🔧Index

Project #	Project Name	What It Does	Key Components
1	**Light-Activated Buzzer**	Buzzer turns ON when it's dark	LDR, Buzzer
2	**Obstacle Alert Buzzer**	Buzzer beeps when obstacle is detected	IR, Buzzer
3	**Light-Following Robot**	Robot moves toward light	LDR, Motors
4	**Object Avoiding Robot**	Robot turns when obstacle detected	IR, Motors
5	**Night Patrol Bot**	Robot only runs in dark	LDR, Motors
6	**Daylight Tracker**	Robot stops when it's bright	LDR, Motors
7	**IR Gesture Motor Control**	Move forward/backward with hand wave	IR, Motors
8	**Sunlight Alarm**	Buzzer rings when sunlight appears	LDR, Buzzer
9	**Invisible Line Bot**	Use IR to follow black tape/line	IR, Motors
10	**Speed Modulation Robot**	Speed varies with light intensity	LDR, Motors
11	**Security Alert**	IR sensor detects motion, triggers buzzer	IR, Buzzer
12	**Line Crossing Alert**	IR detects object passing a line, buzzer rings	IR, Buzzer
13	**Basic Obstacle Counter**	Count how many times obstacle comes	IR
14	**Sound Notification Bot**	Move + beep when IR detects something	IR, Motors, Buzzer

15	**Light Level Indicator**	Output HIGH or LOW based on brightness	LDR
16	**No-Light Movement Bot**	Robot only moves in complete darkness	LDR, Motors
17	**Auto Stop in Light**	Robot runs in dark, stops in light	LDR, Motors
18	**Edge Detection Bot**	Bot stops near edge (simulate using black tape)	IR, Motors
19	**Robot with Alert Horn**	Robot with buzzer alert on obstacle	IR, Motors, Buzzer
20	**Night Entry Alert**	Buzzer sounds if person detected in darkness	LDR, IR, Buzzer

🔧 Project 1: Light-Activated Buzzer

🎯 Objective

When the environment becomes **dark**, the **buzzer will beep**. When it's bright, the buzzer stays OFF.

🧩 Components Needed

Component	Quantity
Arduino Nano	1
LDR Module	1
Buzzer	1
Jumper Wires	As needed
5V Battery	1 (or USB for testing)

🔌 Circuit Diagram / Connections

Arduino Nano Pin	Connect To
`A0`	**LDR Signal** pin
`GND`	**LDR GND** and **Buzzer -**
`5V`	**LDR VCC** and **Buzzer +**
`D3`	**Buzzer Signal** (if using active buzzer module)

> 💡 If your LDR module has an analog output pin (`AO`), connect that to `A0`.

🧠 How It Works

- LDR gives high analog value in **dark** and low in **light**.
- If the LDR value is **greater than threshold**, buzzer turns ON.

```ino
const int ldrPin = A0;
const int buzzerPin = 3;

void setup() {
  pinMode(buzzerPin, OUTPUT);
  Serial.begin(9600); // for debugging
}

void loop() {
  int ldrValue = analogRead(ldrPin);
  Serial.println(ldrValue); // view this in Serial Monitor to set threshold

  if (ldrValue > 700) { // adjust threshold as needed
    digitalWrite(buzzerPin, HIGH);
  } else {
    digitalWrite(buzzerPin, LOW);
  }

  delay(200);
}
```

🧪 Testing

1. Upload code using Arduino IDE.
2. Open Serial Monitor and observe values.
3. Cover the LDR to simulate darkness — buzzer should beep.

✅ Optional Enhancements

- Add a **status LED** to show "Dark Mode Active".
- Use **PWM** to make the buzzer beep intermittently.

🔧 Project 2: Obstacle Alert Buzzer

🎯 Objective

When the **IR sensor detects an obstacle**, the **buzzer beeps** to alert you.

🧩 Components Needed

Component	Quantity
Arduino Nano	1
IR Sensor Module	1
Buzzer	1
Jumper Wires	As needed
5V Battery or USB	1

🔌 Circuit Diagram / Connections

Arduino Nano Pin	Connect To
D2	**IR Sensor OUT**
GND	**IR GND** + **Buzzer -**
5V	**IR VCC** + **Buzzer +**
D3	**Buzzer Signal** (optional if using active buzzer)

> ✅ Most IR modules output **LOW (0)** when obstacle is detected, and **HIGH (1)** when clear.

