DESIGN AND IMPLEMENTATION OF SMART STREET LIGHT AUTOMATION AND FAULT DETECTION SYSTEM

K. MAHESH (23RS5A0408)

K. SWETHA (23RS5A0406)

K. PAVANI (22RS1A0425)

B. PRESHITHA (22RS1A0406)

V. ASHISH (23RS5A0415)

Department of Electronics and Communication Engineering

Jawaharlal Nehru Technological University Hyderabad

University College of Engineering Rajanna Sircilla

Agraharam, Rajanna Sircilla Dist. – 505 302, Telangana

2025

DESIGN AND IMPLEMENTATION OF SMART STREET LIGHT AUTOMATION AND FAULT DETECTION SYSTEM

INDUSTRIAL ORIENTED MINI PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

BY

K. MAHESH (23RS5A0408)

K. SWETHA (23RS5A0406)

K. PAVANI (22RS1A0425)

B. PRESHITHA (22RS1A0406)

V. ASHISH (23RS5A0415)

Department of Electronics and Communication Engineering

Jawaharlal Nehru Technological University Hyderabad

University College of Engineering Rajanna Sircilla

Agraharam, Rajanna Sircilla Dist. – 505 302, Telangana

2025

Jawaharlal Nehru Technological University Hyderabad

University College of Engineering Rajanna Sircilla

Agraharam, Rajanna Sircilla Dist. – 505 302, Telangana

Department of Electronics and Communication Engineering

CERTIFICATE

Date: /06/2025

This is to certify that the project work entitled ***DESIGN AND IMPLEMENTATION A SMART STREET LIGHT AUTOMATION AND FAULT DETECTION SYSTEM*** *is a bonafide work carried out by* ***K. MAHESH, K. SWETHA, K. PAVANI, B. PRESHITHA*** *and* ***V. ASHISH*** *bearing Roll Nos.* ***23RS5A0408 22RS5A0406, 22RS1A0425, 22RS1A0406,*** *and* ***23RS5A0415*** *in partial fulfillment of the requirements for the degree of* ***BACHELOR OF TECHNOLOGY*** *in* ***ELECTRONICS & COMMUNICATION ENGINEERING*** *by the Jawaharlal Nehru Technological University Hyderabad during the academic year 2024-25.*

The results embodied in this report have not been submitted to any other University or Institution for the award of any degree or diploma.

--------------------------	---------------------------	--------------------------
Dr. Dhiraj Sunehra	**Mr. B. Ravi Kumar**	**Dr. Dhiraj Sunehra**
Professor	**Assistant Prof. (C)**	**Professor**
Project Guide	**Project Coordinator**	**Head of the Department I/c.**

ACKNOWLEDGEMENTS

The success in this project would not have been possible without timely help and guidance by many people. We wish to express our sincere gratitude to all those who have helped and guided us for the completion of the project.

It is our pleasure to thank to our Project Guide **Dr. Dhiraj Sunehra, Prof & Head, Department of ECE,** and Project Coordinator **Mr. B. Ravi Kumar, Assistant Professor (C), Department of ECE** for their guidance and suggestions throughout the project, without which we would not have been able to complete this project successfully.

We wish to express our sincere gratitude to **Prof. T. Venugopal, Principal, JNTUH UCER,** for his encouragement and providing facilities to accomplish our project successfully.

We would like to thank our classmates, all faculty members and non-teaching staff of ECE Department, JNTUH UCER for their direct and indirect help during the project work.

Finally, we wish to thank our family members and our friends for their interest and assistance that has enabled to complete the project work successfully.

ABSTRACT

Street light is considered to be one of the vital sources of light used by the people for various purposes. It is commonly used along walkways and streets especially when the environment is dark. In the conventional street light system, identification of the faulty bulb and energy saving by systematic switching on/off requires human intervention and takes lots of time.

The street lights remain in the ON condition with full brightness level throughout the night time even when there is less or no vehicle density. Due to which there is a lot of wastage of Energy.

To overcome this issue, a system was proposed which can identify faulty street lights (non-operating) automatically and send that information along with the location to the android application with the help of IoT technology.

In addition to that systematic switching on/off of street lights and progressive dimming of street lights based on vehicle's movement can be achieved with this which aids in energy saving.

CONTENTS

LIST OF FIGURES

Chapter 1
Introduction

1.1 Introduction

With the rapid growth of urbanization worldwide, cities face increasing challenges in managing resources effectively while striving for sustainability. Among these resources, energy consumption for street lighting stands out as a significant concern due to its substantial impact on both the environment and municipal budgets. Traditional street lighting systems often operate without considering real-time environmental conditions, leading to unnecessary energy wastage and operational inefficiencies. To address these challenges, there is a growing interest in implementing smart solutions for street lighting management. These solutions leverage advanced technologies such as Internet of Things

(IoT) to enhance the efficiency, reliability, and sustainability of urban infrastructure. By integrating sensors, microcontrollers, and connectivity platforms, smart street lighting systems can dynamically adjust lighting levels based on environmental factors and detect faults or malfunctions in real-time. This paper proposes a Smart Street Light Fault Detection System that harnesses IoT technology, NodeMCU microcontroller, Light Dependent Resistor (LDR), street light, and the ThingSpeak to create an intelligent and proactive approach to street lighting management. The system aims to optimize energy usage, improve operational efficiency, and enable timely detection and resolution of faults in street light installations. In this introduction, we will outline the motivations behind developing such a system, discuss the significance of smart street lighting in urban environments, and provide an overview of the proposed solution's architecture and functionalities. Additionally, we will highlight the potential benefits of deploying smart street lighting systems and the contributions of this research to the broader context of smart city initiatives and sustainable urban development.

1.2 Aim of the Project

To design and develop a smart street light system that operates automatically based on environmental and motion conditions, and detects faults in streetlights using IoT-based technology. The primary aim is to automate the control of street lighting based on environmental conditions such as ambient light and motion. The system also includes fault detection capabilities to improve maintenance efficiency.

1.3 Methodology

- Use of NodeMCU (ESP8266) as the main microcontroller.
- LDR sensors to detect ambient light levels.
- IR sensors for motion detection.
- Relay modules to switch streetlights ON/OFF.
- LEDs simulate actual street lights.
- IoT integration for fault detection and monitoring using Wi-Fi.

The methodology for implementing the Smart Street Light Fault Detection System involves several key steps. Firstly, the hardware components including the NodeMCU microcontroller, Light DependentResistor (LDR), and street light are assembled and connected appropriately. The NodeMCU is programmed using the Arduino IDE, integrating libraries for IoT communication and sensor interfacing. The software implementation includes real-time monitoring of ambient light levels using the LDR sensor, with the NodeMCU adjusting the street light output accordingly to optimize energy usage. Fault detection logic is incorporated, utilizing algorithms to detect anomalies such as constant light levels indicative of a fault in the system. Additionally, integration with the ThingSpeak facilitates remote monitoring and management, enabling users to receive fault notifications and take timely actions. Throughout the development process, rigorous testing and validation are conducted to ensure the reliability, functionality, and effectiveness of the system. This methodology encompasses both hardware setup and software development, culminating in a comprehensive solution for smart street lighting with fault detection capabilities .

1.4 Significance of the work

- Minimizes energy consumption by turning lights ON only when needed.
- Reduces human effort through automation.
- Enables proactive maintenance via fault alerts.
- Supports the smart city initiative.

A significant step toward sustainable and intelligent urban infrastructure. This system enhances energy efficiency by automatically controlling streetlights based on ambient light and motion, reducing electricity consumption and operational costs. Real-time fault detection improves maintenance efficiency by immediately identifying issues such as lamp failures or abnormal power usage, ensuring timely repairs and consistent lighting. Such automation not only contributes to environmental conservation by lowering carbon emissions but also enhances public safety by ensuring well-lit streets when needed. The system supports remote monitoring and control, making it ideal for integration into modern smart city frameworks. Moreover, it promotes the use of renewable energy sources like solar power, further boosting sustainability. Overall, this work addresses critical

challenges in urban lighting, combining technology with practical application to create a more reliable, efficient, and eco-friendly lighting system

1.5 Organization of the thesis

This thesis is systematically organized into several chapters to provide a clear and logical understanding of the design and implementation of the smart street light system with automatic control and fault detection. Chapter 1 presents a general introduction to the project, including the aim, objectives, methodology adopted, and the significance of the proposed system in addressing energy efficiency and urban automation. Chapter 2 describes the core processing unit of the system, the ESP8266 NodeMCU microcontroller. It explains its technical specifications, pin configuration, and its role in connecting various hardware components. Chapter 3 focuses on the hardware implementation, providing detailed information about the LDR sensor, IR sensor, LED lights, and other modules used, along with the complete circuit diagram. Chapter 4 discusses the software development aspect, including the programming logic, code structure, and system flowchart. Chapter 5 presents the testing procedures, observations, and system performance under different environmental conditions. Chapter 6 is dedicated to fault detection, explaining how the system identifies non-functional street lights. Finally, Chapter 7 concludes the thesis with a summary of the work completed, key results achieved, challenges encountered, and suggestions for future improvement. This organization ensures a comprehensive understanding of the entire project development process from conception to implementation.

