
BELLS UNIVERSITY OF TECHNOLOGY
CHORDS OF KNOWLEDGE

AUTOMATIC STREET LIGHT SYSTEM

By

GROUP 15

Group Members

EHIMEN ENOCH

2024/13785

OTI MICHAEL

2024/13804

OLAYIWOLA QOYUM

2024/13811

OBANIYI EMMANUEL

2023/12863

POPOOLA FAITH

2025/14537

BART FLOURISH-STIORA

2025/14740

ICT 215

Robotics and Embedded Systems

SUBMITTED TO:

AYUBA MUHAMMAD

TABLE OF CONTENT

ABSTRACT

This project dives into building an Automatic Street Light System using an LDR sensor, an Arduino, and LED lights. Basically, the system turns street lights on when it’s dark and off when there’s enough daylight. This is done to reduces the use of energy and save money. The setup brings together analog sensors, signal tweaks, and some solid microcontroller programming to make sure everything runs smoothly.

I started with circuit simulations in Proteus and Tinkercad, wrote the Arduino code, designed the PCB, and then built the hardware prototype. The simulations helped catch mistakes early, so I didn’t end up frying any components. After that, I made a printed circuit board version, which made everything sturdier and cleaned up all the wiring.

When I tested the system, it reacted well to changes in light and kept the street lit up at night without a hitch. Right now, it only runs one street light and still uses grid power, but the whole setup is pretty affordable and easy to scale up. It’s a solid step toward smarter, more efficient lighting, and it doubles as a hands-on learning project for anyone getting into embedded systems or robotics.

CHAPTER ONE: INTRODUCTION

1.1 Background of the Study

Street lighting is important because it helps in lighting the streets at night which leads to safety at night. In various environment, street lights are still being controlled manually by people and this usually leads to streets lights being left on and off at night of turned on late at night. This usually results in waste of energy, it increases the cost of electricity, etc. With the growing need for smart and efficient systems, automatic control of street lighting has become important especially as the world is advancing more and more into a Digital Era.

This project focuses on the design and simulation of an Automatic Street Light System using sensors and a microcontroller. The system automatically turns the street light on at night and off during the day, all by itself. It also reduces brightness when there is no movement, and increases brightness when motion is detected, this is to conserve energy while still providing safety and visibility on the road.

The project was chosen because it is simple, practical, and actually used in real life. The system is mainly designed for roads, highways and public environments where people and vehicles need to see clearly at night. By letting sensors and automation handle the work, you cut down on human effort and stop wasting electricity.

1.2 Problem Statement

In many environments, street lights are still controlled manually and this causes several issues. For example, sometimes people forget to switch off the lights during the day, which leads to unnecessary power consumption and increases the cost of electricity. At night, the switch may be difficult to locate due to poor lighting, and as a result, the lights may remain off when they are actually needed. These challenges make manual street lighting inefficient and unreliable most of the times.

1.3 Objectives of the Study

General Objective

The general objective of this project is to design and simulate an Automatic Street Light System that controls street lighting using sensors and a microcontroller to make street lighting more efficient and energy saving.

Specific Objectives

The specific objectives are:

1. To design and simulate the automatic street light system using Tinkercad and Proteus.
2. To automatically control street lights based on surrounding light conditions.
3. To reduce the amount energy used and the cost of electricity through automation.
4. To provide clear visibility at night by increasing light brightness when needed.
5. To implement motion-based brightness control in order to conserve energy.
6. To document the complete design process and prepare the project for implementation.

1.4 Research Questions

This project is guided by the following research questions:

1. How can we control street lights without human involvement?
2. How can sensors be used to detect day and night conditions?
3. How can motion detection be used to improve energy efficiency?
4. What is the effect of automation on power consumption and cost?
5. How can microcontrollers be used to control street lighting in real time?

1.5 Significance of the Study

This project shows how automation can solve real world problems such as wasting energy and unreliable street lighting. The Automatic Street Light System helps reduce human effort, saves electricity, lowers costs, and improves road safety at night. It enables students to aware of the practical knowledge of sensors, microcontrollers, simulation tools, and embedded system design, which are important skills used engineering.

1.6 Scope and Limitations

Scope of the Study

This project focuses on designing and simulating an Automatic Street Light System for a single street light using sensors and a microcontroller. The system turns the light on at night, off during the day, and adjusts brightness based on movement. The work covers simulation in Tinkercad and Proteus, plus basic PCB design and coding.

Limitations of the Study

1. It relies solely on electricity and does not include alternative power sources such as solar energy.
2. Advanced features like remote control, scheduling, or integration with a network of lights are not included in this project.
3. The system can control only one street light at a time.

1.7 Organization of the Study

This report has five chapters. Chapter One covers the introduction, problem statement, objectives, research questions, significance, scope, and limitations. Chapter Two looks at related research, existing systems, and the tech behind sensor-based and microcontroller projects. Chapter Three dives into how everything is put together, from block diagrams and circuit design to simulation, PCB work, coding, and building a prototype. Chapter Four shares the results, discussion, and analysis. Chapter Five wraps up with the conclusion, recommendations, what this project adds to the field, its limitations, and some ideas for the future.

CHAPTER TWO: LITERATURE REVIEW

2.1 Introduction

In this literature review, we go through previous research and projects that has done that are related to automatic street lighting and embedded systems. It helps identify what has already been achieved, the techniques used, and gaps that this project will address. This chapter also highlights analog circuit techniques, microcontroller selection, and simulation tools like Tinkercad and Proteus.

2.2 Review of Existing Similar Projects

Here are a few projects related to Automatic Street light System

1. **Simple LDR-based street lights**
    i. These use light-dependent sensors to detect darkness.
    ii. A transistor or relay turns a single lamp on or off.

    Limitation: Usually only one light, no motion detection.

2. **Arduino-controlled street lights**
    i. Arduino reads analog values from an LDR.
    ii. LEDs turn ON at night and OFF during the day.
    iii. Can include multiple lights and PWM control for brightness.

    Limitation: Some systems still lack motion-based energy saving.

3. **PIR + LDR street lights**

i. Combines a light sensor and motion sensor.

ii. Lights remain dim at night unless motion is detected.

iii. Efficient for energy saving.

Limitation: Slightly more complex and expensive.

4. **Smart street light that uses solar panels**

i. This uses solar panels to charge batteries during the day.

ii. At night, the then battery powers the street light based on LDR and motion sensors.

Limitation: Higher cost and requires environmental durability.

5. **Other research articles / projects**

i. Journals on "energy-efficient street lighting" show that automation reduces electricity consumption by up to 40%.

ii. Most systems use microcontrollers (Arduino, PIC, ESP32) and analog circuits (voltage dividers, transistor switches, relays).

2.3 Analog Circuit Techniques in Robotics

Analog circuits are very important when dealing with sensor interface:

i. Voltage dividers: This convert sensor resistance changes (like LDR) into voltage signals readable by microcontrollers.

ii. Transistors: This act as switches to drive LEDs or relays when voltage threshold is reached.

iii. Resistors: This limit current to prevent damage to sensors and actuators.

iv. Capacitors: This filters sensor signals for an easy and smoother readings.

This project uses LDR+10k resistor as voltage divider and a simple digital output to control LEDs.

2.4 Microcontroller Selection and Justification

Arduino Uno R3 was used because:

1. Easy to use and widely supported.
2. Has multiple analog input pins (for LDR).
3. Supports PWM for fading lights.
4. Easy to simulate in Tinkercad.

