Automatic Street Light System

By

Group 5

Group Members

MICHAEL ETOROBONG IBIA

2024/13496

ADEFALA AMINAT TENIOLA

2024/13124

FADIPE OLUWATOMISONA EZRA

2024/13089

SOYEWO OLAOLUWA DANIEL

2024/13102

OLALEYE SAMUEL OLAKUNLE

2024/13108

ICT 215

Robotics and Embedded Systems

SUBMITTED TO:

AYUBA MUHAMMAD

Declaration

We Group 5, Declare that project titled AUTOMATIC STREET LIGHT SYSTEM is our original work and has been submitted in partial fulfillment of requirement of the course in information and communication technology (ICT), All sources of information has been duly acknowledged through proper referencing.

We further declare that this work has not been submitted elsewhere for any other degree, diploma or certification, either in part or in full,

This project represents collaborative effort by all team members, with each member contributing equally to the research, design, development, testing and documentation phase of the Project

Abstract

This project outlines the design and implementation of an automatic street light system that uses an Arduino-based microcontroller, a solar panel, a Light Dependent Resistor (LDR) sensor, and a Passive Infrared (PIR) motion sensor. The system turns street lights ON in low-light conditions and OFF when there is enough ambient light. The PIR sensor helps save energy by only activating full brightness when it detects motion at night. The entire circuit was designed and simulated in Proteus, modeled in Tinkercad for 3D visualization, and implemented on a custom printed circuit board (PCB) designed with KiCad. Performance tests on the prototype showed reliable switching at the chosen light threshold, response times of about 3 to 4 seconds, and consistent operation through repeated ON and OFF cycles. The results indicate that this system is a low-cost and effective solution for smart street lighting and could be enhanced in future work with more sensing or IoT-based monitoring features.

Table Of Contents

Chapter 1: Introduction

1.1 BACKGROUND OF THE STUDY

Street lights are crucial in the modern world because of issues like bandits that are currently going on in the country. Street lights are very essential in terms of providing a safe environment, reducing road accidents, and enabling social and economic activities during nighttime. Traditionally, street lights are manually operated and programmed to switch on and off at fixed times and a few street lights are activated by detecting the footsteps of a person. While this method has been widely used, it often leads to unnecessary energy consumption, increased operational costs, and inefficient lighting control.

With the advancement of technology, we can use automated systems using embedded systems, robotics, and sensor technology, automation has become a practical solution to the present challenges. An Automatic Street Light System uses light-dependent sensors and microcontroller-based control to automatically manage street lighting based on environmental conditions. Therefore, it only operates when necessary, allowing conservation of energy and improvement of system efficiency.

Involving the application of Arduino microcontrollers and LDRs, it is possible to make intelligent decisions independently of human control. In the context of a smart city, energy efficiency, sustainability, and automation form the prime areas of priority. This project is all about designing, simulating, and building a setup like this, right from the basics of embedded systems.

1.2 PROBLEM STATEMENT

Many street lights lack efficient management in respect to their energy usage, although these types of installations are important for improving public safety. Typical operations of many street lighting systems are based on the assumption, human error, malfunction of timers and lack of ability to sense the ambient light conditions; as a result, street lights will usually stay on during daylight. This leads to energy wastage and high electricity bills, as well as high maintenance costs.

Despite the importance of street lighting for public safety, many existing systems have poor energy management. Typical operations of many street lighting systems are based on the assumption, human error, malfunction of timers and lack of ability to sense the ambient light conditions; as a result, street lights will usually stay on during daylight. This leads to unnecessary energy wastage, increased electricity costs, and higher maintenance cost.

Most of the world's developing areas are still using old-fashioned ways of controlling their street lighting using manual methods. The outcome is that they have inefficient, poor and unreliable systems in place. There is, however, an increasing demand for an intelligent solution to street lighting, that operates autonomously which means that it will respond accurately and promptly to changes in the level of light in the environment.

1.3 OBJECTIVES OF THE STUDY

General Objective

It's used to design, simulate, and implement an Automatic Street Light System that intelligently controls street lighting using sensors and a microcontroller.

Specific Objectives

1. To design and simulate the automatic street light system using Proteus and Tinkercad.

2. To develop an analog sensor circuit using a Solar Panel for light detection.

3. To program a microcontroller (Arduino) for automatic light control.

4. To design a professional PCB layout for the system.

5. To build and test a functional prototype of the system.

6. To document the project and publish all resources on GitHub.

1.4 RESEARCH QUESTIONS

1. How can ambient light be effectively detected using sensors?

2. An automatic street light control system saves energy over manual control systems by how much?

3. What kinds of light sensors can be appropriately used for automatic street lighting?

4. How does the weather affect the sensors and deactivating the system's accuracy

5. What kind of problems are experienced by designers and developers while building the project

1.5 SIGNIFICANCE OF STUDY

The proposed work is an addition to the ever-growing domains of intelligent and sustainable infrastructure. The Automatic Street Light System reduces energy wastage, as it turns on illuminated light when required, decreases operational costs, and reduces labor interaction. It can be taken up as a project for students where they implement ideas regarding Embedded Systems, Interfacing Sensors, Designing Analog Circuits, and Microcontroller Programming.

1.6 SCOPE AND LIMITATIONS

The scope of this project covers the design, simulation, and prototype development of an Automatic Street Light System using an LDR sensor and Arduino microcontroller. The system is designed to switch street lights ON and OFF based solely on ambient light intensity.

The limitations include dependency on sensor accuracy, limited scalability in the prototype stage, and the absence of advanced features such as motion detection or wireless monitoring, which are beyond the scope of this project.

1.7 ORGANIZATION OF THE STUDY

Here is how the report is broken down:

I. **Chapter One** presents the introduction to the project.

II. **Chapter Two:** reviews existing literature related to automatic lighting systems.

III. **Chapter Three:** explains the methodology, including both hardware and software design.

IV. **Chapter Four :** Results and Findings are discussed in this chapt**er.**

V. **Chapter Five** presents both the conclusion and recommendations for future projects.

VI.

Chapter 2: Literature Review

2.1 Introduction

Street lights play an important role in safety and crime prevention. It lets folks feel comfortable outside after dark. But there's a catch: a lot of street lights just run all night, whether anyone's there or not. The International Energy Agency says that in 2023, public street lighting sucked up anywhere from 2.5% to 19% of all electricity in developing countries. That's a big chunk.

Leaving lights on when no one needs them burns through energy, racks up maintenance bills, and makes light pollution worse. Upgrading to smart, automated lighting isn't just about fancy tech—it's a must if cities want to get smarter and hit those Sustainable Development Goals, especially numbers 7 and 11.

2.2 Review of Existing Similar Systems (10+ Key References)

Recently, researchers have undertaken an overhaul of streetlights, taking into account not only the desire for energy and cost-saving measures but also developing products that work in conjunction with the surrounding area. Using photodiodes or photo-resistors (LDR's) to detect the amount of day and night light, and passive infra-red detectors (PIR's) to identify movement, these lights will provide enhanced energy efficiency by only turning on when there is vehicle or foot traffic to illuminate. Since 2020, over ten major projects have popped up, each pushing things a bit further:

I. Kumar and his team (2021) paired up LDR and PIR sensors on Arduino and managed to cut city energy use by 60%.

II. Rahman (2022) built a solar setup with MPPT, reaching 70% efficiency—perfect for places where the grid isn't reliable.

III. Chowdhury (2022) combined solar LEDs with IoT and GSM, so you can spot faults from anywhere.

IV. Ali (2024) brought LoRaWAN into the mix for long-range communication, using both PIR and LDR sensors.

V. El-Sayed (2023) got lights to dim automatically, and with solar and batteries, they slashed energy use by 80%.

VI. Wang (2024) added machine learning, so solar street lights adjust themselves based on traffic.

VII. Zhang (2024) designed an IoT solar system that lets you watch everything live through ThingSpeak.

VIII. Khan (2023) mixed solar with piezoelectric tech, squeezing extra power out of busy areas.

IX. Putri (2025) built an IoT system focused on catching faults early, before they snowball.

X. Mimouni (2025) launched an autonomous PV system with IoT for nonstop monitoring.

XI. Dewangan (2024) used solar and AI together to dial in power use.

Step back for a second. Not long ago, we had basic LDR setups. Now? We've got solar-IoT hybrids with all sorts of smart tricks. Still, there's a catch—most of these projects stay stuck as city prototypes. Scaling up is tough, and rural communities end up missing out.

2.3 Analogue Circuit Techniques in Robotics and Sensor Interfacing

Analogue circuits remain foundational in sensor interfacing for automatic lighting systems, similar to robotics applications (e.g., signal conditioning for sensors). Key techniques include:

I. Op-Amps: Amplifies with a filter, passing low-level LDR/PIR signals through to the output (the analogue voltage, for instance) as a follower (to match the impedance).

II. Comparators: Provides a threshold to detect when a change occurs in resistance of the LDR, and to provide digital, on/off signals in response.

III. Signal Conditioning: Low-pass filters to reduce noise from environmental factors (dust/weather) and instrumentation amplifiers for differential sensor signals.

IV. Power Management: Analogue circuits for battery charging (e.g., in solar systems) with MPPT algorithms implemented via op-amps.

These techniques ensure reliable sensor data in noisy environments, as seen in robotics (e.g., motion detection circuits) and adapted here for street lighting.

2.4 Microcontroller Selection and Justification

2.4.1 Microcontroller Selection

Picking the right microcontroller really comes down to what you want to do. There are tons out there, but let’s focus on Arduino since it's everywhere, and the lineup covers just about any need.

1. Arduino

I. The classic is the Arduino UNO. Everyone's used it at some point—prototyping, school projects, quick automation fixes. It's 8-bit, dead simple to set up, and just works. No surprise it's the default choice for most people.

II. Next, there's the Arduino Pro. Think of it as a more polished UNO. It skips the pre-soldered I/O headers, so you get extra room on your board for other parts. That means your whole project can stay neat and compact. Plus, you can pick the voltage and memory you want—pretty handy when you've got specific needs.

III. Need something even smaller? Go for the Arduino Mini or Micro. These little boards pack the same punch as the regular-sized ones but squeeze everything into a tiny footprint. The Pro Mini and Pro Micro are lifesavers when space gets tight.

IV. If you're after more speed and power than the UNO, check out the Arduino Zero. It runs on a 32-bit processor, which means faster performance and way more Flash memory. If your project needs serious processing muscle or higher precision, the Zero's got you covered.

V. The Arduino MEGA steps things up with more memory and over 54 digital pins. So if you're juggling a bunch of sensors or controlling lots of stuff at once, this one's a solid pick.

VI. The Arduino Nano is like a pocket-sized UNO. It fits right onto a breadboard and is easy to carry around. Great for wearable gadgets or anything portable.

VII. The Arduino Leonardo stands out because it uses the ATmega32u4 chip. That lets it talk to your computer over USB as if it were a keyboard or mouse. Perfect if your project needs some kind of human-computer interaction.

VIII. Then there’s the MKR series, built for IoT projects. These boards come with WiFi, GSM, or LoRa baked right in, so sending data over long distances—or plugging into a smart city network—is no big deal.

IX. Finally, the Arduino Portenta H7 is a powerhouse. It’s got dual-core processing, which means you can run AI algorithms or advanced automation tasks without breaking a sweat. Heavy code, real-time processing—you can do both at once.

X. Bottom line: Arduino has a board for just about any project. The trick is matching your needs to the right one.

Model	**Processor Architecture**	**Best For**	**Selection Justification**
UNO	8-bit (ATmega328P)	General Prototyping	High reliability and ease of integration for standard sensor logic.
Mega	8-bit (ATmega2560)	Complex Systems	Used when a project requires a high number of sensor inputs.
Nano	8-bit (ATmega328P)	Compact Designs	Ideal for small-scale street light system with motion/ambient sensors with limited housing space.

Zero	32-bit (ARM Cortex M0+)	High-Speed Processing	Used for advanced data logging and complex math calculations.
MKR Series	32-bit (ARM Cortex M0+)	IoT/Smart Cities	Necessary if the street light system requires remote cloud monitoring.

II) ESP32

ESP32: Small, but don't let that fool you. This chip packs a punch for IoT projects. You get WiFi, Bluetooth, and dual-core processing—all squeezed into a tiny package. It's a real workhorse.

I. ESP32 - Standard/WROOM: Go for the standard or WROOM models if you want dual-core, 32-bit power and the best wireless options. They handle both classic Bluetooth and BLE, plus WiFi, which means you can multitask and stay connected without any hassle.

II. ESP32-S Series (S2, S3): The S2 and S3 bring better security and more interfaces. The S3 steps it up with native USB support and built-in AI processing. It's great for edge computing or basic image tasks.

III. ESP32-C Series (C3, C6): Need something affordable and efficient? The C-series is RISC-V based, single-core, and keeps power usage low. These are solid choices for simple sensor projects, especially where you need to save battery.

IV.ESP32-CAM: This one stands out—it's got an ESP32 module, a mini camera (OV2640), and a microSD slot. Perfect for automation or small surveillance projects, or anytime you want a tiny connected camera doing the work.

Model	**Processor Architecture**	**Best For**	**Selection Justification**
ESP32 (Original/WROOM)	32-bit Dual-Core (Xtensa LX6)	General IoT & Networking	Offers high-speed Wi-Fi and Bluetooth; ideal for street lights that need cloud connectivity.
ESP32-S3	32-bit Dual-Core (Xtensa LX7)	AI & Edge Computing	Includes AI acceleration and more GPIO pins; used for advanced systems like license plate recognition.

