX	24
	A. SOURCE CODE	24
	B. SCREENSHOTS	50
	C. RESEARCH PAPER	57

LISTS OF FIGURES

FIGURE NO	FIGURE NAME	Page No.
3.1	Arduino	7
3.2	Motion sensor(Passive Infrared)	8
3.3	Light sensor	8
3.4	Relays	8
3.5	IoT module(Wi-Fi)	9
3.6	Power supply	9
4.1	Block Diagram	10
4.2	PIR sensor	11
4.3	Arduino Uno	12
4.4	Relay as controlling unit	12
5.1	Architecture Diagram	14
5.2	Data Flow Diagram	15

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
GND	Ground
LDR	Photo Light Dependent Resistor
PIC	Peripheral Interface Controller
PIR	Passive Infrared Sensor
PWM	Pulse Width Modulation

CHAPTER 1

INTRODUCTION

Thanks to scientific advancements, life is now worthwhile. They have given us luxury and comfort. Technology now plays a significant and indispensable role in our daily lives. Technology has advanced tremendously in the previous several years. Thus, the dependence of human life on these technology increases. Among all of its inventions, the creation of electricity is one of the greatest. The use of electricity has become essential to human existence. The earth would stay lifeless and dead without electricity, which is regarded as the soul or life. Home automation is a technology that aims to make human comfort levels higher. Home automation is the process of using technology to automate every amenity and appliance in the house. Energy conservation is necessary since non-renewable resources are becoming scarcer by the day. The most crucial component of all modern society's operations is electricity. Therefore, using non-renewable energy sources wisely is essential. Through the observation and control of many devices, home automation creates energy-efficient spaces. These days, energy-related problems are the principal ones. People in our day and age are often rushing, so they often forget to switch off the lights and fans. This home automation feature, the Automatic Light Control System, saves power in such a scenario. Electrical appliances can automatically turn on or off based on whether people are in a certain region of the room. This system's primary benefit is this.

CHAPTER 2

LITERATURE SURVEY

The basis of any mentoring course is Performance Analysis Systems. With the guidance and assistance of statistical data, an individual will surely progress, which will also help the business that hires him succeed. A background inquiry is done in order to look at comparable existing techniques that are utilised to analyse student performance. Before starting our literature review, we learn about three existing systems that are comparable to the one that is being proposed.

Vibhuti et al. [1] suggested a system that uses an Arduino Uno, WAGO PLC (Programmable Logic Controller), and relay control. PIR sensors allow for the spontaneous functioning of equipment, such as fans, AC units, tube lights, and so forth, based on ambient circumstances. In a real-time implementation, an Android application handles manual control while sensor data handles automated control. However, this paper's challenge lies in the way that both the Arduino Uno and the WAGO PLC are used to control and monitor the devices. All that is needed to complete these tasks is an Arduino Uno.

Maslekar et al.'s [2] smart lighting system makes use of the Raspberry Pi. Fans and lights are being monitored by Raspberry Pi at the same time. Fans and lighting in the room will switch off automatically when no one is there. By adopting this smart lighting system, energy is conserved. This system's experimental data indicate that 50% of the energy is preserved. However, the cost of the Raspberry Pi is more than that of the Arduino Uno. Classroom Automatic Lighting and Control System: Bluetooth, PIR sensor, relay, and electrical light control. When the Bluetooth module is attached to the Arduino Uno, it may transmit voice commands to the Arduino Uno using a mobile Android application, allowing it to turn on or off the light. According to the experimental data, 50% of the energy is preserved.

However, the Bluetooth module can also be removed in order to implement this research [3].

The discussions in [4] discuss an automated room light system that makes use of visitor counter functionality.

The lights in the room turn on and off based on whether people are there.

Manual labour is not required for switching. The PIR sensor detects human presence when someone approaches the room's entry. When someone enters a room using the visitor counter, the counter is incremented, which triggers the microcontroller programme that turns on the room light. When someone leaves the room, the counter moves, which causes the lights to go off. The only thing that turned off in the room was the lights after everyone had departed. The problem with this approach is that only one person should be able to enter a room at a time.

A system that is controlled by an Arduino microcontroller, network connections, and the Modbus industrial protocol was proposed by Vahid et al. [5]. The network connectivity is developed using an Arduino Ethernet shield and a wireless router device. A USB (Universal Serial Bus) cable is used to load Arduino code into the Arduino board, and a particular Android application called "mypro" is used to load the Modbus programme onto mobile or Windows software. Through an Ethernet cable and router, the Arduino Ethernet Shield and mobile device are connected. Connecting to the router makes it simple for the user to monitor and operate the appliances.

<i>Sr. No.</i>	<i>Title</i>	<i>Technology Used</i>
1.	Implementation of Smart Class Room Using WAGO PLC	WAGO PLC and Arduino Uno [1]
2.	Smart Lighting System using Raspberry Pi	Raspberry Pi [2]
3.	Automatic Lighting And Control System For Classroom	PIR sensor, relay and Bluetooth module [3]
4.	Automatics Room Light Controller with Visitor Counter	Infrared(IR) Transmitter, IR Receiver Microcontroller, Bidirectional Visitor counter, section Display Automatic Room Light controller section LCD Display, Relay Darlington pair [4]
5.	Low-Cost Home Automation Using Arduino and Modbus Protocol	Communication Protocol, Arduino Uno [5]

Summary of Literature survey

System Study

2.1 Existing System

These days, electrical energy is essential to human existence. People have been anticipating automation in their daily lives in recent years, and they are still enthusiastic about consuming less energy in order to save money. People are getting lazy and forgetting to turn off the lights when they leave a room, which wastes a lot of energy if the lights are left on while no one is there. In public and private sector businesses, offices, schools, and universities, most individuals are typically not interested in turning off electronic devices like fans and lights.

Disadvantage

- These days, energy-related problems are the major issue.
- Because everyone is rushing these days, individuals frequently forget to switch off the lights and fans.
- Wasted electricity.

2.2 Proposed System

You won't have to bother about flipping the switches on and off each time you want to turn on the lights thanks to this extremely helpful Arduino Automatic Room Lights System project. Relay module, PIR sensor, and Arduino are the three essential parts of the Automatic Room Lights project. The primary tool for identifying people and human movements among the three parts is the PIR Sensor, hence it is the one that is being examined. One significant use of the PIR Sensor is really the Automatic Room Lights project.

Advantage

- Savings on energy.
- Economy of Cost.
- Ecologically Sustainable.
- Practicality for the User.
- Extended Life of the Bulb.

2.3 System Specification

Hardware Requirements

- Microcontroller (Arduino)
- Motion sensors (Passive Infrared)
- Light Sensor (LDR)
- Relays
- IoT module (Wi-Fi)
- Power supply

Software Requirements

Coding Language	:	Embedded C Programming
Operating System	:	Windows 10
IDE	:	Arduino IDE

CHAPTER 3

HARDWARE SPECIFICATION

1. Microcontroller (Arduino)

Developed by Arduino.cc, the Arduino Uno is an open-source microcontroller board that is based on the Microchip ATmega328P microprocessor. Sets of digital and analogue input/output (I/O) pins on the board allow it to be interfaced with other development boards (shields) and other circuits. The board may be programmed using the Arduino IDE (Integrated Development Environment) and a type B USB connector. It includes six analogue I/O pins and fourteen digital I/O pins, six of which can be used for PWM output.

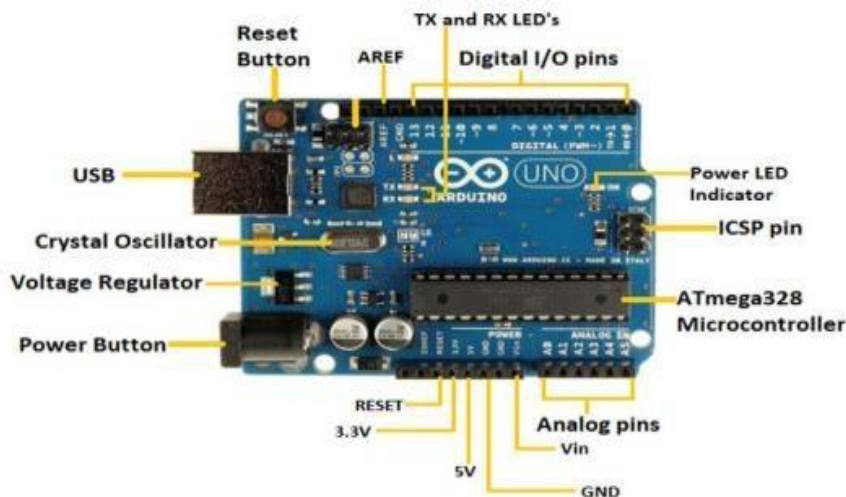


Fig 3.1 Arduino

2. Motion sensors (Passive Infrared)

PIR Sensor, also known as PIR (motion) sensor or IR sensor, is short for passive infrared sensor. It is useful for applications where it is necessary to detect human or particle movement within a specific range. It has been used in many projects and is well-liked by the open-source hardware community for Arduino and Raspberry Pi projects because of its strong function and affordable advantages. All of these might make it easier for newcomers to learn about PIR sensors.



Fig 3.2 Motion Sensor

3. Light Sensor (LDR)

The light intensity is detected using an LDR sensor module. It is connected to the analogue output pin (AO) and digital output pin (DO) on the board, respectively. Depending on the amount of light present, the resistance of the LDR will decrease.

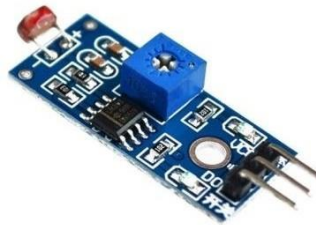


Fig 3.3 Light Sensor

4. Relays

A unique piece of hardware utilised for remote switching is the relay module. Two SPDT relays and one optically separated, broad voltage range input are included in the Relay module. For simple field wiring, they are introduced to screw-type terminal blocks. Relays are switches that may be used electrically or physically to open and close circuits.



Fig 3.4 Relays

5. IoT module (Wi-Fi)

WiFi modules, often referred to as serial to WIFI modules, are part of the Internet of Things' transmission layer. The purpose is to transform a serial port or TTL level into an embedded module with IEEE802 wireless network protocol integrated that can adhere to WiFi wireless network communication standards.



Fig 3.5 IoT module (Wi-Fi)

6. Power supply

In order for the system that has to be powered to function correctly, power supply modules are utilised to convert AC power into useful DC power. Power supplies also condition the DC output to ensure that the DC voltage(s) are noise-, spike-, and interference-free.



Fig 3.6 Power Supply

CHAPTER

SOFTWARE DEVELOPMENT

4.1 METHODOLOGY

Regarding the suggested automated room light control system designed to conserve energy. The suggested design bases the light's intensity on the number of people in the space. The three components of the suggested system design are the Arduino microcontroller circuit, which controls the entire system, the PIR sensor, which detects human presence, and the relay module, which switches the system on automatically.