CHAPTER 2

OVERVIEW OF NODEMCU

2.1 INTRODUCTION

The NodeMCU (Node MicroController Unit) is an open-source software and hardware development environment built around an inexpensive System-on-a-Chip (SoC) called the ESP8266. The ESP8266, designed and manufactured by Espressif Systems, contains the crucial elements of a computer: CPU, RAM, networking (WiFi), and even a modern operating system and SDK. That makes it an excellent choice for Internet of Things (IoT) projects of all kinds.

However, as a chip, the ESP8266 is also hard to access and use. You must solder wires, with the appropriate analog voltage, to its pins for the simplest tasks such as powering it on or sending a keystroke to the “computer” on the chip. You also have to program it in low-level machine instructions that can be interpreted by the chip hardware this level of integration is not a problem using the ESP8266 as an embedded controller chip in mass-produced electronics

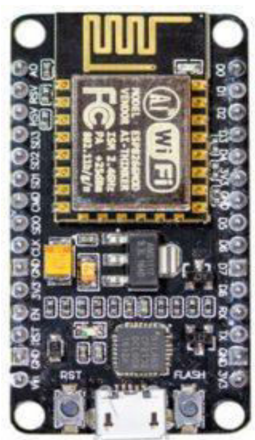

Fig 2.1: NODEMCU ESP8266

2.2 NODEMCU

2.2.1 FEATURES:

The NodeMCU ESP8266 is a low-cost, open-source IoT platform that integrates Wi-Fi capabilities and is designed for rapid prototyping. Here are its main features:

Wi-Fi Connectivity

- Built-in 802.11 b/g/n Wi-Fi module for easy internet access.
- Supports both Station (STA) and Access Point (AP) modes, as well as a combination of both (AP+STA).

Microcontroller

- Based on the ESP8266 chip with a 32-bit processor.
- Clock speed: 80 MHz (can be overclocked to 160 MHz).

Memory

- 4 MB Flash memory for storing code and data.
- 64 KB instruction RAM and 96 KB data RAM.

GPIO Pins

- 17 General Purpose Input/Output (GPIO) pins, programmable for various functions like PWM, I2C, UART, and SPI.

ADC (Analog to Digital Converter)

Single 10-bit ADC pin for reading analog signals (e.g., from sensors).

I/O Interfaces

- UART (Universal Asynchronous Receiver/Transmitter) for serial communication.
- I2C (Inter-Integrated Circuit) for connecting sensors and peripherals.
- SPI (Serial Peripheral Interface) for fast data transfer between devices.

PWM (Pulse Width Modulation)

Supports up to 4 PWM channels for controlling devices like LEDs or motors.

2.2.2 DESCRIPTION

The NodeMCU ESP8266 is an open-source IoT platform designed to facilitate rapid development and deployment of IoT projects. It combines a microcontroller and Wi-Fi module on a single board, making it ideal for smart devices that require internet connectivity. Below is a detailed description of its key aspects:

1. Hardware Overview

- Microcontroller: Powered by the ESP8266 chip, which includes a 32-bit RISC CPU operating at 80 MHz (extendable to 160 MHz).

- Wi-Fi Module: Built-in 802.11 b/g/n Wi-Fi with support for Station (STA), Access Point (AP), and combined AP+STA modes.

- Memory:

- 64 KB instruction RAM

- 96 KB data RAM

- 4 MB flash memory for program storage

- Power Supply: Operates at 3.3V; input voltage range via VIN pin is 4.5V to 10V.

2. GPIO and I/O Interfaces

- GPIO Pins:

- 17 General Purpose Input/Output pins for digital input and output.

- Can be configured for PWM, I2C, SPI, UART, and other functions.

- ADC Pin:

- A single 10-bit ADC for reading analog signals from sensors.

- UART: For asynchronous serial communication.

- I2C and SPI: For interfacing with sensors and peripherals.

3. Programming and Development

- Supported Languages:

- Lua scripting language

- Arduino C/C++ (compatible with the Arduino IDE)

4. Wi-Fi Features

- Supports WPA/WPA2 encryption for secure connections.
- Can serve as a web server or connect to cloud services.

5. Key Features

• Compact and lightweight, making it suitable for embedded systems and Low power consumption.

2.2.3 Pin Description

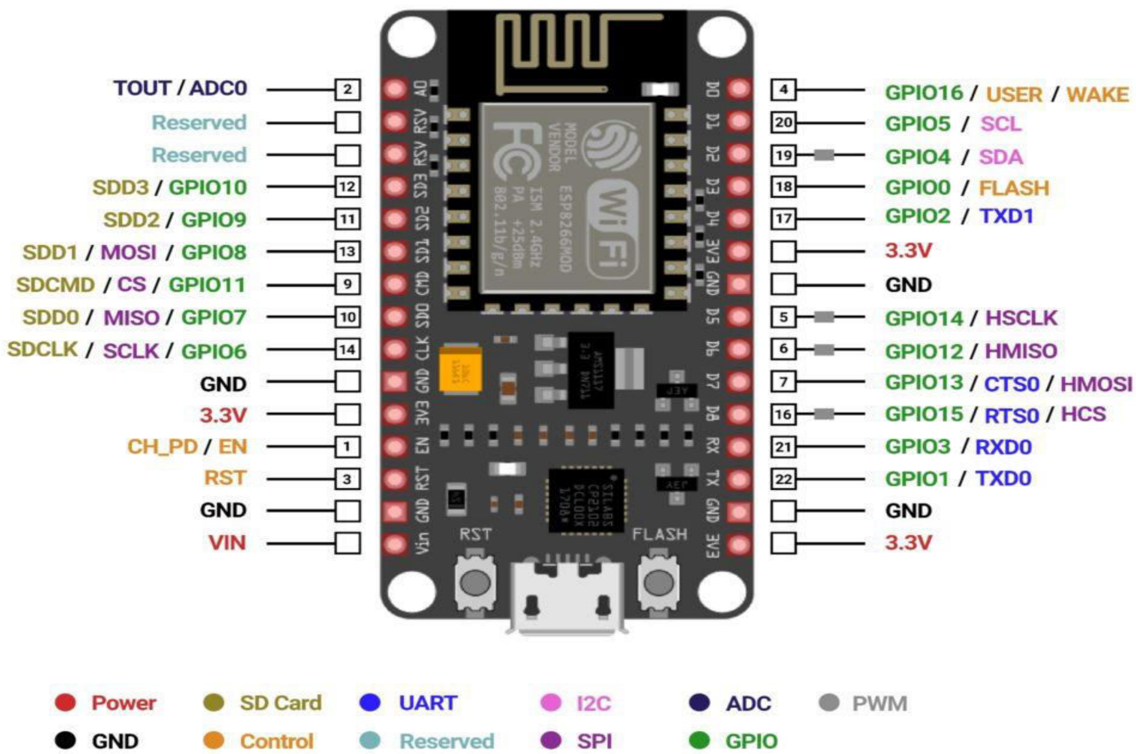

Fig 2.2 Pin Description of NodeMCU Esp8266

The NodeMCU ESP8266 has a total of 30 pins, each serving specific purposes. Below is a detailed description of the key pins and their functions:

1. Power Pins

- VIN: Input power pin (4.5V to 10V). It powers the board through an external power source.
- 3V3: Provides 3.3V output from the onboard voltage regulator. It can be used to power external components.
- GND (Ground): Connects to the ground of the power supply.

2. GPIO (General Purpose Input/Output) Pins

- GPIO Pins:
- The ESP8266 has 17 GPIO pins (D0–D8, SD2, SD3, RX, TX, etc.).
- These pins can be configured as digital input or output.

3. Analog Pin

- A0:
- The only analog pin on the NodeMCU.
- It supports a 10-bit ADC and can measure voltages between 0V and 3.3V.

4. UART (Universal Asynchronous Receiver/Transmitter) Pins

- TX (D10): Transmit pin for serial communication.
- RX (D9): Receive pin for serial communication.

5. I2C (Inter-Integrated Circuit) Pins

- SDA (D2): Data line for I2C communication.
- SCL (D1): Clock line for I2C communication.

6. SPI (Serial Peripheral Interface) Pins

- MOSI (D7): Master Out Slave In – data sent from the NodeMCU to the SPI device.
- MISO (D6): Master In Slave Out – data sent from the SPI device to the NodeMCU.
- SCLK (D5): Clock signal for SPI communication.

- SS (D8): Slave Select – used to select the SPI device.

7. PWM (Pulse Width Modulation) Pins

- GPIO Pins (D0–D8):
- These pins support PWM for controlling devices like LEDs or motors.