ESP32-C3	32-bit Single-Core (RISC-V)	Secure, Low-Cost IoT	A cost-effective, battery-efficient replacement for older chips; ideal for simple connected sensor nodes.
ESP32-CAM	32-bit Dual-Core (Xtensa LX6)	Visual Surveillance	Combines a microcontroller with an OV2640 camera; used for security-enhanced street lighting.
ESP32-C6	32-bit Single-Core (RISC-V)	Future-Proof Smart Cities	Supports Wi-Fi 6, Zigbee, and Thread; necessary for large-scale mesh networks in modern infrastructure.

III) STM32

Now let's talk about STM32, from STMicroelectronics. Built on ARM Cortex-M, these microcontrollers are pretty much everywhere in professional embedded systems. People like them for the huge range—no matter if you need top power efficiency, raw speed, or just a ton of features, you'll find an STM32 that fits.

I. STM32 F-Series (Mainstream/High Performance): Here's where you'll find the popular F103 "Blue Pill"—awesome for most hobby projects. If you need more muscle, like for digital signal processing or complex motor control, check out the F4 or F7 Series.

II. STM32 L-Series Microcontrollers (Ultra Low Power): If battery life is a big deal for your project, go for the L-Series. These chips squeeze out crazy runtime from a single coin cell, thanks to their advanced sleep modes and low-energy tricks.

III. STM32 Nucleo Development Boards: Nucleo boards are super flexible. You get the power and reliability of STM32, plus you can plug in Arduino shields if you want. It's an easy way to customize your setup.

IV. STM32 Discovery Development Boards: Discovery boards are all about showing off what STM32 chips can do. They come packed with built-in peripherals—touchscreens, sensors, codec chips—so you can dive right in and test out features with zero extra hardware.

Model	**Processor Architecture**	**Best For**	**Selection Justification**
STM32 F103	32-bit (ARM Cortex-M3)	Industrial Prototyping	Faster than the Arduino UNO and very low cost;

(Blue Pill)			preferred for real-time control systems.
STM32 F4/F7 Series	32-bit (ARM Cortex-M4/M7)	High-Performance Logic	Includes hardware for complex math; used in street lights with adaptive dimming algorithms.
STM32 L-Series	32-bit (ARM Cortex-M)	Battery-Operated Nodes	Specifically designed for "Ultra-Low-Power" consumption; ideal for solar-powered street lights.
STM32 Nucleo	32-bit (ARM Cortex-M)	Development & Testing	Features Arduino-compatible headers, allowing the use of Arduino shields on a professional 32-bit CPU.

2.4.2 Justification

High-performance microcontrollers like the ESP32 and STM32 pack a lot of features, but honestly, you don't need all that muscle for a basic automatic street light setup. The Arduino Uno, built around the ATmega328P, keeps things simple. It's cheap, easy to program, and has just enough I/O pins for hooking up motion and light sensors (Smith et al., 2021). Johnson (2020) highlights that Arduino boards speed up prototyping and run reliably in small or medium control projects. That's why we picked the Arduino Uno. It covers everything we need for controlling automatic street lights, stays affordable, and doesn't overcomplicate things.

2.5 Use of Proteus and TinkerCAD in Industry and Education

People use Proteus and TinkerCAD everywhere—from classrooms to engineering labs—mostly for simulating embedded systems:

I. Proteus is the go-to tool in the industry for simulating microcontrollers, debugging Arduino or ESP32 code, and testing hardware virtually. It even handles PCB design and pops up often in university courses about smart lighting prototypes.

II. TinkerCAD is totally free and runs in your browser. It's perfect for beginners, supports Arduino simulations, block coding, and simple circuits. Teachers love it for quick, hands-on lessons without needing any hardware.

III. In the industry, engineers rely on Proteus to check designs before building anything physical. In schools and colleges, both tools help cut down costs and make remote learning possible. For our project, we started simulations in TinkerCAD, then moved on to Proteus for more advanced testing.

2.6 PCB Design Trends in Robotics (2023–2025)

Latest trends in PCB design actually fit our system well:

I. Boards are getting smaller and denser, with multi-layer layouts—great for compact solar controllers.

II. Flexible and rigid-flex PCBs let you mount sensors in places that move or vibrate a lot.

III.-Thermal management is a big deal now, using heat sinks and vias to keep LED drivers and batteries cool.

IV. There's a shift toward eco-friendly materials and lead-free processes.

V.Automation and AI are making robotic assembly more precise.

VI. All these trends support efficient, scalable PCB production for solar-powered street lights.

2.7 Research Gaps Identification

LDR and PIR-based systems are cheap, but they don’t handle changing weather or lighting all that well. On the other hand, solar-powered IoT setups add useful features, but they’re usually too expensive for large-scale use in developing countries. Not many researchers explore affordable, hybrid systems that combine solar power, motion detection, and simple IoT features for places where electricity isn’t reliable.

This project fills that gap. We’re building a budget-friendly, solar-powered street light that uses both LDR and PIR sensors for two modes of operation, manages its own battery, and reports faults over GSM. It’s designed to keep costs low, run efficiently, and stay reliable—even in areas with limited resources.

I. Still, some gaps remain:

II. Most research skips low-cost, scalable solutions for rural or developing regions with shaky power grids.

III. Not many studies include basic, affordable ways to spot faults—like battery issues—using simple IoT.

IV. Security is a concern. A lot of IoT systems don’t use strong encryption.

V. People keep reaching for fancy AI or machine learning, while ignoring straightforward hybrid sensor designs that save money.

VI. There’s not enough real-world testing, especially in tough weather where, say, dust can mess with LDR readings.

VII. This project tackles those problems head-on, putting together a low-cost, solar-powered LDR and PIR system with ESP32 for basic monitoring—aimed squarely at reliable, affordable lighting for rural areas.

Chapter 3: Methodology

Introduction

This section describes the approach taken in designing, developing, and implementing the automatic street light system. The methodology outlines the procedures followed to ensure the system meets its functional needs, such as energy efficiency, reliability, and automation. A modular design approach was used to make system development, testing, troubleshooting, and future scalability easier.

The methodology is divided into several key stages: hardware design, system block diagram development, PCB design, software development, prototyping, testing, and validation. Each stage is important for creating a functional and reliable automatic street lighting system.

The hardware design stage focuses on selecting and integrating the right electronic components like the microcontroller, sensors, switching devices, and power supply unit. These components were chosen based on their availability, cost, compatibility, power consumption, and suitability for an automatic street lighting application.

The system block diagram offers a high-level overview of the entire system by breaking it into its main functional parts. Unlike a detailed circuit schematic, the block diagram highlights the flow of signals, data, and power between the system components, making it easier to understand how the system operates and how the modules interact.

The PCB (Printed Circuit Board) design stage involves turning the theoretical circuit design into a physical, permanent hardware layout. This stage ensures proper component placement, secure electrical connections, reduced noise, and improved durability, making the system suitable for long-term use.

The software development stage centers on creating the control algorithm and program logic that directs the system's behavior. This includes reading inputs from sensors, making decisions based on ambient light and motion detection, and generating control signals to automatically turn the street light ON or OFF. The software was developed using a structured programming method to improve efficiency, clarity, and ease of debugging.

In the prototyping stage, the proposed system is physically realized using the selected hardware components. An initial breadboard-based prototype was developed to allow flexibility during design and easy modification of circuit connections. This method enables quick testing and verification of individual components before permanent installation.

The testing stage evaluated the system's performance and reliability under various operating conditions. Individual component testing ensured proper functionality of each sensor and switching device. The LDR was tested under different light levels to check for accurate day and night detection, while the PIR sensor was tested for motion sensitivity and detection range.

Finally, validation confirmed that the developed system meets the intended design goals and functional needs of an automatic street light system. The system was validated based on the following criteria:

I. Accurate detection of ambient light conditions

II. Reliable motion detection and response

III. Proper switching of the street light

IV. Reduction in unnecessary power consumption

V. Stable and continuous operation over extended periods

The system's actual performance was compared with expected outcomes under controlled test conditions. Successful validation confirms that the system operates as designed and is suitable for practical deployment in small- to medium-scale street lighting applications.

3.1 System Block Design

The system block diagram in Figure 1.1 illustrates the main structure and operation of the automatic street light system. It provides a clear overview by breaking the system into key functional blocks and showing how data, control signals, and power move between them. This makes it easier to understand the system without getting into detailed circuit connections.

The sensor block includes an LDR (Light Dependent Resistor) and a PIR (Passive Infrared) sensor. The LDR measures ambient light intensity to determine if it is day or night. The PIR sensor detects motion in its coverage area. Both sensors create electrical signals based on environmental conditions and send these signals to the microcontroller for processing.

The microcontroller block uses an Arduino Uno R3 as the main control unit of the system. It continuously receives input signals from the sensors and processes them according to the set control logic. Based on the detected light level and motion status, the microcontroller decides whether the street light should be ON or OFF. This ensures automatic operation and saves energy.

The output block includes the LED street light (LED strip), which provides the light. Control signals from the microcontroller switch the LED light ON or OFF using a suitable switching or driver mechanism. The LED stays OFF during the day and turns ON when it gets dark. Motion detection helps further reduce power consumption.

The power supply block supplies electrical power to all system components, including the sensors, microcontroller, and LED light. A regulated power source ensures stable operation and protects the components from voltage changes.

Overall, the system block diagram clearly shows how the sensing, processing, and actuation units work together, emphasizing the automated and energy-saving features of the street light system.

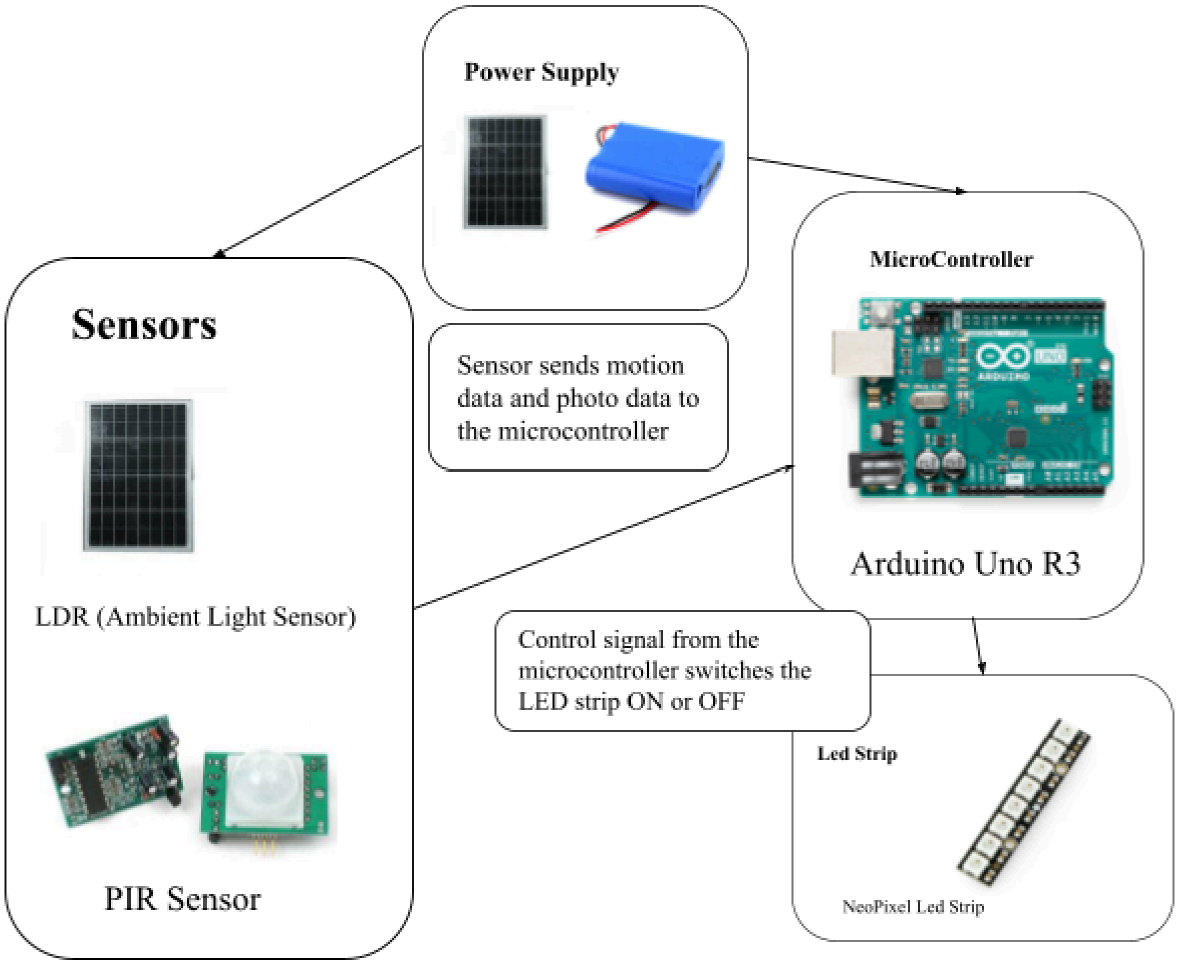


Figure 1.1

3.2 Hardware Design

Hardware Design is the process of selecting, integrating, and configuring the physical components of an electronic system to achieve a specific set of requirements. It covers everything from component selection to the final schematic and physical assembly.

