

IOT-BASED AUTOMATIC STREET LIGHTING SYSTEM

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

By

K. Geethanjali (Reg.No - 42310003)



**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

JEPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 11G

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 **K. Geethanjali (42310003)** who carried out the Project entitled "**IOT-BASED AUTOMATIC STREET LIGHTING SYSTEM**" entitled under my supervision from January 2024 to April 2024.

My
20/4/2024

Internal Guide

Dr. MERCY PAUL SELVAN M.E., Ph.D.

plm

Head of the Department

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

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Jeppiaar Nagar, Rajiv Gandhi Salai,
Chennai – 600 119.

Submitted for Viva voce Examination held on 20 April 2024

R. M. Gpm
20/4/24

Internal Examiner

Y. Nar
20/4/2024

External Examiner

DECLARATION

I, K. GEETHANJALI (42310003), hereby declare that the Project Report entitled "IOT-BASED AUTOMATIC STREET LIGHTING SYSTEM" done by me under the guidance of Dr. MERCY PAUL SELVAN, 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

K. Geethanjali

SIGNATURE OF THE CANDIDATE

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, **Dr. L. Lakshmanan M.E., Ph.D., Head of the Department of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. Mercy Paul Selvan, M.E., Ph.D.,** for her 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

Providing an "IoT-based automatic street lighting system" at night is the goal of this project. All cities and roads must have street lighting because it reduces the risk of accidents and undesired thefts or robberies. Every day, even when no one is there, the street lights run continuously at full power from dusk till dawn. Millions of dollars are spent daily on these street lights across the world in order to supply the necessary power. Conventional light bulbs are quite expensive to replace and maintain. They emit a great deal of heat and require a lot of power to run. All of this increases the demand for the production of energy and, as a result, increases the amount of carbon dioxide that power plants emit into the atmosphere. This technique harms not just unneeded light pollution but also our world. Thousands of streetlights are situated beside major roads and highways. However, the primary issue is that these streetlights account for 25–30% of the city's overall energy use. Dimming the off-peak lights would be an easy and efficient fix. The lights surrounding a detected presence go into normal (bright) mode. Fault light is detected, and the message is sent to the concern person that the light is not working. This would lower the cost of operating public lights and save a significant amount of energy.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	V
	LIST OF FIGURES	IX
	LIST OF ABBREVIATIONS	X
1	INTRODUCTION	1
2	LITERATURE SURVEY	2-3
	2.1 System Study	4
	2.1.1 Existing System	4
	2.1.2 Proposed System	5
	2.1.3 System Specification	6
3	SOFTWARE SPECIFICATION	7
	3.1 Java	7
	3.2 Arduino IDE	8-9
4	SOFTWARE DEVELOPMENT	10
	4.1 Methodology	10

	4.2 Modules Description	11-14
5	SYSTEM DESIGN AND ANALYSIS	15
	5.1 Block Diagram	15
	5.2 Data Flow Diagram	16
6	SYSTEM TESTING AND IMPLEMENTATION	17
	6.1 System Testing	17
	6.1.1 Unit Testing	17
	6.1.2 Integration Testing	18
	6.1.3 Functional Testing	18
	6.1.4 Performance Testing	18
	6.1.5 White Box Testing	19
	6.1.6 Black Box Testing	19
	6.1.7 Acceptance Testing	19
	6.2 Input Design and Output Design	20-21
	6.3 Screen Shots	22-25

7	CONCLUSION	26
	REFERENCES	27-28
	APPENDIX	29
	A. SOURCE CODE	29-37
	B. RESEARCH PAPER	38-45
	C. CERTIFICATE	46

LIST OF FIGURES

FIGURE NO	FIGURE NAME	Page No.
4.1	Arduino uno	11
4.2	ATmega328	12
4.3	NODEMCU	12
4.4	LDR	13
4.5	PIR	13
4.6	LED	14
4.7	Breadboard	14
5.1	Block diagram	15
5.2	Data Flow diagram	16
6.1	Right-side view	22
6.2	Front view	23
6.3	Output image its showing light is not working	24
6.4	Output image its showing light is working	25

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
LED	Light emitting diode
LDR	Photo Light Dependent Resistor
PIC	Peripheral Interface Controller
PIR	Passive Infrared Sensor
PWM	Pulse Width Modulation

CHAPTER 1

INTRODUCTION

Any development site must have street lighting. They may be found in the suburbs and on all main roadways. Every day, even when no one is there, the street lights run continuously at full power from dusk till dawn. Millions of dollars are spent daily on these street lights across the world in order to supply the necessary power. Conventional light bulbs are quite expensive to replace and maintain. They emit a great deal of heat and require a lot of power to run. All of this increases the demand for the production of energy and, as a result, increases the amount of carbon dioxide that power plants emit into the atmosphere. This technique harms not just unneeded light pollution but also our world. The main parts of this project are the LED, IR sensor, Arduino board, and LDR. The system is relatively simple to put into practice. The Arduino board, which powers the whole system, is the project's brain. The IR sensor serves as the project's eyes, detecting the presence of people or cars on the road. The LDR determines if sunlight is there or not, and the LEDs stand in for street lights. When the system detects LDR sunlight throughout the day, it switches off the LED street lights. The system activates the LED street lights when it senses darkness in the evenings. Low-intensity lights will turn on when the street lighting is switched on if the system uses the IR sensor to detect any movement from moving cars or pedestrians on the road. The system activates the high-intensity street lights when it uses the IR sensor to detect movement from cars or pedestrians.

CHAPTER 2

LITERATURE SURVEY

Performance Analysis Systems constitute the foundation of every mentorship programme. An person will undoubtedly advance with the support and direction of statistical data, which will also contribute to the success of the company that employs him. To examine comparable current methods that are used to analyse student performance, a background investigation is conducted. We first get a knowledge of three current systems that bear similarities to the proposed system before beginning our literature review.

Numerous concepts for smart street light systems currently exist. S. Suganya et al. used the most recent advancements in light source technology, such as LED lights, to build a street light beam that can detect vehicle movement utilising the isa sensor system. Moreover, it uses infrared detection technology to provide dynamic flow-based control statistics for street light switching that is automatically controlled depending on light intensity. ZigBee Wireless protocol is also utilised to maintain wireless connection between the control terminal and the street light pole. A timer, operational current size statistics, photodiodes, LEDs, power transistors, and other technologies are also combined.