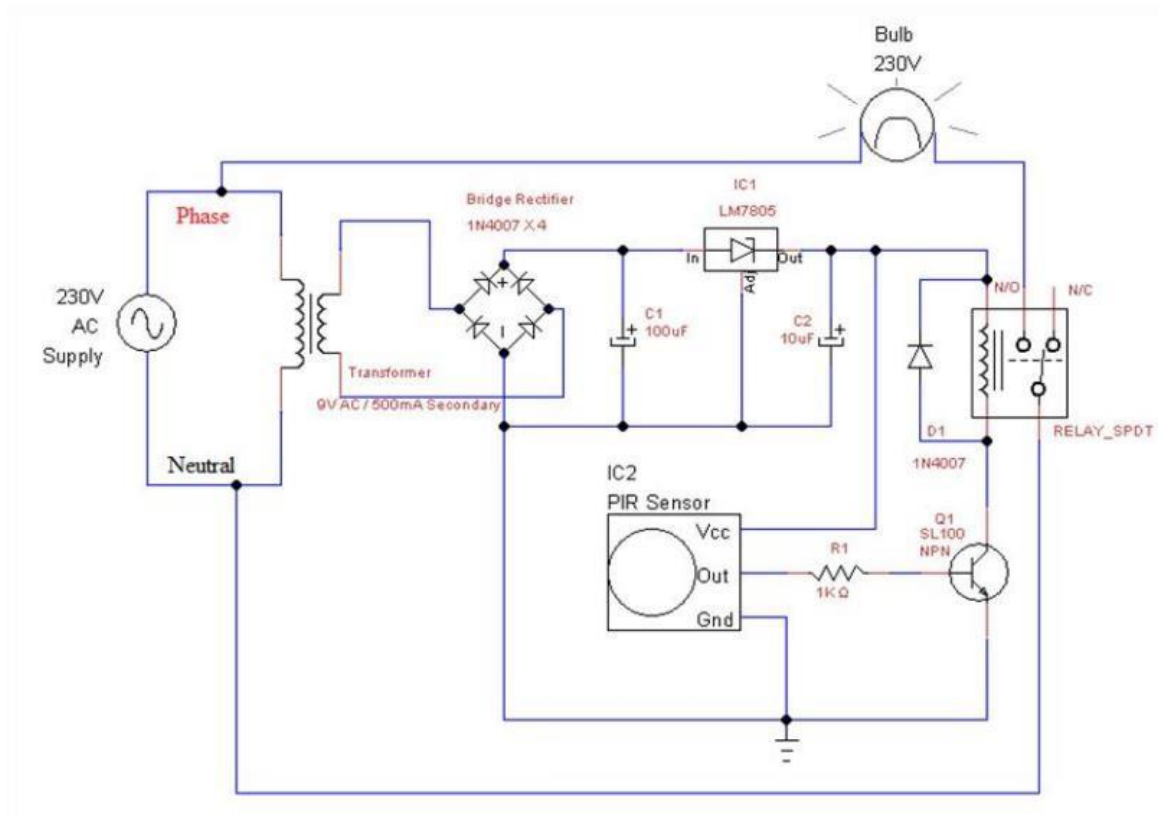


Fig. 4.1 Block Diagram

4.2 MODULES DESCRIPTION

A variety of hardware components make up the automatic light control system, which may be used to process, sense, and regulate appliances. We go over these hardware parts below:

1. PIR Sensor as Sensing Unit

The sensor device is used to gather environmental input characteristics that are needed for automation. To ensure proper functioning in a given section of the room, the following considerations need to be made:

- Constant modifications to human movement
- Conserving (Fiscal)

The PIR (Passive Infrared) sensor utilised in this instance is seen in the provided Figure.

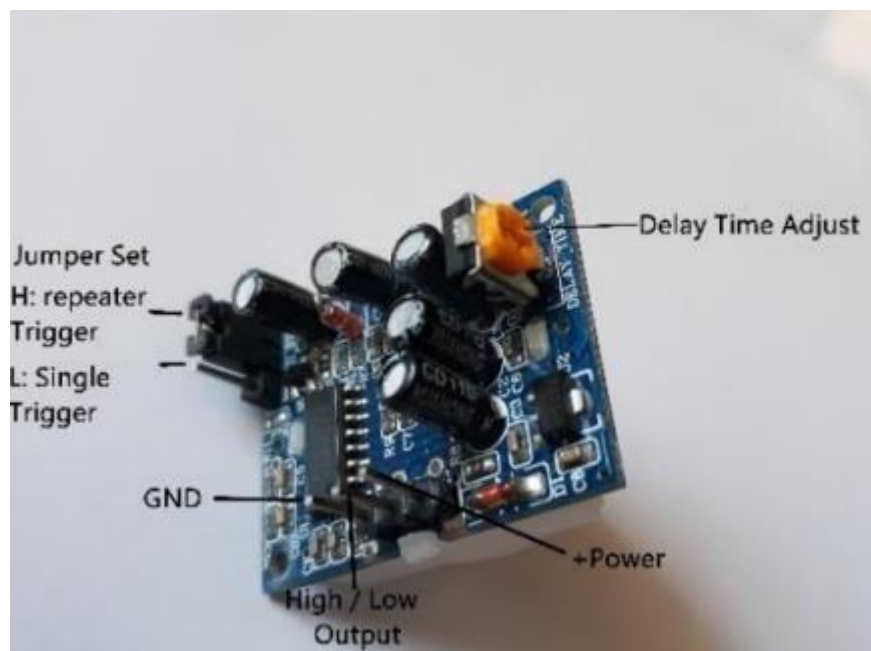


Fig. 4.2 PIR Sensor

Passive infrared, often known as pyroelectric or "IR motion" sensors, is the term used to describe this type of PIR sensor. When a human enters the PIR sensor's sensing range, the sensor begins to detect their movements. This sensor's unique features are its small size, low power consumption, affordability, ease of use, and non-exhaust. As a result, a lot of equipment for home automation employs this sensor.

2. Arduino Uno as Processing Unit

The Arduino Uno microcontroller, seen in the figure, uses the ATmega328 CPU.

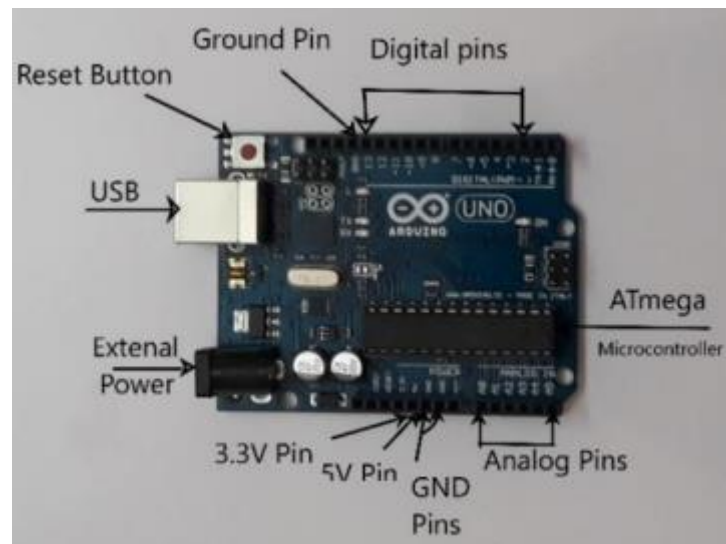


Fig. 4.3 Arduino Uno

It contains 14 digital input or output pins that may be utilised as PWM (Pulse Width Modulation) outputs in addition to 6 analogue input ports. The programming language is unique to it. This microcontroller uses a crystal oscillator frequency of 16MHz. It features ICSP (In Circuit Serial Programming), a reset button, a power barrel jack, and a USB connection that is easy to connect to a computer. The Arduino Uno operates at 5V on each of its pins. This microcontroller's programming language is simple.

3. Relay as Controlling Unit

As seen in Figure, a relay is an electrically powered device.



Fig.4.4 Relay as Controlling unit

Relays are digital switches that have substantially greater voltages and current limits. This gadget is frequently utilised for power preservation. This

device has the advantages of being compact, stable, and long-lasting. It can also be utilised with both ac and dc systems. The three terminals on a relay are the common terminal, the typically open terminal, and the normally closed terminal. The three pins are for the input signal, VCC, and GND.

CHAPTER 5

SYSTEM DESIGN AND ANALYSIS

5.1 Architecture Diagram

An architectural diagram is a visual depiction of the physical implementation of different software system components. It shows the links, limitations, and divisions between the software system's numerous components as well as the architecture of the system as a whole.

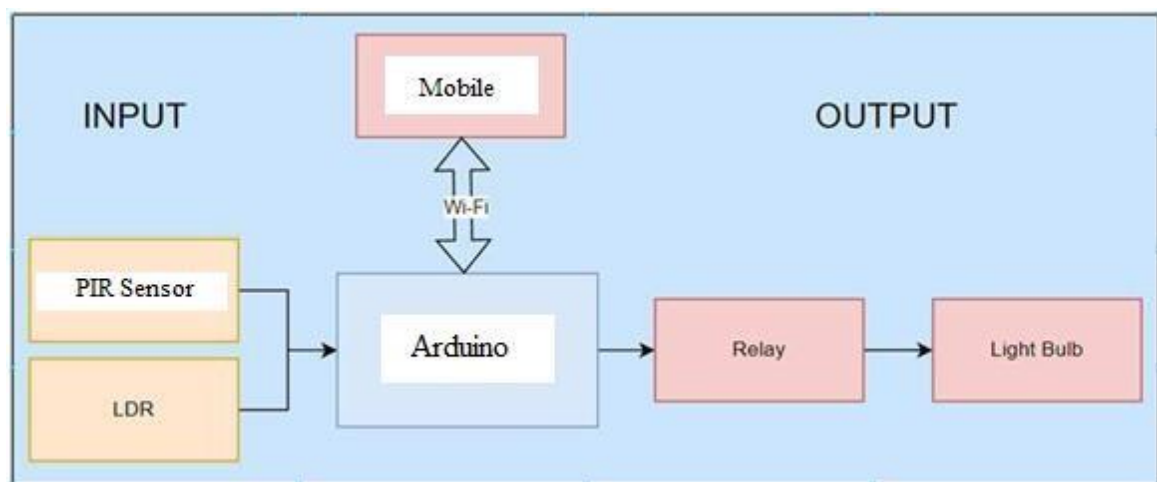


Fig.5.1 Architecture Diagram

Working of the System

The microcontroller Arduino offers an open-source platform for both hardware and software operations. This project is beneficial because it makes use of an Arduino Uno and a PIR sensor, which allows the lights in the room to detect human activity and switch on automatically. The lights will remain on as long as the person is in the room. The PIR Sensor's OUT pin is initially in the LOW mode when there isn't a human in the space. As a result, the room's lights are out, and when someone walks in, the PIR sensor's output increases. The room's infrared (IR) radiation is detected by the PIR sensor. The Data OUT pin of the PIR Sensor is connected to the Arduino Uno's Digital pin 8. Arduino uses this to activate the relay when it hits HIGH. One-o. Relays are active LOW devices, hence the pin on the relay is in the LOW state. The lights will now come on. As long as there is motion in the space, this light will remain in the ON state. If someone leaves the room or naps, the motion in front of the sensor ceases, and the infrared radiation remains unchanged. As a result, the PIR sensor's Data Out pin will be in the LOW mode. This

causes the relay to be turned off. Relay is now operating in HIGH mode. As a result, the room's lights will be off.

