KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY

ELECTRICAL AND ELECTRONICS ENGINEERING(OBUASI)
(YEAR THREE)

COURSE NAME; MICROPROCESSORS (COE 381)

INTELLIGENT HEADLIGHT CONTROL SYSTEM

GROUP FOUR(4) MEMBERS

SAIBA ALHASSAN-1874122

GLORY KOJO GYASI-1872722

AFRIYIE FELIX OPOKU-1871122

QUAYE EBENEZER-1874022

OKYEADEA MAXWELL ADDO-1873522

AWUAH EDMUND OBENG-1872022

TABLE OF CONTENT

Catalog

PROBLEM STATEMENT(ABSTRACT)

1. Background & Problem Statement

Vehicle headlights play a crucial role in night driving, but improper use of high beams can cause glare, leading to poor visibility for other drivers and increasing the risk of accidents. Many vehicles still rely on manual headlight control, which may not always be adjusted correctly or in time. An automatic headlight system that responds to real-time conditions can help improve road safety and reduce driver fatigue.

2. **Motivation & Importance**

With advancements in sensor-based automation, modern vehicles are now incorporating adaptive headlight systems. However, these systems are often expensive and not available in older vehicles. This project aims to develop a cost-effective prototype for an **Intelligent Headlight Control System**, demonstrating how simple electronic components can improve vehicle safety.

3. **Objectives**

The goal of this project is to design and implement an automatic headlight control system using an Arduino Uno, a Light Dependent Resistor (LDR), an Ultrasonic distance sensor, and an LED (simulating headlights). Specifically, the system should:

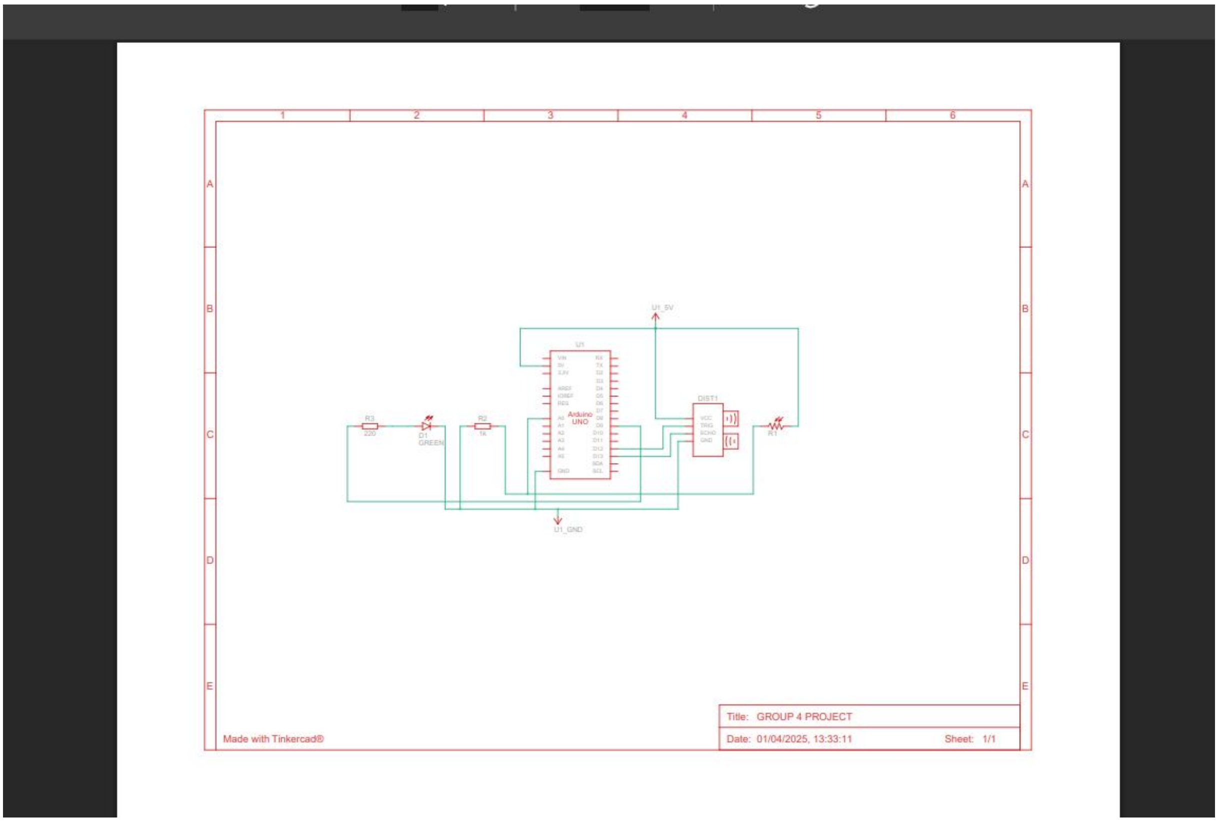
- Detect ambient light levels and automatically turn headlights ON or OFF.
- Identify oncoming vehicles and reduce brightness to minimize glare.
- Operate in a way that is efficient, cost-effective, and adaptable for real-world applications.