A survey of vehicle movement-based street lighting systems was carried out by K. Santha et al. When the system is operating in automated mode, light intensity, brightness, and dimming algorithms are used to control street lighting. Seasonal variations can be taken into account while regulating. To conserve power, it has an automated control pattern and a time-off feature. The PIC microcontroller was used to carry out the complete project. Based on ZigBee, he created an automated remote control system for public lighting. ZigBee modules are used in the system's architecture to assist in identifying and managing malfunctioning lights. It also talks about an intelligent system that makes decisions about on, off, and dimming on its own by monitoring movement of cars or pedestrians and the environment. Both live beings and non-living objects may be detected by using the PIR motion sensor.

OA traffic flow-based street lighting management system design that makes effective use of solar energy was put into place in 2015 by M. Abhishek et al. They employed solar energy, a renewable energy source, for street lighting. They also employed microcontrollers from the 8052 family, which were produced by swapping out regular

LED lights for ones that lower energy usage by three times. There are sensors on both sides of the road that pick up on the movement of the car and communicate with the microcontroller to turn on and off the lights. Here, the street lights are entirely off and only turn on when they sense movement from a moving car. Consequently, the lights are on or off regardless of the time of day thanks to the microcontroller.

Street lighting using an automated tracking system can improve solar energy conversion efficiency, according to research by C. Bhuvaneshwari et al. Here, a sensor that tracks the sun's position is a sensing device that periodically determines the sun's location and signals an amplifier's output depending on the amount of sunlight. The sun tracking sensor is an LDR; its signals are amplified by an amplifier unit, which transforms them from low-level to high-level signals. The comparator receives the output of the amplifier. The amplified circuit is an LM324 IC. After comparing the signals, the comparator instructs the AT89C51 microprocessor.

2.1 System Study

2.1.1 Existing System:

Street illumination is a necessary component of each development site under the current system. They may be found in the suburbs and on all main roadways. Every day, even when no one is there, the street lights run continuously at full power from dusk till dawn. Millions of dollars are spent daily on these street lights across the world in order to supply the necessary power. All of this increases the demand for the production of energy and, as a result, increases the amount of carbon dioxide that power plants emit into the atmosphere. This technique harms not just unneeded light pollution but also our world.

Disadvantage:

- Conventional light bulbs have significant maintenance and replacement expenses.
- They also require a significant amount of power to operate and emit a significant amount of heat.

2.1.2 Proposed System

The main parts of this project are the LED, IR sensor, Arduino board, and LDR. The system is relatively simple to put into practice. The Arduino board, which powers the whole system, is the project's brain. The IR sensor serves as the project's eyes, detecting the presence of people or cars on the road. The LDR determines if sunlight is there or not, and the LEDs stand in for street lights. When the system detects LDR sunlight throughout the day, it switches off the LED street lights. The system activates the LED street lights when it senses darkness in the evenings. When the street lighting is switched on, the low-intensity lights will turn on if the IR sensor is unable to identify any moving cars or persons on the road. The system turns on the street lights at a high intensity when it uses the IR sensor to detect movement from cars or pedestrians. Fault light is detected and the message is sent to the concern person that the light is not working

Advantage

- This would save a significant amount of energy and lower the expense of maintaining public lights.
- Using an infrared sensor, the device detects movement from cars or pedestrians and activates high-intensity street lighting.

2.1.3 System Specification

Hardware Requirements

Hard disk	:	1 TB
RAM	:	4 GB
Processor	:	Core i3
Monitor	:	15’’Color Monitor
Arduino UNO		
ATmega328		
Photo Light Dependent Resistors (LDR)		
PIR Sensors		
LED		
OLatus OL-BREADBOARD-400		

Software Requirements

Coding Language	:	Java
Operating System	:	Windows 10
IDE	:	ARDUINO IDE

CHAPTER 3

SOFTWARE SPECIFICATION

3.1 Code Language – Java using ARDUINO IDE

Java is a widely-used programming language for coding web applications. It has been a popular choice among developers for over two decades, with millions of Java applications in use today. Java is a multi-platform, object-oriented, and network-centric language that can be used as a platform in itself. It is a fast, secure, reliable programming language for coding everything from mobile apps and enterprise software to big data applications and server-side technologies.

1. Game Development

Many popular mobile, computer, and video games are built in Java. Even modern games that integrate advanced technology like machine learning or virtual reality are built with Java technology.

2. Cloud computing

Java is often referred to as WORA – Write Once and Run Anywhere, making it perfect for decentralized cloud-based applications. Cloud providers choose Java language to run programs on a wide range of underlying platforms.

3. Big Data

Java is used for data processing engines that can work with complex data sets and massive amounts of real-time data.

4. Artificial Intelligence

Java is a powerhouse of machine learning libraries. Its stability and speed make it perfect for artificial intelligence application development like natural language processing and deep learning.

5. Internet of Things

Java has been used to program sensors and hardware in edge devices that can connect independently to the internet.

Java is popular because it has been designed for ease of use. Some reasons developers continue to choose Java over other programming languages include:

High quality learning resources

Java has been around for a long time, so many learning resources are available for new programmers. Detailed documentation, comprehensive books, and courses support

developers through the learning curve. In addition, beginners can start writing code in Core Java before moving to Advanced Java.

Inbuilt functions and libraries

When using Java, developers don't need to write every new function from scratch. Instead, Java provides a rich ecosystem of in-built functions and libraries to develop a range of applications.

Active community support

Java has many active users and a community that can support developers when they face coding challenges. The Java platform software is also maintained and updated regularly.

High-quality development tools

Java offers various tools to support automated editing, debugging, testing, deployment, and change management. These tools make Java programming time and cost-efficient.

Platform Independent

Java code can run on any underlying platform like Windows, Linux, iOS, or Android without rewriting. This makes it especially powerful in today's environment, where we want to run applications on multiple devices.