5.2 Data Flow Diagram

A data-flow diagram (DFD) is a graphic tool used to show the movement of data through a system or procedure. The DFD also offers details on the inputs and outputs of each entity as well as the process itself. Control flow, loops, and decision rules are absent from a data-flow diagram. Certain operations based on the data can be shown using a flowchart.

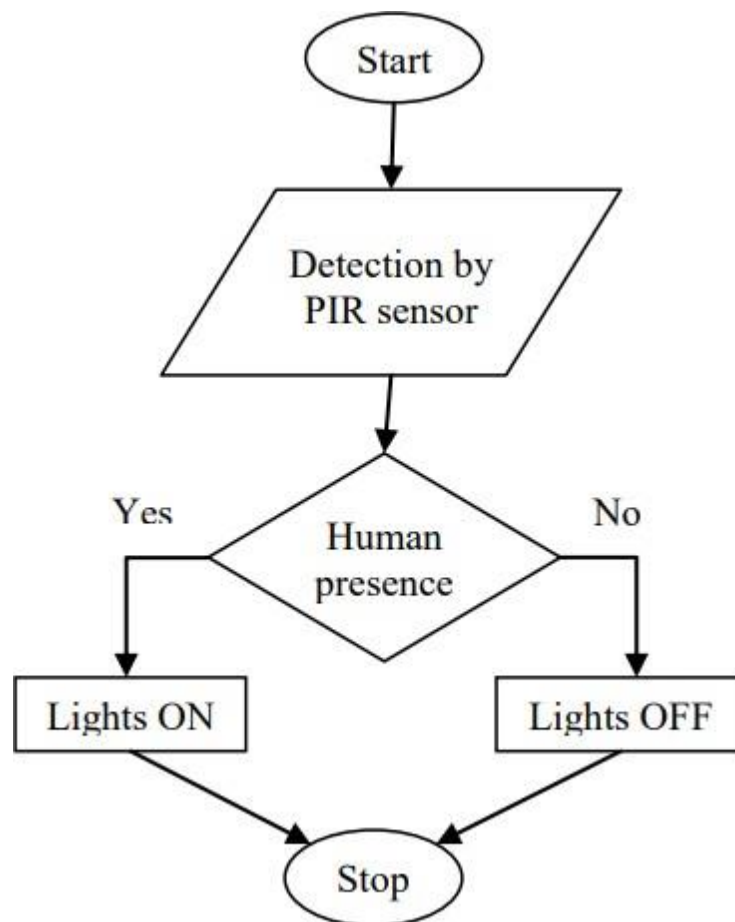


Fig.5.2 Data Flow Diagram

CHAPTER 6

SYSTEM TESTING AND IMPLEMENTATION

6.1 SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS:

Testing is the process of trying to discover every conceivable fault or weakness in a work product. The different type of testing is given below:

6.1.1 UNIT TESTING:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration.

This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.1.2 INTEGRATION TESTING:

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the

components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.1.3 FUNCTIONAL TEST:

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/ Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

6.1.4 PERFORMANCE TESTING:

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

6.1.5 WHITE BOX TESTING:

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

6.1.6 BLACK BOX TESTING:

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

6.1.7 ACCEPTANCE TESTING:

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered

6.2 INPUT DESIGN AND OUTPUT DESIGN

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:’

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
2. Select methods for presenting information.
3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- ❖ Convey information about past activities, current status or projections of the
- ❖ Future.
- ❖ Signal important events, opportunities, problems, or warnings.
- ❖ Trigger an action. Confirm an action.

CHAPTER 7

CONCLUSION & FUTURE WORK

The concept of automated houses has been presented in this work, along with a solution that reduces system power usage. The sensors and household appliances are connected to this automated light management system, which allows for device monitoring and control. An automated house is a large system with several applications and technologies that may be utilised to simply regulate and secure a home. We may infer from the suggested system that a method is used to regulate the room lighting utilising a variety of devices. As a great deal of energy is squandered in daily living these days. This technology can keep energy waste under control and help save a significant amount of electricity. The system's overall effective cost is extremely low.

FUTURE WORK

To improve the accuracy of autonomous lighting systems, a variety of technologies can be applied.

- The GSM (Global System for Mobile) module may be used to receive notifications, adding a professional touch to the system.
- The home may be secured and controlled using a few different types of sensors.
- One such device is a pressure sensor that is installed outside the door to sense occupancy.
- Digital cameras may also be used to identify people's presence through image processing.

REFERENCES

- [1] A. H. Jabire, H. Suleiman, A. A. Noku, P. U. Adamu, D. A. Tsoken, and L. O. Agidike, "Bidirectional Visitors' Counter with Automatic Room Light Controller," *Taraba J. Eng. Technol.*, vol. 2, no. 1, pp. 75–80, 2021.
- [2] A. Maslekar, K. Aparna, K. Mamatha, and T. Shivakumara, "Smart Lighting System using Raspberry Pi," *International Journal of Innovative Research in Science and Technology*, Vol. 4(7), 2015, pp. 5113-5121.
- [3] Castillo-Martinez, J.-A. Medina-Merodio, J.-M. Gutierrez-Martinez, J. Aguado-Delgado, C. de Pablos-Heredero, and S. Oton, "Evaluation and improvement of lighting efficiency in workspaces," *Sustainability*, vol. 10, no. 4, pp. 11-10, 2018.
- [4] Daeho Kim, Junghoon Lee, Yeongmin Jang and Jaesang Cha. "Smart LED lighting system implementation using Human tracking US/IR sensor" 2011 IEEE (ICTC 2011).
- [5] G. Saranya, E. Duraiarasu and S. Manoj, "Universal Smart Lighting System," 2023 IEEE Sustainable Smart Lighting World Conference & Expo (LS18), Mumbai, India, 2023, pp. 1-5, doi: 10.1109/LS1858153.2023.10170266
- [6] Karthikeyan.R.A, [2]piyush.D "Automatic Lighting Using Arduino and PIR Sensor", *International Journal of Computer Science Trends and Technology*, Volume 6, SeptOct2018.
- [7] Abhishek N Vaghela, [2] Bhavin D Gujjar "Automatic Switch using PIR Sensor" *International Journal of Engineering Development and Research*.
- [8] A website for microcontrollers and Arduino Uno components.
- [9] "Mechanicals Room Light Controller with Visitor Counter," accessible at <http://www.projectsof8051.com/automaticroom-light-controller-with-visitor-counter>.
- [10] <https://circuitdigest.com/microcontroller-projects/automatic-room-light-controller-with-bidirectional-visitor-counter-using-arduino>.
- [11] <https://www.projectsof8051.com/automatic-room-light-controller-with-visitor-counter/>
- [12] Kunturwar, Vrushabh and Shinde, Atul and Pokharkar, Ashwini and Shroff, Arpita, Automatic Room lights Control by Using Arduino and IR (April 17, 2020). 2nd International Conference on Communication & Information Processing (ICCIP) 2020.
- [13] Mahesh Kumar Singh" Arduino based: Automatic Room Light Control" department of ECE, Buddha Institute of Technology, Gorakhpur (Uttar Pradesh), India.

- [14] P. Rajakumar, B. C. Kanna, R. Archana, A. Akshaya, K. Abhinaya and M. Boobalan, "Automation of Room Lighting System Using Arduino and PIR Sensor," 2022 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS), Chennai, India, 2022, pp. 1-3, doi: 10.1109/ICPECTS56089.2022.10047213.
- [15] Richu Sam Alex, R Narciss Starbell "Energy efficient Intelligent Street Lighting System using Sensors", International Journal of Engineering and Advanced Technology (IJEAT), Vol-3, Issue 4, April 2014.
- [16] Suresh S., H.N.S. Anusha, T.Rajath, P.Soundarya, and S.V. Prathyusha Vudatha."Automatic Lighting and Control System For Classroom," 2016 International Conference on ICT in Business Industry & Government (ICTBIG).
- [17] S. Fahad, "Bidirectional Visitor counter using Arduino and UltrasonicSensors, Counting Visitors," 2021.<https://www.electronicclinic.com/bidirectional-visitor-counter-usingarduino-and-ultrasonic-sensors-counting-visitors/> (accessed Jan. 20, 2023).
- [18] Vahid Hassanpour, Sedighe Rajabi, Zeinab Shayan, Zahra Hafezi, and Mohammad Mehdi Arefi. The paper "Low-Cost Home Automation Using Arduino and Modbus Protocol" was presented at the 5th International Conference on Control, Instrumentation and Automation (ICCIA) in Shiraz, 2017.
- [19] Vibhuti and Shimi S.L. "Implementation of Smart Class Room Using WAGO PLC" was published in the Proceedings of the Second International Conference on Inventive Systems and Control (ICISC) 2021, held in Coimbatore.

APPENDIX

A. Source Code

```
#include<LiquidCrystal.h>
LiquidCrystal lcd(13,12,11,10,9,8);

#define in 14
#define out 19
#define relay 2

int count=0;

void IN()
{
    count++;
    lcd.clear();
    lcd.print("Person In Room:");
    lcd.setCursor(0,1);
    lcd.print(count);
    delay(1000);
}

void OUT()
{
    count--;
    lcd.clear();
    lcd.print("Person In Room:");
    lcd.setCursor(0,1);
    lcd.print(count);
    delay(1000);
}

void setup()
{
```

```

lcd.begin(16,2);
lcd.print("Visitor Counter");
delay(2000);
pinMode(in, INPUT);
pinMode(out, INPUT);
pinMode(relay, OUTPUT);
lcd.clear();
lcd.print("Person In Room:");
lcd.setCursor(0,1);
lcd.print(count);
}

void loop()
{

    if(digitalRead(in))
    IN();
    if(digitalRead(out))
    OUT();

    if(count<=0)
    {
        lcd.clear();
        digitalWrite(relay, LOW);
        lcd.clear();
        lcd.print("Nobody In Room");
        lcd.setCursor(0,1);
        lcd.print("Light Is Off");
        delay(200);
    }

    else
        digitalWrite(relay, HIGH);

```

```

}

#include <ESP8266WiFi.h>


// Replace with your network credentials

const char* ssid    = "REPLACE_WITH_YOUR_SSID";

const char* password = "REPLACE_WITH_YOUR_PASSWORD";


// Set web server port number to 80

WiFiServer server(80);


// Variable to store the HTTP request

String header;


// Auxiliar variables to store the current output state

String output5State = "off";

String output4State = "off";


// Assign output variables to GPIO pins

const int output5 = 5;

const int output4 = 4;


// Current time