8. Control Pins

- EN (Enable): Enables or disables the chip. Pulling this pin HIGH enables the chip; pulling it LOW disables it.
- RST (Reset): Used to reset the board. Pulling it LOW will reset the NodeMCU.
- WAKE: Used to wake the NodeMCU from deep sleep mode.

9. Other Pins

- SD3, SD2, SD1, CMD: Used for SPI flash memory. Typically not used for general applications.
- D4 (BUILTIN_LED): Built-in LED connected to GPIO2.

2.3 Conclusions

In this chapter, the overview of NodeMCU board including of Introduction, Features, Description are explained. The architecture of NodeMCU Esp8266 board is discussed along with its pin configuration.

Chapter 3

Hardware Description

3.1 Introduction

The hardware includes sensors, LEDs, relay modules, and the NodeMCU for automation and control of the smart street lighting system.The hardware component is the backbone of the smart street light system with automatic control and fault detection. It integrates various sensors and modules to ensure efficient operation, energy conservation, and system reliability. The central controller, typically the ESP8266 NodeMCU, connects all the components and enables wireless data communication and automation. Key sensors include the LDR (Light Dependent Resistor) for ambient light detection, IR sensors for motion detection, and current sensors for fault monitoring. These inputs allow the system to intelligently switch lights on or off depending on environmental conditions and human presence, significantly reducing unnecessary power consumption. The LED lights serve as the primary illumination units, known for their low power usage and high brightness. All components are carefully interfaced using a structured circuit design that ensures proper voltage levels and current ratings. The hardware is designed for real-time operation, durability in outdoor conditions, and easy maintenance for long-term deployment in urban areas.

Hardware Required

1. LDR Sensor
2. LED Light
3. IR Sensor
4. Power supply unit
5. Relay

3.2 Circuit Diagram

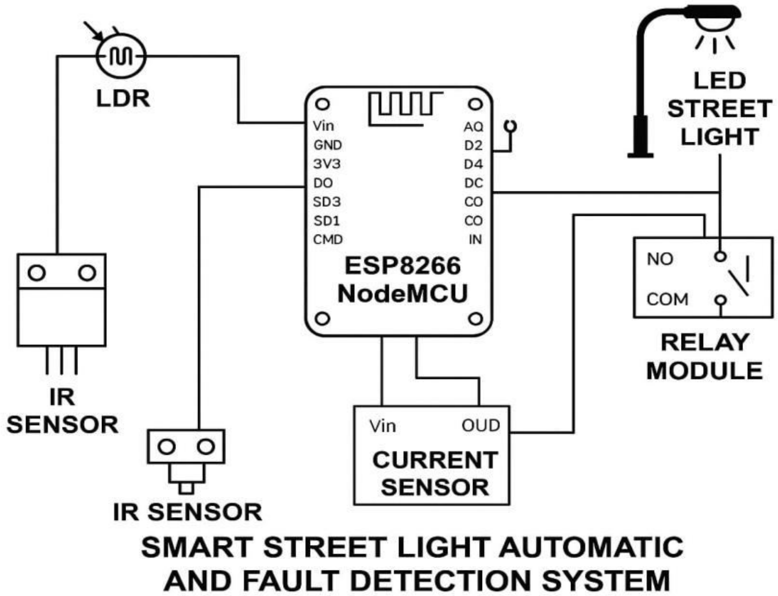

FIG: 3.1 CIRCUIT DIAGRAM

➢ 3.3 LDR SENSOR

3.3.1Introduction of LDR Sensor

A Light Dependent Resistor (LDR) is a sensor that changes resistance according to light intensity. It is used to detect day/night conditions. An LDR (Light Dependent Resistor), also known as a photoresistor, is a crucial sensor in smart street light systems. It is a type of variable resistor whose resistance decreases with increasing light intensity. In dark conditions, the LDR offers high resistance, whereas in bright conditions, its resistance drops significantly. This property allows the LDR to effectively sense the ambient lighting conditions. When integrated with a microcontroller like the ESP8266, the LDR helps in automating the switching of street lights. During daylight, the low resistance indicates sufficient natural light, prompting the system to turn off the street lights. At night or during low light conditions, the high resistance signals the controller to switch the lights on. This smart functionality contributes to energy saving and efficient operation. LDRs are passive, inexpensive, and reliable components, making them ideal for use in outdoor automation projects like smart street lighting systems where light intensity monitoring is essential.

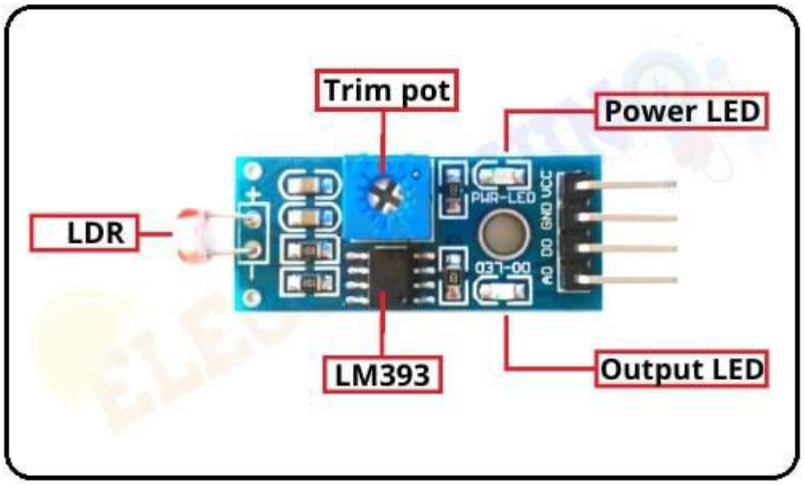

FIG: 3.2 LDR SENSOR

3.3.2Features

Light-Sensitive Resistance: Resistance decreases with increasing light intensity.

* Automatic Light Detection: Ideal for day/night sensing applications.

* Low Power Consumption: Consumes minimal current, suitable for energy-efficient systems.

* Cost-Effective: Inexpensive and easily available in the market.

* Easy to Interface: Simple to connect with microcontrollers like ESP8266, Arduino, etc.

* Analog Output: Provides analog signal that varies with light intensity.

No External Power Needed: Operates directly using the input voltage of the circuit.

* Wide Detection Range: Can sense a broad range of ambient light levels.

* Long Life Span: Passive component with durable performance over time.

* Maintenance-Free: No moving parts, ensuring stable and maintenance-free   operation.

* Temperature Dependent: Slight variation in performance with high temperature.

3.3.3Working principle

The Light Dependent Resistor (LDR) works on the principle of photoconductivity, where its resistance changes based on the intensity of light falling on it. In darkness or low light, the LDR exhibits high resistance, often in the megaohm range, restricting current flow. In bright light, its resistance drops significantly, allowing more current to pass through. This change in resistance can be measured as a varying voltage when connected in a voltage divider circuit. Microcontrollers use this analog voltage to determine ambient light levels and control devices accordingly, such as turning street lights on at night and off during the day.

- **Advantages:**
- Low Cost: Inexpensive and easily available in the market.
- Simple Design: Easy to use and integrate into circuits.
- Low Power Consumption: Ideal for energy-efficient systems.
- Analog Output: Provides smooth variation in signal for precise light detection.
- Easy to Interface: Compatible with microcontrollers like Arduino and ESP8266.
- Long Lifespan: Passive component with no moving parts, offering high durability.
- Effective Light Detection: Accurately detects ambient light changes for automation
- Compact Size: Small and lightweight, suitable for embedded applications.
-
- **3.3.4Pin Description**

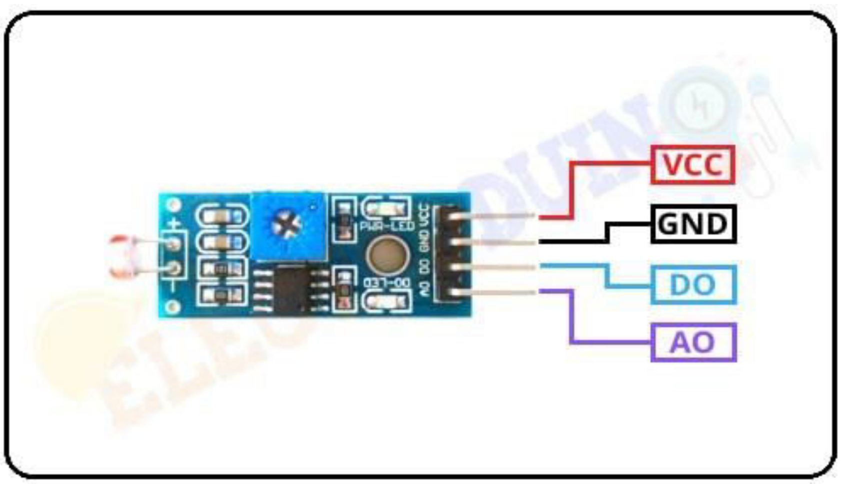

FIG: 3.3 LDR SENSOR PINS

Pin	Label	Description	Connection

1	VCC	Power supply pin for the module Usually operates at 3.3V to 5V.	Connect to 3.3V or 5V on the microcontroller.
2	GND	Ground pin. Completes the circuit and acts as the reference voltage.	Connect to the GND of the microcontroller.
3	OUT	Output pin. Sends digital or analog signal based on light intensity.	Connect to analog pin (A0) or digital pin on the microcontroller, depending on the module.