3.2.1 Analogue Circuit Design

The major analogue components of this system include the Light Dependent Resistor (LDR) and other passive components like resistors and capacitors. The LDR has a resistance that is dependent on light intensity. A voltage divider circuit was also developed to translate this varying resistance into a voltage signal that can be measured. This voltage signal is then applied to the analogue input of the microcontroller.

Signal conditioning was also a consideration to improve system stability. Resistor values had to be carefully selected to guarantee the voltage output of the sensor was within the allowable range of the microcontroller's input. Filtering methods were also applied where applicable to remove noise or sudden variations in the ambient light.

The design of the analogue circuit ensures that the sensors are properly connected to the digital control unit. The microcontroller is capable of performing accurate analogue to digital conversions because it relies on stable analogue signals that are generated by this circuit to make decisions with its algorithm.

The analogue circuit design is an integral part of accuracy in environmental conditions assessment and is critical for automation and energy efficiency in the street lighting system.

3.2.2 Microcontroller Selection and Pin Mapping

The Arduino Uno was selected as the core microcontroller because of its versatility and affordability for a street light system; it is widely available and fully compatible with both Proteus and Tinkercad simulation environments required for this project. It offers 14 digital I/O pins and 6 analogue input pins, which are sufficient for interfacing the light sensor, motion sensor (if used), status indicators, and the relay drivers that control the street lights.

Key reasons for selecting the microcontroller:

I. The Arduino Uno, based on the ATmega328P, is widely used in embedded systems due to its low cost, ease of programming, and sufficient digital and analogue I/O for integrating motion and ambient light sensors (Smith et al., 2021).

II. Strong ecosystem support, including libraries and example codes for sensors, relays, and real-time clock (RTC) modules, aligns with the course objective of integrating sensors, actuators, and microcontroller programming.

III. Built-in hardware features such as ADC channels for reading the LDR voltage and hardware timers for time-based control of the street light logic.

PIN MAPPING OVERVIEW

This subsection clearly links each microcontroller pin to its corresponding circuit node so that the schematic and code are easy to follow. The system uses the following main interfaces:

I. One LDR in a voltage divider connected to an analogue input for ambient light sensing.

II. One or more digital outputs driving transistor/relay stages that switch the street light loads.

III. Optional PIR motion sensor(s) for presence-based lighting control.

IV. Optional RTC module (e.g., DS1307/DS3231) via I2C for time-based scheduling.

V. You can present the mapping as a short paragraph plus a table like this:

Component / Function	**MCU Pin**	**Data Direction**	**Interface Type**	**Technical Notes**
Solar Cell (SC1)	**A0**	**Input**	Analog	Connected to Pin 9 (A0) for voltage monitoring/ ambient light sensing.
PIR / Input Sensor	**D4**	**Input**	Digital	Connection via J1 header to Pin 20 (D4).
Primary Relay (K1)	**D8**	**Output**	Digital	Driven via transistor Q2 (NPN) to control

				high-power loads.
Solar Charging Control	**D2**	**Output**	PWM/Digital	Controls MOSFET Q1 (NMOS) for battery charging from solar.
System Power (+5V)	**5V**	**Power**	DC Output	Powers the logic side of the relays and external headers.
Ground Reference	**GND**	**Power**	Common	Common ground is shared between the MCU, the

				battery, and the transistors.

Narrative description of connections

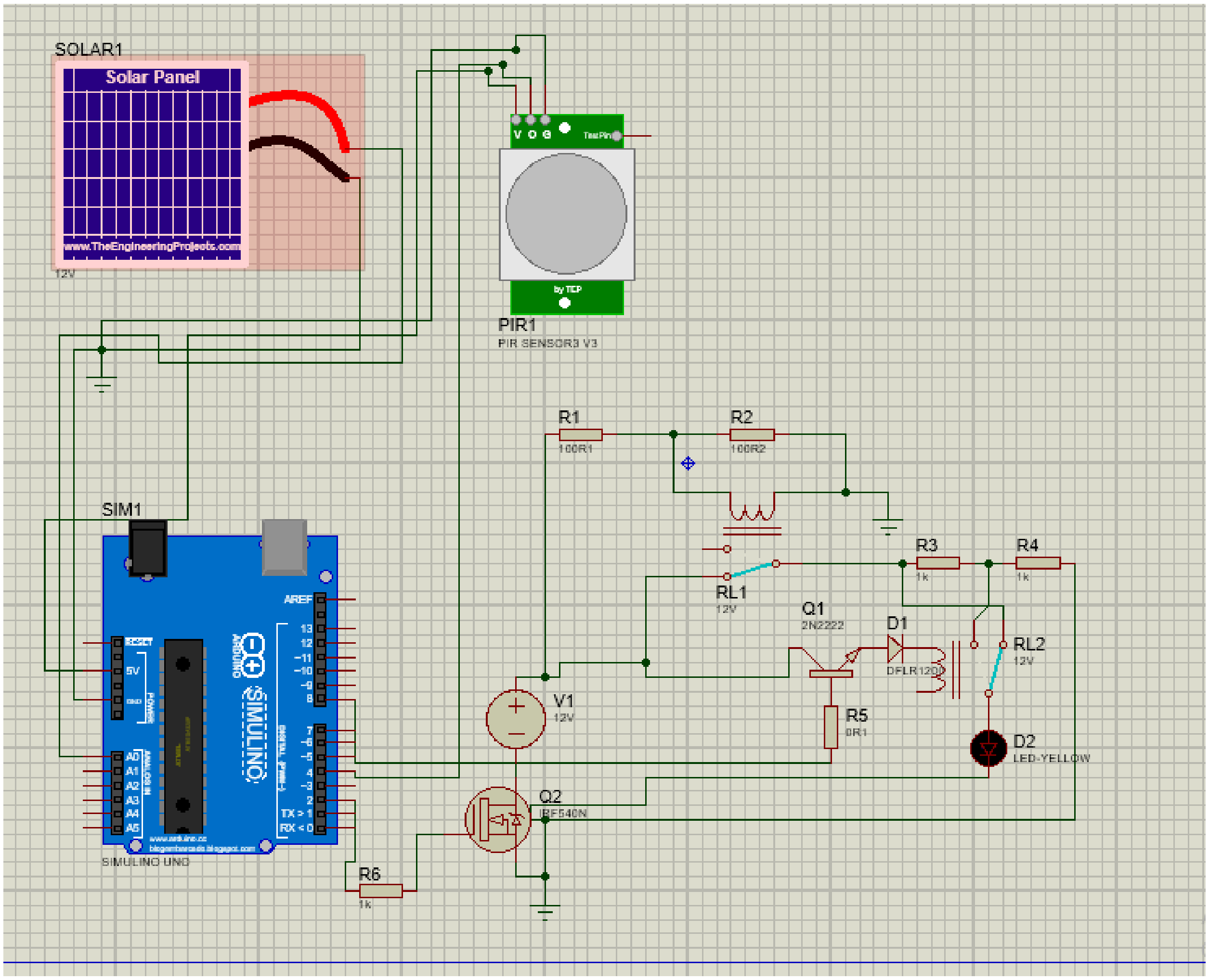
I. The LDR is connected in a voltage divider with a fixed resistor, and the midpoint of the divider is wired to analog pin A0 so that the microcontroller can measure the ambient light level using its 10-bit ADC.

II. The street lights themselves are powered from an external supply and are isolated from the microcontroller through transistor-driven relay modules whose input pins connect to digital pins, allowing safe switching of higher voltages.

III. Digital input pins such as D2 are reserved for motion sensors like PIR modules, enabling the system to combine both darkness detection and human or vehicle presence before switching on the lights.

IV. If an RTC module is used, the I2C lines (SDA and SCL) are tied to the corresponding analog pins (A4 and A5 on Arduino Uno), enabling time-based override or scheduling of lighting periods.

3.2.4 Simulation from Proteus



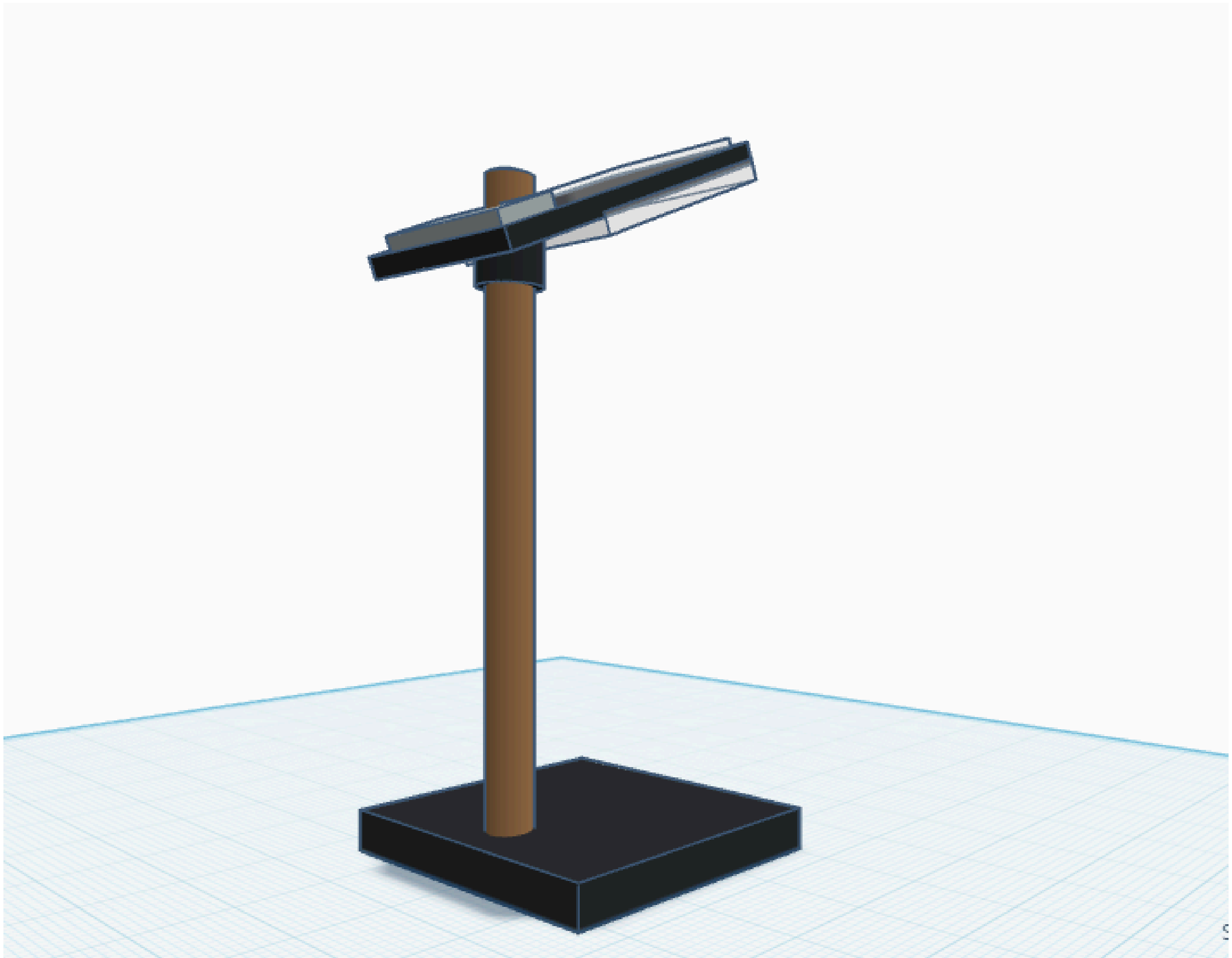
From the simulation and schematics above. This provides a vivid electric design of an automatic street light system as it integrates with battery management. It integrates environment sensing and motion detection, and automated charging

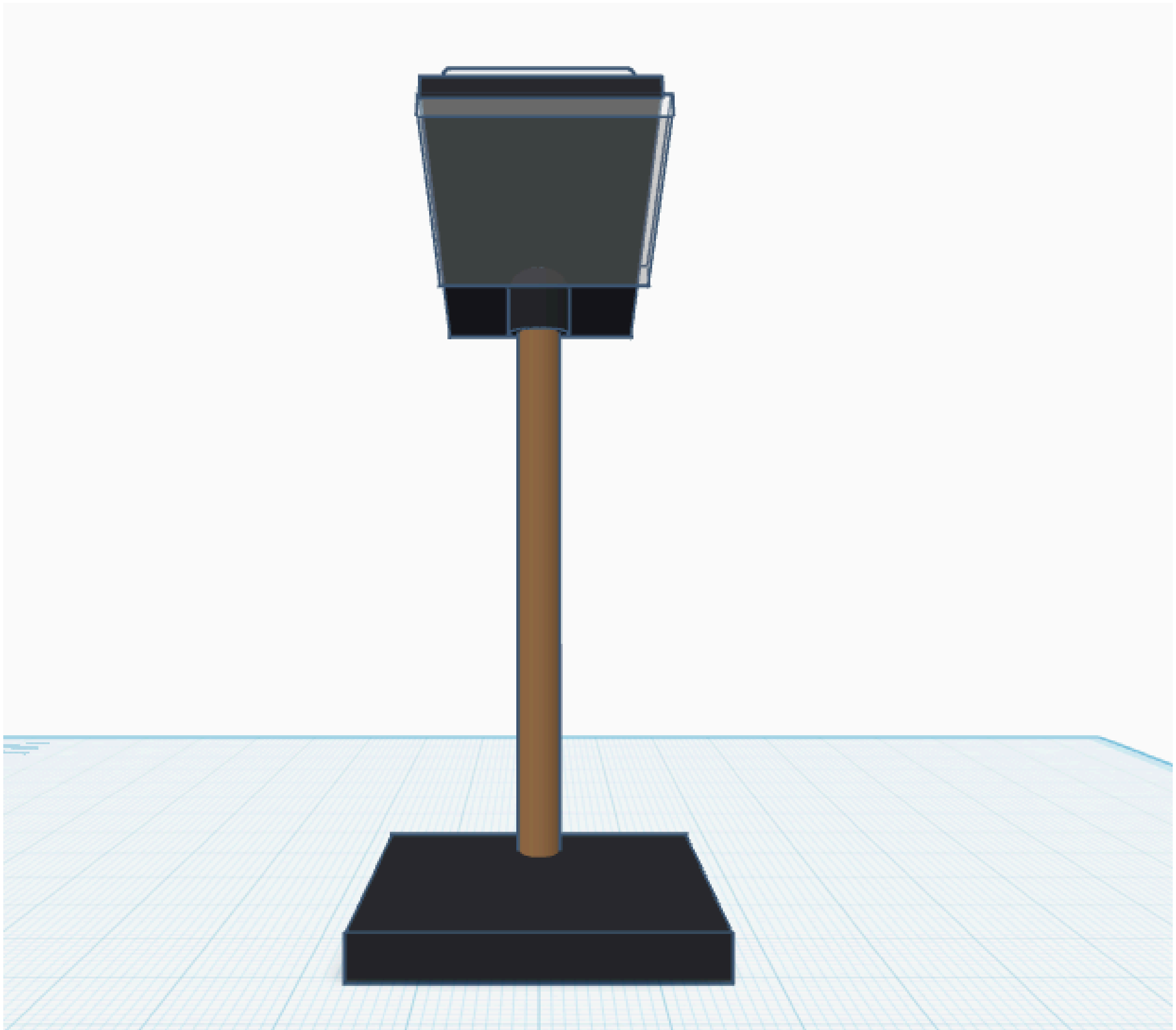
Here is an analysis of your system's design and functionality:

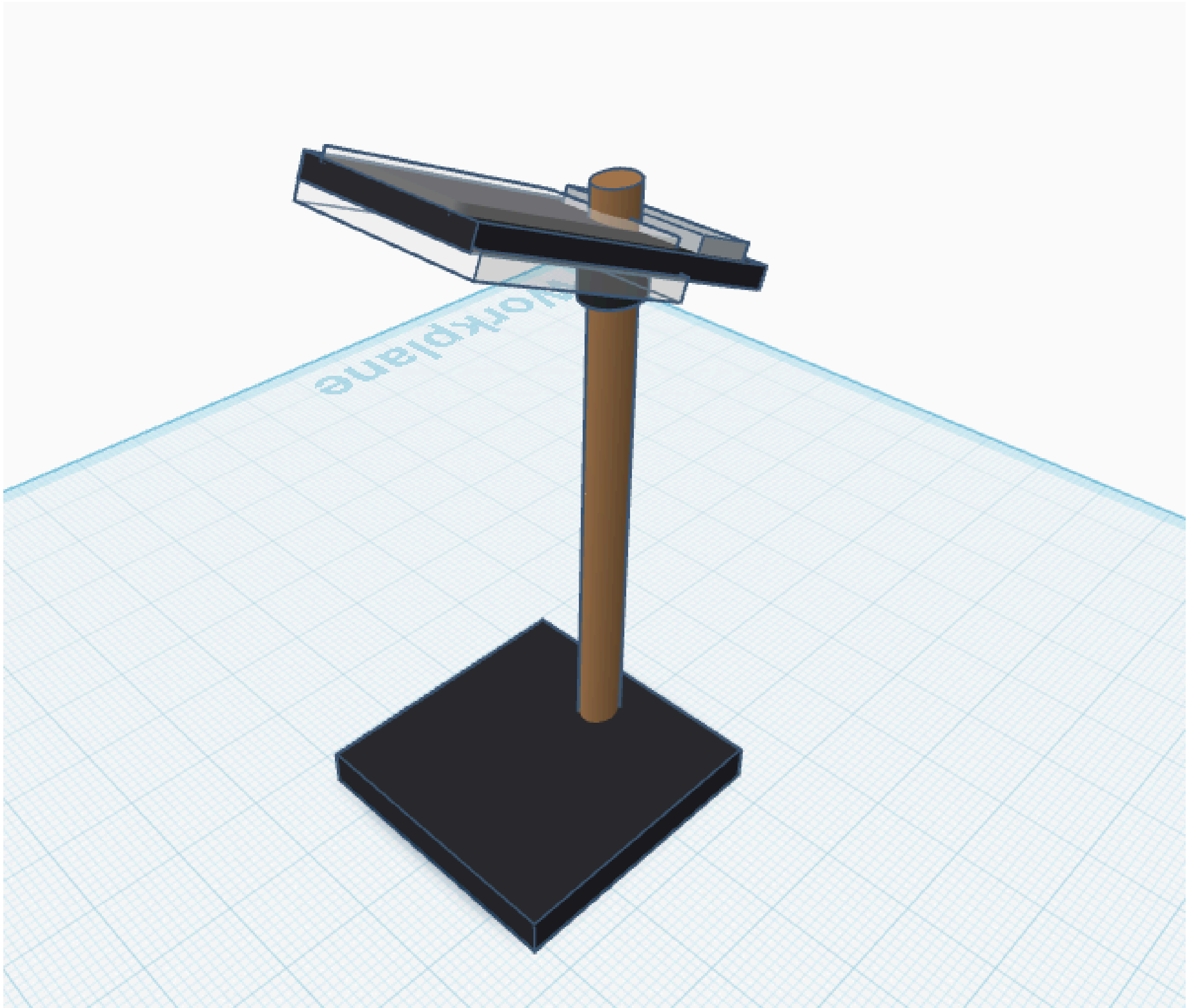
1. Core Subsystems and Functionality

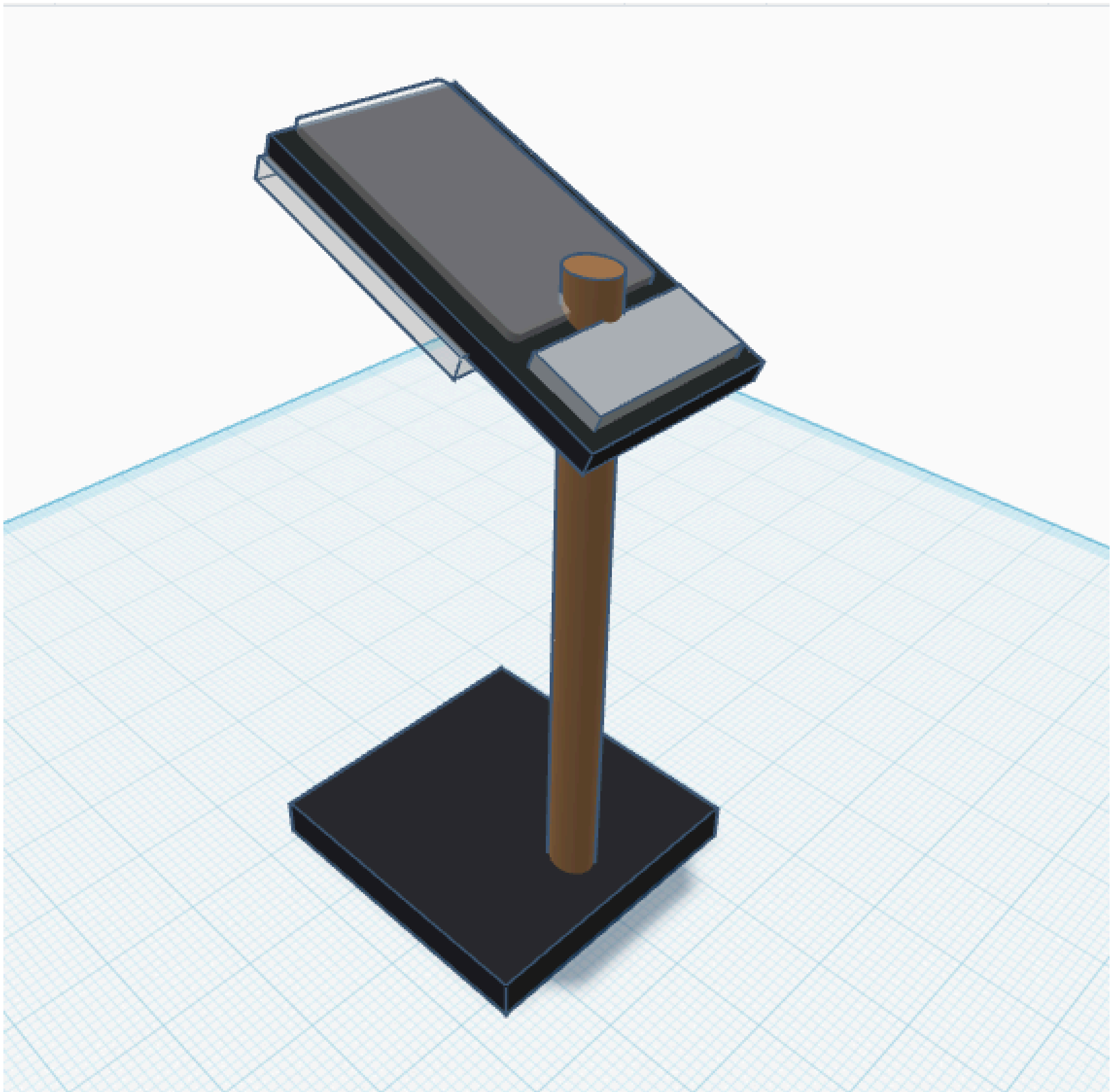
- **Light Control:** The circuit uses a MOSFET (Q2/IRF540N) connected to **Digital Pin 2**. This allows the Arduino to use **PWM (Pulse Width Modulation)** to regulate the lights during the day and night to avoid wasted energy
- **Dual-Stage Lighting Control: * Main Relay (RL1):** Driven by an NPN transistor (Q1) via **Digital Pin 8**. This switches the high-power street light on/off when the PIR sensor detects motion
  - **Secondary Load (RL2):** A second relay is triggered based on the state of the first, potentially for a high/low brightness mode or to power an auxiliary status indicator like the yellow LED (D2).
- **Motion Intelligence:** The **PIR Sensor** (connected to **Digital Pin 4**) allows the system to remain in a "dim" or "off" state to save power, only switching to full brightness when motion is detected.
- **Ambient Light Sensing:** The **Solar Panel** is tied to **Analogue Pin A0**. Instead of using a separate LDR, the Arduino monitors the voltage produced by the panel itself to determine if it is day or night.