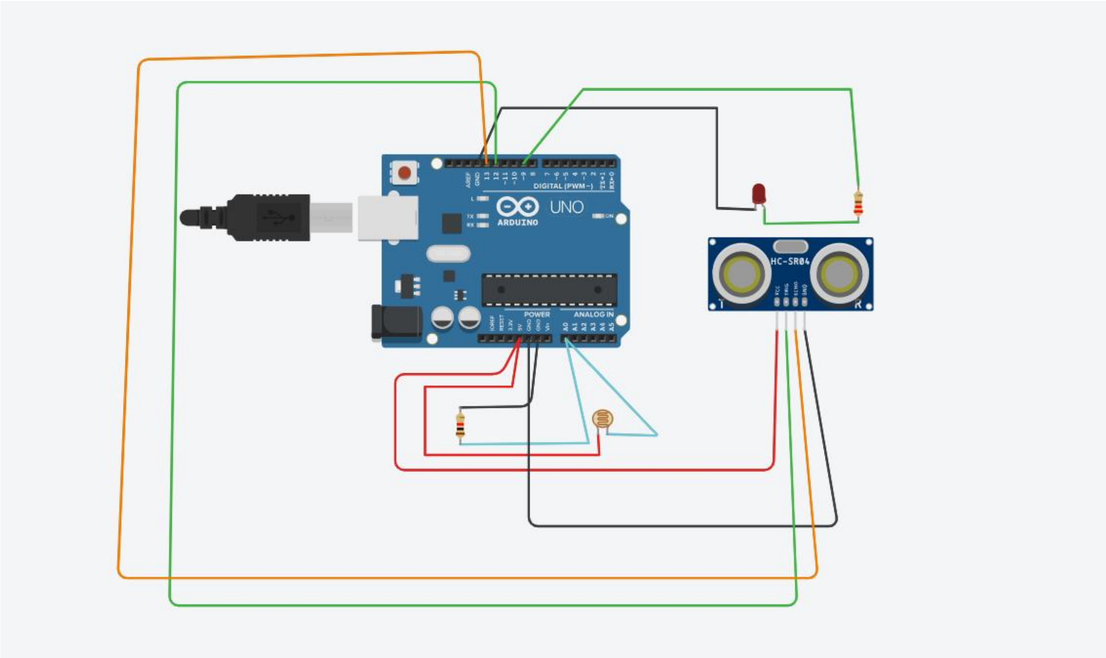
4. **Scope of the Project**

This project is developed as a prototype using Tinkercad simulation. The LDR and Ultrasonic Distance sensor provide real-time inputs, and the Arduino processes these signals to adjust LED brightness via Pulse Width Modulation (PWM). While this project focuses on simulation, the same concept can be applied to real vehicle headlights with minor modifications.

1. System Block Diagram

The system consists of three main components: sensors, a microcontroller (Arduino), and an output (LED as the headlight). Below is a simplified schematic view of the system :





CIRCUIT VIEW

Circuit Diagram & Components Used

2.1 Components List

Component	Specifications
Arduino Uno	Microcontroller
LDR (Light Sensor)	Detects ambient light
Ultrasonic distance sensor	Detec distance of vehicles
LED (Headlight Simulation)	Output light source
Resistors	1kΩ, 230Ω (for current control)
Connecting Wires	For circuit connections

2.2 Circuit Description

- The LDR is connected to Analog Pin A0, with a 230Ω pull-down resistor to stabilize readings.
- The Ultrasonic distance sensor is connected to Digital Pin 12 and 13, providing HIGH or LOW signals for vehicle detection.
- The LED (simulating a headlight) is connected to Digital Pin 9, with Pulse Width Modulation (PWM) controlling its brightness.
- The circuit is powered through the Arduino Uno (5V supply).