- **3.4 LIGHT EMITTING DIODE**

Fig :3.4 Light Emitting Diode

- **3.4.1 Introduction to Light Emitting Diode**

A Light Emitting Diode, or LED, is a semiconductor device that emits light when an electric current pass through it. LEDs are commonly used in various applications such as lighting, displays, and indicators. The technology behind LEDs has advanced rapidly, leading to energy-efficient and long-lasting lighting solutions.

When a voltage is applied to the semiconductor material of an LED, electrons and electron holes recombine, releasing energy in the form of photons. The color of the light emitted by an LED depends on the materials used in the semiconductor structure. LEDs are known for their efficiency in converting electrical energy into light, making them a popular choice for lighting applications.

3.4.2 Features:

- LEDs are capable of rapid on/off switching, allowing them to be used for high-speed data transmission in fiber optic communications.
- LEDs are highly energy-efficient, converting up to 50% of input energy into light compared to just 10% for incandescent bulbs.
- LEDs have a long lifespan, with some models rated for over 50,000 hours of use.
- LEDs are available in a wide range of colors, including red, green, blue, and white, by using different semiconductor materials.
- LEDs are compact and durable, making them suitable for use in a variety of applications from indicator lights to high-power illumination.

3.4.3 Working principle

The working principle of an LED (Light Emitting Diode) is based on the concept of electroluminescence, which occurs when current flows through a semiconductor material, causing it to emit light. An LED consists of a diode, typically made from materials such as gallium arsenide or gallium phosphide. When a voltage is applied across the LED, electrons in the semiconductor material are energized and jump from a high-energy state to a lower-energy state. This movement of electrons releases energy in the form of photons, which we perceive as light.

LEDs are designed to emit light in specific colors based on the energy bandgap of the semiconductor material used. When the LED is forward-biased (positive voltage on the anode and negative on the cathode), it allows current to pass through, triggering the light emission. The efficiency and longevity of LEDs make them ideal for a wide range of applications, including displays, lighting, and indicators.

- **Advantages:**
- Energy Efficient – LEDs consume very low power compared to incandescent and fluorescent lights.
- Long Lifespan – LEDs have a much longer operational life, often lasting over 50,000 hours.
- Low Heat Emission – LEDs produce very little heat, making them safer and more efficient.
- Fast Switching – LEDs turn on instantly without any warm-up time.
- Low Voltage Operation – They can operate on low voltage supplies, making them suitable for battery-powered devices.

- **3.4.4Pin Description**

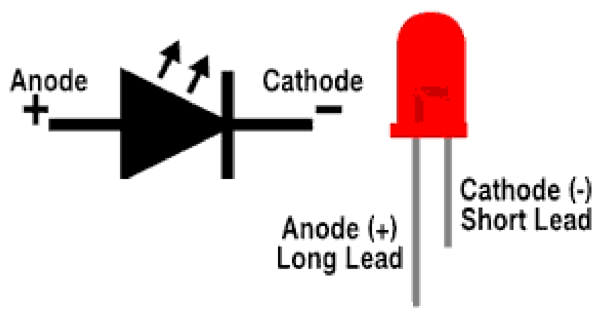

FIG: 3.5 LED PINS

A standard LED typically has two pins:

<u>1. Anode (Positive Pin):</u>

This is the longer leg of the LED.

It must be connected to the positive terminal of the power supply.

It allows the current to enter the LED.

<u>2. Cathode (Negative Pin):</u>

This is the shorter leg of the LED.

It must be connected to the negative terminal (ground) of the power supply.

It allows the current to exit the LED.

➢ 3.5 IR SENSOR

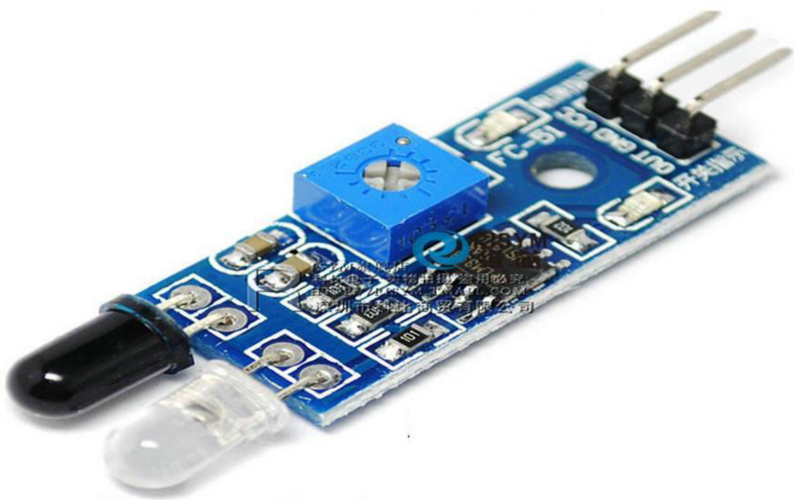

FIG:3.6 IR SENSOR

3.5.1 Introduction to IR Sensor

An Infrared (IR) sensor is an electronic device that measures and detects infrared radiation in its surroundings. IR radiation is a form of electromagnetic radiation with wavelengths longer than visible light but shorter than microwaves, typically in the range of 700 nm to 1 mm. IR sensors are widely used in automation, robotics, and electronics due to their ability to detect objects and measure heat or motion without direct contact.

IR sensors consist mainly of two components: an IR transmitter (usually an IR LED) and an IR receiver (such as a photodiode or phototransistor). The transmitter emits infrared light, which reflects off an object and is detected by the receiver. When an object comes in proximity, the reflected IR light intensity changes, and this change is captured by the sensor circuitry to trigger a response.

- There are two main types of IR sensors:
  Active and Passive. Active IR sensors both emit and detect infrared radiation, while passive IR sensors only detect infrared radiation emitted from objects, commonly used in motion

detectors. IR sensors are used in a variety of applications, including automatic doors, line-following robots, obstacle detection, flame detection, and even in medical instruments. Their low cost, energy efficiency, and versatility make them an essential component in many embedded systems and IoT-based projects.

3.5.2 Features:

- Detects infrared radiation from objects
- Can sense motion, distance, or obstacles
- Consists of an IR LED (transmitter) and a photodiode or phototransistor (receiver)
- Provides analog or digital output
- Compact and lightweight design
- Easy to interface with microcontrollers
- Low power consumption

3.5.3 Working principle:

The IR sensor works on the principle of emission and detection of infrared light. It consists of an IR LED (transmitter) that emits infrared light and a photodiode or phototransistor (receiver) that detects the reflected light from nearby objects. When there is no object in front of the sensor, the IR light does not reflect back to the receiver. However, when an object is present, the emitted IR light reflects off its surface and is detected by the receiver. The sensor then converts the intensity of the reflected light into an electrical signal, which can be processed to determine the presence, distance, or movement of the object.

Advantages:

1.Low cost – Affordable and widely available for various applications.

2. Fast response time – Quickly detects motion or objects.

3. Simple and compact – Easy to use and integrate into small electronic systems.

3.5.4 Pin Description

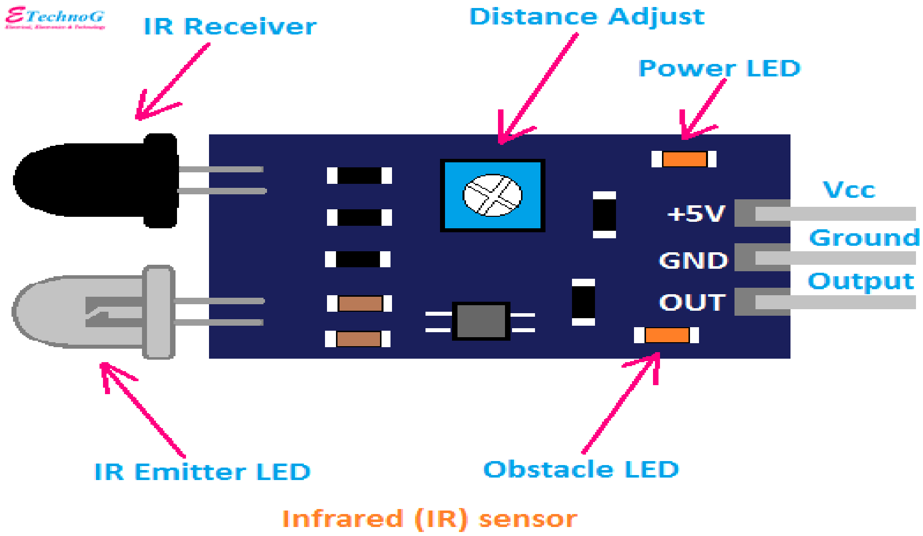

FIG:3.7 IR SENSOR PINS

A typical IR sensor module (used for obstacle detection or line tracking) usually has 3 pins. Here's a description of each:

1. VCC (Power Supply Pin):

This pin is used to supply power to the IR sensor module.