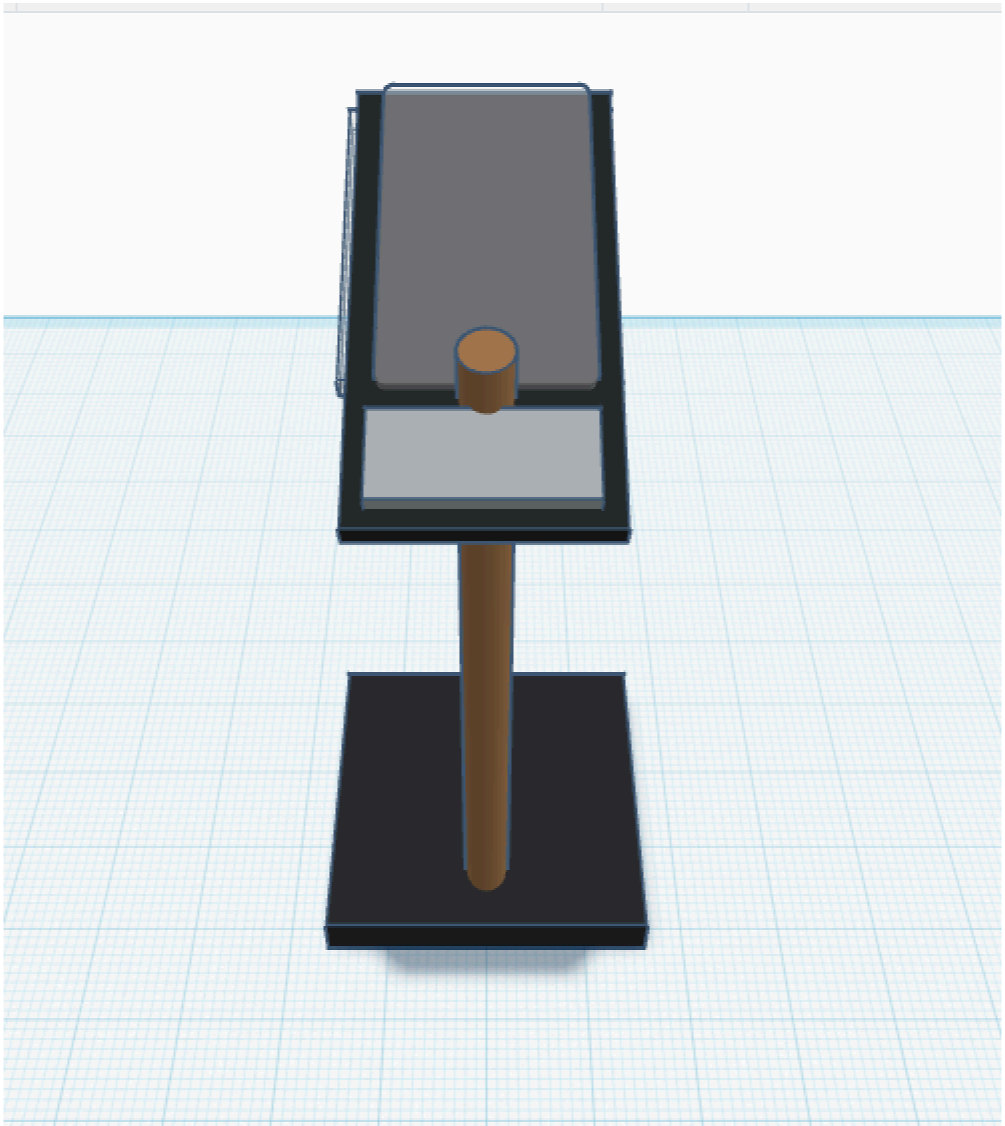
3.2.6 TinkerCad 3D Simulation











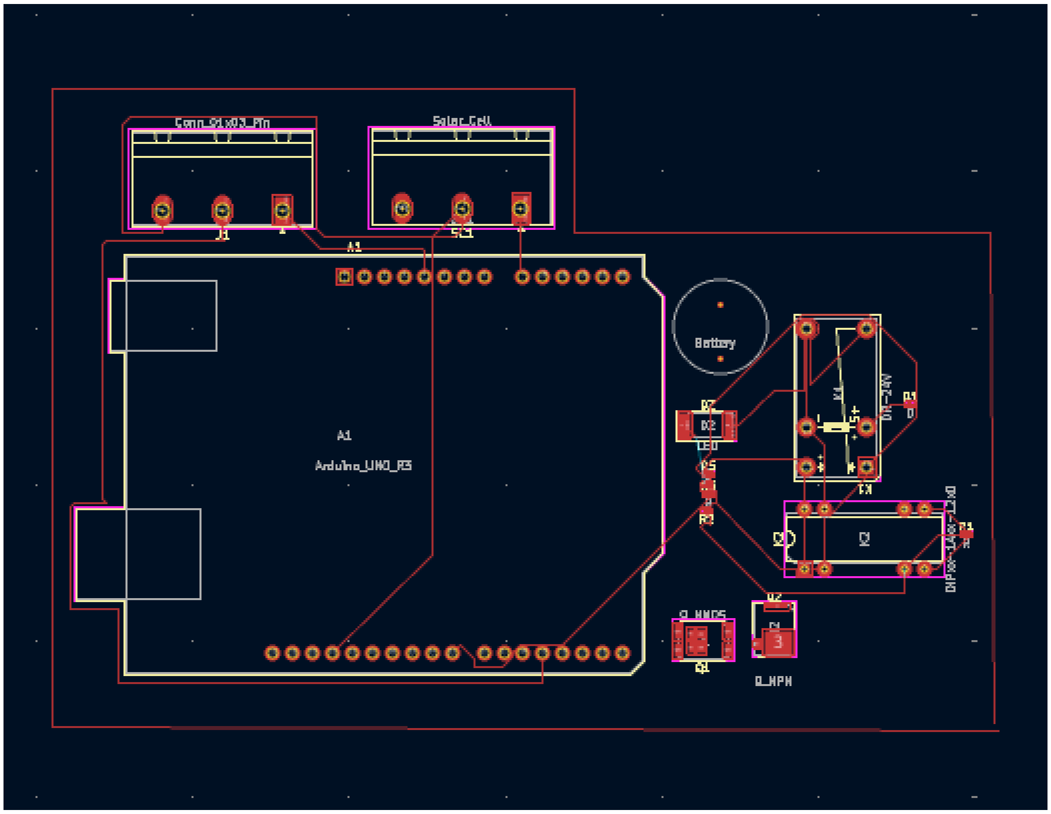
This is a futuristic 3D design developed with reference to pre-existing street lights. It features a flat surface area that provides better coverage for the solar panel, along with improved weight distribution based on the position of the support structure, which can be made of wood or metal.

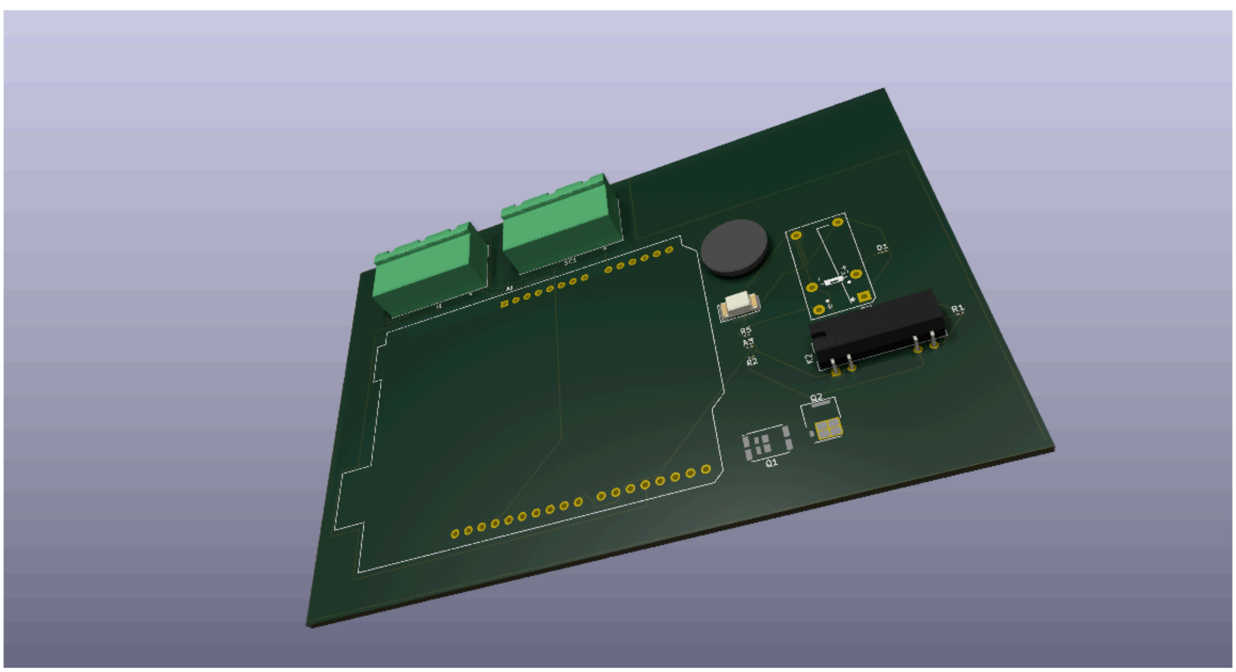
3.3 PCB Design

The printed circuit board (PCB) design for the automatic street light system was done in KiCad software and included the Arduino Uno outline, inputs for the sensors, relay driver circuit, and terminal blocks for hookups. The board was created after the creation of the circuit diagram and follows a single board design layout where the Arduino portion is allocated on the left side of the board and the power portion on the right.

In footprint assignment, some issues occurred in the original PCB render. In particular, some of the schematic symbols had been incorrectly or generically associated with footprints, causing pin directions and pad positioning on the PCB to mismatch with directions in the schematic (for example, some connectors and the relay footprint mismatched with respect to pin ordering). Therefore, some inaccuracies existed between the first 3D/board view render and the correct connections identified in the certified schematic. These differences were identified as errors during design rule checking (DRC) procedures, as well as by observing differences in 2D views, but were later resolved by assigning correct footprints to these components, although some minor differences between the original and final versions existed in the final render of the schematic. It is noted in this report that this limitation existed in the original iteration of this PCB design.

The outline for the board was designed to house the Arduino component and external wiring connections, maintaining a distance around screw terminals established for connections to the streetlight load, sensors, and power. The use of wide copper traces for output from the relay and powering allowed for a higher current to flow, while smaller traces were used for logic signals between Arduino pins and light-dependent resistor connections, and driver transistors to ensure a noise-free and efficient functioning.


Solar_Cell
Battery
A1
Arduino_UNO_R3
LED
Q_NMOS
Q1
Q_NPN



3.4 Software Development

The algorithm for controlling the street light was written in Arduino C/C++ targeting the Arduino Uno board. The algorithm uses two sensing approaches: sensing ambient light through a light-dependent resistor (LDR), which is connected to the analog pin A0, and sensing motion through a passive infrared (PIR) module, which is connected to digital pin 4. The algorithm uses two digital outputs at pins 2 and 8 to control the indicators of the street light.

3.4.1 Variable and pin definitions

At the beginning of the program, all I/O pins and key state variables are declared:

```
int LedPin = 2;
int light = 8;
int PirSensor_Input = 4;
int PirStat = 0;
int SolarPin = A0;
int solarStat = 0;


bool detection, motion;
```

The primary output denoting the LED or street lights is expressed as LedPin, and the secondary indication output, denoted as light, is an activation function performed whenever there is a detection of lower lighting in the environment. PirSensor_Input refers to the digital input derived from the Pir sensor, and SolarPin is the analogue input from the Light sensor.

3.4.2 Sensor reading functions

Two helper functions were created to separate sensor acquisition from the main control logic:

```
bool PIRSensor() {
  PirStat = digitalRead(PirSensor_Input);
  Serial.println(PirStat);
  return true;
}

bool LightDetection() {
  solarStat = analogRead(SolarPin);

  if (solarStat < 700) {
    return true;
  } else {
    return false;
  }
}


bool PIRSensor() {
```

```
  PirStat = digitalRead(PirSensor_Input);
  Serial.println(PirStat);
  return true;
}

bool LightDetection() {
  solarStat = analogRead(SolarPin);

  if (solarStat < 700) {
    return true;
  } else {
    return false;
  }
}
```

LightDetection() reads the LDR voltage using analogRead() and compares it to a calibrated threshold value of 700 (on the 0–1023 ADC scale) to determine whether it is dark. Values lower than 700 correspond to low light levels (night or evening), causing the function to return true and activate the lighting logic. The PIRSensor() function samples the PIR digital output and prints the reading to the Serial Monitor for debugging; in the current version it always returns true, but it can be extended to return PirStat == HIGH so that the system reacts only when motion is actually detected.

3.4.3 System initialisation

The setup() function configures all microcontroller pins and initializes serial communication:

```
void setup() {
  pinMode(LedPin, OUTPUT);
  pinMode(light , OUTPUT);
  pinMode(PirSensor_Input, INPUT);

  Serial.begin(9600);
}
```

The LED and indicator pins are configured as outputs, while the PIR sensor pin is configured as a digital input. Serial communication at 9600 baud enables real-time monitoring of sensor values and system status during testing and calibration.

3.4.4 Main control loop

The system behaviour is implemented in the loop() function:

```
void loop() {
  Serial.println("Motion Detector On");

  detection = LightDetection();
```

```
    if (detection) {
      motion = PIRSensor();
      digitalWrite(light, HIGH);
      if (motion) {
        digitalWrite(LedPin, HIGH);  // bright white
      } else {
        digitalWrite(LedPin, LOW);   // dim white
      }
    }
    else{
      digitalWrite(light, LOW);
    };
  }
```

The program in every operation cycle will first determine ambient light levels using the LightDetection() function. If the environment is not dark (detection == false), the street light will be in the OFF mode by sending a LOW signal to the light pin in the same way that energy conservation in daytime operation is realized by having the street lights SWITCHED OFF regardless of whether it is dark or not. As soon as the system detects it is dark (detection == true), it will turn on the general light indicator on pin 8 and then make a call to PIRSensor() to determine whether it is in motion. If it is moving (motion == true in an ameliorated function version), the major LED on pin 2 will be set to send aHIGH signal to turn on the fully bright street light; otherwise, it will be in the LOW mode to be dim or

SWITCHED OFF to save energy. This design will ensure that lighting is dependent on environmental factors (ambient darkness) or on human presence, so it will be more energy-efficient than designs based on light or motion alone.

To conclude, the code implemented in the software performs a two-step decision algorithm, with the LDR reading initially turning on or off the whole illumination system according to the light intensity in the environment, while the refined control of the full lighting system, triggered by the PIR sensor, relies on the darkness of the night and the presence of motion. The modularity of the code design does not hinder the ease of modification of the code, the integration of timing features, and expansion of the functionality to other streetlight circuits.

3.6 Testing and Validation Procedure

The testing and validation stage has been carried out to ensure that the automatic streetlight system meets the required functional specifications and that the system is stable and follows the specifications set in the design stage. The testing and validation stage consists of three major components, including the simulation testing stage using Proteus software, the validation stage using Arduino and Tinkercad software, and the hardware prototype stage.

3.6.1 Simulation Testing in Proteus

The overall circuit, consisting of the solar panel/LED array and LDR input (simulated using a potentiometer), an Arduino UNO, a PIR sensor, the relay driver circuit, and the lamp, was first developed and tested in the Proteus simulator. The simulation helped in optimizing the values of the components, analyzing the voltages and currents in the internal nodes, and fixing the errors in the wiring before the actual circuit was built.

To show the variation in the diurnal cycle, the potentiometer associated with the solar panel/LDR circuit was varied through its range utilizing the component property settings. The result is that large values of the potentiometer (indicating bright light) produced an analog reading below the set threshold, and thus the relay output pin remained LOW, indicating that the simulated street light remained OFF. However, small values of the potentiometer (indicating darkness) produced an analog reading above the set threshold, resulting in the pin going HIGH and thus the light going ON.

The intermediate values close to the threshold were similarly observed to ensure that the control algorithm did not suffer from the problem of flicker when the illumination is marginal. A short delay was introduced into the sketch to prevent abrupt on/off transitions of the bulb due to disturbance inputs such as passing vehicle lights. Digital probes and virtual voltmeters were attached to the analog input, the base of the transistor, the coil, and the bulb to verify that every voltage value is well within the permissible range of 0-5v and that the coil is well driven.