Security

Users can download untrusted Java code over a network and run it in a secure environment in which it cannot do any harm. Untrusted code cannot infect the host system with a virus nor can it read or write files from the hard drive. The security levels and restrictions in Java are also highly configurable.

All programming languages are a means to communicate with machines. Machine hardware only responds to electronic communication. High-level programming languages like Java act as a bridge between human language and hardware language. To use Java, a developer needs to understand two things:

1. Java language and APIs

This is the front-end communication between the developer and the Java platform.

2. Java Virtual Machine

This is the back-end communication between the Java platform and the underlying hardware. Let's look at each of these in detail below.

Java defines the syntax and semantics of the Java programming language. This includes the basic vocabulary and rules used to write algorithms such as primitive data types, if/else blocks, loops, etc.

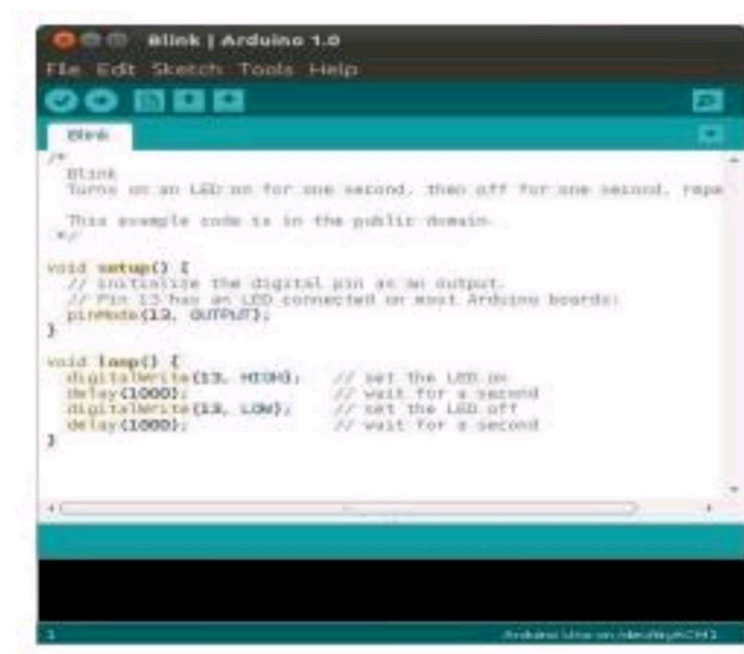
APIs are important software components bundled with the Java Platform. These are pre-written Java programs that can plug and play existing functionality into your own code. For example, you could use Java APIs to get the date and time, perform mathematical operations, or manipulate text.

Any Java application code written by a developer will typically combine new and pre-existing code from Java APIs and Java libraries.

When natural programming languages were first developed, they fell into two broad categories, depending on how they communicated with the underlying hardware.

1. Compilers: The complete program is written in natural English-like syntax with compilers, and the language then compiles (or translates) the entire code into machine code. The compiled code is then run on the hardware.
2. Interpreters: With interpreters, every high-level code statement is interpreted into machine code on the fly. Written statements are run immediately by the hardware before looking at the next statement.

The Arduino Software (IDE) is open-source software and makes coding and uploading to the board easy. Running on a device other than Windows, MAC OS, Linux. The environment is written in Java and can be used with any Arduino board before running the Java IDE software to be installed on the machine.



CHAPTER 4

SOFTWARE DEVELOPMENT

4.1 METHODOLOGY

The Arduino's analogue pin is linked to the LDR. detects whether or not sunlight is present in order to control LEDs.

Condition 1: The LDR delivers high resistance and functions as an insulator when there is sufficient sunlight present. In this instance, the Arduino will automatically turn off all of the LEDs (street lights) by reading the high values of the analogue output from the LDR.

Condition 2: The LDR senses darkness in the absence of sunshine, provides low resistance, and functions as a conductor. In this instance, the LED (street lights) is automatically turned on by the Arduino once it receives the low analogue input values from the LDR.

Condition 3: The PIR sensor also activates at the same moment and begins to identify any moving cars or pedestrians on the road. Two PIR sensors are used in this project, and they are wired to the Arduino's digital pins. In addition, we utilise four LEDs, each linked to the Arduino's PWM ports, to symbolise the street lights. Two LEDs are controlled by each PIR sensor. One of the PIR sensors will go LOW (0) in output when it detects the position of a vehicle or human. The Arduino then used the Pulse Width Modulation (PWM) technology to boost the two LEDs' light intensity after reading the sensor's low output value. The PIR sensors' output will switch to High (1) if they do not detect any humans, vehicles, or obstacles. After that, the Arduino read the sensor's high output value. Using the Pulse Width Modulation (PWM) method, the Arduino now lowers the LED light's intensity.

Condition 4: At night, during off-peak hours, street lights fade to lower settings; but, when cars, bikes, or pedestrians are recognised, the lights automatically illuminate. LDR also detect the fault light and send the message that the light is not working.

4.2 MODULES DESCRIPTION

Arduino Uno

With the typical size factor of the UNO family, the Arduino UNO WIFI Rev.2 is the easiest way to get started with basic IoT. For many common IoT application scenarios, the Arduino UNO WiFi Rev.2 is your one-stop shop. It may be used to establish a sensor network linked to your home or business router or to create a Bluetooth® Low Energy device that delivers data to a mobile phone.

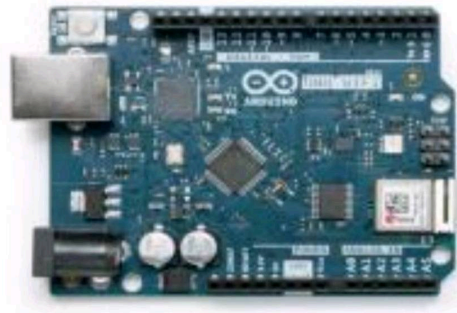


Fig:4.1: Arduino uno

The Arduino UNO WiFi Rev. 2 is equipped with 14 digital input/output pins, including 6 analogue inputs, 5 PWM output pins, a USB port, a power connector, an ICSP header, and a reset button. Everything required to sustain the microcontroller is contained in it. Simply use a USB connection to connect it to your computer, or an AC adapter or battery to power it, to get started.