🧠 How It Works

- IR sensor detects reflection when an object is nearby.
- Output pin goes **LOW** on detection.
- Arduino triggers the **buzzer** when it receives a **LOW signal** from the IR sensor.

🗒️ Arduino Code

```
const int irPin = 2;
const int buzzerPin = 3;

void setup() {
  pinMode(irPin, INPUT);
  pinMode(buzzerPin, OUTPUT);
  Serial.begin(9600); // For monitoring IR value
}

void loop() {
  int irValue = digitalRead(irPin);
  Serial.println(irValue); // Check output in Serial Monitor

  if (irValue == LOW) {  // Obstacle detected
    digitalWrite(buzzerPin, HIGH);
  } else {
    digitalWrite(buzzerPin, LOW);
  }

  delay(100);
}
```

🧪 Testing Steps

1. Power the setup via USB or 5V battery.
2. Keep your hand or an object near the IR sensor.
3. Buzzer should beep when obstacle is detected.

✅ Optional Tweaks

- Add delay-based intermittent buzzer.
- Blink an LED instead of buzzer for silent testing.

🔧 Project 3: Light-Following Robot

🎯 Objective

The robot **moves toward a light source** (like a torch or sunlight). When it's dark, it stops.

🧩 Components Needed

Component	Quantity
Arduino Nano	1
LDR Module	1
BO Motors	2
Tyres	2
5V Battery Pack	1
Jumper Wires	As needed

> ❗ *We will run the motors directly through digital pins as you don't have a motor driver. So motors must be small (low current)*.

🔌 Wiring Guide

Arduino Nano Pin	Connect To
A0	LDR AO (analog output)
D4	Motor 1 (+)
D5	Motor 2 (+)
GND	Motor 1 & 2 (–), LDR GND
5V	LDR VCC

> ⚠️ Connect motors directly **only if they are low-power BO motors** (like yellow BO). Otherwise use external driver or transistor.

🧠 Logic

- If **light is detected**, motors turn ON → robot moves.
- If **no light**, motors OFF → robot stops.

🗒️ Arduino Code

cpp

```
const int ldrPin = A0;
const int motor1 = 4;
const int motor2 = 5;

void setup() {
  pinMode(motor1, OUTPUT);
  pinMode(motor2, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int lightValue = analogRead(ldrPin);
  Serial.println(lightValue); // Use to determine threshold

  if (lightValue < 500) {  // Bright area
    digitalWrite(motor1, HIGH);
    digitalWrite(motor2, HIGH);
  } else {  // Dark area
    digitalWrite(motor1, LOW);
    digitalWrite(motor2, LOW);
  }

  delay(100);
}
```

🧪 Testing

1. Upload the code.
2. Shine a flashlight or use daylight on the LDR → robot moves forward.
3. Cover the LDR → robot stops.

✅ Enhancements

- Add two LDRs (left/right) for **directional light-following**.
- Add buzzer for sound alert in darkness.

🔧 Project 4: Object Avoiding Robot

🎯 Objective

The robot **moves forward** and **avoids obstacles** using an IR sensor. When an object is detected, it **stops or turns**.

🧩 Components Needed

Component	Quantity
Arduino Nano	1
IR Sensor Module	1
BO Motors	2
Tyres	2
5V Battery	1
Jumper Wires	As needed

🔌 Wiring Guide

Arduino Nano Pin	Connect To
D2	IR Sensor OUT
D4	Motor 1 (+)
D5	Motor 2 (+)
GND	Motor – wires + IR GND
5V	IR VCC

🧠 Logic

- If IR detects nothing (no obstacle): motors ON → robot moves forward.

- If IR detects obstacle (LOW signal): motors OFF → robot stops or turns.

🗒️ Arduino Code

cpp

```
const int irPin = 2;
const int motor1 = 4;
const int motor2 = 5;

void setup() {
  pinMode(irPin, INPUT);
  pinMode(motor1, OUTPUT);
  pinMode(motor2, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int irValue = digitalRead(irPin);
  Serial.println(irValue);  // Use Serial Monitor for debugging

  if (irValue == HIGH) {
    // No obstacle: move forward
    digitalWrite(motor1, HIGH);
    digitalWrite(motor2, HIGH);
  } else {
    // Obstacle detected: stop
    digitalWrite(motor1, LOW);
    digitalWrite(motor2, LOW);
  }

  delay(100);
}
```

🧪 Testing

1. Upload code and power your setup.
2. IR sensor facing forward — robot should move when clear.
3. Place an object in front — robot stops.