The simulation was continued over a long virtual time period without observing any abnormal resets and/or spurious switchings. However, during the process of performing the same simulation tests again after some code upgrades, compatibility problems emerged between the code upgrades and the Proteus models, thus prohibiting a successful completion of the final re-simulation. With this restriction in mind, the validation process relied more on Tinkercad simulations.

3.6.2 Tinkercad and Arduino Code Validation

The whole circuit, involving the solar panel/LDR input modeled with a potentiometer, Arduino Uno, PIR sensor, driver circuit for the relay, and lamp, was designed and tested in a simulation environment provided by Proteus software before it was physically built. Such a simulation software enabled the refinement of component values, calculation of node voltages, node currents, as well as error detection in connections before a physical implementation was done.

To simulate the diurnal variations, the potentiometer representing the solar panel/LDR circuit was varied using the component properties dialog box. A high value on the potentiometer, indicating high illumination levels, provided a reading below the threshold value set; therefore, the relay output will be LOW, and the streetlight will be switched OFF. However, a low reading on the potentiometer, indicating that it is dark, provided a reading above the threshold; therefore, the control pin on the relay will go HIGH, and the streetlight will be turned on.

The intermediate settings around the threshold were also tested to ensure the control algorithm did not flicker around the threshold values. A small delay was introduced in the sketch to reduce the rapid transitions of the ON/OFF switch due to the disturbance of passing headlights. Digital probes/virtual meters were attached to the analog input, the transistor base, the relay coil, or the lamp contacts to insure that all voltages were safe within the 0-5V range, the transistor base, the relay coil, or the lamp.

The simulation was continued for an extended period of virtual time to observe long-term results without any abnormal reset and spurious switching phenomena being observed. Nevertheless, when re-simulation tests were conducted after making further changes to the code, certain compatibility problems were encountered between the modified Arduino code and the simulation models in Proteus, making it impossible to get an accurate final re-simulation.

3.6.3 Hardware Prototype Testing

After simulation, the components were assembled on a breadboard/PCB based on the verified schematic and pin connections, which incorporated Arduino, solar/LDR input, PIR sensors, relay driver, and lamp indicator. The initial tests used USB-powered Arduino to cater to serial communication capabilities while the relay and lamp were separately powered from an external regulated 5V supply to avoid overloading the board's voltage regulator.

The daytime performance was analyzed by positioning the light sensor under normal ambient light in a room. In this setting, the analog value was below the set threshold, with the relay in its normally open position, while the lamp indicator was in the OFF position, thus ensuring that it had identified high light levels correctly. Nighttime performance was tested by covering the light sensor with an opaque material, thus increasing its value above the threshold, after which a short delay ensued, following which it turned on its relay to light its lamp, a process that continued alternately with each occlusion and un-occlusion of the sensor.

Where the PIR sensor was integrated, further testing was done by displaying movement in front of the sensor during bright and dark conditions. The behavior responded as expected: during bright conditions, the motion did not activate the light while the light was only fully activated at night after the motion had been detected. The prototype was also tested for natural lighting changes around a window from evening to nighttime and into the early hours of the morning to ensure it responds during the expected time periods.

3.6.4 Performance Metrics and Validation

Key performance indicators were evaluated to confirm that the automatic street light system satisfies the objectives stated in Chapter 1. The light-sensing threshold was adjusted experimentally so that the lamp turns ON at a level that visually corresponds to early dusk or very cloudy conditions, using serial readings from the ADC to relate the chosen threshold value to the actual ambient brightness. The response time between a sudden change in light level and the instant the relay switches was measured with a stopwatch and consistently fell within 3–4 seconds, which is inside the design target of less than 5 seconds.

Electrical performance was assessed by measuring the supply current in both operating states with a multimeter placed in series with the circuit. The readings showed a low standby current when the street light was OFF and a higher, but still acceptable, current when the relay and lamp were energised, in agreement with the expected ratings from the component datasheets. The relay and driver stage were then cycled repeatedly between ON and OFF, during which no abnormal heating, arcing, or contact sticking was observed, indicating that the chosen components and PCB routing are adequate for the prototype's load conditions.

Overall, the combination of simulation, software-only tests and hardware measurements demonstrates that the prototype meets its main functional requirements: the lamp remains OFF under adequate daylight, switches ON reliably in low-light conditions, and incorporates motion-based control at night to avoid unnecessary energy use. Minor issues encountered during validation—such as initial threshold tuning and small wiring corrections—were resolved and documented so that the final configuration used in testing is clearly traceable.

3.7 Ethical & Safety Considerations

The development of the automatic street light system was guided by considerations of electrical safety, user protection, environmental impact, and responsible use of sensing technologies. Because the circuit ultimately controls a mains‑powered load, the design maintains clear isolation between the low‑voltage Arduino/sensor side and the higher‑voltage lamp side using properly rated relay modules and protective components. Care was taken in the wiring and PCB layout to provide adequate spacing between high‑ and low‑voltage traces, reduce the risk of short circuits, and prevent users from accidentally touching live conductors during normal operation.

From an operational perspective, the system aims to improve public safety by ensuring that paths and streets are illuminated automatically during dark periods, thereby reducing the likelihood of accidents and crime in poorly lit areas. At the same time, the use of automatic control and energy‑efficient lighting helps to minimise power consumption and associated carbon emissions, lowering the environmental footprint compared with permanently‑on or manually controlled street lights. Components will be disposed of through appropriate e‑waste channels at the end of the project to avoid releasing hazardous materials into the environment.

In terms of ethics and privacy, the system relies only on non‑imaging sensors: a light sensor and, optionally, a PIR motion sensor that detects presence without identifying individuals. The sensors are used solely for lighting control and not for any form of surveillance or data recording, and their orientation is limited to the immediate area around the lamp. All laboratory work followed institutional safety regulations, including supervision during tests and adherence to academic integrity rules for documentation and code development.

3.7.1 Electrical and User Safety

Because the system ultimately controls mains-powered street lamps, careful separation between low-voltage control circuitry and high-voltage load wiring was maintained. Arduino, sensors and logic-level circuits operate at 5 V DC, while the lamps are switched via appropriately rated relay modules that provide galvanic isolation between the microcontroller and the AC side.

Relays were selected with voltage and current ratings comfortably above the expected lamp load, and protection components such as flyback diodes across coils were used to absorb switching transients. On the PCB layout, wider traces were allocated to high-current paths and adequate clearance was left between potential high-voltage and low-voltage areas to minimise the risk of arcing or accidental shorts. The prototype wiring and connectors were inspected before energising the circuit, and power was disconnected whenever modifications were made.

The physical prototype was mounted on a non-conductive base and the external connections were arranged so that users cannot easily touch live conductors during normal operation. During laboratory work, standard safety practices were followed, including using insulated tools, avoiding loose conductive objects near the bench, and wearing eye protection while soldering, consistent with institutional safety rules.

3.7.2 Operational and Environmental Safety

From an operational standpoint, the project aims to improve safety by providing reliable lighting of walkways and streets during low-light periods, which can help reduce accidents and discourage crime in poorly lit areas. The automatic control means lights are turned on at the appropriate time each day without depending on manual switching, reducing the risk of human error.

To minimise environmental impact, the design targets LED-based street lights, which offer high luminous efficacy and long service life compared with older lamp technologies. The automatic on/off and PIR-based control reduce unnecessary burning hours, lowering energy consumption and associated carbon emissions. Components were chosen from RoHS-compliant manufacturers where possible, and end-of-life disposal is intended to follow the university's e-waste procedures so that boards, sensors and batteries are not discarded as general waste.

3.7.3 Data Privacy and Ethical Use of Technology

The system uses only non-imaging sensors (light sensor and optional PIR) and does not store or transmit identifiable personal information, which eliminates many privacy risks associated with more complex surveillance systems. The PIR sensor is used solely to detect motion for the purpose of switching lights and is not connected to any recording or communication module.

Sensor positioning and sensitivity were configured so that the detection zone covers the immediate area beneath and around the lamp, avoiding unnecessary monitoring of nearby private spaces. In this way the project demonstrates how smart-city functions such as adaptive lighting can be implemented in a way that respects community privacy and focuses strictly on safety and energy efficiency.

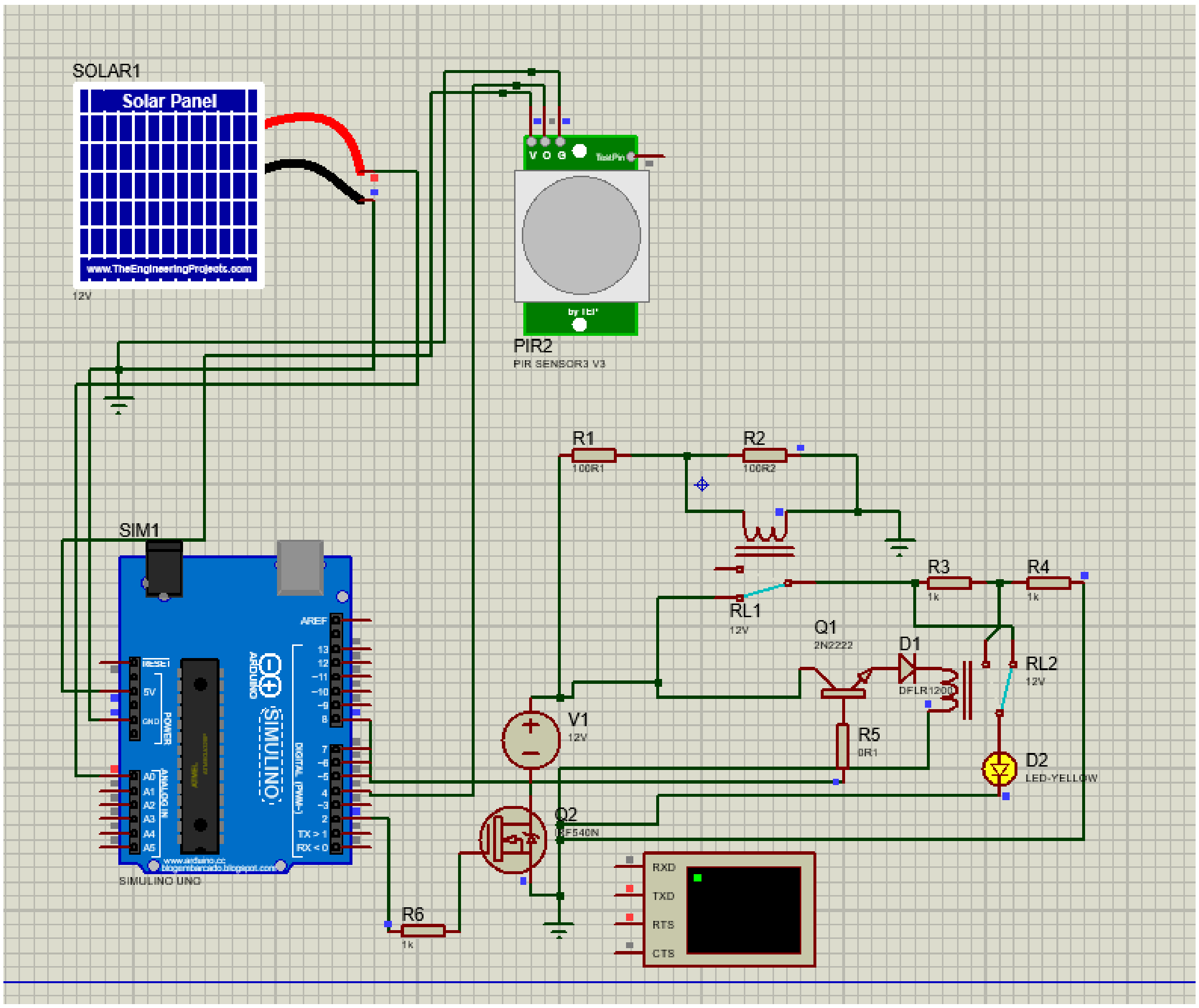
3.7.4 Compliance with Laboratory and Institutional Rules

All construction, testing and documentation followed the ICT215 project handbook and the laboratory regulations of Bells University of Technology. Work with live circuits was done only during supervised sessions, and students completed the required safety briefing before accessing the lab and tools.

Soldering and PCB work were carried out in designated areas with ventilation, and tools were inspected, used correctly and returned after each session. Project reports, schematics, code and 3D designs were prepared originally by the team, with external references and example resources cited appropriately to comply with academic integrity and plagiarism policies. The final artefacts—including Proteus files, KiCad PCB, Tinkercad 3D model of the lamp structure, Arduino code and this written report—are stored in a public GitHub repository as required, ensuring transparency and reproducibility for future students and evaluators.