Other microcontrollers like ESP32 or STM32 could also work but are more complex and not really necessary for a simple street light system.

2.5 Use of Proteus and Tinkercad in Industry/Education

- Proteus**:** This is a professional tool for simulating electronics, analog and digital circuits, and microcontroller programming.
- Tinkercad**:** Browser-based, beginner-friendly platform for simulating Arduino circuits and sensors.
- Both tools allow:

i. Testing circuits before physical implementation
ii. Debugging and optimization
iii. Visualization of hardware behavior

2.6 PCB Design Trends in Robotics

Before we talk about the trends, we need to understand what PCB actually means

Printed Circuit Board (PCB) is a platform in electronics that supports and connects components electrically like resistors, capacitors and chips using conductive pathways, all etched on a non-conductive surface.

Design trends

i. Sustainability: In PCB designs, eco-friendly materials are used and manufacturing processes favour the environment.
ii. Flexibility: In other for bending and stretching, PCB designs have evolved and have become flexible.
iii. Miniaturization: Power is also needed in smaller devices so in other to achieve such feats High-Density Interconnect Printed Circuit Board where created to pack more power into smaller spaces
iv. This project can show how a single-light control board could be made ready for PCB**,** even if simulation only.

2.7 Research Gap Identification

i. Most existing LDR-based street lights are **single-light only**, with no motion-based energy saving.

ii. Some Arduino solutions exist, but many tutorials lack **proper simulation screenshots, analog circuit explanations, or PCB planning**.

iii. This project addresses these gaps by:

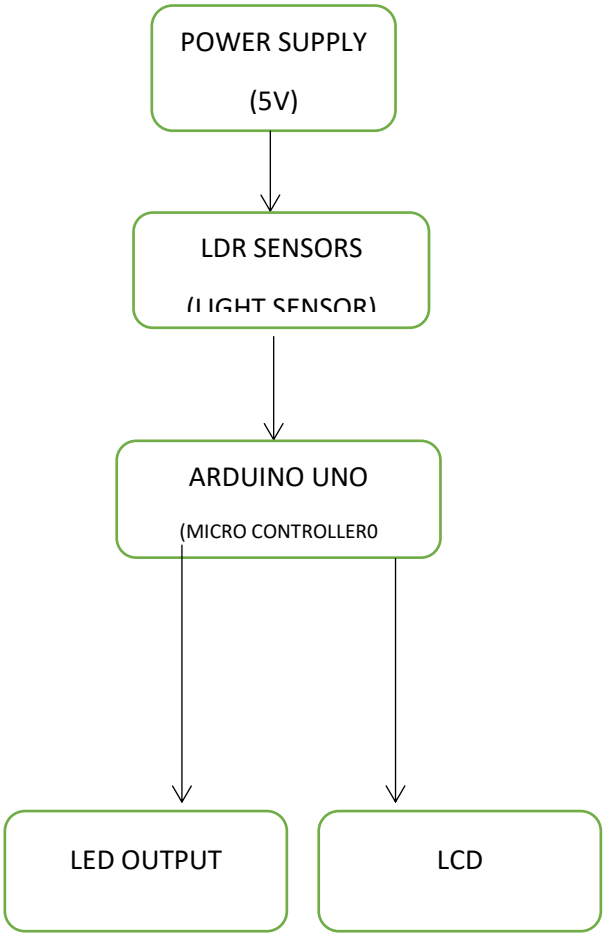
1. Simulating the circuit in Tinkercad
2. Explaining analog and digital integration
3. Designing a conceptual PCB for the street light

CHAPTER THREE: METHODOLOGY

3.1 System Block Diagram

The automatic street light system is designed to operate based on the environments light intensity.

The block diagram of the system is as follows:



Explanation**:**

i. LDR (Light Sensor): This detects the amount of light in the surrounding environment. High resistance indicates darkness, low resistance indicates brightness.
ii. Arduino R3: This reads analog voltage from LDR, compares it to a threshold, and decides whether to turn the LED ON or OFF.
iii. LED (Street Light): This simulates a street light, turning ON in darkness and OFF in daylight.
iv. LCD (Display Unit): This displays the output whether LIGHT OFF or LIGHT ON

This block diagram clearly shows the flow of information from sensing to action.

3.2 Hardware Design

3.2.1 Analog Circuit Design

In this project, the Light Dependent Resistor (LDR) was connected together with a 10 kΩ resistor to form a voltage divider. The aim of this circuit is to convert light intensity into a variable voltage that can be read by the Arduino.

The connections were made as follows:

i. The 10 kΩ resistor was connected to A0, and the other end was connected to GND.
ii. One leg of the LDR was connected to **5 V**.
iii. The second leg of the LDR was placed in the same row as the A0.

How it works

i. During the day, the resistance of the LDR becomes very small. This makes the voltage at A0 drop.

ii. During the night time, the LDR resistance increases. This makes the voltage at A0 rise.

The Arduino reads this voltage and decides when to turn the street light ON or OFF.

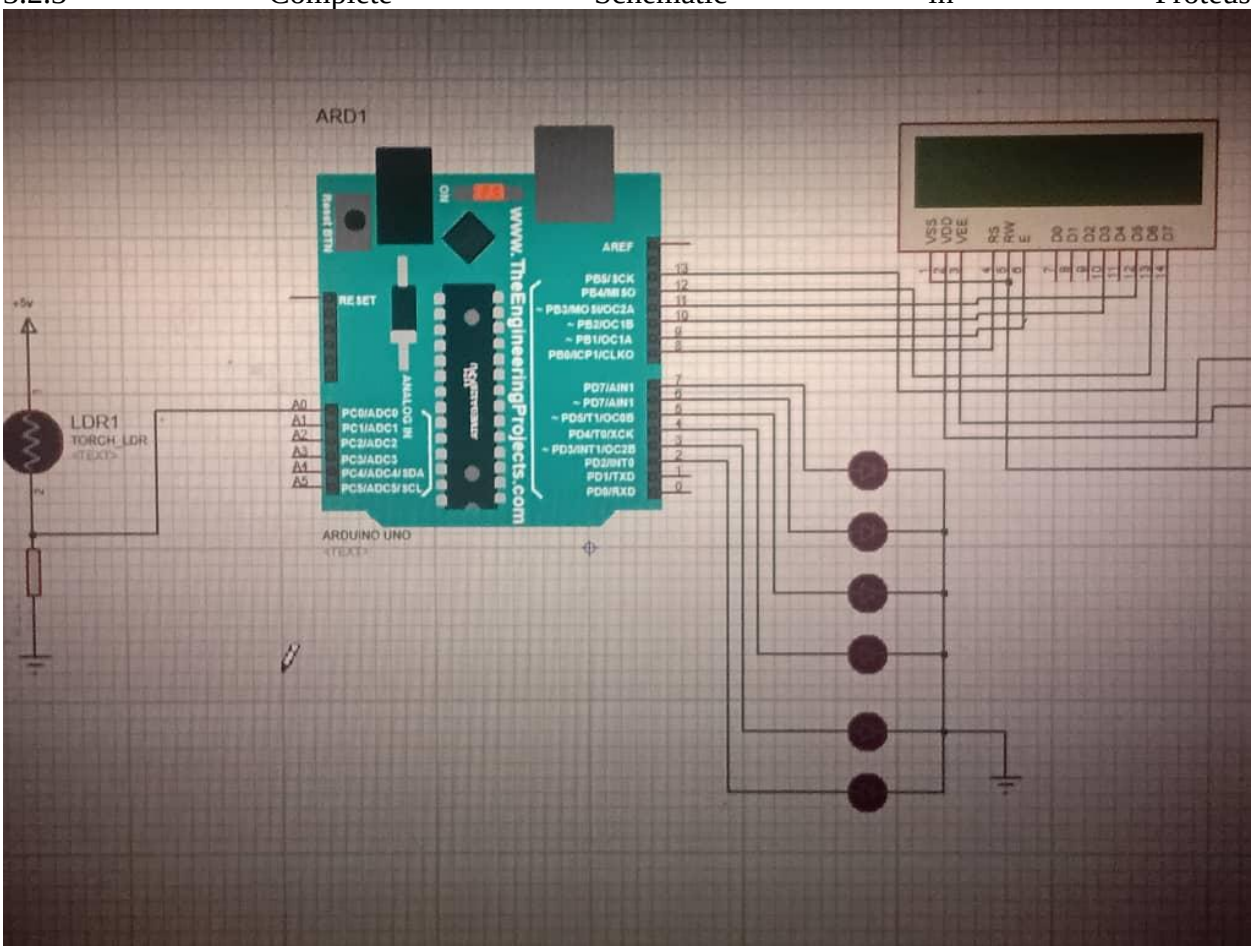
3.2.2 Microcontroller Selection & Pin Mapping