ATmega328

The bootloader uses up 0.5 KB of the 32KB ATmega328. Additionally, it features one KB of EEPROM and two additional KRAM, both of which can be read and written with the EEPROM library. You may utilise Arduino code in your own embedded project without a real Arduino board by using this gadget.

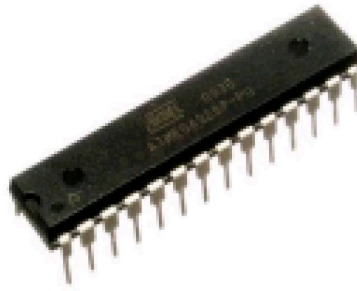


Fig:4.2: ATmega328

NODEMCU

Since NodeMCU is an open-source platform, anybody may construct, modify, and alter its hardware architecture. The ESP8266 chip, which supports WiFi, is what makes up the NodeMCU Dev Kit/board. The TCP/IP-enabled, low-cost ESP8266 Wi-Fi chip was created by Espressif Systems. The PCB of the NodeMCU Dev Kit features analogue (A0) and digital (D0-D8) Arduino pins. Serial communication protocols, such as SPI, I2C, and UART, are supported. We can link it with serial devices, including touch screens, SD cards, GPS modules, RTC chips, HMC5883 magnetometers, MPU-6050 gyro metres with accelerometers, and LCD displays with I2C support.

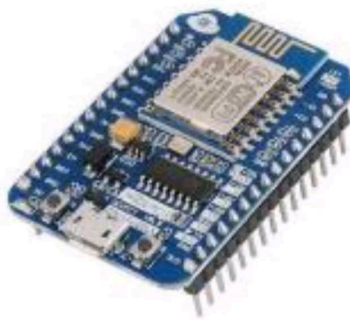


Fig:4.3: NODEMCU

Photo Light Dependent Resistors (LDR)

A device whose resistance factor depends on electromagnetic radiation is known as a light-dependent resistor (LDR), commonly referred to as a photo resistor. They are light-sensitive gadgets that resemble human eyes because of this. They can also be referred to as photocells, conductive cells, or photoconductors. They are constructed from very

resistant semiconductor materials. The photoconductivity concept underlies LDR operation. An optical phenomena known as photoconductivity occurs when a substance actually absorbs light, causing the conductivity of the material to drop. On the other hand, the LDR becomes less resistant to light, which causes current to flow to the bases of the first and second transistors. The circuit's sensitivity may be adjusted by adjusting the preset resistance, which increases or decreases the resistance. Arduino receives a reply from LDR.



Fig:4.4: LDR

PIR Sensors

Passive infrared sensors, or PIR sensors, are another name for them. Nearby objects generate infrared light, which may be detected using a passive infrared (PIR) sensor. PIR sensors do not release infrared radiation from LEDs; they merely detect it.



Fig:4.5: PIR

Two strips of pyroelectric material (pyroelectric sensor) make up passive infrared sensors. An infrared filter (which filters out all other light wavelengths), Fresnel lens, which concentrates light from many directions into a single point, and case, which shields the sensor from other environmental factors like humidity

LED

When turned on, a junction diode produces light, which is known as an LED. The LED's holes allow electrons to recombine and release energy in the form of photons, which produce light, when voltage is applied to the wires. It is a two-wire semiconductor light source as a result.

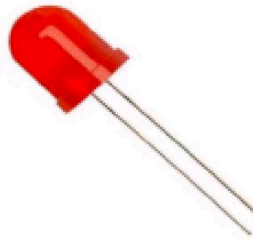


Fig:4.6: LED

OLatus OL-BREADBOARD-400

All you need a breadboard is a board for creating circuits or prototyping. It enables you to build circuits without soldering by arranging parts and connections on the board. A breadboard's holes manage your connections by electrically joining the components or wires inside the board while also supporting them physically. Additionally, a letter ranging from A to J is labelled in each column, making it simple to refer to the instructions (much like squares on a chessboard).

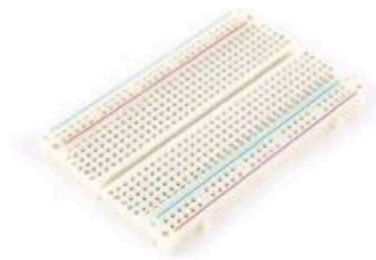


Fig:4.7: breadboard

CHAPTER 5

SYSTEM DESIGN AND ANALYSIS

5.1 Block Diagram

A block diagram is a system diagram where the main components or functions are represented by blocks joined by lines indicating the connections between the blocks. In engineering, they are widely utilised in process flow diagrams, software, hardware, and electrical design.