```

```

unsigned long currentTime = millis();

// Previous time

unsigned long previousTime = 0;

// Define timeout time in milliseconds (example: 2000ms = 2s)

const long timeoutTime = 2000;


void setup() {

  Serial.begin(115200);

  // Initialize the output variables as outputs

  pinMode(output5, OUTPUT);

  pinMode(output4, OUTPUT);

  // Set outputs to LOW

  digitalWrite(output5, LOW);

  digitalWrite(output4, LOW);


  // Connect to Wi-Fi network with SSID and password

  Serial.print("Connecting to ");

  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {

    delay(500);

    Serial.print(".");

  }

```

```

// Print local IP address and start web server

Serial.println("");

Serial.println("WiFi connected.");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

server.begin();

}

void loop(){

WiFiClient client = server.available(); // Listen for incoming clients

if (client) { // If a new client connects,

Serial.println("New Client."); // print a message out in the serial port

String currentLine = ""; // make a String to hold incoming data from the client

currentTime = millis();

previousTime = currentTime;

while (client.connected() && currentTime - previousTime <= timeoutTime) { // loop
while the client's connected

currentTime = millis();

if (client.available()) { // if there's bytes to read from the client,

char c = client.read(); // read a byte, then

Serial.write(c); // print it out the serial monitor

header += c;

```

```

if (c == '\n') {           // if the byte is a newline character

// if the current line is blank, you got two newline characters in a row.

// that's the end of the client HTTP request, so send a response:

if (currentLine.length() == 0) {

// HTTP headers always start with a response code (e.g. HTTP/1.1 200 OK)

// and a content-type so the client knows what's coming, then a blank line:

client.println("HTTP/1.1 200 OK");

client.println("Content-type:text/html");

client.println("Connection: close");

client.println();


// turns the GPIOs on and off

if (header.indexOf("GET /5/on") >= 0) {

Serial.println("GPIO 5 on");

output5State = "on";

digitalWrite(output5, HIGH);

} else if (header.indexOf("GET /5/off") >= 0) {

Serial.println("GPIO 5 off");

output5State = "off";

digitalWrite(output5, LOW);

} else if (header.indexOf("GET /4/on") >= 0) {

Serial.println("GPIO 4 on");

output4State = "on";

```

```

digitalWrite(output4, HIGH);

} else if (header.indexOf("GET /4/off") >= 0) {

Serial.println("GPIO 4 off");

output4State = "off";

digitalWrite(output4, LOW);

}


// Display the HTML web page

client.println("<!DOCTYPE html><html>");

client.println("<head><meta name=\"viewport\" content=\"width=device-width, initial-
scale=1\">");

client.println("<link rel=\"icon\" href=\"data:;\">");

// CSS to style the on/off buttons

// Feel free to change the background-color and font-size attributes to fit your preferences

client.println("<style>html { font-family: Helvetica; display: inline-block; margin: 0px
auto; text-align: center;}");

client.println(".button { background-color: #195B6A; border: none; color: white;
padding: 16px 40px;");

client.println("text-decoration: none; font-size: 30px; margin: 2px; cursor: pointer;}");

client.println(".button2 {background-color: #77878A;}</style></head>");


// Web Page Heading

client.println("<body><h1>ESP8266 Web Server</h1>");

```

```

// Display current state, and ON/OFF buttons for GPIO 5

client.println("<p>GPIO 5 - State " + output5State + "</p>");

// If the output5State is off, it displays the ON button

if (output5State=="off") {

client.println("<p><a href='\"/5/on\"'><button class='\"button\"'>ON</button></a></p>");

} else {

client.println("<p><a href='\"/5/off\"'><button class='\"button  
button2\"'>OFF</button></a></p>");

}

// Display current state, and ON/OFF buttons for GPIO 4

client.println("<p>GPIO 4 - State " + output4State + "</p>");

// If the output4State is off, it displays the ON button

if (output4State=="off") {

client.println("<p><a href='\"/4/on\"'><button class='\"button\"'>ON</button></a></p>");

} else {

client.println("<p><a href='\"/4/off\"'><button class='\"button  
button2\"'>OFF</button></a></p>");

}

client.println("</body></html>");

// The HTTP response ends with another blank line

client.println();

```

```

// Break out of the while loop

break;

} else { // if you got a newline, then clear currentLine

currentLine = "";

}

} else if (c != '\r') { // if you got anything else but a carriage return character,

currentLine += c;    // add it to the end of the currentLine

}

}

}

// Clear the header variable

header = "";

// Close the connection

client.stop();

Serial.println("Client disconnected.");

Serial.println("");

}

}

#include<LiquidCrystal.h>
LiquidCrystal lcd(13,12,11,10,9,8);

#define in 14
#define out 19
#define relay 2

int count=0;

```

```

void IN()
{
    count++;
    lcd.clear();
    lcd.print("Person In Room:");
    lcd.setCursor(0,1);
    lcd.print(count);
    delay(1000);
}

void OUT()
{
    count--;
    lcd.clear();
    lcd.print("Person In Room:");
    lcd.setCursor(0,1);
    lcd.print(count);
    delay(1000);
}

void setup()
{
    lcd.begin(16,2);
    lcd.print("Visitor Counter");
    delay(2000);
    pinMode(in, INPUT);
    pinMode(out, INPUT);
    pinMode(relay, OUTPUT);
    lcd.clear();
    lcd.print("Person In Room:");
    lcd.setCursor(0,1);
    lcd.print(count);
}

```

```

void loop()
{

    if(digitalRead(in))
    IN();
    if(digitalRead(out))
    OUT();

    if(count<=0)
    {
        lcd.clear();
        digitalWrite(relay, LOW);
        lcd.clear();
        lcd.print("Nobody In Room");
        lcd.setCursor(0,1);
        lcd.print("Light Is Off");
        delay(200);
    }

    else
        digitalWrite(relay, HIGH);

}

#include <ESP8266WiFi.h>

// Replace with your network credentials

const char* ssid    = "REPLACE_WITH_YOUR_SSID";

const char* password = "REPLACE_WITH_YOUR_PASSWORD";

// Set web server port number to 80

```

```

WiFiServer server(80);

// Variable to store the HTTP request
String header;

// Auxiliar variables to store the current output state
String output5State = "off";
String output4State = "off";

// Assign output variables to GPIO pins
const int output5 = 5;
const int output4 = 4;

// Current time
unsigned long currentTime = millis();

// Previous time
unsigned long previousTime = 0;

// Define timeout time in milliseconds (example: 2000ms = 2s)
const long timeoutTime = 2000;

void setup() {

  Serial.begin(115200);

  // Initialize the output variables as outputs

```

```

pinMode(output5, OUTPUT);

pinMode(output4, OUTPUT);

// Set outputs to LOW

digitalWrite(output5, LOW);

digitalWrite(output4, LOW);


// Connect to Wi-Fi network with SSID and password

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {

    delay(500);

    Serial.print(".");

}

// Print local IP address and start web server

Serial.println("");

Serial.println("WiFi connected.");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

server.begin();

}

void loop(){

```

```

WiFiClient client = server.available(); // Listen for incoming clients

if (client) {                                // If a new client connects,

  Serial.println("New Client.");              // print a message out in the serial port

  String currentLine = "";                    // make a String to hold incoming data from the client

  currentTime = millis();

  previousTime = currentTime;

  while (client.connected() && currentTime - previousTime <= timeoutTime) { // loop
    while the client's connected

    currentTime = millis();

    if (client.available()) {                  // if there's bytes to read from the client,

      char c = client.read();                  // read a byte, then

      Serial.write(c);                          // print it out the serial monitor

      header += c;

      if (c == '\n') {                          // if the byte is a newline character

        // if the current line is blank, you got two newline characters in a row.

        // that's the end of the client HTTP request, so send a response:

        if (currentLine.length() == 0) {

          // HTTP headers always start with a response code (e.g. HTTP/1.1 200 OK)

          // and a content-type so the client knows what's coming, then a blank line:

          client.println("HTTP/1.1 200 OK");

          client.println("Content-type:text/html");

          client.println("Connection: close");

```

```

client.println();

// turns the GPIOs on and off

if (header.indexOf("GET /5/on") >= 0) {

Serial.println("GPIO 5 on");

output5State = "on";

digitalWrite(output5, HIGH);

} else if (header.indexOf("GET /5/off") >= 0) {

Serial.println("GPIO 5 off");

output5State = "off";

digitalWrite(output5, LOW);

} else if (header.indexOf("GET /4/on") >= 0) {

Serial.println("GPIO 4 on");

output4State = "on";

digitalWrite(output4, HIGH);

} else if (header.indexOf("GET /4/off") >= 0) {

Serial.println("GPIO 4 off");

output4State = "off";

digitalWrite(output4, LOW);

}

// Display the HTML web page

client.println("<!DOCTYPE html><html>");

```

```

client.println("<head><meta name=\"viewport\" content=\"width=device-width, initial-
scale=1\">");

client.println("<link rel=\"icon\" href=\"data;\">");

// CSS to style the on/off buttons

// Feel free to change the background-color and font-size attributes to fit your preferences

client.println("<style>html { font-family: Helvetica; display: inline-block; margin: 0px
auto; text-align: center;});");

client.println(".button { background-color: #195B6A; border: none; color: white;
padding: 16px 40px;");

client.println("text-decoration: none; font-size: 30px; margin: 2px; cursor: pointer;});");

client.println(".button2 {background-color: #77878A;}</style></head>");


// Web Page Heading

client.println("<body><h1>ESP8266 Web Server</h1>");


// Display current state, and ON/OFF buttons for GPIO 5

client.println("<p>GPIO 5 - State " + output5State + "</p>");

// If the output5State is off, it displays the ON button

if (output5State=="off") {

client.println("<p><a href=\"/5/on\"><button class=\"button\">ON</button></a></p>");

} else {

client.println("<p><a href=\"/5/off\"><button class=\"button
button2\">OFF</button></a></p>");

}

```

```

// Display current state, and ON/OFF buttons for GPIO 4

client.println("<p>GPIO 4 - State " + output4State + "</p>");

// If the output4State is off, it displays the ON button

if (output4State=="off") {

client.println("<p><a href=\"/4/on\"><button class=\"button\">ON</button></a></p>");

} else {

client.println("<p><a href=\"/4/off\"><button class=\"button
button2\">OFF</button></a></p>");

}

client.println("</body></html>");


// The HTTP response ends with another blank line

client.println();

// Break out of the while loop

break;

} else { // if you got a newline, then clear currentLine

currentLine = "";

}

} else if (c != '\r') { // if you got anything else but a carriage return character,

currentLine += c;    // add it to the end of the currentLine

}

}

```

```

}

// Clear the header variable

header = "";

// Close the connection

client.stop();

Serial.println("Client disconnected.");

Serial.println("");

}

}

```

```

#include<LiquidCrystal.h>
LiquidCrystal lcd(13,12,11,10,9,8);

#define in 14
#define out 19
#define relay 2

int count=0;

void IN()
{
    count++;
    lcd.clear();
    lcd.print("Person In Room:");
    lcd.setCursor(0,1);
    lcd.print(count);
    delay(1000);
}

void OUT()
{