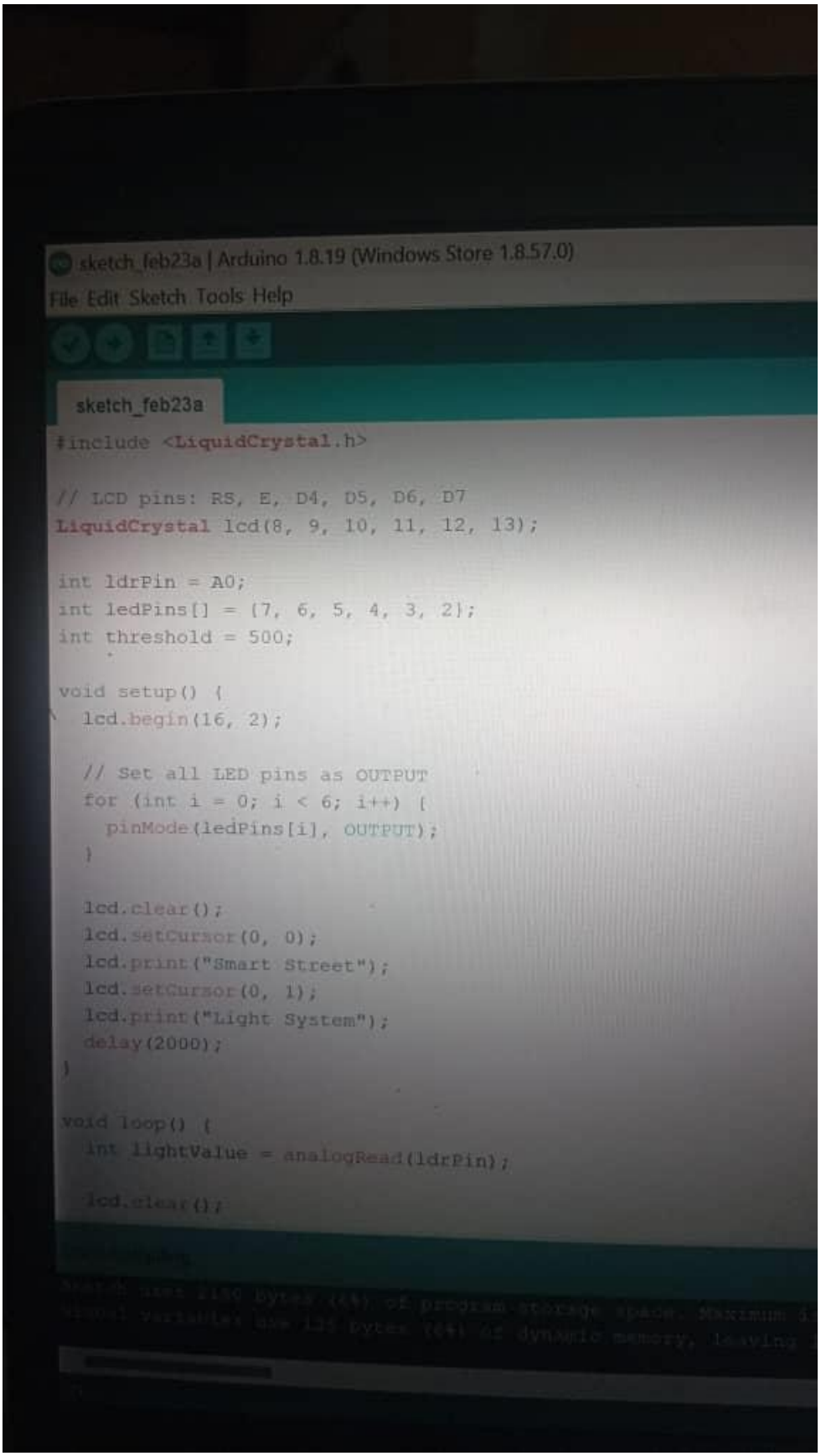
i. Arduino Uno R3 is used for its simple and easy and it is compatible with TinkerCAD.