3. System Operation & Working Principle

The Arduino Uno serves as the processing unit, continuously reading data from the LDR and IR sensor to determine the required headlight brightness.

3.1 Light-Based Headlight Control (LDR)

If the ambient light is bright, the headlights remain OFF to save power.

If the environment is dark, the headlights automatically turn ON.

3.2 Vehicle Detection (Ultrasonic distance Sensor)

If no vehicle is detected, the headlights operate at normal brightness.

If an oncoming vehicle is detected, the headlights automatically dim to avoid glare.

4. Software Implementation

The system is programmed using Arduino IDE, and the logic is implemented as follows:

1. Read sensor values from the LDR (A0) and UD sensor (D12,13).

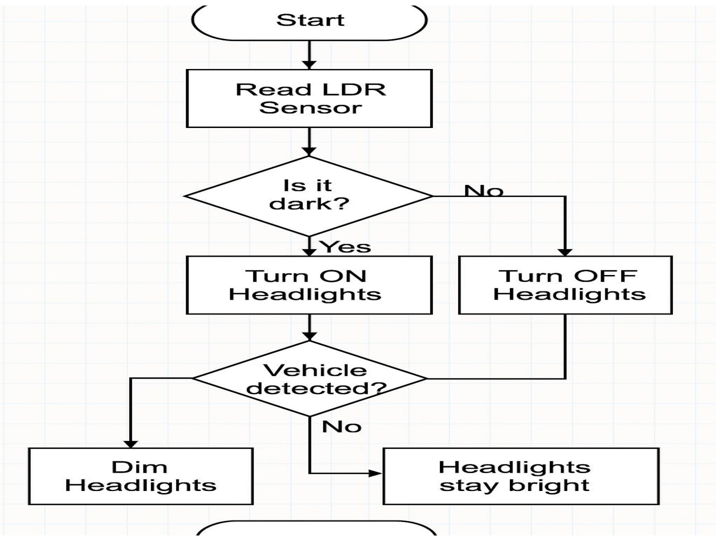
2. Determine light conditions and control the LED brightness accordingly.

3. Use Pulse Width Modulation (PWM) on Pin D9 to adjust brightness levels.

4. Serial Monitor debugging is used to check sensor readings.

5. Flowchart of the System

Below is a simplified flowchart of our system



6. Simulation in Tinkercad

The circuit was designed and tested in Tinkercad to verify functionality.

Different light conditions were simulated using the LDR slider.

The Ultrasonic distance sensor’s state was manually toggled to check vehicle detection.

Adjustments were made to resistor values and LED brightness levels to ensure safe current limits.

6. Code Implementation

This section provides details on how the **Intelligent Headlight Control System** was programmed. It includes the **final Arduino code**, an explanation of **sensor readings**, the difference between **analog and digital inputs**, and the **PWM control logic for LED brightness adjustment**.