✅ Enhancement Ideas

- Add random turning behavior when obstacle is detected.
- Combine with LDR for **light-aware** obstacle avoider.

🔧 Project 5: Night Patrol Bot

🎯 Objective

The robot uses an **LDR module** to sense darkness. It **starts patrolling (moving forward)** when it's **dark**, and **stops in daylight**.

🧩 Components Needed

Component	Quantity
Arduino Nano	1
LDR Module	1
BO Motors	2
Tyres	2
5V Battery	1
Jumper Wires	As needed

🔌 Wiring Connections

Arduino Nano Pin	Connect To
A0	LDR Analog OUT
D4	Motor 1 (+)
D5	Motor 2 (+)
GND	LDR GND, Motor –
5V	LDR VCC

🧠 Logic

- Read **light intensity** from the LDR.

- If the environment is **dark**, turn ON motors (move).
- If it's **bright**, stop the robot.

🗒️ Arduino Code

cpp

```
const int ldrPin = A0;
const int motor1 = 4;
const int motor2 = 5;

void setup() {
  pinMode(motor1, OUTPUT);
  pinMode(motor2, OUTPUT);
  Serial.begin(9600); // For debugging light value
}

void loop() {
  int ldrValue = analogRead(ldrPin);
  Serial.println(ldrValue);  // View in Serial Monitor

  if (ldrValue > 700) {  // Dark environment
    digitalWrite(motor1, HIGH);
    digitalWrite(motor2, HIGH);
  } else {  // Bright light
    digitalWrite(motor1, LOW);
    digitalWrite(motor2, LOW);
  }

  delay(100);
}
```

🧪 Testing

1. Upload code to your Nano.
2. Block light (use a box, cloth, or cover the LDR) → robot moves.
3. Bring it to light or shine torch → robot stops.

🔧 Project 6: Daylight Tracker Bot

🎯 Objective

The robot **moves only when there is light** and **stops in the dark**. This simulates a robot that **tracks or follows daylight**.

🧩 Components Required

Component	Quantity
Arduino Nano	1
LDR Module	1
BO Motors	2
Tyres	2
5V Battery	1
Jumper Wires	As needed

🔌 Wiring Diagram

Arduino Nano Pin	Connect To
A0	LDR Analog OUT
D4	Motor 1 (+)
D5	Motor 2 (+)
GND	LDR GND, Motor –
5V	LDR VCC

> ⚠️ Same wiring as Project 5 – just different logic.

🧠 Logic

- LDR detects **bright light** → robot moves.
- In **darkness**, robot stays still.

🗒️ Arduino Code

cpp

```
const int ldrPin = A0;
const int motor1 = 4;
const int motor2 = 5;

void setup() {
  pinMode(motor1, OUTPUT);
  pinMode(motor2, OUTPUT);
  Serial.begin(9600); // Optional: Monitor LDR value
}

void loop() {
  int lightValue = analogRead(ldrPin);
  Serial.println(lightValue); // Tune your threshold by watching
values

  if (lightValue < 500) {  // Bright light
    digitalWrite(motor1, HIGH);
    digitalWrite(motor2, HIGH);
  } else {  // Dark environment
    digitalWrite(motor1, LOW);
    digitalWrite(motor2, LOW);
  }

  delay(100);
}
```

🧪 Testing

1. Upload and power the Nano.
2. Shine a torch or go to sunlight → robot moves forward.
3. Cover the LDR → robot stops.

🔧 Project 7: IR Gesture-Based Motor Control

🎯 Objective

Use a simple **hand wave or object** in front of the IR sensor to **start** or **stop** the motors. It acts like a **touchless remote**.

🧩 Components Needed

Component	Quantity
Arduino Nano	1
IR Sensor	1
BO Motors	2
Tyres	2
5V Battery	1
Jumper Wires	As needed

🔌 Wiring Connections

Arduino Nano Pin	Connect To
D2	IR Sensor OUT
D4	Motor 1 (+)
D5	Motor 2 (+)
GND	IR GND + Motor –
5V	IR VCC

🧠 Logic

- IR Sensor detects object (like your hand).

- On **detection**, toggle motor state (ON ⇄ OFF).
- Uses a flag to remember the current state of the robot.