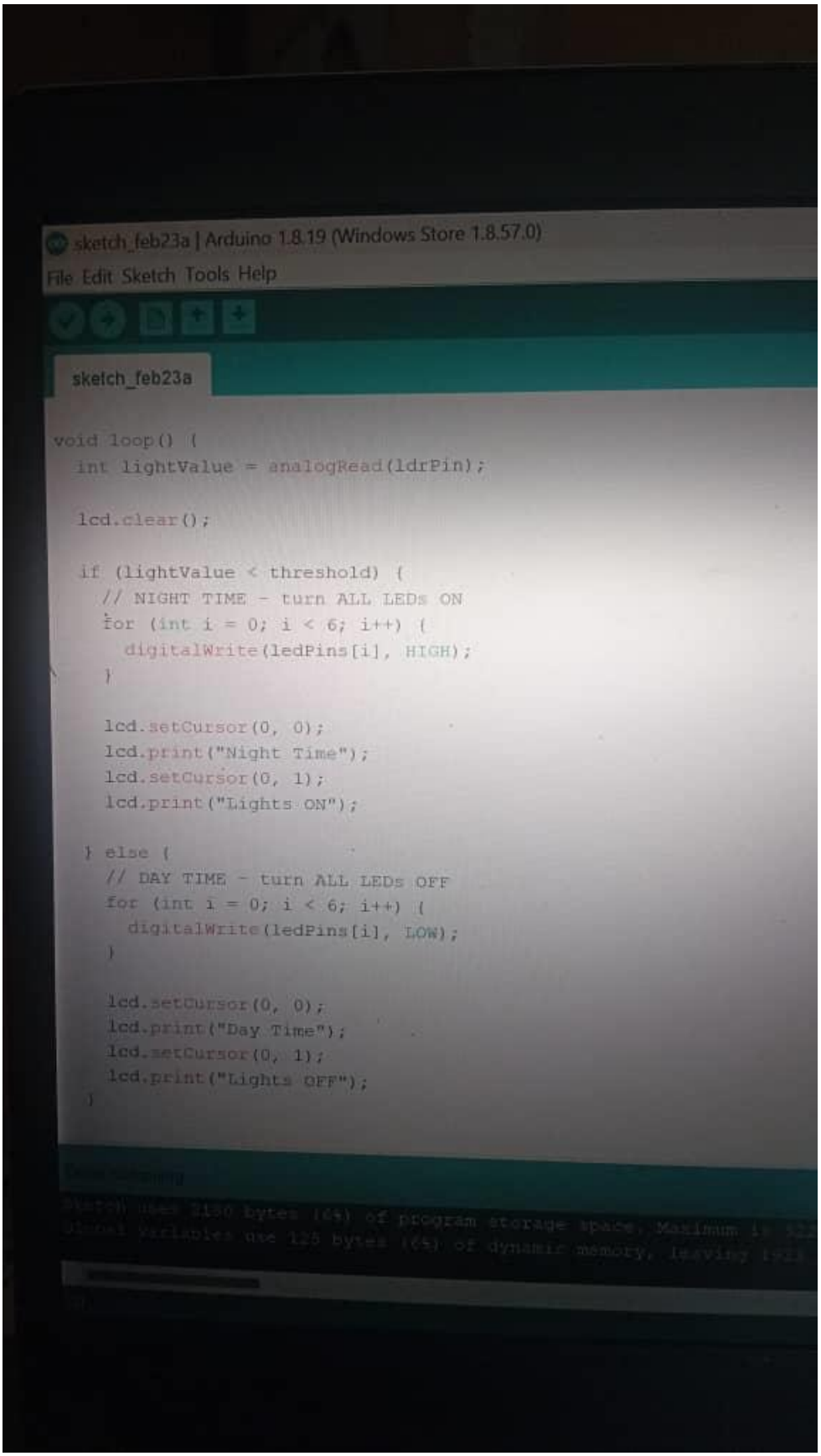
ii. Pin mapping: This is the pin mapping for this project in **TinkerCAD**

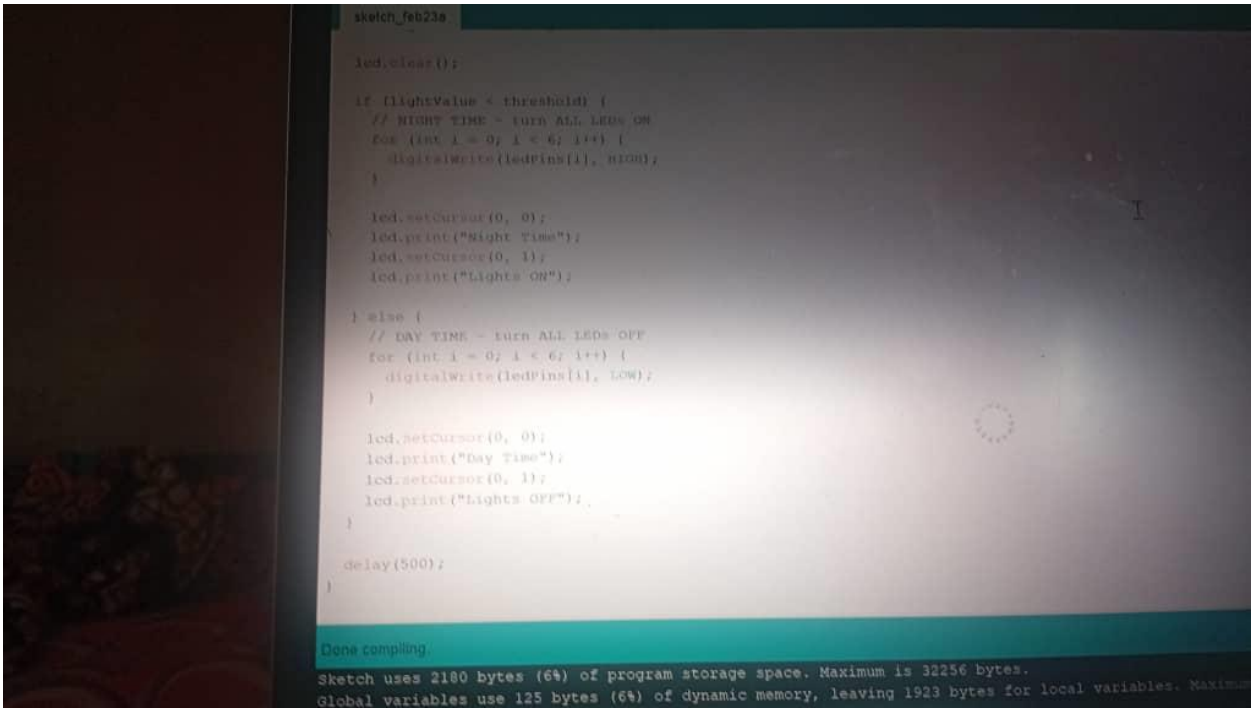
Components	**Arduino pin placement**
LDR	5v, A0
LED	GND
10 kΩ resistor	A0, GND
220Ω resistor	2, 3, 4, 5, 6, 7