It is typically connected to +5V or +3.3V, depending on the module specifications.

2. GND (Ground Pin):

This is the ground or negative terminal.

It must be connected to the ground (GND) of the power supply or microcontroller.

3. OUT (Output Pin):

This pin gives the digital output signal from the IR sensor.

When the IR light reflects back from an object (i.e., an obstacle is detected), the sensor sends a LOW (0) or HIGH (1) signal, depending on the design.

It is connected to a microcontroller's digital input pin for reading sensor data

➢ 3.6 POWER SUPPLY UNIT

3.6.1 Introduction

FIG:3.8 POWER SUPPLY UNIT

A Power Supply Unit (PSU) is an essential component in any electronic system, responsible for providing the required electrical power to all the connected devices and components. It converts the incoming electrical energy from a source, such as an AC mains supply or a battery, into a suitable form—typically a stable DC voltage—required for the proper functioning of electronic circuits.

In embedded systems, like smart street lights or IoT-based devices, the power supply unit plays a crucial role in ensuring that microcontrollers, sensors, and actuators operate reliably and efficiently. Most electronic components work on low voltage DC (like 3.3V, 5V, or 12V), whereas the mains electricity available is AC (typically 220V or 110V). Therefore, a power supply unit includes steps such as transforming, rectifying, filtering, and regulating the input power to provide a consistent output.

There are different types of power supply units, such as linear and switching power supplies. For microcontroller-based projects, regulated power supplies using voltage regulators like LM7805 are commonly used to ensure steady 5V output.

A reliable PSU improves the overall durability and performance of the system, protecting it from voltage fluctuations and power surges, which is especially important in real-time or outdoor applications like smart lighting systems

3.6.2 Features

1. Voltage Regulation: Maintains a constant output voltage despite changes in input voltage or load conditions.

2. Current Limiting Protection: Prevents the circuit from drawing excessive current, protecting components from damage.

3. Overvoltage Protection (OVP): Shuts down or limits the voltage if it exceeds a safe threshold.

4. Short-Circuit Protection: Automatically cuts off the output if a short circuit is detected to prevent damage.

5. High Efficiency: Converts input power to output with minimal energy loss, especially in switching power supplies.

3.6.3 Working principle:

A Power Supply Unit (PSU) works by converting high-voltage AC (Alternating Current) from the mains into a low-voltage, stable DC (Direct Current) required by electronic components. The process begins with a step-down transformer that reduces the AC voltage to a lower level. This is followed by a rectifier circuit (typically using diodes) that converts AC to pulsating DC.

The output then passes through a filter circuit, often using capacitors, to smooth out the fluctuations and reduce ripple. Finally, a voltage regulator (like the LM7805) ensures a constant and regulated DC output, regardless of variations in load or input voltage.

This regulated DC power is then supplied to microcontrollers, sensors, and other modules in the system. The PSU may also include protection features like over-voltage, over-current, and short-circuit protection to safeguard the connected components and ensure reliable operation in various conditions

Advantages:

1. Stable Voltage Output:

Provides a consistent and regulated DC voltage, essential for the reliable operation of electronic devices.

2. Protects Components:

Includes safety features like over-voltage, over-current, and short-circuit protection to prevent damage to circuits.

3. Improves System Reliability:

Ensures smooth functioning of the entire system by preventing voltage fluctuations.

4. Energy Efficient:

Modern power supplies, especially switch-mode types, consume less power and generate less heat.

3.6.4 Pin Description

general pin description for such power supply units:

1. VCC (Input Voltage Pin)

Description: This pin is used to supply the input voltage to the power supply unit.

Typical Input: Often, this pin is connected to an external AC or DC source, depending on the design. For an AC-to-DC converter, it would be connected to the AC mains (like 110V/220V); for DC-to-DC, it would be connected to a higher DC voltage, such as 12V or 24V.

2. GND (Ground Pin)

Description: The ground pin is used to complete the circuit by providing a common reference point for the power supply. It is the negative terminal of the power supply.

Connection: This pin is connected to the system's ground (GND), which serves as a return path for the current.

3. VOUT (Output Voltage Pin)

Description: This pin provides the regulated output voltage to the connected circuit or load. It’s the final DC voltage that is used to power components like microcontrollers, sensors, etc.

Typical Output: This could be a regulated voltage like 5V, 12V, or 3.3V, depending on the PSU design

➢ 3.7 RELAY MODULE

3.7.1Introduction

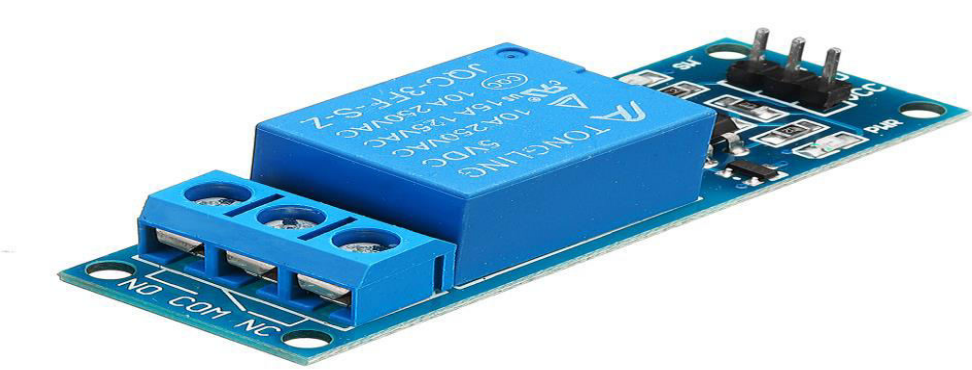

FIG: 3.9 RELAY MODULE

A relay is an electromechanical switch used to control a high-power circuit using a low-power signal. It operates by using an electromagnet to mechanically open or close electrical contacts. When a small current flows through the relay's coil, it generates a magnetic field that pulls a switch, allowing a larger current to flow through another circuit. Relays are widely used in automation, control systems, and electrical protection circuits. They are especially useful when isolating the low-voltage control side (like a microcontroller) from the high-voltage load side (like lights or motors). Common types include electromagnetic relays, solid-state relays, and time-delay relays. Their ability to control multiple devices from a single input makes them essential in smart systems and industrial applications.

3.7.2 Features

1. Electrical Isolation:

Separates control circuit (low voltage) from the load circuit (high voltage).

2. Low Power Control:

Can switch high-current devices using a small control current

3. High Load Capacity:

Capable of handling large voltages and currents, depending on the type.

4. Mechanical or Solid-State Operation:

Available in both electromechanical and solid-state versions.

3.7.3 Working principle:A relay is an electrically operated switch. Its fundamental working principle involves an electromagnet. When an electric current flows through the coil of the electromagnet, it generates a magnetic field. This magnetic field attracts a movable armature, which is typically spring-loaded. As the armature moves, it either closes or opens a set of electrical contacts.These contacts form a separate circuit, allowing the relay to control a high-power circuit using a low-power control signal. When the current to the electromagnet is removed, the magnetic field collapses, and the spring returns the armature to its

original position, thereby changing the state of the contacts again. This isolation between the control circuit and the switched circuit is a key advantage of relays.

Advantages:

1. Electrical Isolation:

Provides safe separation between the control circuit and the high-power load.

2. Low Power Operation:

Can control large electrical loads with a small input signal or current.

3. Cost-Effective:

Relays are inexpensive and widely available for various applications.

4. Versatile Usage:

Suitable for switching AC or DC loads, motors, lights, and other devices.

5. Multiple Contact Control:

A single relay can control multiple circuits using different contact points.

3.7.4 Pin Description

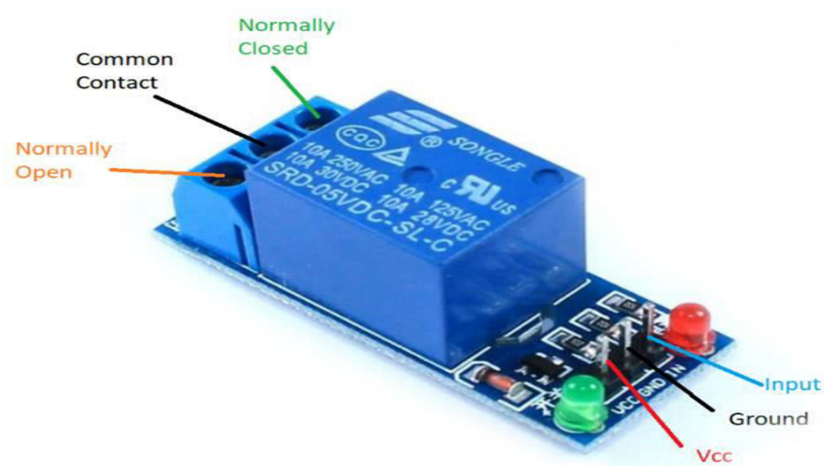

FIG:3.10 RELAY MODULE PINS

A typical electromechanical relay has several pins, each with a specific function. The main categories are:

* Coil Pins (e.g., 85 & 86 or A1 & A2): These are the input terminals for the relay's electromagnet. Applying a control voltage across these pins energizes the coil, creating a magnetic field that actuates the internal switch mechanism. Polarity might matter if an internal diode is present.