6.1 Final Arduino Code

```
const int LDR_PIN = A0;    // LDR connected to analog pin A0
const int LED_PIN = 9;     // LED connected to digital pin 9 (PWM pin)
const int TRIG_PIN = 12;   // Trigger pin for ultrasonic sensor
const int ECHO_PIN = 13;   // Echo pin for ultrasonic sensor

long duration;
int distance;
int ldrValue;
int ldrBrightness;
int ledBrightness;

void setup() {
  Serial.begin(9600);  // Initialize serial communication
  pinMode(LED_PIN, OUTPUT);  // Set LED pin as output
  pinMode(TRIG_PIN, OUTPUT); // Set trigger pin as output
  pinMode(ECHO_PIN, INPUT);  // Set echo pin as input
}

void loop() {
  // Read the value from the LDR (0 to 1023)
  ldrValue = analogRead(LDR_PIN);

  // Debugging: Print the LDR value for inspection
  Serial.print("LDR Value: ");
  Serial.println(ldrValue);

  // Map the LDR value to a PWM range (0 to 255) for LED brightness
  // When more light is detected (higher LDR value), dim the headlights (lower PWM)
  // When less light is detected (lower LDR value), brighten the headlights (higher
PWM)
  ledBrightness = map(ldrValue, 0, 1023, 255, 0);  // More light = dimmer, less light =
brighter

  // Trigger the ultrasonic sensor to send a pulse
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
```

```
  // Measure the duration of the pulse
  duration = pulseIn(ECHO_PIN, HIGH);

  // Calculate the distance (in cm)
  distance = duration * 0.034 / 2;

  // Print the distance for debugging
  Serial.print("Distance: ");
  Serial.println(distance);

  // If a vehicle is detected within 20 cm, dim the headlights
  if (distance < 20) {  // Vehicle is very close
    ledBrightness = map(distance, 0, 20, 255, 50);  // Dim LED when close vehicle
detected
    Serial.println("Vehicle Detected, Headlights Dimmed");
  }

  // Ensure LED brightness is within the PWM range (0 to 255)
  ledBrightness = constrain(ledBrightness, 0, 255);

  // Set the LED brightness based on the final value
  analogWrite(LED_PIN, ledBrightness);

  // Delay before the next loop
  delay(500);
}


const int LDR_PIN = A0;    // LDR connected to analog pin A0
const int LED_PIN = 9;     // LED connected to digital pin 9 (PWM pin)
const int TRIG_PIN = 12;   // Trigger pin for ultrasonic sensor
const int ECHO_PIN = 13;   // Echo pin for ultrasonic sensor

long duration;
int distance;
int ldrValue;
int ledBrightness;
const int ldrThreshold = 600;  // Adjust based on your environment
const int maxBrightness = 254; // Maximum LED brightness

unsigned long previousMillis = 0;
const long interval = 100;  // Adjusted interval for faster updates

void setup() {
  Serial.begin(9600);  // Initialize serial communication
  pinMode(LED_PIN, OUTPUT);  // Set LED pin as output
  pinMode(TRIG_PIN, OUTPUT); // Set trigger pin as output
```

```
  pinMode(ECHO_PIN, INPUT);  // Set echo pin as input
}

void loop() {
  unsigned long currentMillis = millis();

  // Only update every 'interval' milliseconds
  if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;

    // Read the value from the LDR (0 to 1023)
    ldrValue = analogRead(LDR_PIN);
    Serial.print("LDR Value: ");
    Serial.println(ldrValue);

    // If daytime (LDR detects high brightness), turn off LED completely
    if (ldrValue > ldrThreshold) {
      analogWrite(LED_PIN, 0);
      Serial.println("Daytime detected, LED turned OFF");
    }
    else {  // Nighttime scenario
      Serial.println("Nighttime detected, adjusting LED brightness...");

      // Trigger the ultrasonic sensor
      digitalWrite(TRIG_PIN, LOW);
      delayMicroseconds(2);
      digitalWrite(TRIG_PIN, HIGH);
      delayMicroseconds(10);
      digitalWrite(TRIG_PIN, LOW);

      // Measure distance
      duration = pulseIn(ECHO_PIN, HIGH);
      distance = duration * 0.034 / 2;

      Serial.print("Distance: ");
      Serial.println(distance);

      // Adjust LED brightness based on distance (with a minimum brightness of 20)
      if (distance <= 100) {  // If vehicle is within 100 cm
        ledBrightness = map(distance, 100, 0, maxBrightness, 20);  // Map distance to
brightness
      } else {
        ledBrightness = maxBrightness;  // If further than 100cm, LED at max brightness
      }

      // Apply the brightness to LED
      analogWrite(LED_PIN, ledBrightness);
```

```
    // Print LED brightness for feedback
    Serial.print("LED Brightness: ");
    Serial.println(ledBrightness);
   }
  }
}
```

6.2 Explanation of Key Sections

Sensor Readings & LED Control Logic

- The system reads sensor values, processes the data, and adjusts the LED brightness accordingly:

- LDR (Light Dependent Resistor) measures ambient light to determine if it's day or night.

- IR sensor detects approaching vehicles to reduce headlight brightness and prevent glare.

- LED control logic determines whether the headlights should be fully on, dimmed, or off based on the sensor values.

Analog vs. Digital Inputs

Component	Input type	description
LDR(Light sensor	Analog(A0)	Reads continous light intensity values

Ultrasonic distance sensor(Vehicle Detection)	Digital(D12,D13)	maasures distnace of vehicles

· **LDR uses an analog pin** because light intensity varies continuously.

Ultrasonic distance **sensors use a digital pin** because it only needs to detect **vehicle presence (YES/NO)**.

PWM Control for Brightness Adjustment

PWM Value (0-255)	Brightness level	When used
0	OFF	During daytime
80	Dimmed	When a vehicle is detected
200	Full brightness	At night with no vehicles

How it works?

analogWrite(LED_Pin, dimBrightness); // Dim brightness (80)

analogWrite(LED_Pin, fullBrightness); // Full brightness (200)

analogWrite(LED_Pin, 0); // Turn off LED

PWM (Pulse Width Modulation) allows us to control **brightness smoothly** rather than just turning the LED **on/off**.