🗒️ Arduino Code

cpp

```
const int irPin = 2;
const int motor1 = 4;
const int motor2 = 5;

bool motorState = false;
bool previousState = HIGH;

void setup() {
  pinMode(irPin, INPUT);
  pinMode(motor1, OUTPUT);
  pinMode(motor2, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int currentIR = digitalRead(irPin);

  if (currentIR == LOW && previousState == HIGH) {
    // Gesture detected (falling edge)
    motorState = !motorState; // Toggle motor state
    delay(200); // Debounce delay
  }

  previousState = currentIR;

  // Apply motor state
  digitalWrite(motor1, motorState ? HIGH : LOW);
  digitalWrite(motor2, motorState ? HIGH : LOW);
}
```

🧪 Testing

1. Power on your setup.
2. Wave your hand in front of the IR sensor.
3. The robot starts moving. Wave again to stop.

🔧 Project 8: Sunlight Alarm

🎯 Objective

When **sunlight or any bright light** hits the **LDR**, the **buzzer will ring**. Useful for:

- Morning wake-up alarm using sunlight
- Light intrusion detection

🧩 Components Needed

Component	Quantity
Arduino Nano	1
LDR Module	1
Buzzer	1
Jumper Wires	As needed
5V Battery or USB	1

🔌 Wiring Connections

Arduino Nano Pin	Connect To
A0	LDR Analog OUT
D3	Buzzer (+)
GND	LDR GND, Buzzer (–)
5V	LDR VCC

🧠 Logic

- LDR reads light level.

- If **light intensity > threshold**, the **buzzer turns ON**.
- Otherwise, buzzer stays OFF.

🗒️ Arduino Code

cpp

```
const int ldrPin = A0;
const int buzzerPin = 3;

void setup() {
  pinMode(buzzerPin, OUTPUT);
  Serial.begin(9600); // For observing light values
}

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

  if (lightValue < 500) {  // Bright light or sunlight
    digitalWrite(buzzerPin, HIGH);
  } else {
    digitalWrite(buzzerPin, LOW);
  }

  delay(100);
}
```

🧪 Testing

1. Upload the code.
2. Shine a flashlight or move the LDR into sunlight → buzzer will beep.
3. Cover the LDR → buzzer stops.

🔧 Project 9: Invisible Line Bot (Line Follower Using IR Sensor)

🎯 Objective

The robot will **follow a black line** on a white surface using the **IR sensor**. When it sees black, it goes straight; when it sees white, it stops or turns.

🧩 Components Needed

Component	Quantity
Arduino Nano	1
IR Sensor Module	1
BO Motors	2
Tyres	2
5V Battery	1
Jumper Wires	As needed

🧠 Working Principle

- **IR sensor** reflects from white surface → sensor gives **HIGH (1)**.
- **IR sensor** absorbs on black line → sensor gives **LOW (0)**.
- Robot moves when IR sees black line.

🔌 Wiring Connections

Arduino Nano Pin	Connect To
D2	IR Sensor OUT
D4	Motor 1 (+)
D5	Motor 2 (+)

GND	IR GND + Motor –
5V	IR VCC

🗒️ Arduino Code

cpp

```
const int irPin = 2;
const int motor1 = 4;
const int motor2 = 5;

void setup() {
  pinMode(irPin, INPUT);
  pinMode(motor1, OUTPUT);
  pinMode(motor2, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int irValue = digitalRead(irPin);
  Serial.println(irValue);  // Check HIGH or LOW on white/black

  if (irValue == LOW) {
    // IR sees black line → follow
    digitalWrite(motor1, HIGH);
    digitalWrite(motor2, HIGH);
  } else {
    // IR sees white → stop or turn
    digitalWrite(motor1, LOW);
    digitalWrite(motor2, LOW);
  }

  delay(100);
}
```

🧪 Testing

1. Create a black line on white sheet (e.g., black tape).
2. Point IR sensor downward and place the robot on the line.
3. It should move forward on black and stop when off the line.

🔧 Project 10: Speed Modulation Robot

🎯 Objective

The robot's **motor speed increases in bright light** and **slows down in darkness** – like an energy-efficient daylight robot!

🧩 Components Required

Component	Quantity
Arduino Nano	1
LDR Module	1
BO Motors	2
Tyres	2
5V Battery	1
Jumper Wires	As needed

🧠 Working Principle

- The **LDR** detects light intensity and gives an analog value.
- The Arduino reads this value and uses **PWM (analogWrite)** to control motor speed.
- More light → higher speed; less light → slower speed.

🔌 Wiring Diagram

Arduino Nano Pin	Connect To
A0	LDR AO (Analog)
D5 (PWM)	Motor 1 (+)
D6 (PWM)	Motor 2 (+)

GND	LDR GND, Motor –
5V	LDR VCC

⚠️ Both `D5` and `D6` are PWM-capable pins needed for speed control.