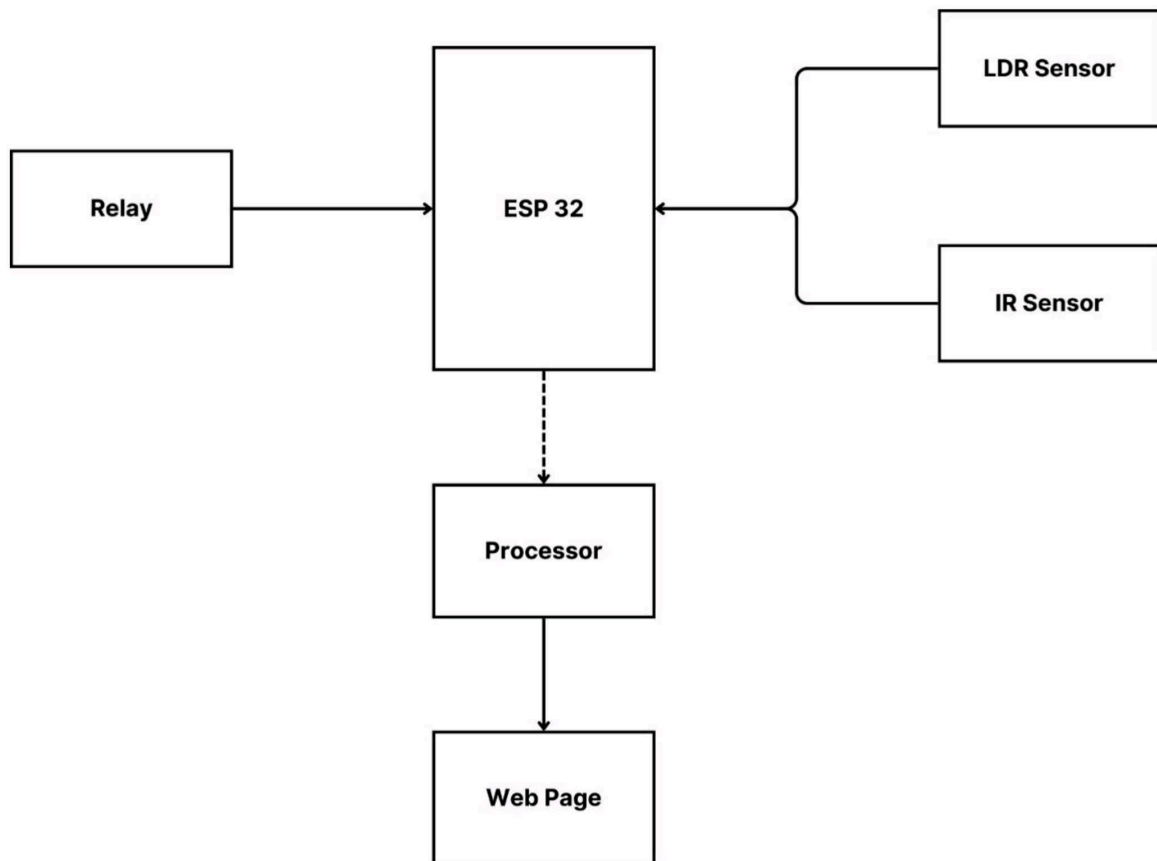


Fig:5.1: Block diagram

5.2 Data Flow Diagram

A data-flow diagram (DFD) is a visual aid for illustrating how data moves through a system or process. Information on each entity's inputs and outputs as well as the process itself are also provided by the DFD. A data-flow diagram lacks decision rules, loops, and control flow. A flowchart can be used to illustrate certain operations based on the data.

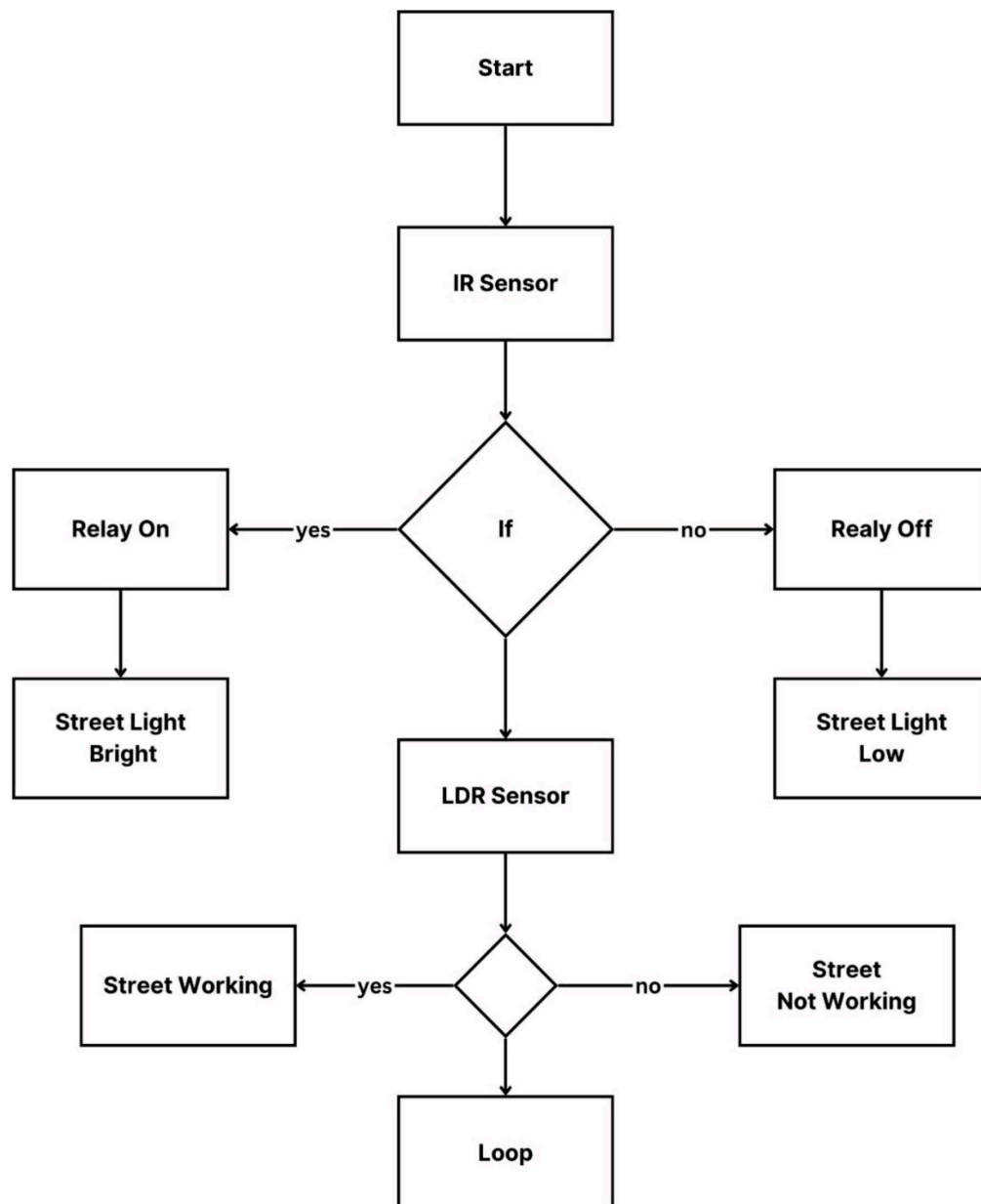


Fig:5.2: Data flow diagram

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 satisfactory, 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 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 in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is 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, 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.