7. Results and Discussion

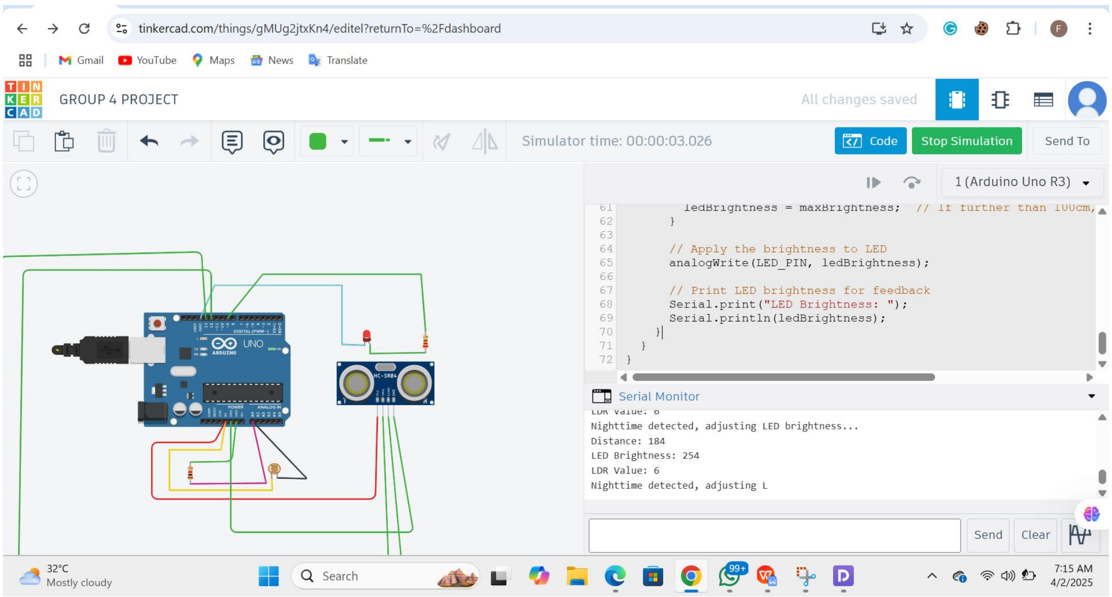
7.1 Simulation Results

After implementing the **Intelligent Headlight Control System** in **Tinkercad**, the following observations were made:

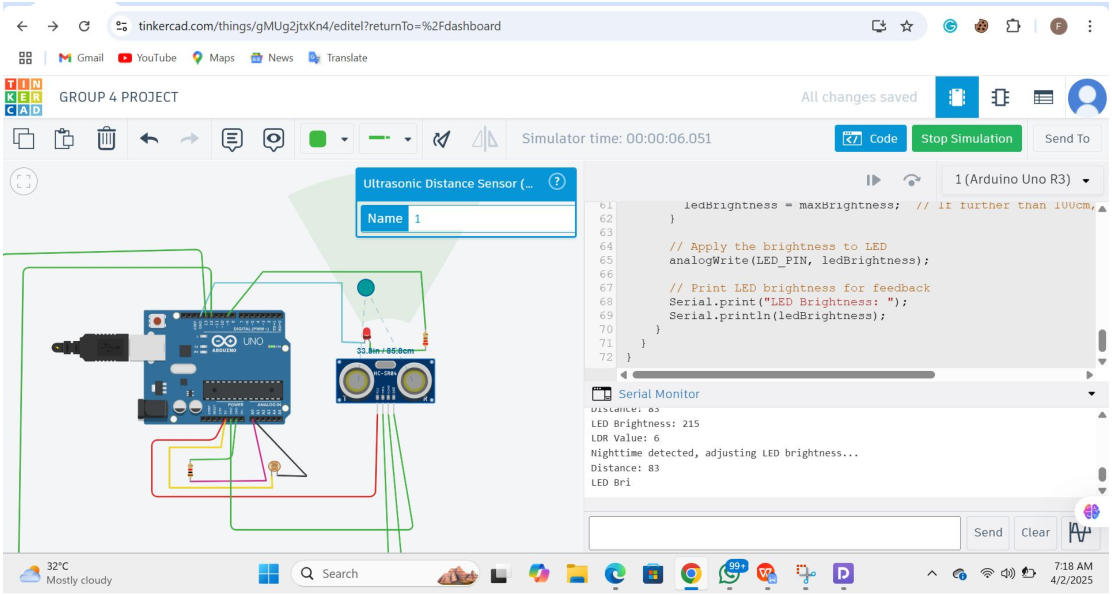
✅ **Headlights turned ON automatically in low light conditions**
✅ **Headlights turned OFF in bright environments**
✅ **When a vehicle was detected at night, the headlights dimmed to reduce glare**
✅ **The system responded instantly to changes in light intensity and vehicle detection**
✅ **After correcting resistor values, the circuit operated within safe current limits**