🗒️ Arduino Code

cpp

```
const int ldrPin = A0;
const int motor1 = 5;
const int motor2 = 6;

void setup() {
  pinMode(motor1, OUTPUT);
  pinMode(motor2, OUTPUT);
  Serial.begin(9600);
}

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

  // Map light value (0-1023) to PWM range (0-255)
  int speed = map(lightVal, 0, 1023, 0, 255);

  analogWrite(motor1, speed);
  analogWrite(motor2, speed);

  delay(100);
}
```

🧪 Testing

1. Shine a flashlight on the LDR → motors spin fast.

2. Dim or cover the LDR → motors slow down or stop.

🔧 Project 11: IR-Based Security Alert

🎯 Objective

If an **object or person crosses** in front of the **IR sensor**, the **buzzer rings** — like a motion alarm for a gate, drawer, or door.

🧩 Components Required

Component	Quantity
Arduino Nano	1
IR Sensor Module	1
Buzzer	1
Jumper Wires	As needed
5V Battery / USB	1

🧠 Working Principle

- IR sensor emits infrared and detects reflected light.
- When a person or object comes close, **IR OUT goes LOW**.
- Arduino turns ON the **buzzer** when detection occurs.

🔌 Wiring Diagram

Arduino Nano Pin	Connect To
D2	IR Sensor OUT
D3	Buzzer + (Signal)
GND	IR GND, Buzzer –
5V	IR VCC

📌 If using active buzzer module, connect D3 directly to signal pin.

🗒️ Arduino Code

cpp

```
const int irPin = 2;
const int buzzerPin = 3;

void setup() {
  pinMode(irPin, INPUT);
  pinMode(buzzerPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int irValue = digitalRead(irPin);
  Serial.println(irValue);  // Check IR output status

  if (irValue == LOW) { // Object detected
    digitalWrite(buzzerPin, HIGH);
  } else {
    digitalWrite(buzzerPin, LOW);
  }

  delay(100);
}
```

🧪 Testing

1. Upload the code.
2. Bring your hand close to the IR sensor → buzzer beeps.
3. Remove hand → buzzer stops.

🔧 Project 12: Line Crossing Alert

🎯 Objective

The buzzer **beeps briefly** whenever something **crosses the IR sensor's line of sight** — great for entrances, bookshelves, drawers, etc.

🧩 Components Needed

Component	Quantity
Arduino Nano	1
IR Sensor	1
Buzzer	1
Jumper Wires	As needed
5V Battery / USB	1

🧠 Working Principle

- IR sensor detects motion in front (e.g., a person or hand).
- When object crosses, sensor's output goes **LOW**.
- Arduino **beeps the buzzer for a short duration**.

🔌 Wiring Guide

Arduino Nano Pin	Connect To
D2	IR Sensor OUT
D3	Buzzer + (Signal)
GND	IR GND, Buzzer –
5V	IR VCC

🗒️ Arduino Code

cpp

```
const int irPin = 2;
const int buzzerPin = 3;
bool buzzerActive = false;

void setup() {
  pinMode(irPin, INPUT);
  pinMode(buzzerPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int irValue = digitalRead(irPin);
  Serial.println(irValue);

  if (irValue == LOW && !buzzerActive) {
    digitalWrite(buzzerPin, HIGH);
    buzzerActive = true;
    delay(300); // Short beep
    digitalWrite(buzzerPin, LOW);
    delay(100); // Prevent retrigger
  }

  if (irValue == HIGH) {
    buzzerActive = false; // Reset when object gone
  }

  delay(50);
}
```

🧪 Testing

1. Upload the code.

2. Wave your hand quickly in front of the IR sensor.
3. You should hear a **short "beep"** each time something crosses.

✅ Use Cases

- Mini entry alert at door
- Shelf/drawer access buzzer
- People/object crossing counter (with enhancement)

🔧 Project 13: Basic Obstacle Counter

🎯 Objective

Each time something (like a hand or object) crosses the IR sensor, the Arduino will **count and display** the total number of detections in the **Serial Monitor**.

🧩 Components Needed

Component	Quantity
Arduino Nano	1
IR Sensor	1
Jumper Wires	As needed
USB (for Serial Monitor)	1

> 🔔 *No buzzer or motors in this one — just counting events.*

🧠 Working Principle

- IR sensor detects object and gives **LOW** output.
- When LOW is detected, the counter is incremented.
- A flag is used to **prevent multiple counts** from a single crossing (debouncing).