* Common Pin (COM, e.g., 30): This is the central switching point for the load circuit. It moves between the Normally Closed (NC) and Normally Open (NO) contacts.

* Normally Open (NO) Pin (e.g., 87): When the relay coil is de-energized, there is no connection between the Common and NO pins. When the coil is energized, the Common pin switches to connect with the NO pin, completing the circuit.

* Normally Closed (NC) Pin (e.g., 87a): When the relay coil is de-energized, the Common pin is connected to the NC pin, forming a closed circuit. When the coil is energized, this connection is broken as the Common switches to the NO pin.

These pin configurations allow relays to act as electrically controlled switches, isolating control circuits from higher power loads.

CHAPTER 4

SOFTWARE TOOLS

4.1 Introduction

The successful implementation of the soldier position tracking and Health monitoring system using NODEMCU relies on a synergy between hardware and software tools. These tools facilitate the programming, debugging, and management of the system components, ensuring seamless functionality and integration.

Key software tools include:

1. Arduino IDE:

- Used for programming the ESP8266 NodeMCU microcontroller.
- Offers an intuitive environment for writing and uploading code, enabling the control of sensors, motors, and displays.

2. Embedded C Programming Language:

- Provides efficient and compact code to manage real-time operations of the microcontroller.
- Essential for implementing the logic behind sensor readings, barrier control, and display updates.

4.2 Arduino IDE

The Arduino Integrated Development Environment (IDE) is an open-source software tool used for writing, compiling, and uploading code to Arduino-compatible microcontrollers, such as the ESP8266 NodeMCU utilized in this project. It serves as the primary interface for developers to interact with the hardware, facilitating easy and efficient programming. Arduino IDE interface is shown in Fig. 4.1.

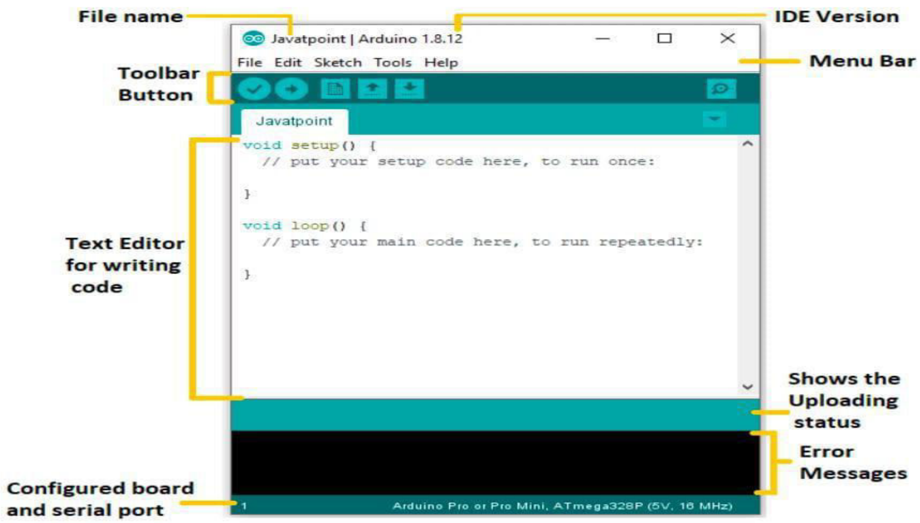

FIG:4.1 Arduino IDE

Steps to follow for Arduino programming IDE:

Step1: Choose a suitable Arduino board and connect it to PC using USB cable.

Step 2: Arduino IDE Software should be downloaded and installed from the website.

Step 3: Provide power supply. Most Arduino boards have USB as a power source. Adaptor can also be a choice. The power LED named PWR glows on the power supply.

Step 4: Start Arduino IDE. Open the installed software by double click.

Step 5: Initiate a project. It can be done in two ways:

1.A new project can be designed.

2.An existing project can be reviewed.

Step 6: Type of Arduino board needs to be selected. Any discrepancy can be eliminated while uploading code if a compatible board is chosen. Board can be chosen from the toolbar.

Step 7: Configuration of Serial port. See the port to which Arduino is connected and select that port like COM3.

Step 8: The program can now be uploaded to the board. The code is finally dumped and the result can be verified. Done uploading is seen on screen if upload is successful.

FIG:4.2 Arduino IDE Tool Bar

4.3 CODING {PROJECT CODE}

```
#include <ESP8266WiFi.h>
#include <Wire.h>
 #include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
#define pir D3
#define pump D7
const char* ssid     = "project"; // Your ssid
const char* password = "project1"; // Your Password
char status;
double T,P,p0,a;
int e=0;
int b=0;
int c=0;
int d=0;
WiFiServer server(80)
void setup() {
Serial.begin(9600);
pinMode(D7, OUTPUT);
pinMode(D6, OUTPUT);
  pinMode(D0, INPUT);
  pinMode(D8, INPUT);
  pinMode(D5, INPUT);
  digitalWrite(D7, HIGH);
  digitalWrite(D6, HIGH);
lcd.init(); // initialize the lcd
lcd.backlight();
lcd.setCursor(0,0);
```

```
 lcd.print("IoT based Smart");
  lcd.setCursor(0,1);
  lcd.print("Street light   ");
  delay(2000);
Serial.print("Connecting to ");
Serial.println(ssid);
WiFi.begin(ssid, password);
//lcd.begin(16, 2);
 while (WiFi.status() != WL_CONNECTED)
{
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WiFi is connected");
server.begin();
Serial.println("Server started");
lcd.setCursor(0,0);
lcd.print("IP Address          ");
lcd.setCursor(0,1);
lcd.print(WiFi.localIP());
Serial.println(WiFi.localIP());
delay(4000);
lcd.clear();
}
void loop() {
WiFiClient client = server.available();
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println("Connection: close");  // the connection will be closed after completion of the
response
client.println("Refresh: 10");  // update the page after 10 sec
client.println();
client.println("<!DOCTYPE HTML>");
client.println("<html>");
client.println("<style>html { font-family: Cairo; display: block; margin: 0px auto; text-align:
center;color: #FFFFFF; background-color: #0066FF;}");
client.println("body{margin-top: 50px;}");
client.println("h1 {margin: 50px auto 30px; font-size: 50px; text-align: center;}");
client.println(".side_adjust{display: inline-block;vertical-align: middle;position: relative;}");
```

```
client.println(".text1{font-weight: 180; padding-left: 15px; font-size: 50px; width: 170px;
text-align: left; color: #FFFFFF;}");
client.println(".data1{font-weight: 180; padding-left: 80px; font-size: 50px;color:
#FFFFFF;}");
client.println(".text2{font-weight: 180; font-size: 50px; width: 170px; text-align: left; color:
#FFFFFF;}");
client.println(".data2{font-weight: 180; padding-left: 150px; font-size: 50px;color:
#FFFFFF;}");
client.println(".text3{font-weight: 180; font-size: 50px; width: 170px; text-align: left; color:
#FFFFFF;}");
client.println(".data3{font-weight: 180; padding-left: 150px; font-size: 50px;color:
#FFFFFF;}");
client.println(".text4{font-weight: 180; font-size: 50px; width: 170px; text-align: left; color:
#FFFFFF;}");
client.println(".data4{font-weight: 180; padding-left: 150px; font-size: 50px;color:
#FFFFFF;}");
client.println(".text5{font-weight: 180; font-size: 50px; width: 170px; text-align: left; color:
#FFFFFF;}");
client.println(".data5{font-weight: 180; padding-left: 150px; font-size: 50px;color:
#FFFFFF;}");
client.println(".data{padding: 10px;}");
client.println("</style>");
client.println("</head>");
client.println("<body>");
client.println("<div id=\"webpage\">");
client.println("<h1>IoT based Smart Street Light and Fault Monitoring System</h1>");
if(digitalRead(D0)==LOW)
  {
   lcd.setCursor(0,0);
  lcd.print("Day time...     ");
   digitalWrite(D7, HIGH);
  digitalWrite(D6, HIGH);
 client.println("<div class=\"data\">");
client.println("<div class=\"side_adjust text1\">Status:Day</div>");
client.println("<div class=\"side_adjust data1\">");
client.println("</div>");
client.println("<div class=\"data\">");
delay(500);
  }
  if(digitalRead(D0)!=LOW)
```

```
 {
   lcd.setCursor(0,0);
  lcd.print("Night time...     ");
 client.println("<div class=\"data\">");
client.println("<div class=\"side_adjust text1\">Status:Night</div>");
client.println("<div class=\"side_adjust data1\">");
client.println("</div>");
client.println("<div class=\"data\">");
delay(500);
  }
if(digitalRead(D8)==LOW)
  {
   lcd.setCursor(0,1);
  lcd.print("No fault          ");
 client.println("<div class=\"data\">");
client.println("<div class=\"side_adjust text2\">No_Fault</div>");
client.println("<div class=\"side_adjust data2\">");
client.println("</div>");
client.println("<div class=\"data\">");
delay(500);
  }
 if(digitalRead(D8)!=LOW)
  {
   lcd.setCursor(0,1);
  lcd.print("Fault detected    ");
 client.println("<div class=\"data\">");
client.println("<div class=\"side_adjust text2\">Fault_detected</div>");
client.println("<div class=\"side_adjust data2\">");
client.println("</div>");
client.println("<div class=\"data\">");
delay(500);
  }
 if(digitalRead(D0)!=LOW && digitalRead(D5)!=LOW)
  {
  digitalWrite(D6, LOW);
  digitalWrite(D7, HIGH);
   client.println("<div class=\"data\">");
client.println("<div class=\"side_adjust text3\">No_vehicles_near_street_light</div>");
client.println("<div class=\"side_adjust data3\">");
client.println("</div>");
```

```
client.println("<div class=\"data\">");
delay(500);
  }
 if(digitalRead(D0)!=LOW && digitalRead(D5)==LOW)
  {
  digitalWrite(D6, LOW);
  digitalWrite(D7, LOW);
   client.println("<div class=\"data\">");
client.println("<div class=\"side_adjust text3\">Vehicles_passing_near_street_light</div>");
client.println("<div class=\"side_adjust data3\">");
client.println("</div>");
client.println("<div class=\"data\">");
delay(500);
}
client.println("</div>");
client.println("<div class=\"data\">");
client.println("</body>");
client.println("</html>");
}
```