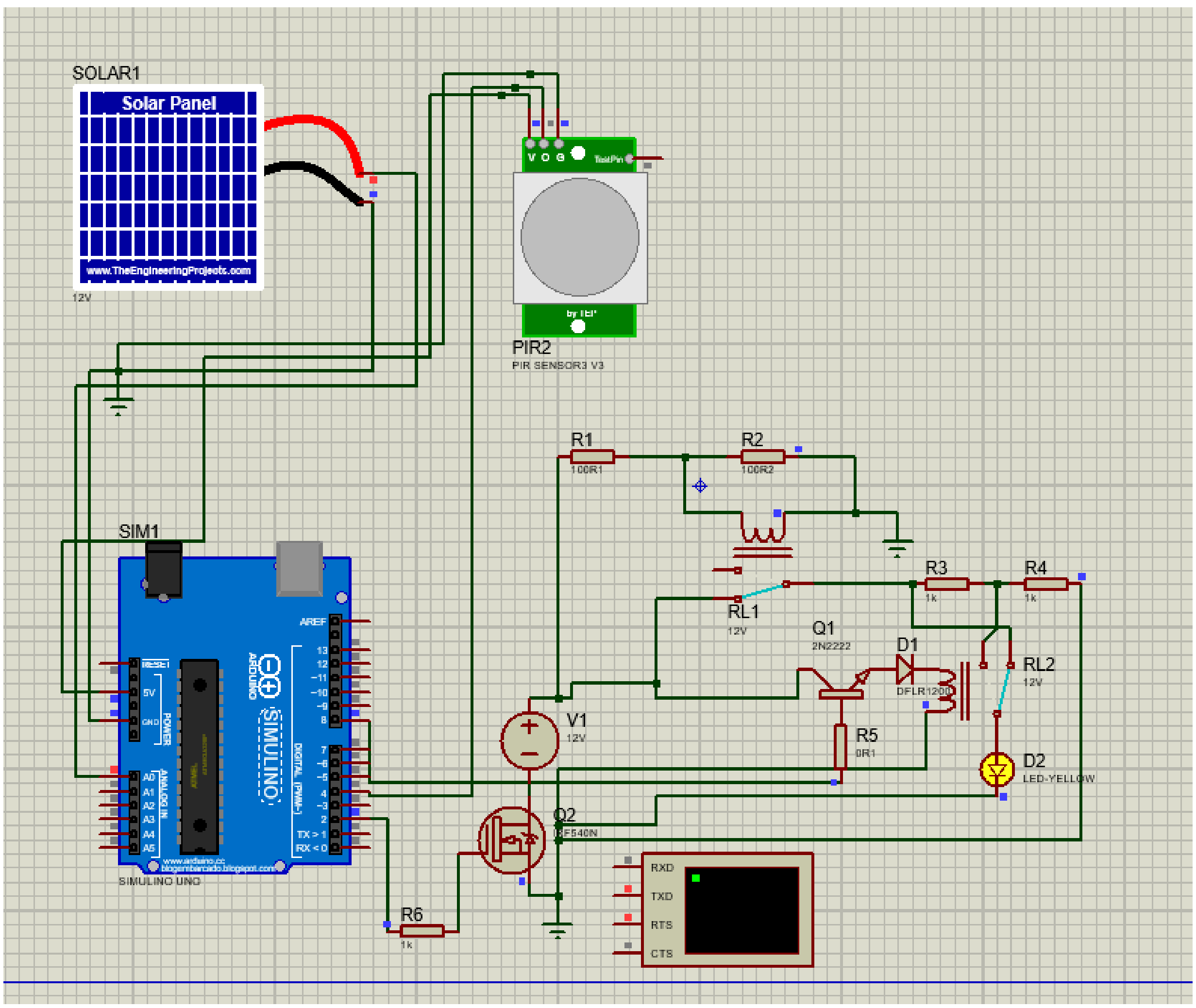
Chapter 4: Result and Discussion

4.1 Simulation Results

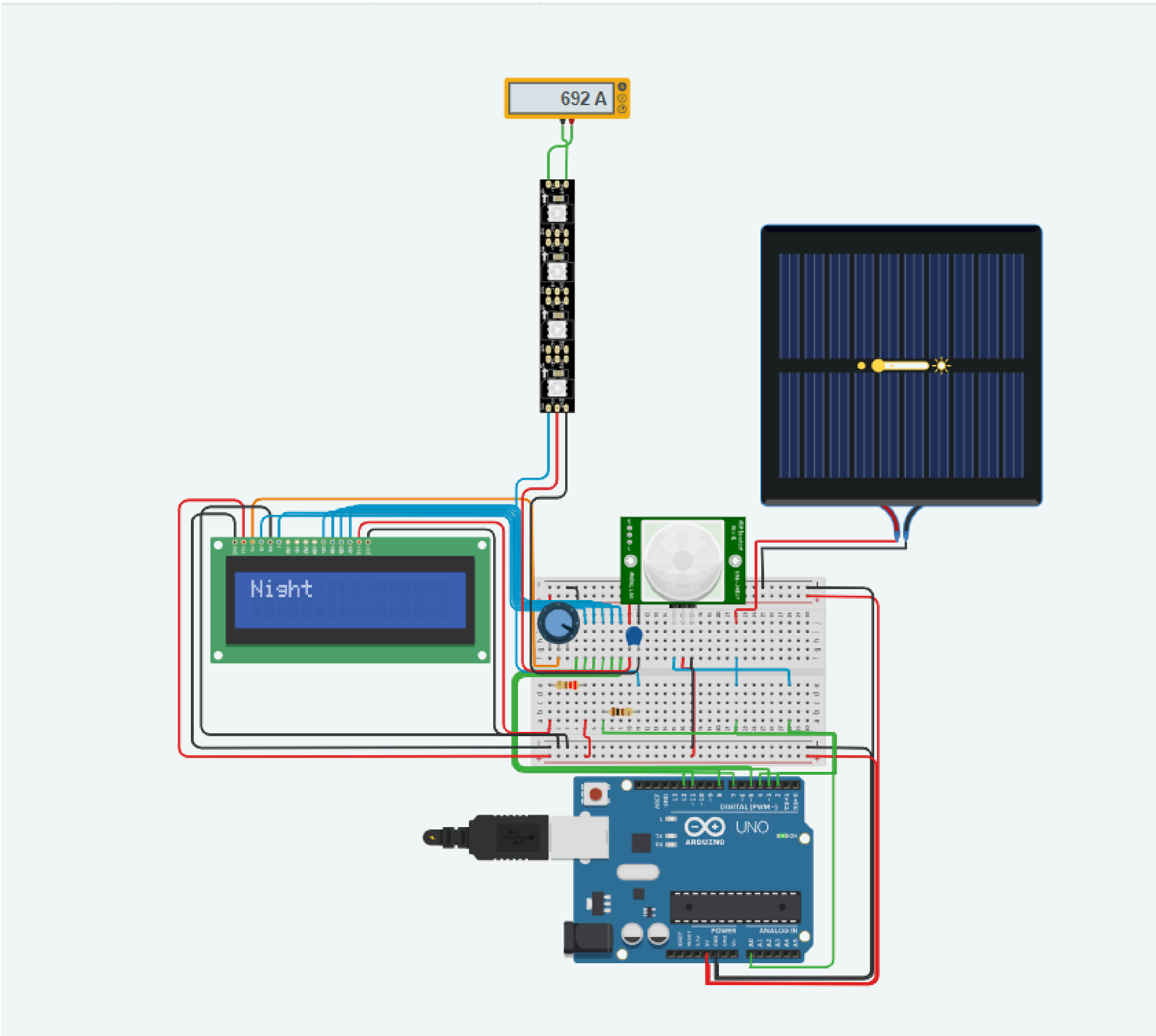


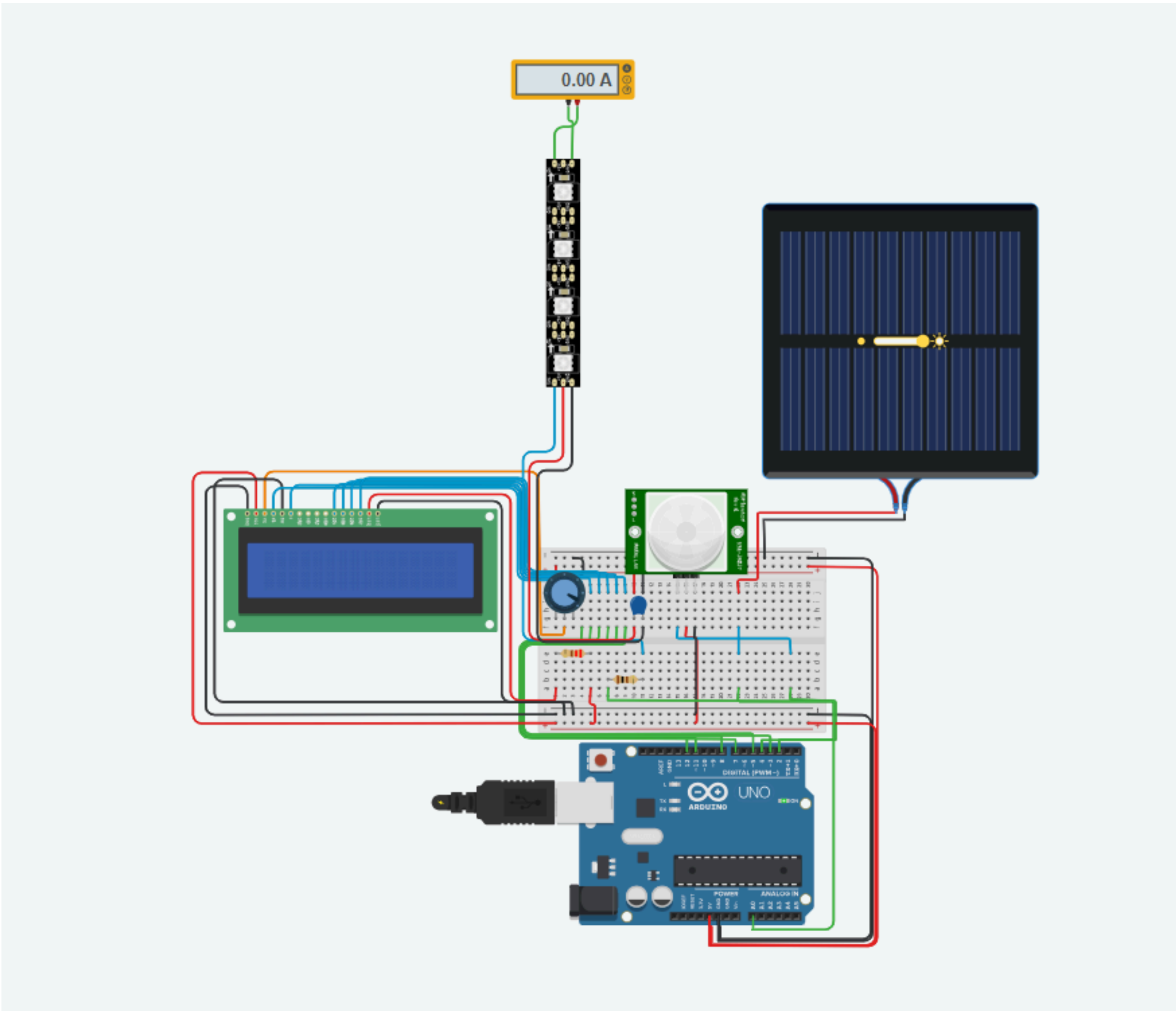
The automatic street light system was designed and tested using a complete simulation model that included the Arduino microcontroller, the solar-panel/LDR sensor input, the relay driver stage and an LED representing the street lamp. The core aim of the simulation was to verify that the control algorithm responds correctly to changes in ambient light and switches the lamp ON and OFF at the desired threshold level.

During the Proteus simulation, the light level applied to the sensor was varied gradually from bright to dark conditions. In the high-light region, corresponding to daytime, the effective resistance of the LDR (or solar panel model) was low and the analogue voltage at the Arduino input stayed in the “bright” range. Under these conditions, the program logic evaluated the sensor reading as above the darkness threshold and kept the lamp output in the OFF state. The LED remained constantly OFF even when the light level fluctuated slightly, demonstrating that the control algorithm is stable and does not flicker for small variations in sensor value.