🔌 Wiring Connections

Arduino Nano Pin	Connect To
D2	IR Sensor OUT
GND	IR GND
5V	IR VCC

🗒️ Arduino Code

cpp

```
const int irPin = 2;
int counter = 0;
bool objectPreviouslyDetected = false;

void setup() {
  pinMode(irPin, INPUT);
  Serial.begin(9600);
  Serial.println("Obstacle Counter Ready...");
}

void loop() {
  int irValue = digitalRead(irPin);

  if (irValue == LOW && !objectPreviouslyDetected) {
    counter++;
    Serial.print("Object Count: ");
    Serial.println(counter);
    objectPreviouslyDetected = true;
    delay(200); // Debounce delay
  }

  if (irValue == HIGH) {
    objectPreviouslyDetected = false;
  }

  delay(50);
}
```

🧪 Testing Steps

1. Upload code and open **Serial Monitor** (set to 9600 baud).

2. Move your hand across the IR sensor.
3. You'll see: `Object Count: 1`, `Object Count: 2`, and so on.

✅ Enhancements

- Add buzzer beep on each count.
- Add reset feature using a button.
- Display count on LCD (if added later).

🔧 Project 14: Sound Notification Bot

🎯 Objective

When the IR sensor detects an obstacle, the robot will:

- **Start moving**
- **Sound the buzzer**

This is like a **patrolling alert bot** that reacts to nearby objects.

🧩 Components Required

Component	Quantity
Arduino Nano	1
IR Sensor	1
BO Motors	2
Tyres	2
Buzzer	1
5V Battery	1
Jumper Wires	As needed

🔌 Wiring Connections

Arduino Nano Pin	Connect To
D2	IR Sensor OUT
D3	Buzzer (+)
D4	Motor 1 (+)
D5	Motor 2 (+)

GND	IR GND, Buzzer –, Motor –
5V	IR VCC

🧠 Logic

- When an **object is detected**, robot **moves forward** and **buzzer sounds**.
- When no object, robot and buzzer are both OFF.

🧾 Arduino Code

cpp

```
const int irPin = 2;
const int buzzerPin = 3;
const int motor1 = 4;
const int motor2 = 5;

void setup() {
  pinMode(irPin, INPUT);
  pinMode(buzzerPin, OUTPUT);
  pinMode(motor1, OUTPUT);
  pinMode(motor2, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int irValue = digitalRead(irPin);
  Serial.println(irValue);

  if (irValue == LOW) { // Object detected
    digitalWrite(buzzerPin, HIGH);
    digitalWrite(motor1, HIGH);
    digitalWrite(motor2, HIGH);
  } else { // No object
    digitalWrite(buzzerPin, LOW);
```

```
    digitalWrite(motor1, LOW);
    digitalWrite(motor2, LOW);
  }

  delay(100);
}
```

🧪 Testing

1. Upload the code.
2. Place hand/object in front of the IR sensor → robot moves and buzzer beeps.
3. Remove the object → everything stops.

🔧 Project 15: Light Level Indicator

🎯 Objective

The Arduino will read values from the **LDR** and print the **light level in real-time** in the **Serial Monitor**. This can help you understand how much light is present.

🧩 Components Needed

Component	Quantity
Arduino Nano	1
LDR Module	1
Jumper Wires	As needed
USB (for Serial Monitor)	1

> 🔔 This is a non-movement project – useful for testing light environments.

🧠 Working Principle

- LDR gives an **analog value** based on brightness:
    - **Low value = Bright light**
    - **High value = Darkness**
- Arduino reads this and prints it on Serial Monitor.

🔌 Wiring Connections

Arduino Nano Pin	Connect To
A0	LDR AO (Analog Out)
GND	LDR GND

5V LDR VCC

🧾 Arduino Code

cpp

```
const int ldrPin = A0;

void setup() {
  Serial.begin(9600);
  Serial.println("Light Level Indicator Ready...");
}

void loop() {
  int lightLevel = analogRead(ldrPin);
  Serial.print("LDR Value: ");
  Serial.println(lightLevel);

  delay(300); // Adjust for smoother updates
}
```

🧪 Testing Steps

1. Upload the code and open **Serial Monitor** (set to 9600 baud).
2. Shine a torch on the LDR → value drops.
3. Cover the LDR → value increases.