CHAPTER 5
RESULTS AND DISCUSSION

5.1 Introduction

In this chapter we discuss the project outputs and the sequential process of the project, i.e., schematic diagram of setup, explanation of experimental setup, and results of the project.

5.2 Schematic Diagram

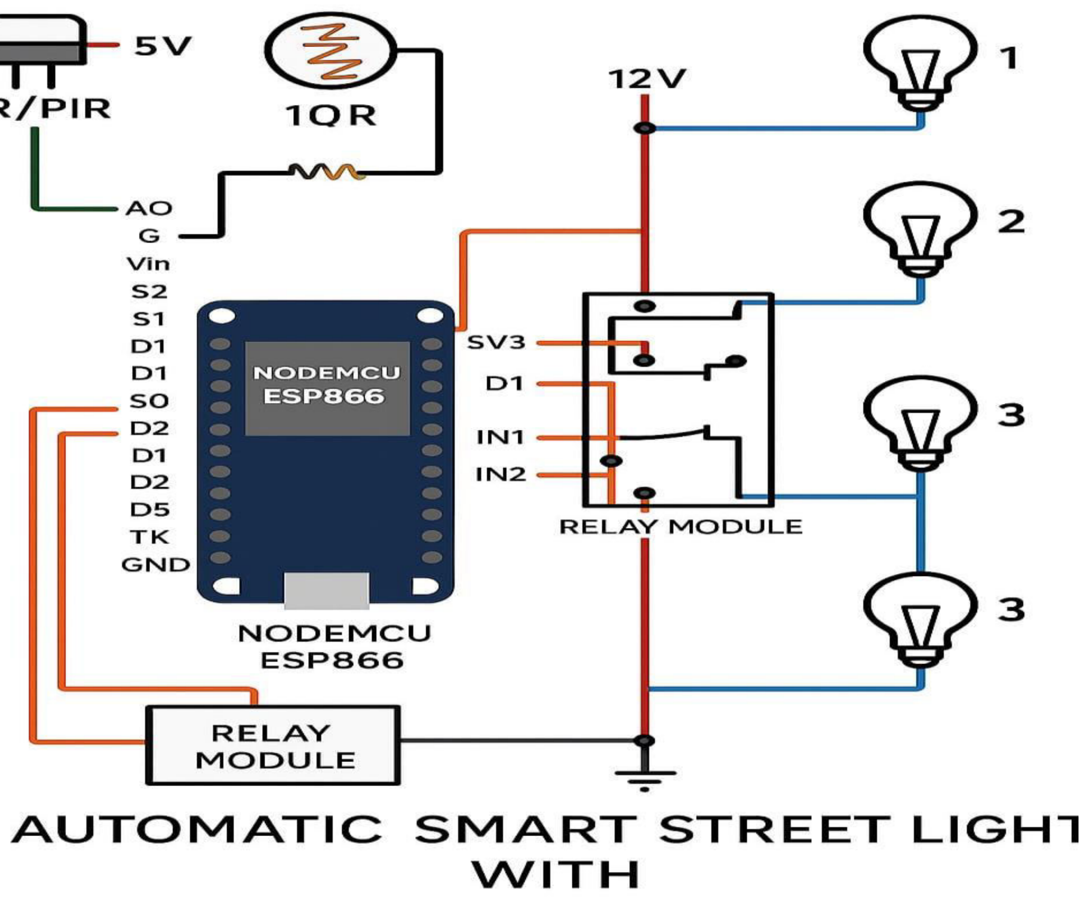

FIG:5.1 SCHEMATIC DIAGRAM

5.3 Experimental Setup

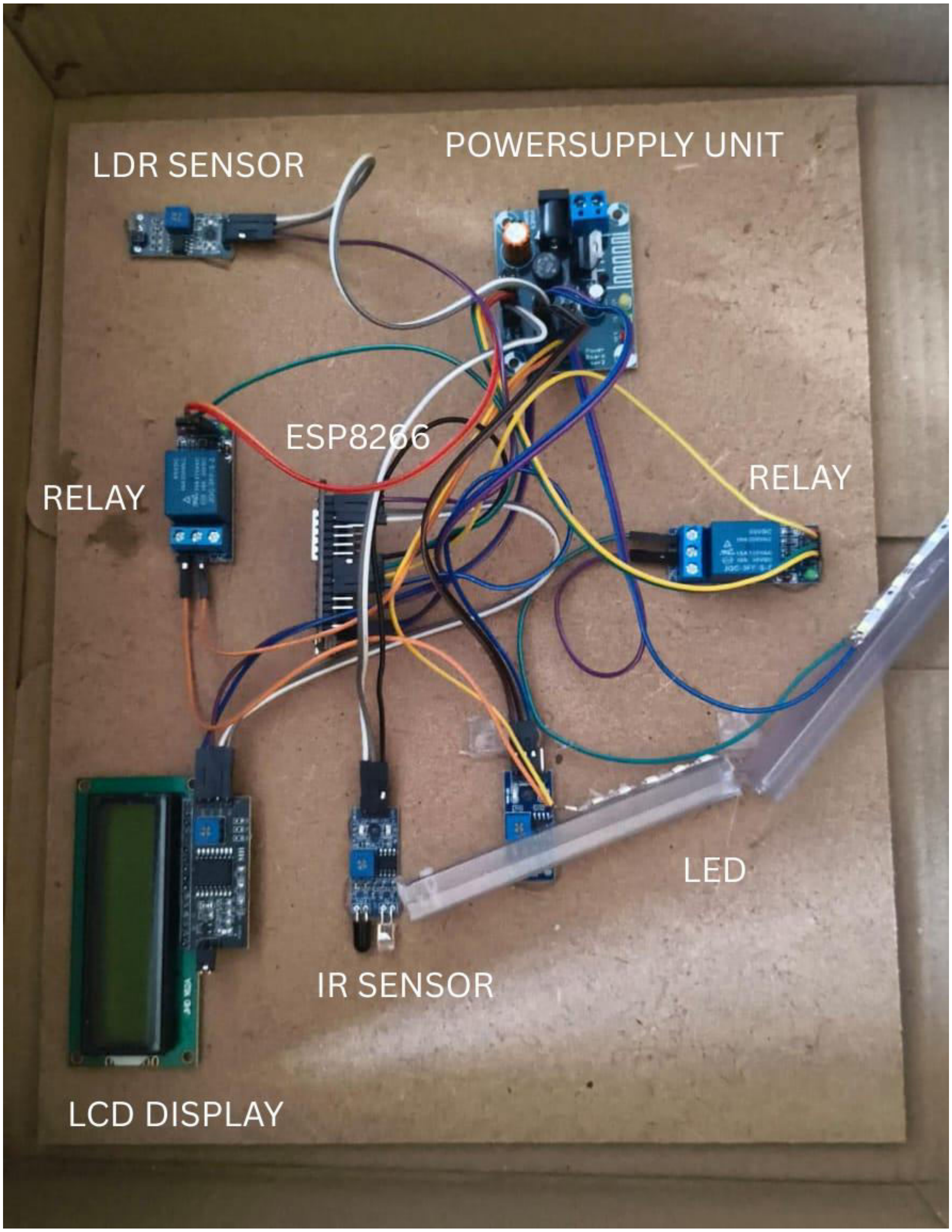

Fig 5.2 Experimental Setup

- **5.4 FLOW CHART**

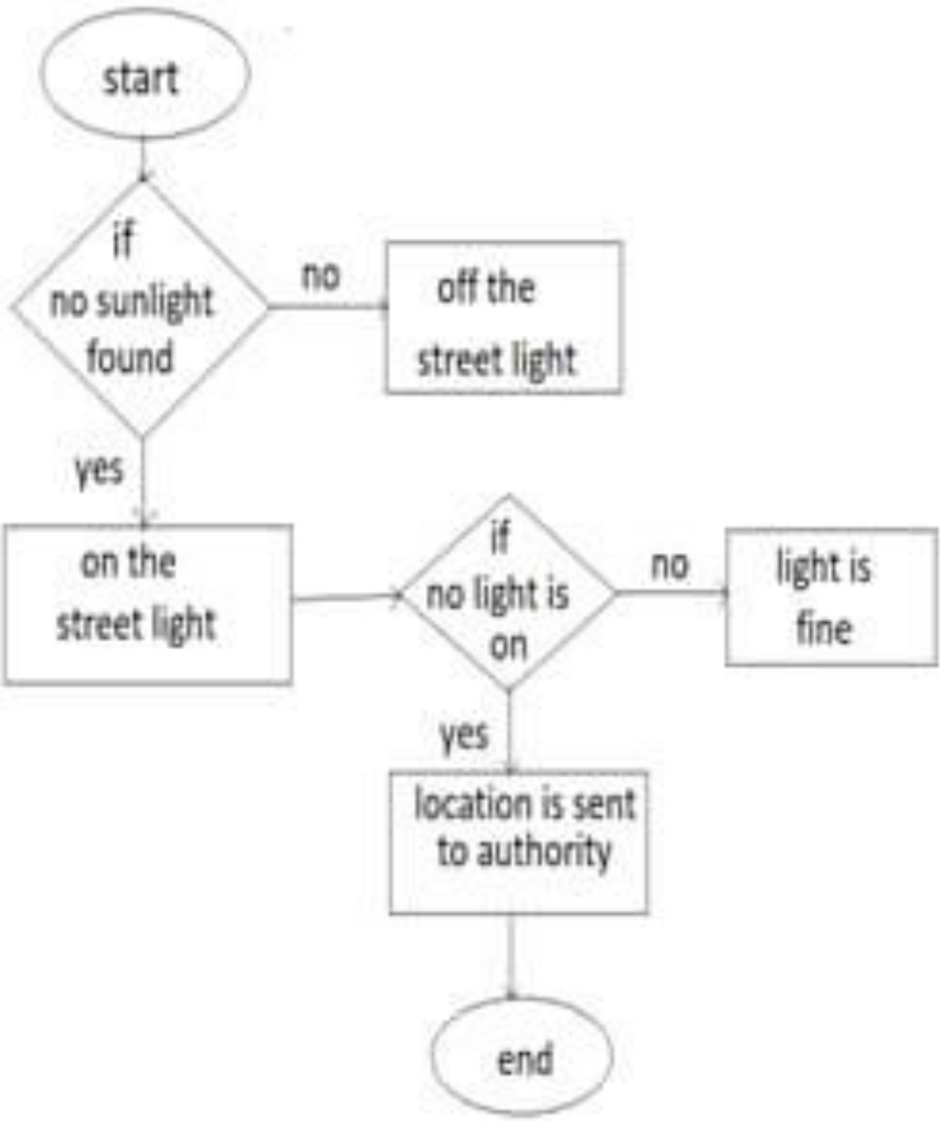

FIG:5.3 FLOW CHART

5.5 Results

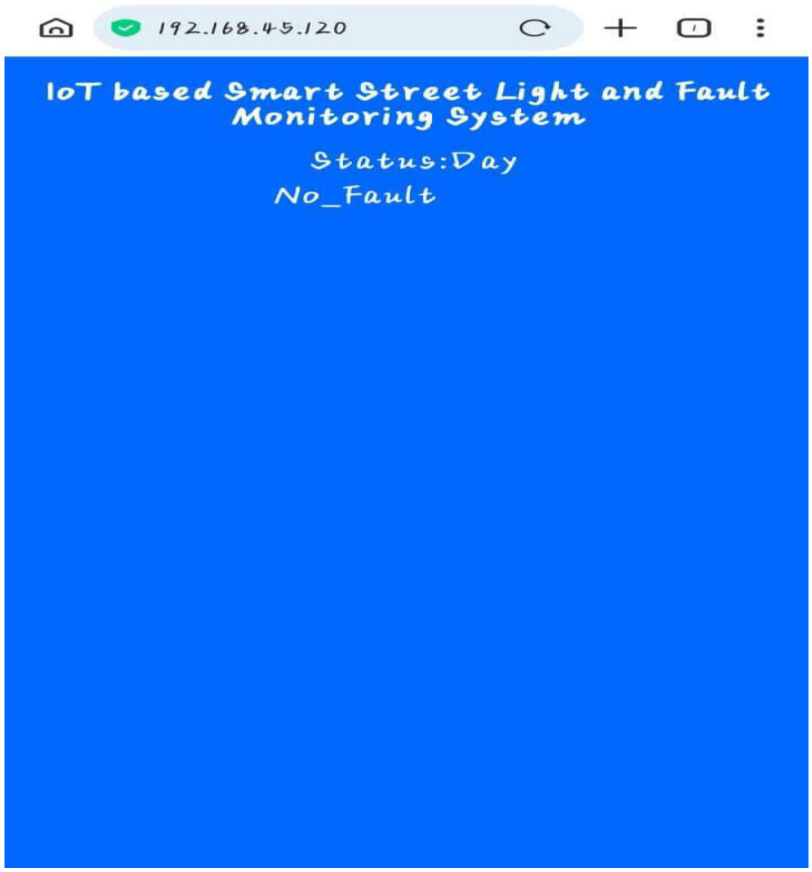

FIG:5.4 OUTPUT ON WEBPAGE

This NodeMCU-based system automates street lights using LDRs for ambient light detection. Current sensors identify faults like bulb failures or wiring issues. A local web page, hosted by NodeMCU's Wi-Fi, provides real-time light status, power consumption, and fault alerts, enabling remote monitoring and efficient maintenance, saving energy and improving safety

A smart street light system using NodeMCU allows remote control and monitoring via a web page. It integrates fault detection capabilities to identify malfunctioning lights, enabling quick maintenance and optimizing energy consumption.Track the Real time Data.

5.6 Conclusion

In this chapter we have seen the schematic diagram, experimental setup of the project and their results of various operations

CHAPTER 6

CONCLUSION

The smart street light system developed in this project successfully demonstrates an energy-efficient and intelligent lighting solution for modern urban infrastructure. By integrating sensors like the LDR and IR with the ESP8266 microcontroller, the system is capable of automatically switching street lights ON during low-light conditions and OFF during daylight, thereby reducing unnecessary power consumption. Additionally, the inclusion of motion detection ensures that lights activate only when vehicles or pedestrians are present, further conserving energy.