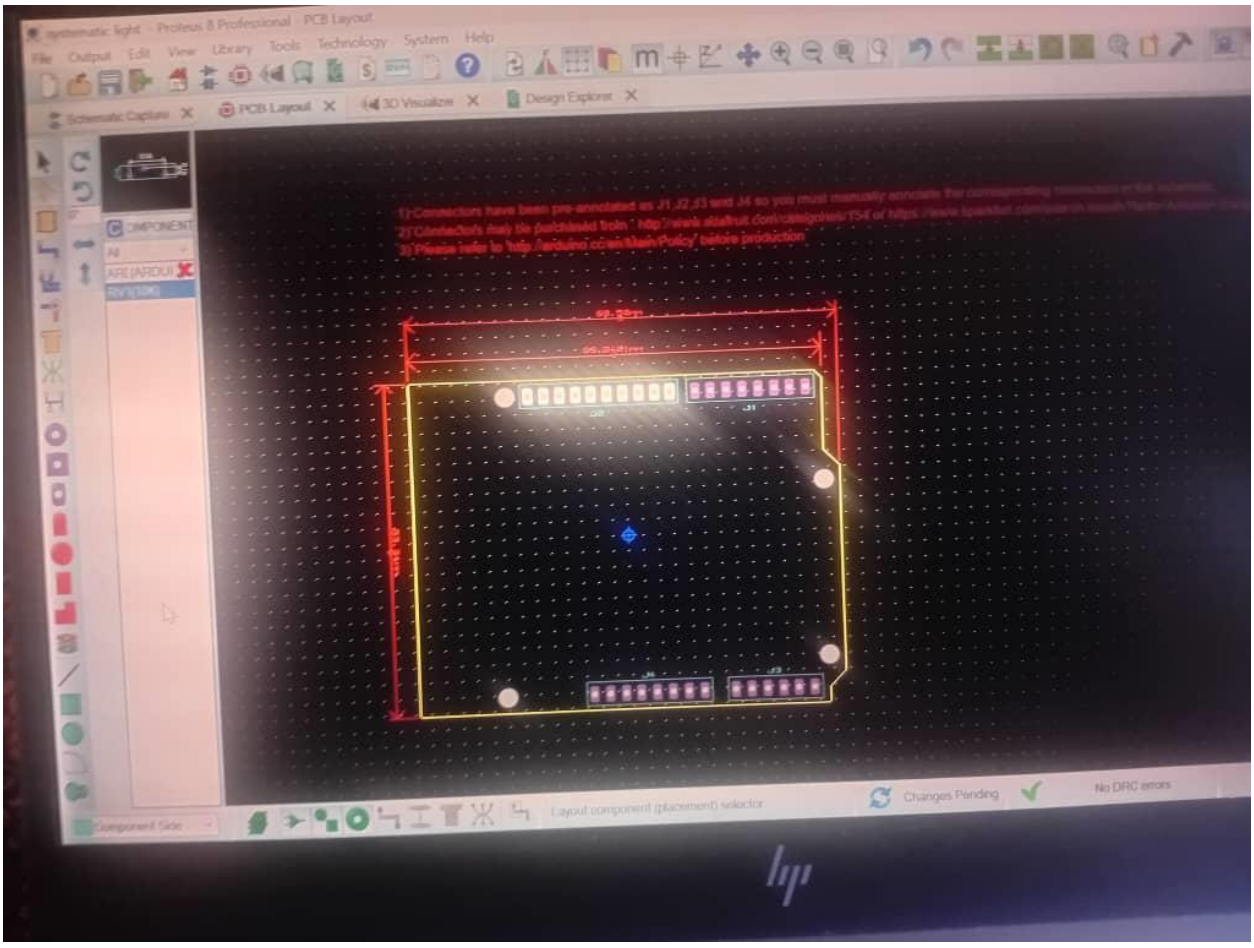
3.2.3 Complete Schematic in Proteus




sketch_feb23a | Arduino 1.8.19 (Windows Store 1.8.57.0)
File Edit Sketch Tools Help
sketch_feb23a
#include <LiquidCrystal.h>
// LCD pins: RS, E, D4, D5, D6, D7
LiquidCrystal lcd(8, 9, 10, 11, 12, 13);
int ldrPin = A0;
int ledPins[] = {7, 6, 5, 4, 3, 2};
int threshold = 500;
void setup() {
lcd.begin(16, 2);
// Set all LED pins as OUTPUT
for (int i = 0; i < 6; i++) {
pinMode(ledPins[i], OUTPUT);
}
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Smart Street");
lcd.setCursor(0, 1);
lcd.print("Light System");
delay(2000);
}
void loop() {
int lightValue = analogRead(ldrPin);
lcd.clear();


sketch_feb23a | Arduino 1.8.19 (Windows Store 1.8.57.0)
File Edit Sketch Tools Help
sketch_feb23a
void loop() {
int lightValue = analogRead(ldrPin);
lcd.clear();
if (lightValue < threshold) {
// NIGHT TIME - turn ALL LEDs ON
for (int i = 0; i < 6; i++) {
digitalWrite(ledPins[i], HIGH);
}
lcd.setCursor(0, 0);
lcd.print("Night Time");
lcd.setCursor(0, 1);
lcd.print("Lights ON");
} else {
// DAY TIME - turn ALL LEDs OFF
for (int i = 0; i < 6; i++) {
digitalWrite(ledPins[i], LOW);
}
lcd.setCursor(0, 0);
lcd.print("Day Time");
lcd.setCursor(0, 1);
lcd.print("Lights OFF");
}


sketch_feb23a
led.clear();
if (lightValue < threshold) {
// NIGHT TIME - turn ALL LEDs ON
for (int i = 0; i < 6; i++) {
digitalWrite(ledPins[i], HIGH);
}
led.setCursor(0, 0);
led.print("Night Time");
led.setCursor(0, 1);
led.print("Lights ON");
} else {
// DAY TIME - turn ALL LEDs OFF
for (int i = 0; i < 6; i++) {
digitalWrite(ledPins[i], LOW);
}
led.setCursor(0, 0);
led.print("Day Time");
led.setCursor(0, 1);
led.print("Lights OFF");
}
delay(500);
}
Done compiling.
Sketch uses 2180 bytes (6%) of program storage space. Maximum is 32256 bytes.
Global variables use 125 bytes (6%) of dynamic memory, leaving 1923 bytes for local variables.


Schematic Capture
PCB Layout
3D Visualizer
Design Explorer
Changes Pending
No DRC errors

The entire circuit was drawn and simulated in Proteus.

Components used:

i. Arduino Uno
ii. LDR
iii. 10 kΩ resistor
iv. 220 Ω resistor
v. LED/Relay
vi. 5 V source
vii. Ground reference

The schematic clearly shows the LDR voltage divider feeding the analog pin and the output signal controlling the LED/relay.

3.2.4 Simulation Results in Proteus

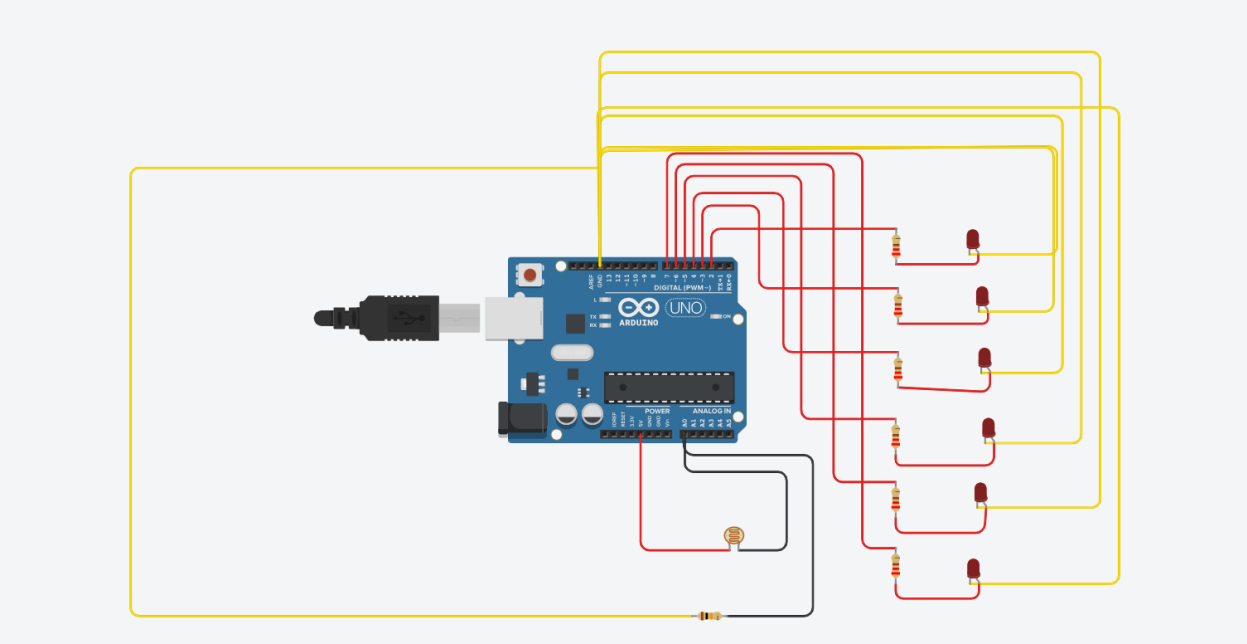
The simulation was performed and the LDR light level was varied.