Typical LDR range: **Bright = ~100–300**, **Dark = ~700–1023**

✅ Enhancements

- Add color-coded comments: Bright / Dim / Dark
- Connect a buzzer to alert if light is too low (for security lighting)

🔧 Project 16: No-Light Movement Bot

🎯 Objective

The robot uses the **LDR sensor** to detect complete darkness. If it's fully dark, it **starts moving**. If there's any light, it **stops immediately**.

🧩 Components Required

Component	Quantity
Arduino Nano	1
LDR Module	1
BO Motors	2
Tyres	2
5V Battery	1
Jumper Wires	As needed

🧠 Working Logic

- Read the LDR value from analog pin.
- If it's **above a high threshold** (very dark), move motors.
- If light is present, stop the motors.

🔌 Wiring Connections

Arduino Nano Pin	Connect To
A0	LDR AO
D4	Motor 1 (+)
D5	Motor 2 (+)

GND	LDR GND, Motor –
5V	LDR VCC

🗒️ Arduino Code

cpp

```
const int ldrPin = A0;
const int motor1 = 4;
const int motor2 = 5;

void setup() {
  pinMode(motor1, OUTPUT);
  pinMode(motor2, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int ldrValue = analogRead(ldrPin);
  Serial.print("Light Level: ");
  Serial.println(ldrValue);

  if (ldrValue > 800) { // Fully dark
    digitalWrite(motor1, HIGH);
    digitalWrite(motor2, HIGH);
  } else {
    digitalWrite(motor1, LOW);
    digitalWrite(motor2, LOW);
  }

  delay(100);
}
```

🧪 Testing Instructions

1. Upload code and power the robot.

2. Cover the LDR completely → robot moves.
3. Shine any light → robot stops immediately.

🔧 Project 17: Auto Stop in Light

🎯 Objective

The robot will:

- **Move in darkness**
- **Immediately stop when light is detected**

This can be used for **stealth movement**, **security patrol**, or **motion traps**.

🧩 Components Required

Component	Quantity
Arduino Nano	1
LDR Module	1
BO Motors	2
Tyres	2
5V Battery	1
Jumper Wires	As needed

🧠 Working Logic

- **LDR reads light** continuously.
- If **light value is high (dark)** → motors ON.
- If **light value is low (bright)** → motors OFF.

🔌 Wiring Connections

Arduino Nano Pin	Connect To
A0	LDR Analog Out
D4	Motor 1 (+)
D5	Motor 2 (+)
GND	LDR GND, Motor –
5V	LDR VCC

🧾 Arduino Code

cpp

```
const int ldrPin = A0;
const int motor1 = 4;
const int motor2 = 5;

void setup() {
  pinMode(motor1, OUTPUT);
  pinMode(motor2, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int ldrValue = analogRead(ldrPin);
  Serial.print("Light Level: ");
  Serial.println(ldrValue);

  if (ldrValue > 700) { // Dark enough to move
    digitalWrite(motor1, HIGH);
    digitalWrite(motor2, HIGH);
  } else {
    digitalWrite(motor1, LOW);
    digitalWrite(motor2, LOW);
  }

  delay(100);
}
```

🧪 Testing

1. Upload code and power the robot.
2. Block the LDR → robot starts moving.
3. Shine a flashlight → robot stops instantly.

✅ Use Case Ideas

- **Night-time exploration bot**
- **Trap-style robot that moves only in dark**
- **Smart robot for blackout scenarios**

🔧 Project 18: Edge Detection Bot

🎯 Objective

The robot will move forward until it **detects a black edge (or table edge)** — then it **immediately stops** to avoid falling or crossing the boundary.

🧩 Components Needed

Component	Quantity
Arduino Nano	1
IR Sensor	1
BO Motors	2
Tyres	2
5V Battery	1
Jumper Wires	As needed

🧠 How It Works

- IR sensor looks **downward** at the surface.
- White surface reflects IR → output is **HIGH**.
- Black line/edge absorbs IR → output goes **LOW**.
- Robot stops immediately when IR sees **black (edge)**.

🧠 Suggested Setup

- Place black tape at the edge of a white sheet or table.
- Mount the IR sensor facing **downward** toward the surface.