```

```

count--;
  lcd.clear();
  lcd.print("Person In Room:");
  lcd.setCursor(0,1);
  lcd.print(count);
  delay(1000);
}

void setup()
{
  lcd.begin(16,2);
  lcd.print("Visitor Counter");
  delay(2000);
  pinMode(in, INPUT);
  pinMode(out, INPUT);
  pinMode(relay, OUTPUT);
  lcd.clear();
  lcd.print("Person In Room:");
  lcd.setCursor(0,1);
  lcd.print(count);
}

void loop()
{

  if(digitalRead(in))
  IN();
  if(digitalRead(out))
  OUT();

  if(count<=0)
  {
    lcd.clear();
    digitalWrite(relay, LOW);
    lcd.clear();
  }
}

```

```

    lcd.print("Nobody In Room");
    lcd.setCursor(0,1);
    lcd.print("Light Is Off");
    delay(200);
}

else
    digitalWrite(relay, HIGH);

}

#include <ESP8266WiFi.h>

// Replace with your network credentials

const char* ssid    = "REPLACE_WITH_YOUR_SSID";

const char* password = "REPLACE_WITH_YOUR_PASSWORD";


// Set web server port number to 80

WiFiServer server(80);


// Variable to store the HTTP request

String header;


// Auxiliar variables to store the current output state

String output5State = "off";

String output4State = "off";

```

```

// Assign output variables to GPIO pins

const int output5 = 5;

const int output4 = 4;


// Current time

unsigned long currentTime = millis();

// Previous time

unsigned long previousTime = 0;

// Define timeout time in milliseconds (example: 2000ms = 2s)

const long timeoutTime = 2000;


void setup() {

  Serial.begin(115200);

  // Initialize the output variables as outputs

  pinMode(output5, OUTPUT);

  pinMode(output4, OUTPUT);

  // Set outputs to LOW

  digitalWrite(output5, LOW);

  digitalWrite(output4, LOW);


  // Connect to Wi-Fi network with SSID and password

  Serial.print("Connecting to ");

  Serial.println(ssid);

```

```

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {

  delay(500);

  Serial.print(".");

}

// Print local IP address and start web server

Serial.println("");

Serial.println("WiFi connected.");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

server.begin();

}

void loop(){

  WiFiClient client = server.available(); // Listen for incoming clients

  if (client) {                                // If a new client connects,

    Serial.println("New Client.");              // print a message out in the serial port

    String currentLine = "";                    // make a String to hold incoming data from the client

    currentTime = millis();

    previousTime = currentTime;

    while (client.connected() && currentTime - previousTime <= timeoutTime) { // loop
      while the client's connected

```

```

currentTime = millis();

if (client.available()) {      // if there's bytes to read from the client,

char c = client.read();      // read a byte, then

Serial.write(c);              // print it out the serial monitor

header += c;

if (c == '\n') {              // if the byte is a newline character

// if the current line is blank, you got two newline characters in a row.

// that's the end of the client HTTP request, so send a response:

if (currentLine.length() == 0) {

// HTTP headers always start with a response code (e.g. HTTP/1.1 200 OK)

// and a content-type so the client knows what's coming, then a blank line:

client.println("HTTP/1.1 200 OK");

client.println("Content-type:text/html");

client.println("Connection: close");

client.println();

// turns the GPIOs on and off

if (header.indexOf("GET /5/on") >= 0) {

Serial.println("GPIO 5 on");

output5State = "on";

digitalWrite(output5, HIGH);

} else if (header.indexOf("GET /5/off") >= 0) {

Serial.println("GPIO 5 off");

```

```

output5State = "off";

digitalWrite(output5, LOW);

} else if (header.indexOf("GET /4/on") >= 0) {

Serial.println("GPIO 4 on");

output4State = "on";

digitalWrite(output4, HIGH);

} else if (header.indexOf("GET /4/off") >= 0) {

Serial.println("GPIO 4 off");

output4State = "off";

digitalWrite(output4, LOW);

}


// Display the HTML web page

client.println("<!DOCTYPE html><html>");

client.println("<head><meta name=\"viewport\" content=\"width=device-width, initial-
scale=1\">");

client.println("<link rel=\"icon\" href=\"data:;\">");

// CSS to style the on/off buttons

// Feel free to change the background-color and font-size attributes to fit your preferences

client.println("<style>html { font-family: Helvetica; display: inline-block; margin: 0px
auto; text-align: center;}");

client.println(".button { background-color: #195B6A; border: none; color: white;
padding: 16px 40px;");

client.println("text-decoration: none; font-size: 30px; margin: 2px; cursor: pointer;}");

```

```

client.println(".button2 {background-color: #77878A;}</style></head>");

// Web Page Heading

client.println("<body><h1>ESP8266 Web Server</h1>");

// Display current state, and ON/OFF buttons for GPIO 5

client.println("<p>GPIO 5 - State " + output5State + "</p>");

// If the output5State is off, it displays the ON button

if (output5State=="off") {

client.println("<p><a href=\""/5/on\"><button class=\"button\">ON</button></a></p>");

} else {

client.println("<p><a href=\""/5/off\"><button class=\"button  
button2\">OFF</button></a></p>");

}

// Display current state, and ON/OFF buttons for GPIO 4

client.println("<p>GPIO 4 - State " + output4State + "</p>");

// If the output4State is off, it displays the ON button

if (output4State=="off") {

client.println("<p><a href=\""/4/on\"><button class=\"button\">ON</button></a></p>");

} else {

client.println("<p><a href=\""/4/off\"><button class=\"button  
button2\">OFF</button></a></p>");

}

```

```

}

client.println("</body></html>");


// The HTTP response ends with another blank line

client.println();

// Break out of the while loop

break;

} else { // if you got a newline, then clear currentLine

currentLine = "";

}

} else if (c != '\r') { // if you got anything else but a carriage return character,

currentLine += c;    // add it to the end of the currentLine

}

}

}

// Clear the header variable

header = "";

// Close the connection

client.stop();

Serial.println("Client disconnected.");

Serial.println("");

}

}