6.3 Screen Shots



Fig:6.1: Right-side view(LDR,PIR connected to light through relay and arduino uno)



Fig:6.2: front view(LDR,PIR connected to light through reay and arduino uno)

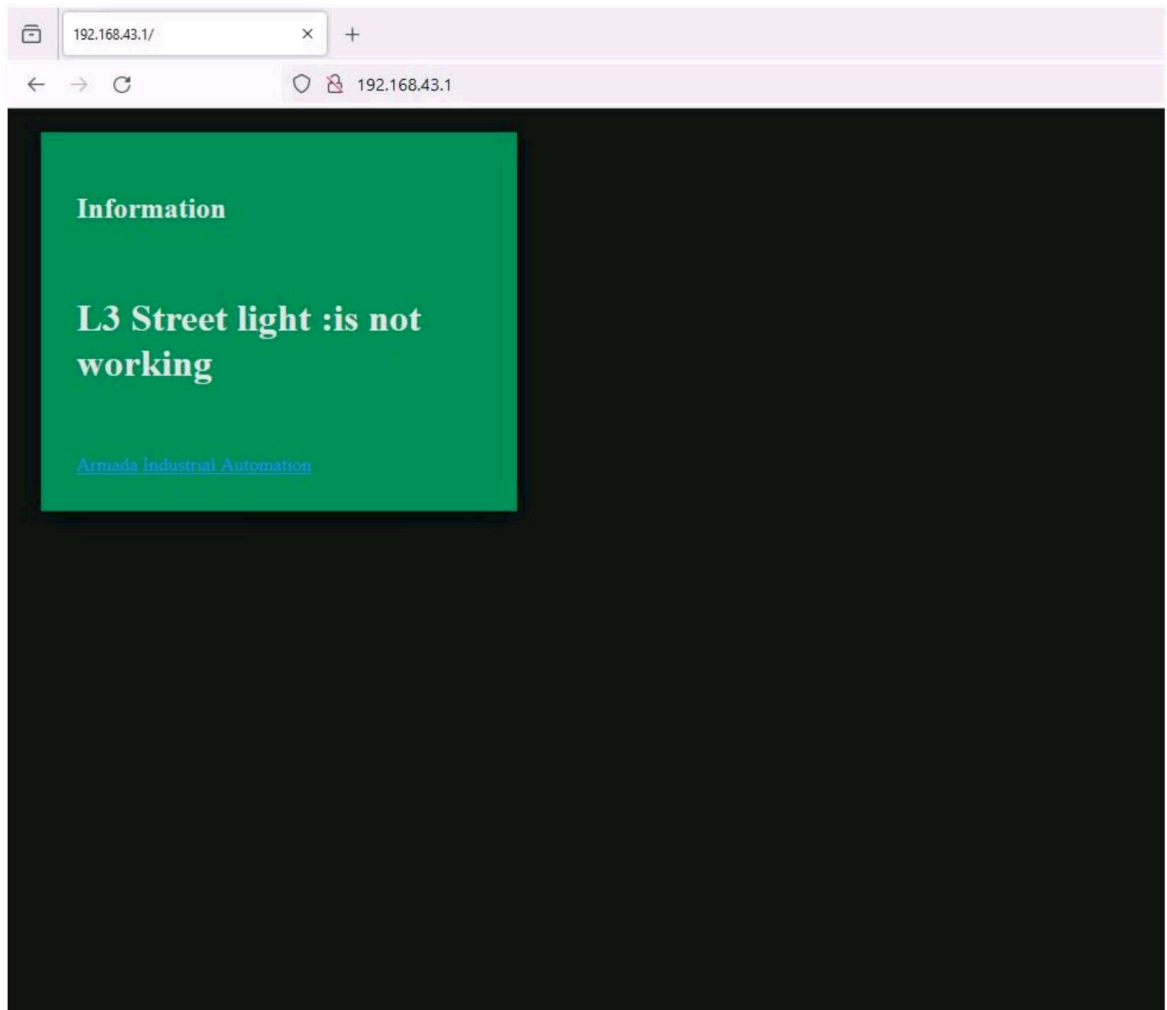


Fig:6.3: Output image its showing light is not working

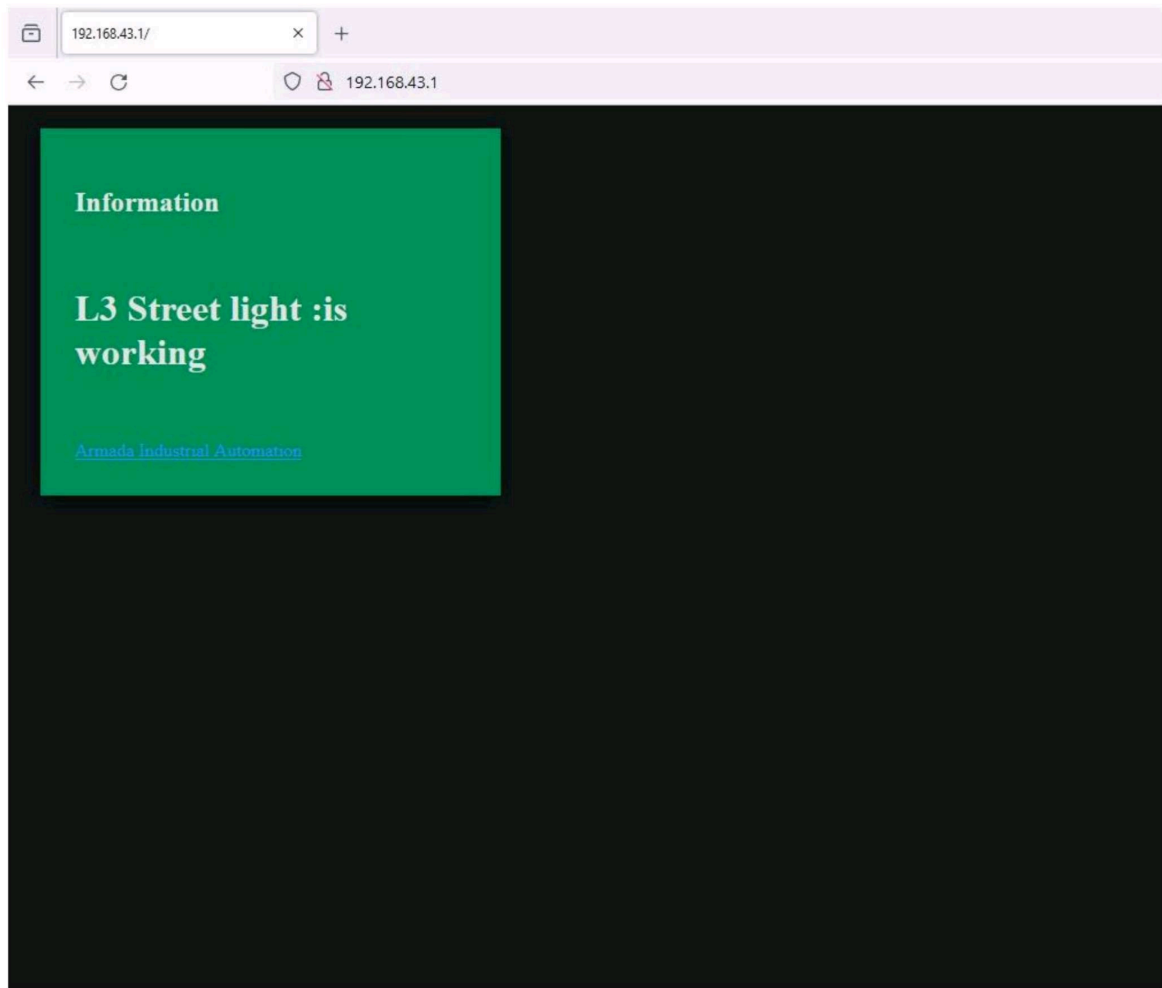


Fig:6.4: Output image its showing light is working

CHAPTER 7

CONCLUSION

The "IOT Based Smart Lighting" project has the potential to be the safest, efficient, and environmentally friendly method of preventing energy waste. It also allows for anytime, anywhere access to sunlight status data. It saves energy and fixes the issue that the planet has been having lately. This paper covers a wide range of applications for lighting parking lots in several big spaces, including retail malls, colleges, and enterprises. The essay offers a number of benefits that could outweigh the existing restrictions. Because of the incredibly short payback time, taking into account the long-term advantages and consequently the initial value would never be an issue. Fault light is detected, and message is sent to the concern person that the light is not working.

REFERENCES

- [1] J.C. Downey, "Discussion on street lighting," in Transactions of the South African Institute of Electrical Engineers, vol. 48, no. 1, Jan. 1957, pp. 36-38.
- [2] In Volume 3 of the IOSR Journal of Electrical and Electronics Engineering (IOSR- JEEE), S. Jadhav presents "An energy efficient pedestrian aware smart street lighting system."
- [3] In the International Workshop on Mobile Ad-Hoc Systems, A. Lavric, V. Popa, C. Males, and I. Finis conducted a performance research on ZigBee wireless sensor network topologies for street lighting management systems. I W
- [4] P.Y. Chen, Y.H. Liu, Y.T. Yau, and H.C. Lee, The IEEE International Conference on Sustainable Energy Technologies, Singapore, 2008, "Development of energy efficient street lighting system."
- [5] "Smart SSL: Application of IoT/CPS design platforms in LED-based street-lighting luminaires," A. Szalai, T. Szabó, P. Horváth, A. Timár, and A. Poppe, 2016 IEEE Lighting Conference of the Visegrad Countries (Lumen V4), Karpacz, Poland,
- [6] Prasad, R. "Energy Efficient Smart Street Lighting System in Nagpur Smart City using IoT-A Case Study," Fifth International Conference on Fog and Mobile Edge Computing (FMEC), Paris, France, 2020.