🔌 Wiring Connections

Arduino Nano Pin	Connect To
D2	IR OUT
D4	Motor 1 (+)
D5	Motor 2 (+)
GND	IR GND + Motor –
5V	IR VCC

🗒️ Arduino Code

cpp

```
const int irPin = 2;
const int motor1 = 4;
const int motor2 = 5;

void setup() {
  pinMode(irPin, INPUT);
  pinMode(motor1, OUTPUT);
  pinMode(motor2, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int irValue = digitalRead(irPin);
  Serial.println(irValue);

  if (irValue == HIGH) { // White surface
    digitalWrite(motor1, HIGH);
    digitalWrite(motor2, HIGH);
  } else { // Black line or edge
    digitalWrite(motor1, LOW);
    digitalWrite(motor2, LOW);
  }
```

```
  delay(100);
}
```

🧪 Testing

1. Place the bot on a white surface with black border/tape.
2. Upload the code and power up.
3. Robot moves forward and **stops at the black edge**.

✅ Use Cases

- Prevent falling from table or stairs.
- Mini line-stopper robot.
- Can be extended for full **line-following + stopping bot**.

🔧 Project 19: Robot with Alert Horn

🎯 Objective

When the robot detects an **obstacle** (like a wall or hand), it will:

- **Move forward**
- **Sound an alert horn (buzzer)**

Useful for **alert patrol bots** or **presence notifiers**.

🧩 Components Needed

Component	Quantity
Arduino Nano	1
IR Sensor	1
BO Motors	2
Buzzer	1
Tyres	2
5V Battery	1
Jumper Wires	As needed

🧠 Working Principle

- IR sensor detects nearby obstacle → OUT = LOW
- Arduino turns **ON motors and buzzer**
- No obstacle → both stay OFF

🔌 Wiring Connections

Arduino Nano Pin	Connect To
D2	IR OUT
D3	Buzzer (+)
D4	Motor 1 (+)
D5	Motor 2 (+)
GND	IR GND, Buzzer –, Motor –
5V	IR VCC

🧾 Arduino Code

cpp

```
const int irPin = 2;
const int buzzerPin = 3;
const int motor1 = 4;
const int motor2 = 5;

void setup() {
  pinMode(irPin, INPUT);
  pinMode(buzzerPin, OUTPUT);
  pinMode(motor1, OUTPUT);
  pinMode(motor2, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int irValue = digitalRead(irPin);
  Serial.println(irValue);

  if (irValue == LOW) { // Obstacle detected
    digitalWrite(buzzerPin, HIGH);
    digitalWrite(motor1, HIGH);
    digitalWrite(motor2, HIGH);
  } else { // No obstacle
    digitalWrite(buzzerPin, LOW);
```

```
    digitalWrite(motor1, LOW);
    digitalWrite(motor2, LOW);
  }

  delay(100);
}
```

🧪 Testing

1. Upload and power up your bot.
2. Bring your hand near the IR sensor.
3. Robot starts moving and buzzer beeps.
4. Remove hand → everything stops.

🔧 Project 20: Night Entry Alert

🎯 Objective

This system will trigger a **buzzer alert only when**:

- It's **dark** (LDR condition), **AND**
- An **object/person is detected** (IR sensor)

Perfect for:

- **Nighttime security**
- **Room intrusion alerts**
- **Light-sensitive traps**

🧩 Components Needed

Component	Quantity
Arduino Nano	1
LDR Module	1
IR Sensor	1
Buzzer	1
Jumper Wires	As needed
5V Battery / USB	1

🧠 Working Logic

- **LDR** checks if it's dark (analogRead > threshold)
- **IR sensor** checks for object (digitalRead == LOW)

- If **both conditions are met**, **buzzer rings**

🔌 Wiring Diagram

Arduino Nano Pin	Connect To
A0	LDR Analog OUT
D2	IR OUT
D3	Buzzer (+)
GND	All GNDs
5V	LDR VCC, IR VCC

🗒️ Arduino Code

cpp

```
const int ldrPin = A0;
const int irPin = 2;
const int buzzerPin = 3;

void setup() {
  pinMode(irPin, INPUT);
  pinMode(buzzerPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int lightValue = analogRead(ldrPin);
  int irValue = digitalRead(irPin);

  Serial.print("LDR: ");
  Serial.print(lightValue);
  Serial.print(" | IR: ");
  Serial.println(irValue);
```

```
  if (lightValue > 700 && irValue == LOW) {
    // It's dark AND motion/object detected
    digitalWrite(buzzerPin, HIGH);
  } else {
    digitalWrite(buzzerPin, LOW);
  }

  delay(100);
}
```

🧪 Testing Steps

1. Upload the code.
2. Cover the LDR to simulate darkness.
3. Wave hand near IR sensor — buzzer beeps.
4. If it's bright, waving hand won’t trigger buzzer.