```

B. Screen Shots

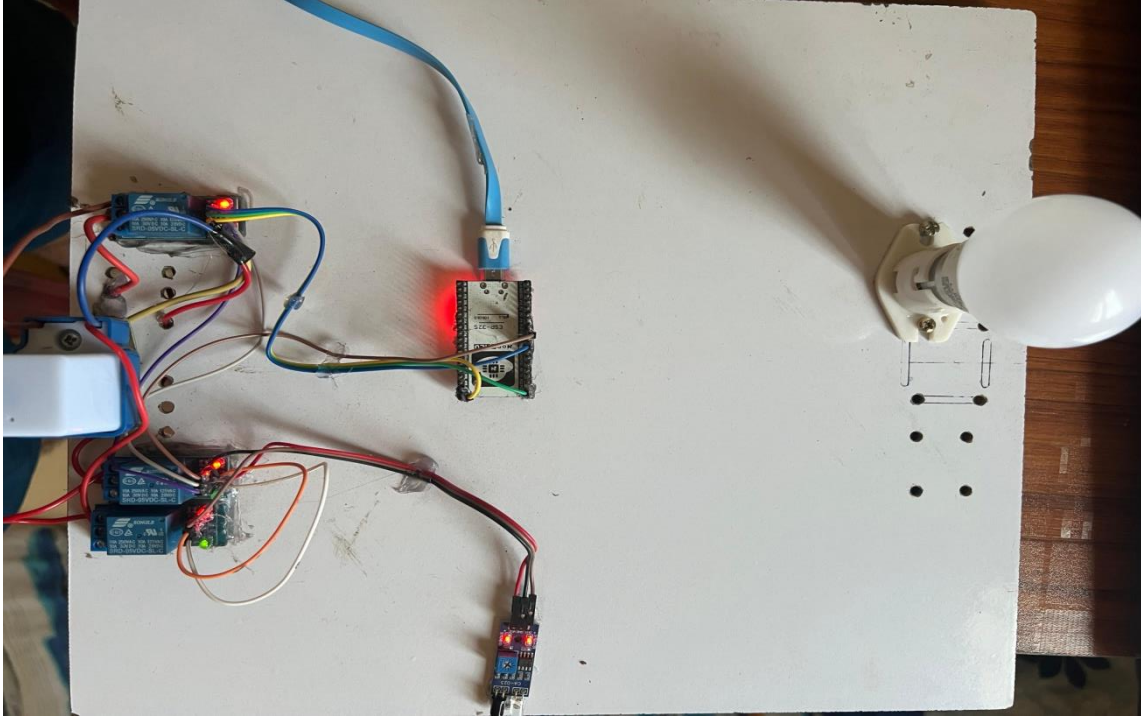


Fig.1 Hardware image

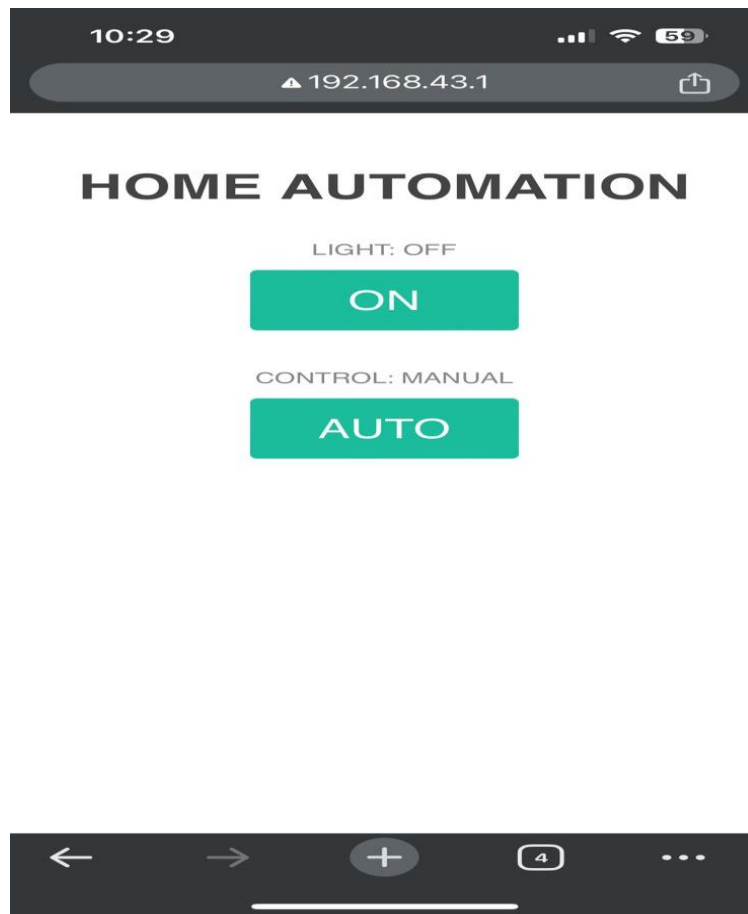


Fig.2 Manual and Automatic switches

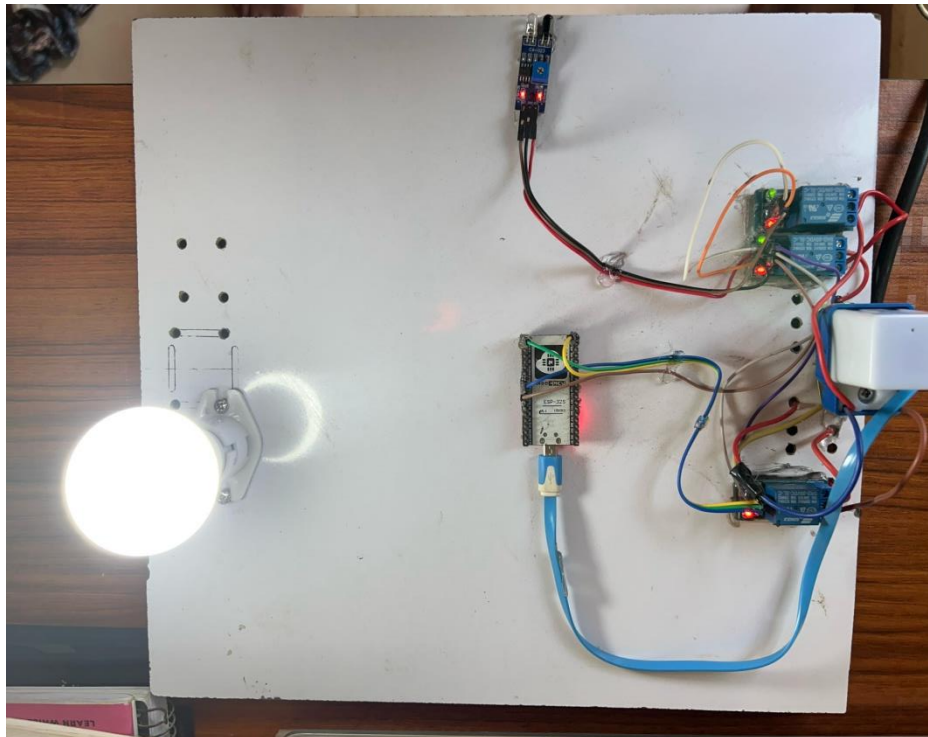
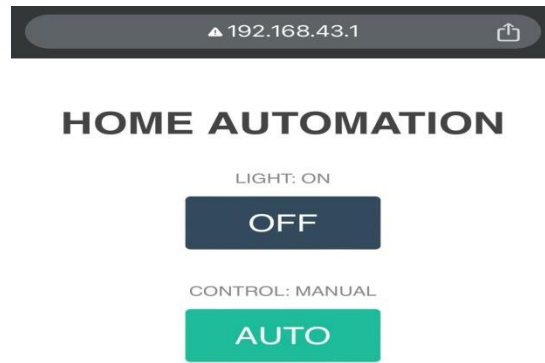


Fig.3 when the switch is ON

HOME AUTOMATION

LIGHT: OFF

ON

CONTROL: MANUAL

AUTO

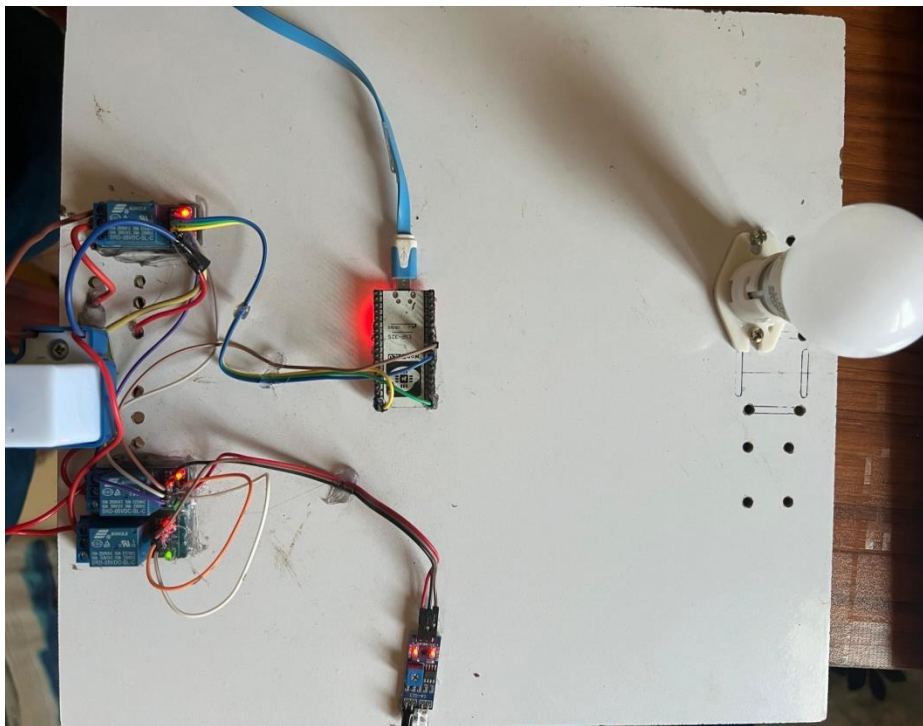
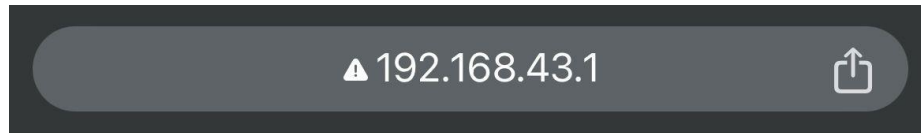