7.2

Test condition	**Expected Output**	**Actual outcome**	**Pass/fail**
Bright environment (LDR high value)	Headlights OFF	Headlights OFF	✅ Pass
Dark environment, no vehicle	Headlights ON (Full brightness)	**Headlight ON**	✅ Pass
Dark environment, vehicle detected	Headlights Dimmed	Headlights Dimmed	✅ Pass
Incorrect resistor value (Low resistance)	**High current warning**	**High current warning**	❌ Fail (Corrected)
Correct resistor value (230Ω)	**Safe current**	No warning	✅ Pass



Screenshot taken during the night when no car is detected



Screenshot of when vehicle is detected (headlights are dimming)

7.3 Challenges Encountered

❌ Encountered a short circuit along the way while connecting the 230Ω resistor in series with the LED

Challenges Encountered

During the development and simulation of the **Intelligent Headlight Control System**, several challenges were encountered:

❌ Sensor Selection & Calibration

- **Initial IR Sensor Limitations** – The IR sensor was unreliable in detecting vehicles under different lighting conditions, leading to the decision to switch to an **ultrasonic sensor** for better accuracy.
  LDR Sensitivity Issues – The **LDR's readings fluctuated**, requiring careful calibration to ensure proper brightness adjustment.

❌ Hardware & Circuit Challenges

- **High Current Warning** – Initial resistor values caused excessive current through the LED, triggering warnings in Tinkercad. Adjusting to **appropriate resistor values (e.g., 220Ω & 4.7kΩ)** resolved this issue.
  Correct Wiring & Pin Assignments – Ensuring **proper connections** for the ultrasonic sensor and LDR was crucial to avoid **incorrect readings**.

❌ Software & Coding Errors

- **PWM Control for Headlights** – Mapping LDR values to **LED brightness** required tuning to achieve smooth dimming.

Distance Measurement Accuracy – The ultrasonic sensor **sometimes gave unstable readings**, requiring averaging techniques for better accuracy.

❌ Simulation Limitations

- **IR Sensor Adjustments in Tinkercad** – The **IR sensor's behavior was difficult to simulate**, leading to its replacement.

Despite these challenges, adjustments in software, and circuit design helped in achieving a **working prototype** of the Intelligent Headlight Control System.

8. Conclusion & Improvement

8.1 Conclusion

The Intelligent Headlight Control System was successfully designed and simulated to automate vehicle headlights based on ambient light conditions and vehicle detection. The system effectively:
✅ Turns ON headlights in low-light conditions.
✅ Turns OFF headlights in bright environments.
✅ Dims headlights when a vehicle is detected to reduce glare and improve road safety.

The **Intelligent Headlight Control System** successfully automates vehicle headlights by detecting oncoming cars and adjusting brightness accordingly. The system enhances road safety by preventing glare, improving night-time visibility, and reducing driver fatigue. The combination of an **LDR (Light Dependent Resistor) for ambient light sensing** and an **ultrasonic sensor for vehicle detection** ensures an adaptive and reliable headlight control mechanism. The project was successfully simulated using **Tinkercad**, demonstrating its effectiveness in real-world scenarios..

8.2 Impoovement

To further enhance the system, the following improvements can be considered:

✅ **Enhanced Vehicle Detection** – Using a **radar or camera-based system** to improve vehicle detection accuracy in diverse conditions.
✅ **Adaptive Headlight Control** – Implementing **servo motors** to dynamically adjust the headlight angle based on road curvature.
✅ **Machine Learning Integration** – Training an AI model to predict optimal headlight intensity based on traffic and weather conditions.
✅ **Real-World Testing** – Deploying the system in an actual vehicle for **real-time performance evaluation** and adjustments.
✅ **Energy Efficiency** – Integrating **low-power components** and **solar charging** for sustainable operation.

With these improvements, the **Intelligent Headlight Control System** can become a **more efficient, accurate, and user-friendly solution** for modern vehicles.