Observations:

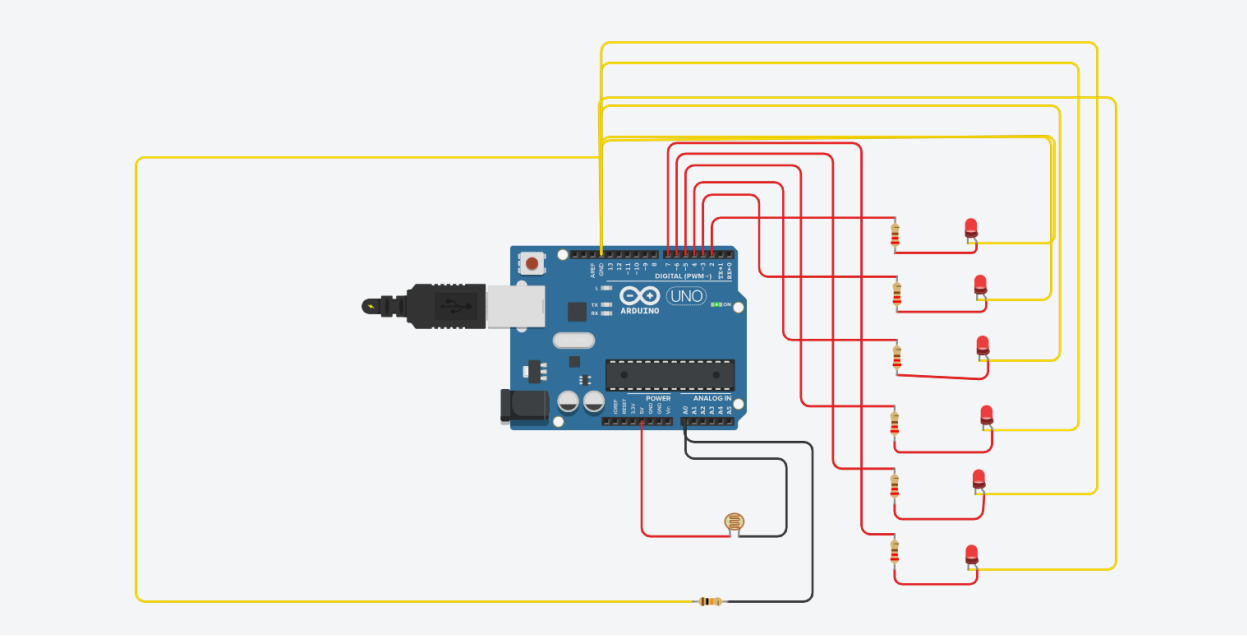
- When light intensity was high, the LED remained OFF.
- When light intensity decreased, the LED turned ON automatically.
- The ADC readings varied proportionally with illumination.

3.2.5 Tinkercad 3D Simulation

The same circuit was re-created in Tinkercad to visualize hardware connections in a practical form.



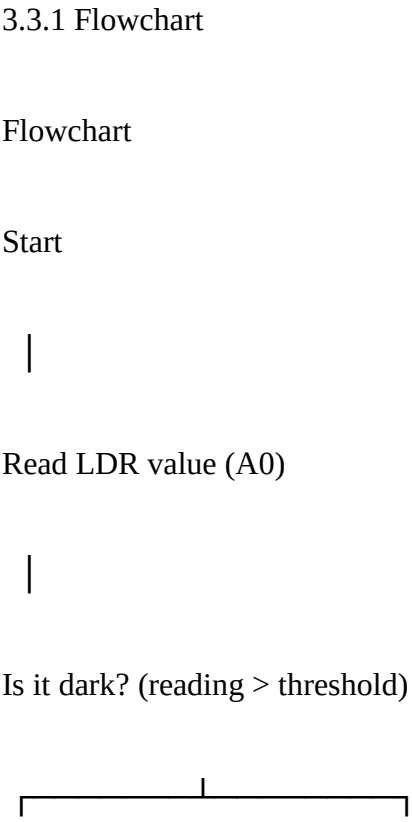
The Serial Monitor displayed live analog readings while adjusting light levels. The LED responded exactly as expected.



3.3 Software Development

3.3.1 Flowchart

Flowchart



3.3.2 Arduino Code

```
int ldrPin = A0;


int streetLights[6] = {2, 3, 4, 5, 6, 7};

int threshold = 500;

void setup() {
  for (int i = 0; i < 6; i++) {
    pinMode(streetLights[i], OUTPUT);
  }
  Serial.begin(9600);
}

void loop() {
  int lightValue = analogRead(ldrPin);
  Serial.println(lightValue);

  if (lightValue < threshold) {
    for (int i = 0; i < 6; i++) {
      digitalWrite(streetLights[i], HIGH);
    }
  } else {
    for (int i = 0; i < 6; i++) {
      digitalWrite(streetLights[i], LOW);
    }
  }

  delay(500);
}
```

3.4 Testing and Validation Procedure

Testing focused on:

1. Sensor response to different light levels
2. ADC accuracy

3. Stability of output switching
4. Power consumption
5. Noise susceptibility

Procedure:

1. Vary light intensity gradually
2. Record sensor readings
3. Observe LED/relay behavior
4. Compare results with expected threshold response
5. Repeat multiple times for consistency