Fig.4 When the Switch is OFF



HOME AUTOMATION

LIGHT: OFF



CONTROL: AUTO



Fig.5 Turn on Auto control

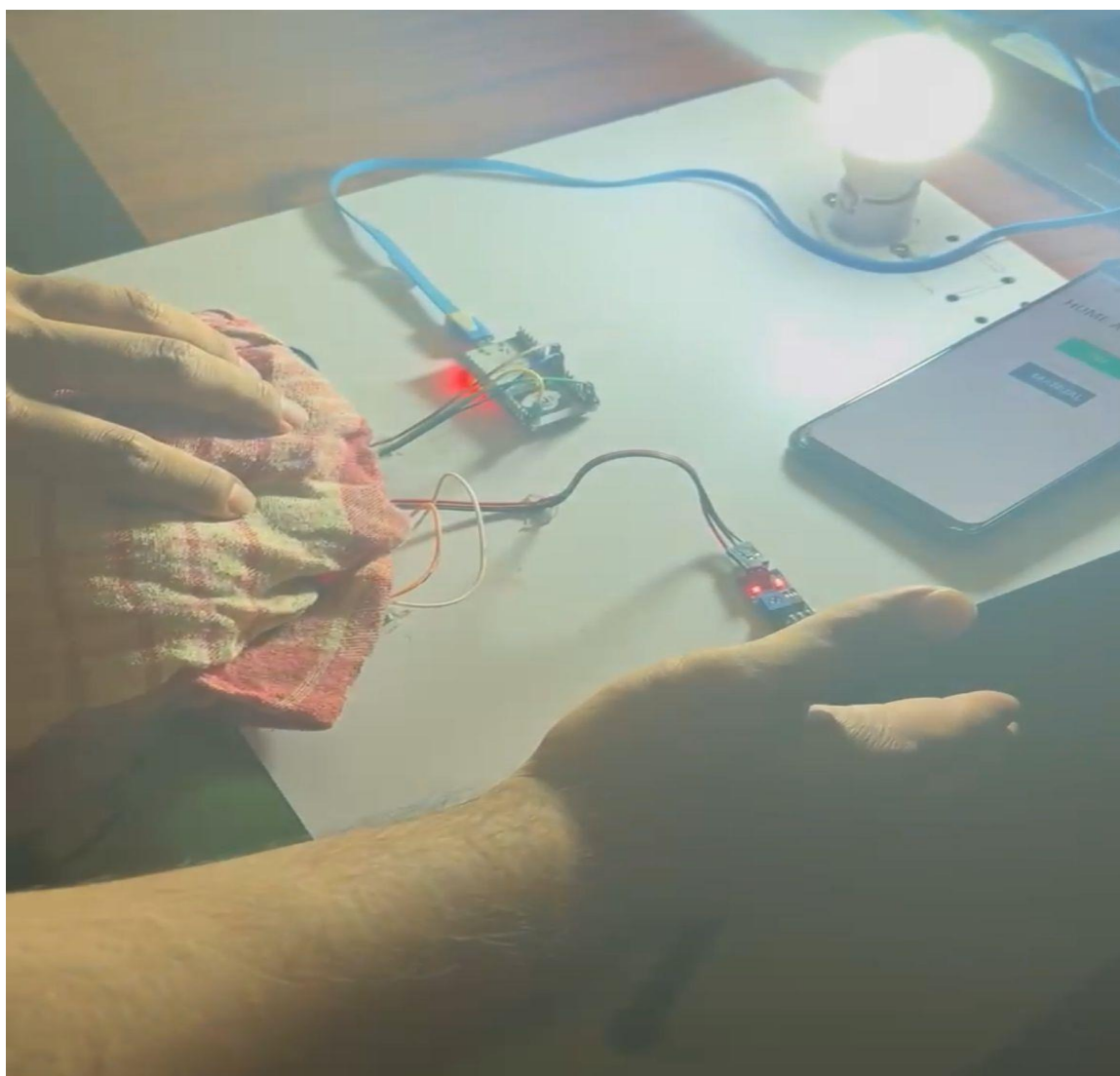


Fig.6 With the presence of movement

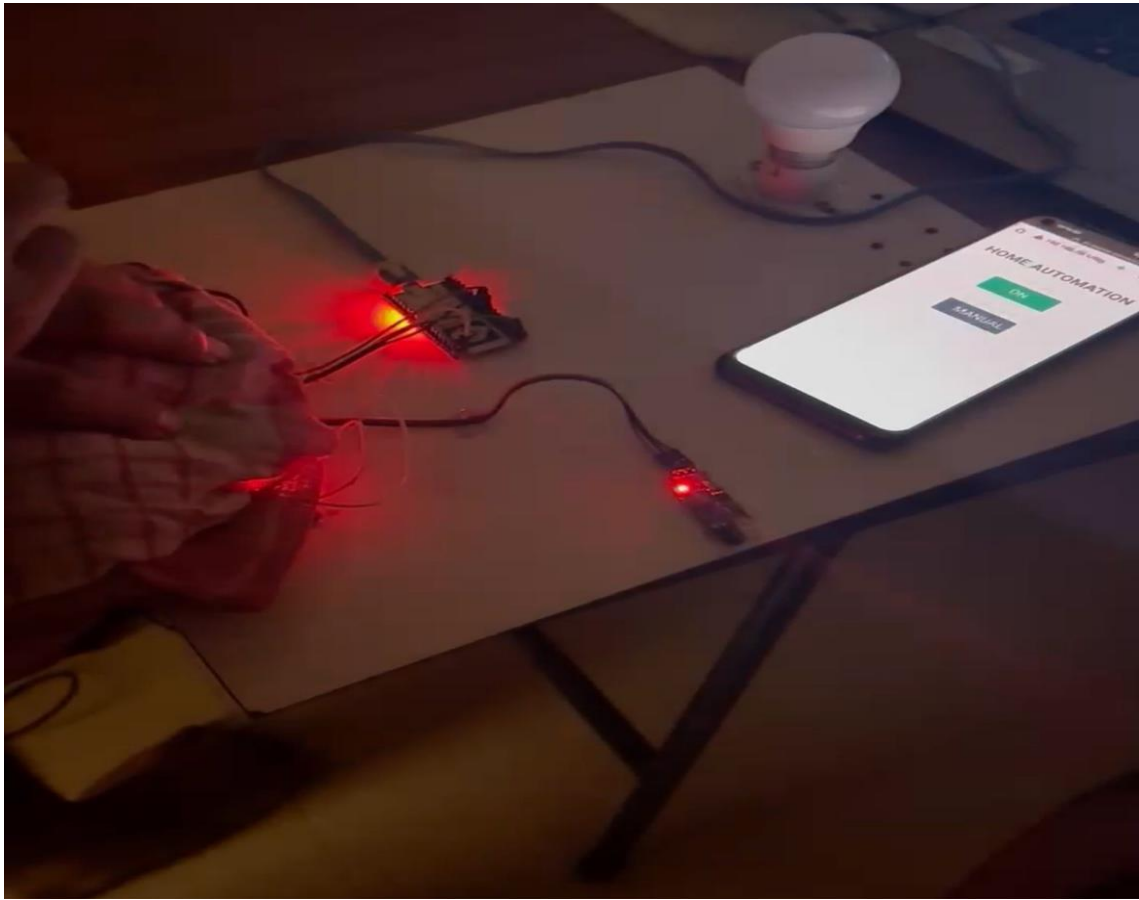


Fig.7 No movement detection

AUTOMATIC ROOM LIGHTS CONTROL BY USING ARDUINO DESIGN

^[1]PG Scholar Moirangthem Sanathoi Singh (Dept. of CSE),
Sathyabama Institute of Science and Technology
Sanathoimoirangthem552@gmail.com

^[2]Prof. Dr.L.Lakshmanan (Head Department of CSE),
Sathyabama Institute of Science and Technology
Lakshmanan.cse@sathyabama.ac.in

Abstract— An autonomous light control system that employs an Arduino and a PIR (passive infrared) sensor may turn on and off the lights in a house based on its ability to detect human presence. This system can be used in restrooms, garages, classrooms, stairwells, and other locations that need light but aren't usually occupied by humans. Additionally, there is no need to worry about power costs since, in the absence of a person, the lights go out and bill payment must be made in accordance with usage. This study suggests an autonomous light control system that turns on the lights in the room using an Arduino and a PIR sensor. The primary three components of this system are a relay module, a PIR sensor, and an Arduino Uno. The PIR sensor, which aids in the detection of human presence, is the component that powers the system the most out of all of them. The suggested technique can save at least 50% of electricity, according to experimental data.

Keywords— automatic room light, LDR Sensor, PIR Sensor, and Arduino.

I. INTRODUCTION

Thanks to scientific advancements, life is now worthwhile. They have given us luxury and comfort. Technology now plays a significant and indispensable role in our daily lives. Technology has advanced tremendously in the previous several years. Thus, the dependence of human life on these technology increases. The growth of electricity is one of its biggest inventions overall. The use of electricity has become essential to human existence. The earth would stay lifeless and dead without electricity, which is regarded as the soul or life. Home automation is a technology that aims to make human comfort levels higher. Home automation is the process of using technology to automate every amenity and appliance in the house. Energy conservation is necessary since non-renewable resources are becoming scarcer by the day. The most crucial component of all modern society's operations is electricity. Therefore, using non-renewable energy sources wisely is essential. Through the observation and control of many devices, home automation creates energy-efficient spaces. These days, energy-related problems are the principal ones. People in our day and age are often

rushing, so they often forget to switch off the lights and fans. This home automation feature, the Automatic Light Control System, saves power in such a scenario. Electrical appliances can automatically turn on or off based on whether people are in a certain region of the room. This system's primary benefit is this.

II. LITERATURE SURVEY

The basis of any mentoring course is Performance Analysis Systems. With the guidance and assistance of statistical data, an individual will surely progress, which will also help the business that hires him succeed. A background inquiry is done in order to look at comparable existing techniques that are utilised to analyse student performance. Before starting our literature review, we learn about three existing systems that are comparable to the one that is being proposed.

Vibhuti et al. [1] suggested a system that uses an Arduino Uno, WAGO PLC (Programmable Logic Controller), and relay control. PIR sensors allow for the spontaneous functioning of equipment, such as fans, AC units, tube lights, and so forth, based on ambient circumstances. In a real-time implementation, an Android application handles manual control while sensor data handles automated control. However, this paper's challenge lies in the way that both the Arduino Uno and the WAGO PLC are used to control and monitor the devices. All that is needed to complete these tasks is an Arduino Uno.

The Raspberry Pi is used in the smart lighting system of Maslekar et al. [2]. Raspberry Pi is monitoring lights and fans simultaneously. When no one is in the room, the fans and lights will turn off automatically. Energy is saved by using this smart lighting system. The experimental results for this system show that 50% of the energy is retained. But the Raspberry Pi is more expensive than the Arduino Uno. Bluetooth, PIR sensor, relay, and electrical light control are all part of the classroom's automatic lighting and control system. When the Arduino Uno has been linked to the Bluetooth module, it may transmit voice commands to the Arduino Uno using a mobile Android application, allowing it to turn on or off the light. According to the experimental data, 50% of the energy is preserved.

However, the Bluetooth module can also be removed in order to implement this research [3]. The discussions in [4] discuss an automated room light system that makes use of visitor counter functionality.

Depending on whether somebody is present, the room's lights will either turn on or off.

Manual labour is not required for switching. The PIR sensor detects human presence when someone approaches the room's entry. The visitor counter is tallied whenever a person uses it to enter a room; this causes the microcontroller program to be triggered, turning on the room light. The counter shifts when someone leaves the room, which results in

the lights to go off. The only thing that turned off in the room was the lights after everyone had departed. The problem with this approach is that only one person should be able to enter a room at a time.

A system that is controlled in Vahid et al. proposed the Modbus industrial protocol, network connections, and an Arduino microcontroller [5]. The network connectivity is developed using an A wireless router device in addition to an Arduino Ethernet shield. A USB (Universal Serial Bus) cable is used to load Arduino code into the Arduino board, and a particular Android application called "Myron" is used to load the Modbus programme onto mobile or Windows software. Through an Ethernet cable and router, the Arduino Ethernet Shield and mobile device are connected. Connecting to the router makes it simple for the user to monitor and operate the appliances.

III. SYSTEM IMPLEMENTATION

Existing System

These days, electrical energy is essential to human existence. Individuals are nonetheless passionate about using less energy to save money, even though they have been anticipating automation in their daily lives in recent years. The majority of individuals are usually not in public or private sector offices, organizations, schools, or institutions

interested in turning off electronic devices like fans and lights.

Disadvantage

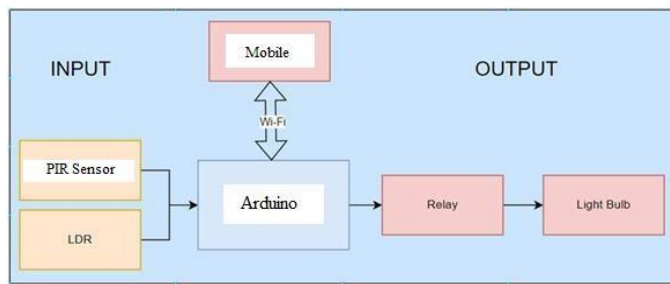
- These days, energy-related problems are the major issue.
- Because everyone is rushing these days, individuals Often forget to toggle off the fans and lights
- Wasted electricity.

Proposed System

You won't have to bother about flipping the switches on and off each time you want to turn on the lights thanks to this extremely helpful Arduino Automatic Room Lights System project. Relay module, PIR sensor, and Arduino are the three essential parts of the Automatic Room Lights project. The primary tool for identifying people and human movements among the three parts is the PIR Sensor, hence it is the one that is being examined. One significant use of the PIR Sensor is really the Automatic Room Lights project.

Advantage

- Savings on energy.
- Economy of Cost.
- Ecologically Sustainable.
- Practicality for the User.
- Extended Life of the Bulb.



Proposed Architecture

Methodology

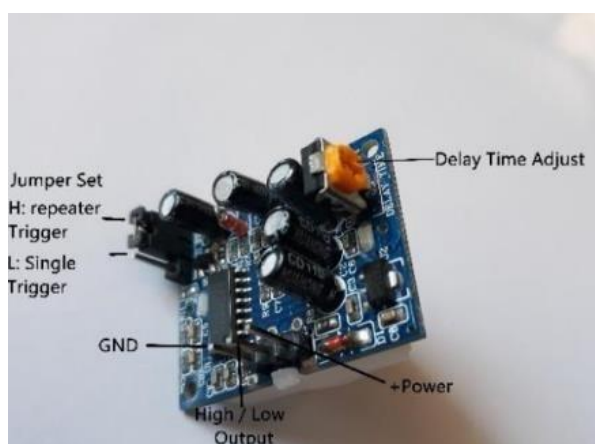
A variety of hardware components make up the automatic light control system, which may be used to process, sense, and regulate appliances. We go over these hardware parts below:

1. PIR Sensor as Sensing Unit

The sensor device is used to gather environmental input characteristics that are needed for automation. To ensure proper functioning in a given section of the room, the following considerations need to be made:

- Constant modifications to human movement
- Conserving (Fiscal)

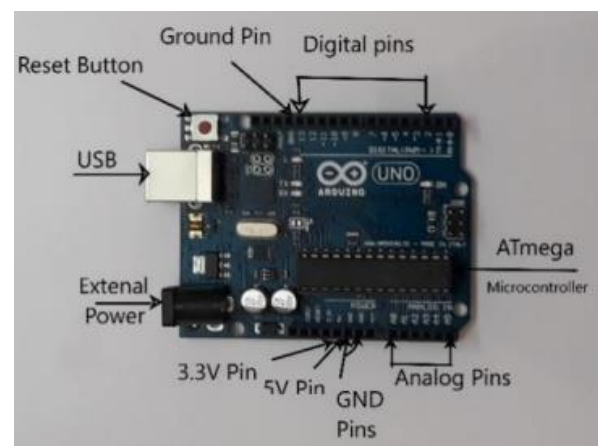
The PIR (Passive Infrared) sensor utilised in this instance is seen in the provided Figure.



Passive infrared, often known as pyroelectric or "IR motion" sensors, is the term used to describe this type of PIR sensor. When a human enters the PIR sensor's sensing range, the sensor begins to detect their movements. This sensor's unique features are its small size, low power consumption, affordability, ease of use, and non-exhaust. As a result, a lot of equipment for home automation employs this sensor.