- [7] "Energy and Economic Analysis of Smart Technologies on Street Lighting System," by Modabbir and A. Mohammad, presented at the 7th International Conference on Advanced Computing and Communication Systems (ICACCS), July 2021 in Coimbatore, India.
- [8] In 2021 (IJEACS), S. Kushdeep Kaur, M.Bheema Singh "An energy saving, aware smart street light system."
- [9] In the International Workshop on Embedded IOT, lee Xiang , Park Shin Hae, and Louis Maxwell conducted a performance research on Embedded wireless sensor network for street Lighting Organize systems.
- [10] Shobitha S.Kumar, Parithya Agarwal, Sherin Mervin National conference in delhi IEE, "Evolution of Energy Efficient street light system." In Aug 2022
- [11] "Smart Energy Efficiency: Application of IoT design platforms in LED street-lighting system," Aser D. Silas, Sibi Kashyap, Sindhya 2019 IEEE Advance Embedded Technology at Vishakhapatnam, Andhra Pradesh.

- [12] Ramnath, Sunil Duththa, Lokin Shinde, Fifth National Conference on Mobile computing and embedded system, Noida, 2020 “Energy and smart system technology on Street Light”
- [13] “Smart Energy Saving Technologies on Street Lighting System," by Moideen and Mohamadha Jumana, presented at the 7th International Conference on Advanced Information and Communication Systems (ICAICS), July 2021 in Coimbatore, India.

APPENDIX

A. Source Code

```
void read_ldr()
{
    unsigned int adc_value=0;
    adc_value=ADC_Read(0);
    light = 100 – adc_value/10.24;
    if(light>=80) // SWITCH of the light when light is 80 percent
    {
        PORTB.F1=0;
    }
    else
    {
        PORTB.F1=1;
    }
}

void main()
{
    TRISB=0X00;
    PORTB=0X00;
    Adc_Init();
    while (1)
    {
        read_ldr();
    }
}

int smooth;
int LDR;
int threshold = 40;//sun's intensity
int brightness = 0;
int ledState = 0;
```

```

int sensor1 = 11;
int sensor2 = 8;
int sensor3 = 9;
int led1=5;
int led = 6;
int led2=2;
int carPresent = 0;
int carPresent1 = 0;
float beta = 0.65;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  pinMode(sensor1, INPUT);
  pinMode(sensor2, INPUT);
  pinMode(sensor3, INPUT);
  pinMode(led,OUTPUT);
  pinMode(led1,OUTPUT);
  pinMode(led2,OUTPUT);
}
void loop() {
  smooth = smooth - (beta * (smooth - analogRead(A0)));
  delay(1);
  LDR = round(((float)smooth / 1023) * 100);
  if (LDR <= 40)
    brightness=0;
  else
  {
    brightness = map(LDR, 40, 100, 0, 255);
  }
  checkSensors();
  if (carPresent == 1)
  {
    ledState = 1;
    digitalWrite(led,HIGH);