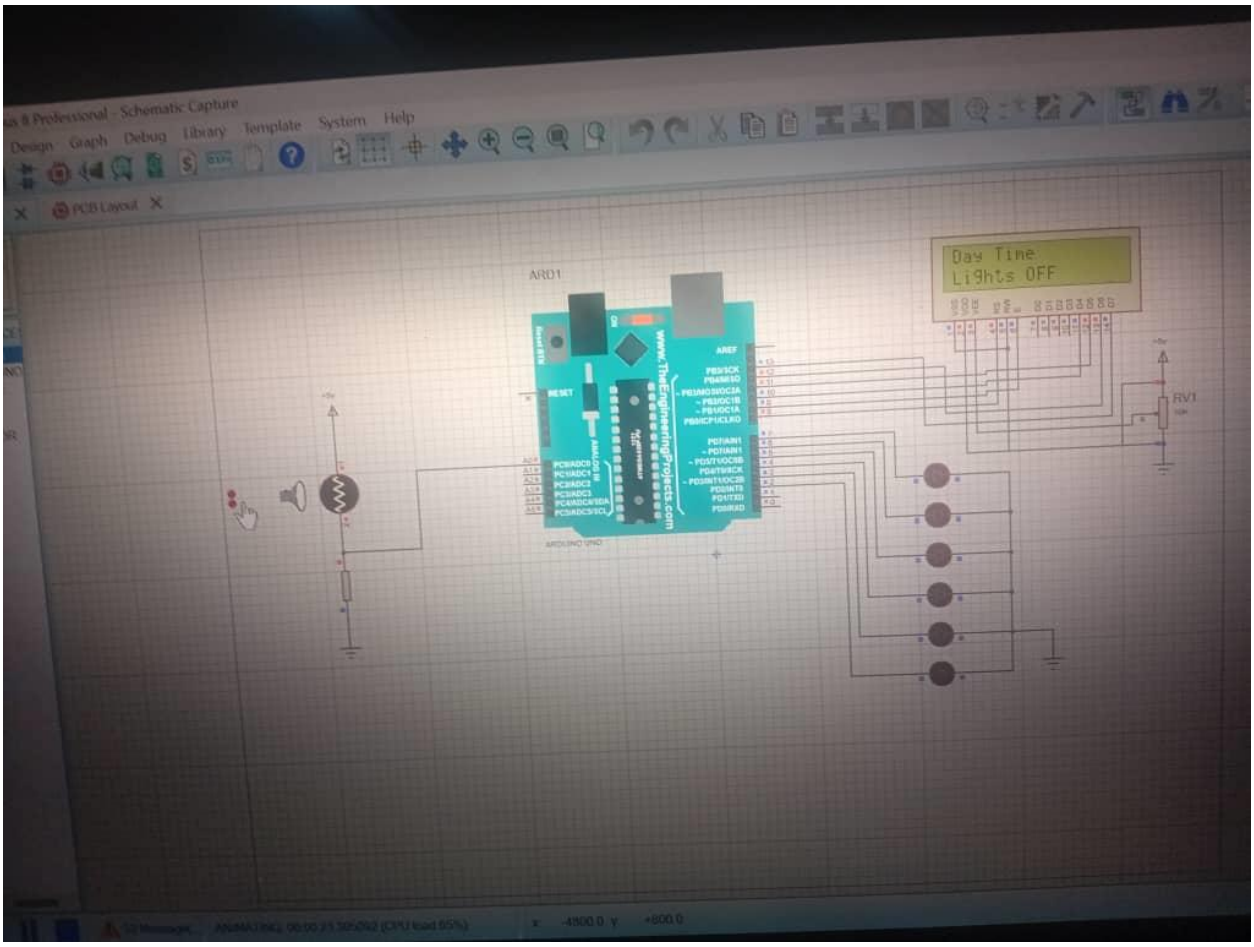
3.5 Ethical & Safety Considerations

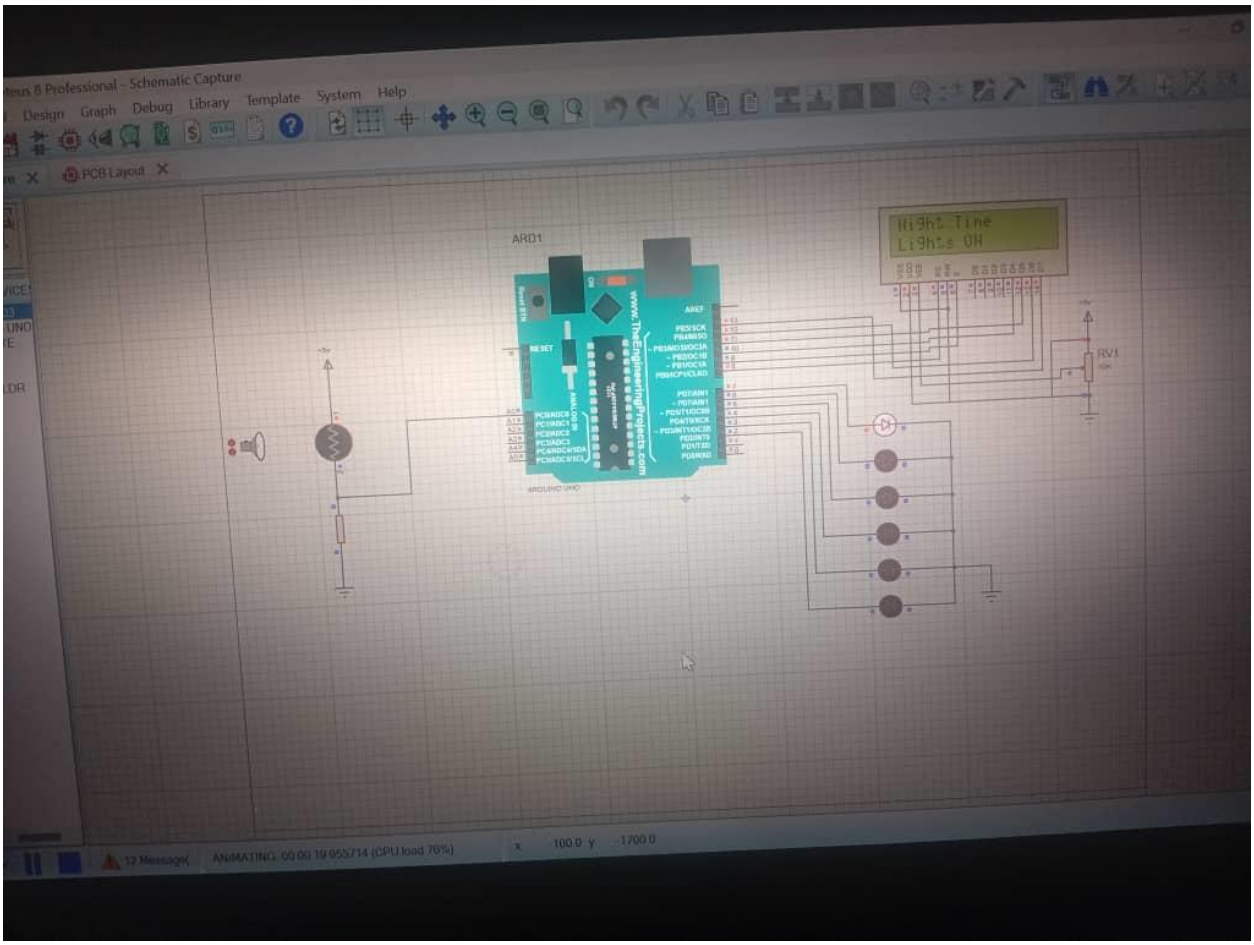
1. 5volts is used to prevent electric shock.
2. Be careful so as not to harm your eyes from direct exposure to bright light
3. Ensure you follow the proper disposal guidelines.
4. Ensure the device is not hazardous to the environmental.