2. Arduino Uno as Processing Unit

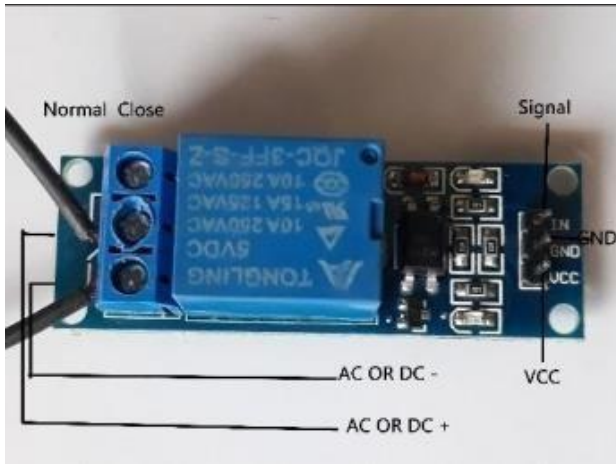
The Arduino Uno microcontroller, seen in the figure, uses the ATmega328 CPU.



In addition to 6 analogue input ports, it has 14 digital input or output pins that can be used as PWM (Pulse Width Modulation) outputs. Its programming language is distinct from others. The crystal oscillator frequency used by this microcontroller is 16MHz. It has an easy-to-connect USB connection to a computer, a reset button, a power barrel connector, and ICSP (In Circuit Serial Programming). All of the pins on the Arduino Uno run at 5V. There is a simple programming language for this microcontroller.

3. Relay as Controlling Unit

As seen in Figure, a relay is an electrically powered device.



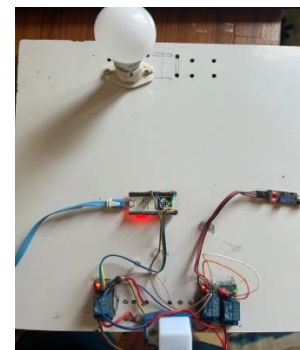
Relays are digital switches that have substantially greater voltages and current limits. This gadget is frequently utilised for power preservation. The advantages of this device are its long lifespan, stability, and compact size. Additionally, it is applicable to both ac and dc systems. A relay has three terminals: the common terminal, the ordinarily closed terminal, and the usually open terminal. The input signal, VCC, and GND are connected to the three pins.

Implementation

The microcontroller Arduino offers an open-source platform for both hardware and software operations. This project is useful since it combines a PIR sensor and an Arduino Uno to detect the presence of people in the room and turn on the lights automatically. As long as the person is in the room, the lights are going to remain on. When no one is in the room, the PIR Sensor's OUT pin is formerly in the LOW mode. As a result,

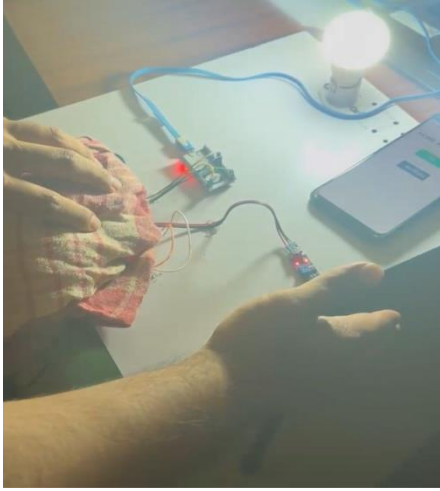
The PIR sensor's output rises when someone enters the room while the lights are off. The PIR sensor picks up infrared (IR) radiation from the room. The Arduino Uno's Digital pin 8 is linked to the PIR Sensor's Data OUT pin. This is how Arduino triggers the relay when it reaches the HIGH setting. One-o. Relays are active LOW devices, hence the pin on the relay is in the LOW state. The lights will now come on. As long as there is motion in the space, this light will remain in the ON state. If someone leaves the room or naps, the motion in front of the sensor ceases, and the infrared radiation remains unchanged. Consequently, the Data Out pin of the PIR sensor will be in the LOW mode. As a result, the relay is turned off. Right now, the relay is in the HIGH mode. As a result, the room's lights will be off.

Experimental Results



Manually switches OFF/ON



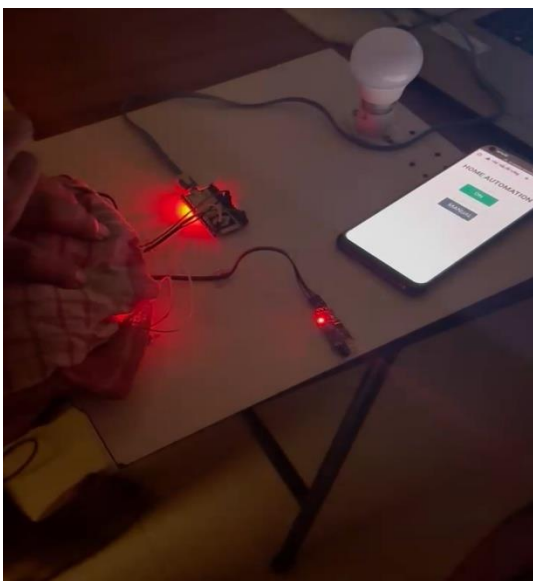


PIR sensors sense motion when there is no movement within the room by detecting variations in the infrared radiation that moving things emit. It sends a signal to the Arduino or microcontroller, indicating that someone is present in the room. As a result, the Arduino activates the relay or switches connected to the room lights, turning them on. If no motion is detected (after the last detected motion), the Arduino triggers the relay or switches to turn off the lights. This is a built-in energy-saving feature that ensures the lights are only on when the room is occupied.

When the PIR sensor detects movement in the room, it sends a signal to the Arduino indicating that motion has been detected. The Arduino responds to the signal by activating the lights, typically by sending a signal to a relay module that switches the lights on. The light remains on for a predefined duration to provide illumination in response to the detected movement. After a certain period of inactivity, the Arduino turns off the lights to conserve energy.

IV. CONCLUSIONS

The concept of automated houses has been presented in this work, along with a solution that reduces system power usage. The sensors and household appliances are connected to this automated light management system, which allows for device monitoring and control. An automated house is a large system with several applications and technologies that may be utilised to simply regulate and secure a home. The proposed system suggests that a number of devices are employed in a technique to control the lighting in the room. Because so much energy is wasted in contemporary living. This technology can keep energy waste under control and help save a significant amount of electricity. The system's overall effective cost is extremely low.



To improve the accuracy of autonomous lighting systems, a variety of technologies can be applied.

- Receiving notifications via the GSM (Global System for Mobile) module gives the system a more polished appearance.
- The home may be secured and controlled using a few different types of sensors.
- One such device is a pressure sensor that is installed outside the door to sense occupancy.

Digital cameras may also be used to identify people's presence through image processing.

REFERENCES

- [1] Abhishek N Vaghela, Bhavin D Gujjar "Automatic Switch using PIR Sensor" International Journal of Engineering Development and Research.
- [2] A. H. Jabire, H. Suleiman, A. A. Noku, P. U. Adamu, D. A. Tsoken, and L. O. Agidike, "Bidirectional Visitors' Counter with Automatic Room Light Controller," *Taraba J. Eng. Technol.*, vol. 2, no. 1, pp.75–80, 2021.
- [3] "Automatic Lighting and Control System For Classroom," 2016 International Conference on ICT in Business Industry & Government (ICTBIG), Suresh S., H.N.S. Anusha, T.Rajath, P.Soundarya, and S.V. Prathyusha Vudatha.
- [4] Castillo-Martinez, J.-A. Medina-Merodio, J.-M. Gutierrez-Martinez, J. Aguado-Delgado, C. de Pablos-Herederó, and S. Oton, "Evaluation and improvement of lighting efficiency in workspaces," *Sustainability*, vol. 10, no. 4, pp. 11-10, 2018.
- [5] Daeho Kim, Junghoon Lee, Yeongmin Jang and Jaesang Cha. "Smart LED lighting system implementation using Human tracking US/IR sensor" 2011 IEEE (ICTC 2011).
- [6] G. Saranya, E. Duraiarasu and S. Manoj, "Universal Smart Lighting System," 2023 IEEE Sustainable Smart Lighting World Conference & Expo (LS18), Mumbai, India, 2023, pp. 1-5, doi: 10.1109/LS1858153.2023.10170266.
- [7] Karthikeyan.R.A, [2] piyush.D "Automatic Lighting Using Arduino and PIR Sensor", *International Journal of Computer Science Trends and Technology*, Volume 6, SeptOct2018.
- [8] Kunturwar, Vrushabh and Shinde, Atul and Pokharkar, Ashwini and Shroff, Arpita, Automatic Room lights Control by Using Arduino and IR (April 17, 2020). 2nd International Conference on Communication & Information Processing (ICCIP) 2020.
- [9] "Mechanicals Room Light Controller with Visitor Counter, "accessible at <http://www.projectsof8051.com/automaticroom-light-controller-with-visitor-counter>.
- [10] Mahesh Kumar Singh" Arduino based: Automatic Room Light Control" department of ECE, Buddha Institute of Technology, Gorakhpur (Uttar Pradesh), India.
- [11] P. Rajakumar, B. C. Kanna, R. Archana, A. Akshaya, K. Abhinaya and M. Boobalan, "Automation of Room Lighting System Using Arduino and PIR Sensor," 2022 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS), Chennai, India, 2022, pp. 1-3, doi: 10.1109/ICPECTS56089.2022.10047213.
- [12] Richu Sam Alex, R Narciss Starbell "Energy efficient Intelligent Street Lighting System using Sensors", *International Journal of Engineering and Advanced Technology (IJEAT)*, Vol-3, Issue 4, April 2014.

- [13] S. Fahad, "Bidirectional Visitor counter using Arduino and UltrasonicSensors, Counting Visitors," 2021.<https://www.electronicclinic.com/bidirectional-visitor-counter-usingarduino-and-ultrasonic-sensors-counting-visitors/> (accessed Jan. 20, 2023).
- [14] "Smart Lighting System using Raspberry Pi," A. Maslekar, K. Aparna, K. Mamatha, and T. Shivakumara, International Journal of Innovative Research in Science and Technology, Vol. 4(7)2015, pp. 5113-5121I.
- [15] The article "Implementation of Smart Class Room Using WAGO PLC" was published in the Proceedings of the Second International Conference on Inventive Systems and Control (ICISC) 2021, held in Coimbatore. The authors are Vibhuti and Shimi S.L.
- [16] The paper "Low-Cost Home Automation Using Arduino and Modbus Protocol" was presented at the 5th International Conference on Control, Instrumentation and Automation (ICCIA) in Shiraz, 2017 by Vahid Hassanpour, Sedighe Rajabi, Zeinab Shayan, Zahra Hafezi, and Mohammad Mehdi Arefi.
- [17] <https://www.projectsof8051.com/automatic-room-light-controller-with-visitor-counter/>
- [18] <https://circuitdigest.com/microcontroller-projects/automatic-room-light-controller-with-bidirectional-visitor-counter-using-arduino>