```

```

digitalWrite(led1,HIGH);
analogWrite(led,brightness);
analogWrite(led1,brightness);
}
else if (carPresent == 0)
{
ledState = 0;
digitalWrite(led,HIGH);
//digitalWrite(led1,HIGH);
analogWrite(led,ledState);
//analogWrite(led1,ledState);
if(carPresent1 == 1)
{
ledState = 1;
if(ledState == 1)
{

analogWrite(led1,brightness);
analogWrite(led2,brightness);
}
}
else if (carPresent1 == 0)
{
ledState = 0;
digitalWrite(led1,HIGH);
digitalWrite(led2,HIGH);
analogWrite(led1,ledState);
analogWrite(led2,ledState);
}
}
String data = (String)ledState+","+ (String)brightness+"";
Serial.print(data);
// Serial.print(digitalRead(sensor1));
// Serial.print("\t");

```

```

// Serial.print(digitalRead(sensor2));
// Serial.print("\t");
// Serial.print(ledState);
// Serial.print("\t");
// Serial.println(brightness);
delay(100);
}
void checkSensors()
{
if (digitalRead(sensor1) == 0)//Car captured in 1st sensor
{
if (digitalRead(sensor2) == 1)//Car still didnt reach the 2nd sensor
carPresent = 1;
}
else if (digitalRead(sensor2) == 0)//Car reached the 2nd sensor
{ //No cars detected behind the first car
if (digitalRead(sensor1) == 1)
{
carPresent = 0;
carPresent1 = 1;
}
else if (digitalRead(sensor1) == 0 )
{
analogWrite(led,brightness);
analogWrite(led1,brightness);
analogWrite(led2,brightness);
digitalWrite(led,HIGH);
digitalWrite(led1,HIGH);
digitalWrite(led2,HIGH);
}
}
else if(digitalRead(sensor3) == 0)//car reached the 3rd sensor
{
//No cars detected behind the first car

```

```

if (digitalRead(sensor2) == 1)
{
carPresent = 0;

carPresent1 = 0;
}
else if (digitalRead(sensor2) == 0 )
{
carPresent = 0;
carPresent1 = 1;
}
}
}
int light;
void read_ldr()
{
unsigned int adc_value=0;
adc_value=ADC_Read(0);
light = 100 – adc_value/10.24;
if(light>=80) // SWITCH of the light when light is 80 percent
{
PORTB.F1=0;
}
else
{
PORTB.F1=1;
}
}
void main()
{
TRISB=0X00;
PORTB=0X00;
Adc_Init();

```

```

while (1)
{
  read_ldr();
}
}

int smooth;
int LDR;
int threshold = 40;//sun's intensity
int brightness = 0;
int ledState = 0;
int sensor1 = 11;
int sensor2 = 8;
int sensor3 = 9;
int led1=5;
int led = 6;
int led2=2;
int carPresent = 0;
int carPresent1 = 0;
float beta = 0.65;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  pinMode(sensor1, INPUT);
  pinMode(sensor2, INPUT);
  pinMode(sensor3, INPUT);
  pinMode(led,OUTPUT);
  pinMode(led1,OUTPUT);
  pinMode(led2,OUTPUT);
}
void loop() {
  smooth = smooth - (beta * (smooth - analogRead(A0)));
  delay(1);
  LDR = round(((float)smooth / 1023) * 100);
  if (LDR <= 40)

```

```

brightness=0;
else
{
brightness = map(LDR, 40, 100, 0, 255);
}
checkSensors();
if (carPresent == 1)
{
ledState = 1;
digitalWrite(led,HIGH);
digitalWrite(led1,HIGH);
analogWrite(led,brightness);
analogWrite(led1,brightness);
}
else if (carPresent == 0)
{
ledState = 0;
digitalWrite(led,HIGH);
//digitalWrite(led1,HIGH);
analogWrite(led,ledState);
//analogWrite(led1,ledState);
if(carPresent1 == 1)
{
ledState = 1;
if(ledState == 1)
{
analogWrite(led1,brightness);
analogWrite(led2,brightness);
}
}
else if (carPresent1 == 0)
{
ledState = 0;
digitalWrite(led1,HIGH);

```

```

digitalWrite(led2,HIGH);
analogWrite(led1,ledState);
analogWrite(led2,ledState);
}
}
String data = (String)ledState+","+ (String)brightness+"";
Serial.print(data);
// Serial.print(digitalRead(sensor1));
// Serial.print("\t");
// Serial.print(digitalRead(sensor2));
// Serial.print("\t");
// Serial.print(ledState);
// Serial.print("\t");
// Serial.println(brightness);
delay(100);
}
void checkSensors()
{
if (digitalRead(sensor1) == 0)//Car captured in 1st sensor
{
if (digitalRead(sensor2) == 1)//Car still didnt reach the 2nd sensor
carPresent = 1;
}
else if (digitalRead(sensor2) == 0)//Car reached the 2nd sensor
{ //No cars detected behind the first car
if (digitalRead(sensor1) == 1)
{
carPresent = 0;
carPresent1 = 1;
}
else if (digitalRead(sensor1) == 0 )
{
analogWrite(led,brightness);
analogWrite(led1,brightness);

```

```

analogWrite(led2,brightness);
digitalWrite(led,HIGH);
digitalWrite(led1,HIGH);
digitalWrite(led2,HIGH);
}
}
else if(digitalRead(sensor3) == 0)//car reached the 3rd sensor
{
//No cars detected behind the first car
if (digitalRead(sensor2) == 1)
{
carPresent = 0;
carPresent1 = 0;
}
else if (digitalRead(sensor2) == 0 )
{
carPresent = 0;
carPresent1 = 1;
}
}
}

```