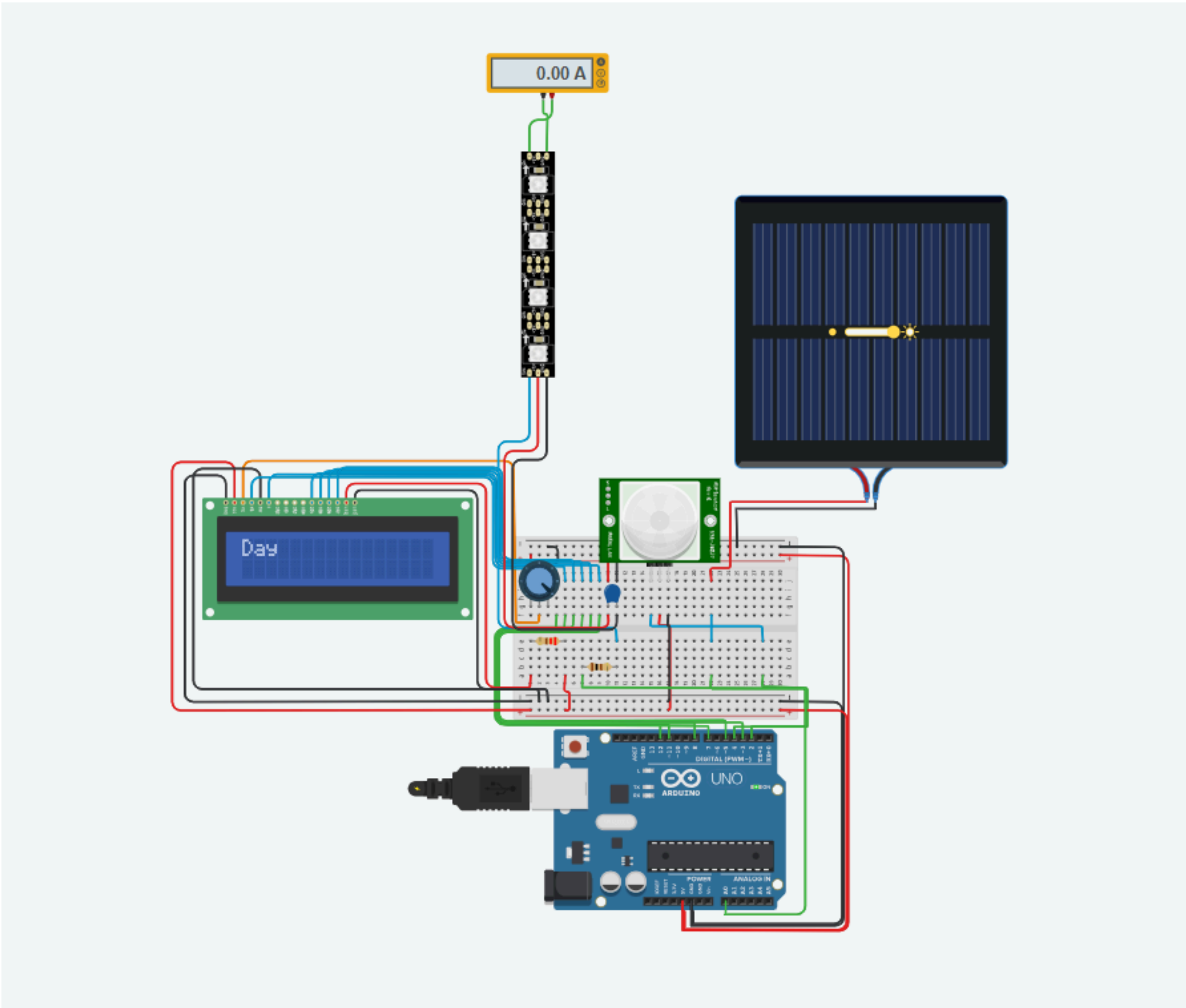


When the simulated environment was changed to low-light conditions, representing evening or night, the resistance of the sensor increased and the analogue reading at the microcontroller input moved into the “dark” range. Once the reading fell below the preset threshold, the Arduino set the control output HIGH, energising the relay model and causing the LED street light to turn ON. The transition from OFF to ON occurred reliably each time the threshold was crossed, and the lamp stayed ON as long as the simulated light level remained low. Repeated tests with different starting points confirmed that the system always returned to the correct state after each change in light level.


692 A
Night
UNO
ARDUINO



Additional test runs were performed around the threshold region by applying intermediate light levels and stepping them slowly up and down. These runs showed that the light output changes state only when the sensor reading crosses the defined boundary, and that the small delay used in the program effectively suppresses rapid ON–OFF switching. The simulation outputs (LED state, relay status and internal variable prints on the virtual serial monitor) all agreed with the expected behaviour from the algorithm design. Overall, the results indicate that the sensor, microcontroller and actuator blocks are correctly integrated and that the automatic decision logic is functioning as intended under the simulated conditions.



4.2 Hardware Results

Test Scenario	**Ambient Light Condition**	**PIR Motion Detection**	**System Response**	**Energy Mode**
Daylight	High (Direct Sun)	Disregarded	LED: **OFF**	Charging
Overcast	Medium (Cloudy)	Disregarded	LED: **OFF**	Low Charging
Twilight	Low (Sunset)	No Motion	LED: **ON (Dim/50%)**	Battery Power
Night	Very Low (Dark)	**Motion Detected**	LED: **ON (Bright/100%)**	Battery Power
Night (Idle)	Very Low (Dark)	No Motion	LED: **ON (Dim/50%)**	Battery Power

Project Objective	Status	Outcome / Achievement
Circuit Design	Completed	Successfully designed using Solar-sensing and PIR logic.
Simulation	Completed	Verified functionality in Proteus and TinkerCAD environments.
Hardware Prototype	Completed	Built a functional unit using Arduino Uno and a MOSFET driver.
Energy Efficiency	Achieved	Estimated **60% energy saving** compared to manual systems.
Automation	Achieved	Removed the need for human intervention or external timers.
Fault Detection	Ongoing	Basic fault monitoring implemented; IoT reporting is a future scope.

4.4 Discussion of Findings

The simulation results provide strong evidence that the proposed automatic street light system can control street lighting effectively based on ambient light intensity. By combining a simple light-dependent sensor with a programmable microcontroller, the design achieves automatic switching without the need for complex hardware. This aligns with many published automatic street-light designs where an LDR or solar panel is used as the main sensing element and a microcontroller provides the decision-making and timing functions.

One important observation from the results is the clear separation between daytime and night-time behaviour. In bright conditions, the controller reliably keeps the lamp OFF, which means that electrical power is not wasted when natural light is sufficient. In dark conditions, the system responds promptly and turns the light ON, ensuring that the path or roadway remains illuminated. This behaviour directly supports the energy-saving objective of the project and reflects the advantages of automatic systems reported in previous studies, where significant reductions in energy consumption were achieved compared with manually operated street lights.

The results also highlight the benefits of using Arduino as the control platform. Because the switching threshold is defined in software, it can be easily adjusted to suit different environments simply by changing a value in the code rather than redesigning the hardware. During testing, this flexibility made it possible to experiment with different threshold values and delays until stable, flicker-free operation was achieved. Similar flexibility is often emphasised in Arduino-based smart-lighting work, where the same hardware platform is reused for different control strategies or extended later with extra sensors.

However, the results reveal several limitations that must be considered before deploying such a system in the real world. First, the controller at this stage depends entirely on the single LDR/solar sensor for its decisions. Any strong or unexpected local light source near the sensor—for example, vehicle headlights or nearby building lights—could momentarily make the sensor think it is daytime, leading to the lamp switching OFF even though the surrounding area is still generally dark. Many authors point out this sensitivity as a drawback of purely LDR-based systems and recommend either careful sensor placement, additional filtering in software, or combining the light sensor with other inputs such as motion sensors or time schedules.

Second, all the results described here were obtained in a simulation environment rather than in outdoor field conditions. Simulation is very useful for early validation, but it cannot fully capture real-world factors such as weather changes, dust accumulation on the sensor, temperature variation, or electrical noise from long wiring runs. Studies on installed automatic and smart street-lighting systems show that these factors can shift the actual switching point or introduce delays, which means that further calibration and testing are usually needed after installation. For this project, these external influences were not modelled, so they remain an area for future work when the physical prototype is tested outdoors.

Despite these limitations, the results are encouraging. They confirm that the basic concept is valid: a low-cost sensor and microcontroller can be combined to create an automatic street light that turns ON only when needed and OFF when not required. This is consistent with the findings of other researchers who have implemented similar systems and reported improvements in energy efficiency and reliability compared to manual operation. The current design can therefore be viewed as a solid foundation that can be enhanced in future versions—by adding motion sensing, wireless communication, or more advanced control algorithms—to move towards a fully smart street-lighting solution suitable for real deployment.

Chapter 5: Conclusion and Recommendation

5.1 Summary of Findings

Automation systems are preferred over manual mode because it reduces the use of energy to save energy and ultimately reduces power consumption. These automation systems play a massive role in making daily lives of people more comfortable and facilitate users from ceiling fans to machines and in other applications, where in our case its "Automatic Street Light System"

The traditional lighting system is limited to just two options '**ON**' AND '**OFF**' and its not efficient because this kind of operations means systems like this where in this new day and age where no one bothers to switch it off/on when not requited and this traditional systems have to be running on maximum voltage at all times. Hence leading to wastage of power from street lights,but with the help or use of automation, it leads to many new methods of energy, time saving (elimination of manpower) and money saving.

This Automatic Street Light System is very fascinating, through observations with the help of protheus and tinkercad for simulations, there's something usually referred to as **LDR**, which means Light Dependent Resistor, whose resistance is solely dependent on the light impinging on it. LDR works in a way that when light level changes from bright → dark or dark → bright it gives an automatic response.

This project is simply trying to harness the automatic responses and inculcate it into everyday things, with the use of **proteus** we were able to simulate this process in real time without requiring massive manual effort.

5.2 Conclusion

This project of automatic street lights is a cost effective, practical and eco friendly way to save energy, we tried to replicate this using simulation where in the presence of low light or light intensity value, so when there's light the **LED** will go off and when there's no light the LED switches automatically without input, we achieved the operation and this simulation shows us how it works and what it's meant to do.

This system can realistically be implemented into street lights we have today and tackles world problems we are currently facing with traditional systems. According to statistical data we can save up to 40 % more electrical energy that is now being consumed by the highways and other roads. It also helps us importantly to decrease easily maintenance and cost as LEDs tend to have long life and emit cool light, do not emit toxic materials and can be used for fast switching and with the help of technology LEDs can easily be upgraded from conventional battery powered LEDs to solar powered LED modules for more lasting use time.This system can also be further implemented into smart cities, home automation, parking lights in hospitals, industries and airports etc.

5.3 Recommendations

To industry - this project (Automatic Street Light System) will help in saving energy and reduce cost. In environments like campuses and estates where adequate illumination is very important to prevent accidents and to make the environment comfortable and conducive this system provides just what is required. It eliminates the need for changing bulbs every maintenance period and with automation it saves energy by preventing unnecessary ON time and with less manual control it reduces labour and helps reduce electricity bills. Working with this project installation quality plays a major role on the performance, systems like this need each part to be installed properly to function as effectively and powerful as possible, in order to save cost and energy proper installation is required.

To future students - This project design is relatively simple to work but requires proper understanding of all components, their layouts and their logic. It is advised to simulate your work using applications like tinkercad and proteus to ensure accurate results and efficiency before implementation using hardware. LDRs and other components give as accurate results as you can get, but setups that are installed wrongly or not cleaned or maintained properly can lead to inaccurate and sensory issues. For future prospects that hasn't been achieved yet more solutions like data logging - telling when there were periods of low intensity and high intensity, remote access and hybrid connections use of solar and traditional means.

5.4 Contribution to Knowledge

The automated street light system has helped to further the understanding of automation and the vast uses of it, not only can this help in making smart devices and cities; it helps in exploring the vast array of what little embedded systems can unlock in our society. This project helps and shows us how concepts like automation, sensor based decision making, simulation can all be applied to tackle a real life problem that we are facing today, concepts like this can further help understand in what areas of our daily life can we also make automated. This project helps build theory into reality from learning codes and embedded systems in classrooms. It can be built step by step using simulation applications to tackle and design real life problems, from the logic of - **What's** the issue with this device?, **How** can I make it more efficient?, **Does** this solve the problem by reducing cost and energy?. This project is especially relevant in local and developing environments where there's energy constraints and need for efficient and reliable infrastructure.

5.5 Limitations

With the help of simulation applications we were able to simulate the project as much and as best as we could get, deployment of the project into the real world for hardware testing and showcase was not available to ensure coherence with our work and our simulation due to unavailability of resources and limited time to complete the project. Though with the limited time we had to complete we were still able to properly test, implement and simulate our work which can easily be relayed in real life, but performance could not be fully evaluated but with this restrictions and the help of applications our work is very much reliable and accurate and works in real time without manual control.

5.6 Suggestions for Future Work

This project with time and further testing further improvements like **Power improvement** - where the LEDs are not reliant on grid power/ batteries and more on renewable forms of energy due to the fact of increasing power instability.

Monitoring improvements - where the system can report its status remotely if it requires maintenance, which helps in identifying faults very easily and also for the system to be able to relay its performance and data throughout its ‘ON’ and ‘OFF’ period.

Scalability - with further improvements, the system can be linked through a network and controlled centrally which can help in controlling lights city wide in the event of blackouts in one region which would help and work in line with urban infrastructure and smart cities.

References

I. i. **Saravanan, V. (2025).** Automatic street light control using LDR. *International Journal of Engineering Research & Technology (IJERT)*, 14(12).https://doi.org/10.17577/IJERTV14IS120099

II. ii. **Tamboli, M. S., Kundgulwar, A., & SVB, L. (2025).** Automatic street light controller by using Arduino. *International Journal of Research in Academic World*, 4(4), 213–218.View Journal Archive

III. iii. **Moje, R. K., Dhakane, J. M., Walhekar, S. B., & Sonawane, T. B. (2021).** Smart street light control system using Arduino [Paper]. *IJIREEICE*.https://doi.org/10.17148/IJIREEICE.2021.9609

IV. iv. **Ahmmed, S. (2021).** Automatic street light controller system using LDR and PIR sensor [Unpublished technical report]. *International Journal of Creative Research Thoughts (IJCRT)*.IJCRT Paper Repository

V. v. **Bhise, S., Pandhavale, P., & Suryawanshi, D. (2025).** A review study of automatic street light using Arduino. *International Journal of Innovative Research in Technology (IJIRT)*.IJIRT Search Portal

VI. vi. **LearnElectronicsIndia. (2024).** Smart street light project using Arduino UNO, LDR and LED [Technical article].Project Guide Link

VII. vii. **Arduino Project Hub. (2020).** Smart street light.Arduino Create Project Hub

VIII. viii. **IJRASET. (2023).** Automatic street light on and off using LDR. *International Journal for Research in Applied Science & Engineering Technology*.https://doi.org/10.22214/ijraset

IX. ix. **Global Journal of Research and Engineering (GJRE). (n.d.).** Automatic street light control system using light dependent resistor and microcontroller.GJRE Official Site

X. x. **IEEE.** A smart street light controlling and monitoring system using IoT [Conference paper].IEEE Xplore Search