CHAPTER FOUR: RESULST AND DISCUSSION

4.1 Simulation Results

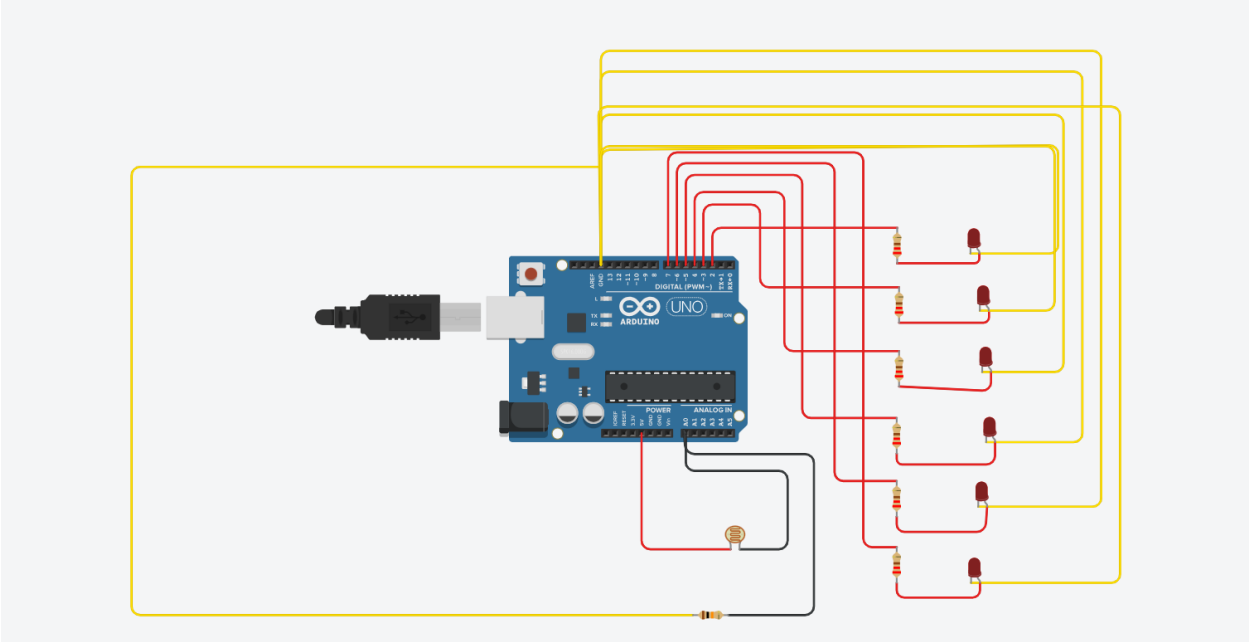
4.1.1 Proteus Simulation Results



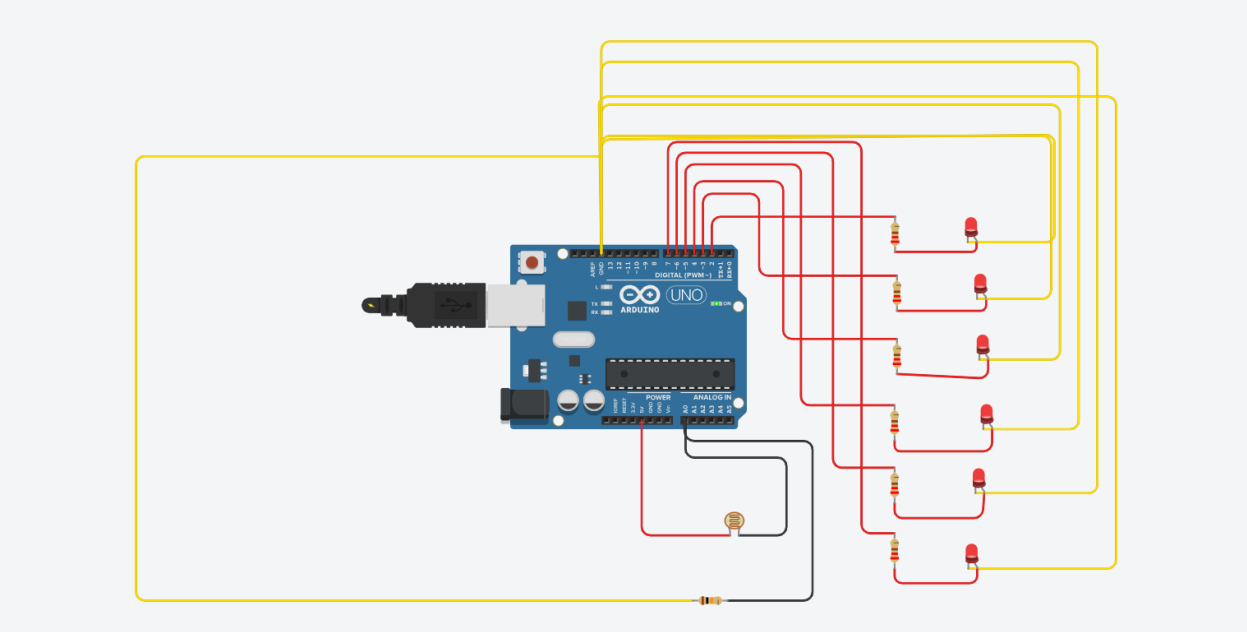


Our circuit was first tested in Proteus before implementation. The LDR resistance was made to represent **day** and **night** conditions.

4.1.2 Tinkercad Simulation Results



The circuit was re-created and tested in tinkercad for accuracy and functionality



4.2 Discussion and Findings

Here are our findings:

1. The use of LDR sensors helps to reduce cost of electricity.
2. The use of microcontrollers reduces the involvement of humans when it comes to switching on and off of the street light
3. Simulation before implementing helps to reduce design error

4.3 Problems Encountered and Solutions

Problem

1. LEDs didn't turn on due to wrong wiring.
2. System not working
3. Proteus failed to run initially

Solution

1. Carefully arranged the wires
2. Carefully inspecting wires and connecting them properly
3. Switched to Tinkercad for simulation

CHAPTER FIVE: CONCLUSING AND RECOMMENTION

5.1 Summary of Findings

We built an Automatic Street Light System with an LDR sensor, an Arduino, and LED lights. It worked just like we hoped.

Here's what stood out after running simulations and testing the hardware:

1. The LDR picked up changes in light around it.
2. The Arduino took those readings and handled the lights on its own.
3. Power use dropped a lot because the lights only switched on when they actually needed to.
4. Testing everything in Proteus and Tinkercad made it easy to find mistakes before we started wiring things up.
5. All in all, this setup proved dependable, affordable, and flexible enough to work in both cities and the countryside.

5.2 Conclusion

The automatic street light system designed in this project proves that streets lights can also be automated using electronic components and microcontrollers

This System:

1. Conserves electrical energy
2. Reduce operational cost
3. Improves night visibility and safety

4. Eliminates human dependency on manual switching

5.3 Recommendations

For Industry

1. Use automatic light systems in streets, campuses, estates, and car parks to prevent wastage of electricity.
2. Add solar panels and rechargeable batteries to rely less on the grid.
3. Protect outdoor sensors and electronics with weather-proof casings
4. Include smart communication modules like LoRa or Wi-Fi for remote monitoring.

For Future Students

1. Start simulations early to avoid last-minute errors.
2. Document every step — wiring, code, screenshots, results.
3. Test circuits in stages rather than building everything at once.
4. Understand the code instead of copying and pasting.
5. Always back up files on GitHub as required.

5.4 Contribution to Knowledge

This project contributes to academic and practical knowledge by:

- Demonstrating integration of analog sensing with microcontroller control,
- Showing how simulations reduce prototyping risk,
- Providing a simple low-cost solution for automatic lighting,

- Illustrating PCB migration from breadboard to permanent hardware.

It also serves as a learning framework for future automation projects.

5.5 Limitations

Despite its success, the project has some limitations:

1. Depends on electricity supply (no solar backup).
2. LDR readings can be slightly affected by dust, shadows, or sudden light changes.
3. No wireless monitoring or fault detection.

5.6 Suggestions for Future Work

Future improvements may include:

1. Adding solar charging and battery storage.
2. Introducing motion detection for extra brightness when movement is detected.
3. Adding IoT connectivity for remote monitoring and data logging.
4. Using more advanced sensors such as photodiodes or light-intensity modules.

References

1. For learning Tinkercad
    i. https://www.tinkercad.com/learn/circuits

ii. https://www.youtube.com/watch?v=L-BWfZEaeps&t=12s

2. PBC

i. https://www.wonderfulpcb.com/blog/emerging-trends-popular-pcbs-used-in-artificial-intelligence/

3. Proteus

i. https://youtu.be/I815ae7xtko?si=HKsK6Hr0xEMFYpPC