A significant feature of the project is the fault detection mechanism, which allows the system to monitor the working status of individual lights and identify any malfunctions. This feature can greatly assist in timely maintenance, reducing downtime and enhancing public safety.

The implementation proves to be cost-effective, reliable, and scalable, making it suitable for real-world deployment in streets, highways, and public areas. The use of IoT-compatible components also opens the possibility of remote monitoring and control in future enhancements. Overall, this project contributes toward the development of smart cities by promoting automation, energy saving, and maintenance efficiency.

This project successfully demonstrates a practical approach to solving real-world problems such as excessive power usage and delayed maintenance in street lighting. The automatic control feature ensures that street lights operate only when necessary, while the fault detection mechanism improves system reliability and safety. The system was built using low-cost components, making it highly affordable for rural and urban municipalities. With further enhancements like wireless monitoring or solar integration, this system can become an essential part of modern smart infrastructure.

REFERENCES

- S. K. Sharma, Introduction to Embedded Systems, McGraw-Hill Education, 2nd Edition, 2017.
- M. A. Mazidi, The 8051 Microcontroller and Embedded Systems, Pearson Education, 2nd Edition
- Circuit Diagram Tool – EasyEDA, Used for designing the project schematic. Available at: https://easyeda.com/
- Smart Street Lighting System Based on IoT, International Journal of Engineering Research & Technology (IJERT), Vol. 9 Issue 06, 2020.
- 6. Street Light Monitoring and Fault Detection using IoT, International Research Journal of Engineering and Technology (IRJET), Vol. 7, Issue 4, April 2020.