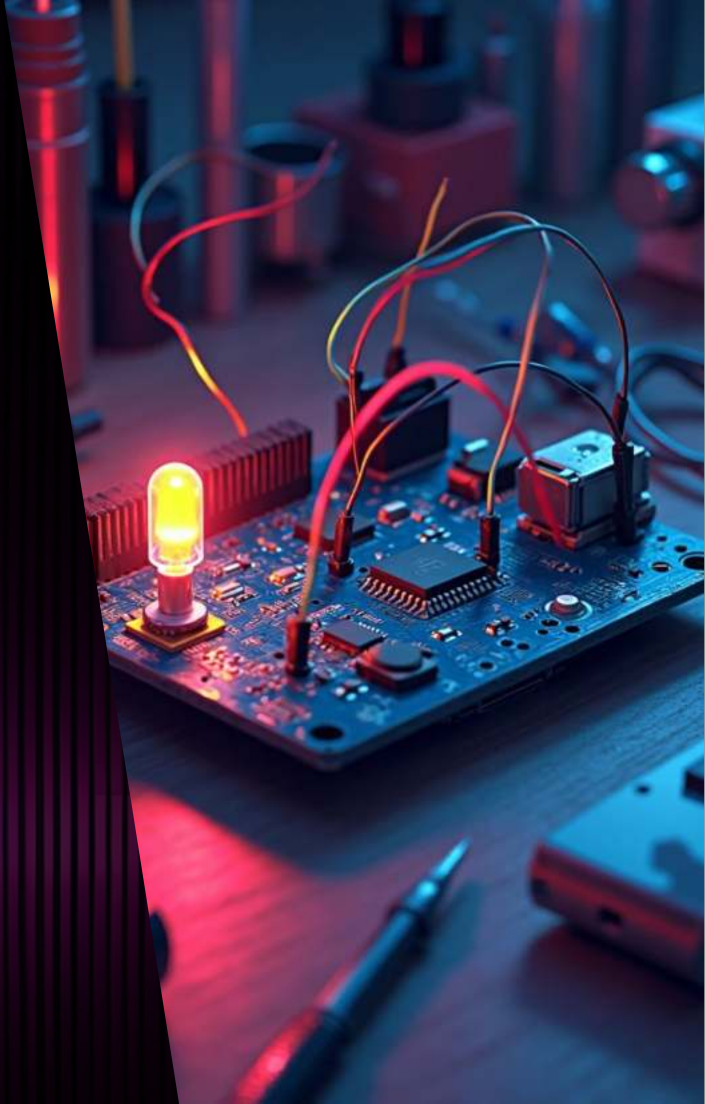
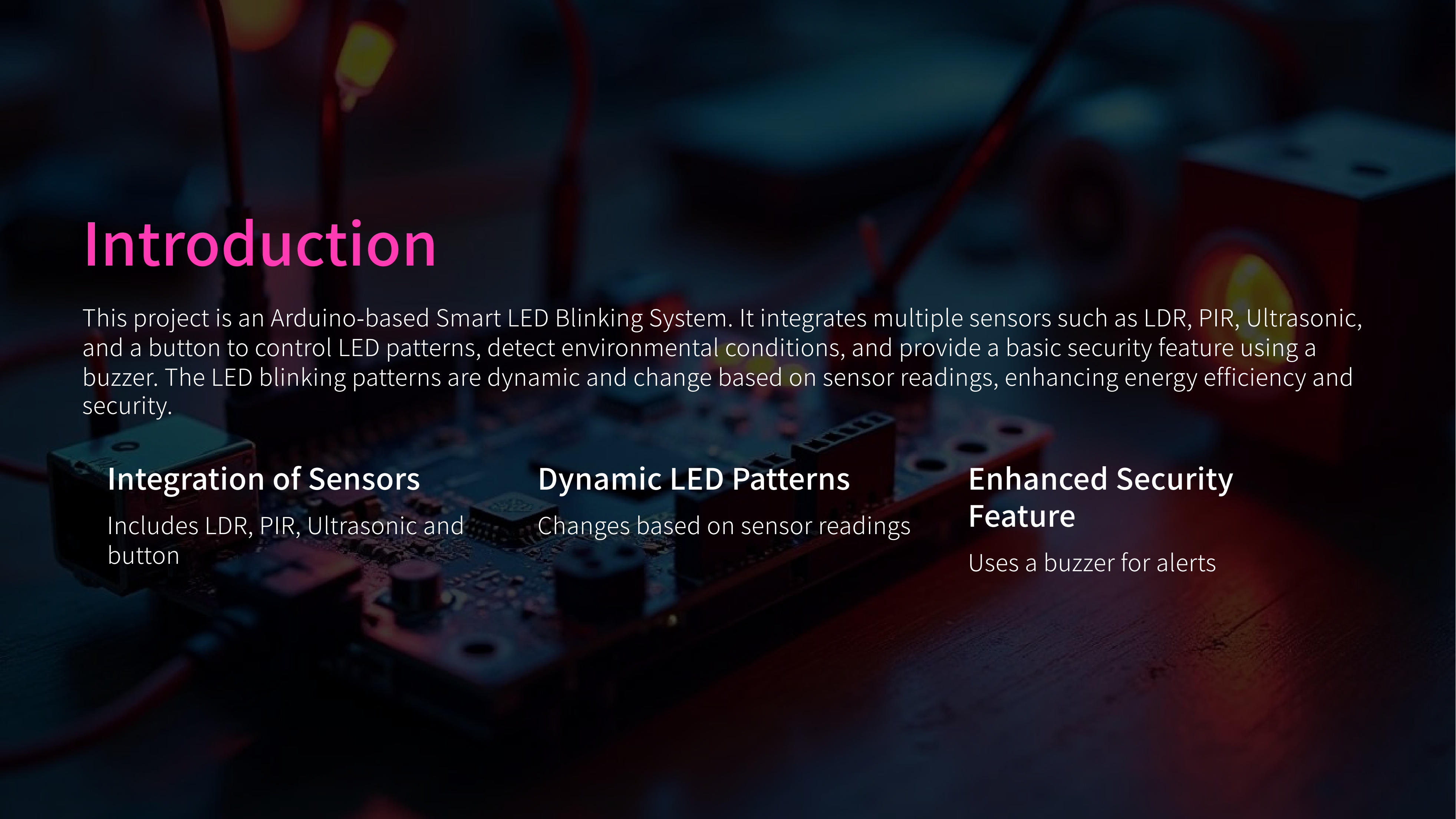
LED Blinking System





Introduction

This project is an Arduino-based Smart LED Blinking System. It integrates multiple sensors such as LDR, PIR, Ultrasonic, and a button to control LED patterns, detect environmental conditions, and provide a basic security feature using a buzzer. The LED blinking patterns are dynamic and change based on sensor readings, enhancing energy efficiency and security.

Integration of Sensors

Includes LDR, PIR, Ultrasonic and button

Dynamic LED Patterns

Changes based on sensor readings

Enhanced Security Feature

Uses a buzzer for alerts

Components Required

Arduino Uno

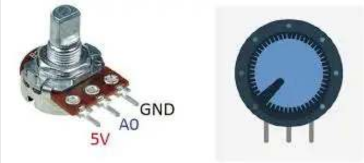
Main microcontroller board to control all components.

LEDs (x3)

Light-emitting diodes used for blinking patterns.

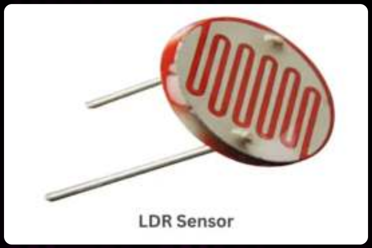
Push Button

User input to switch between LED patterns.



Potentiometer

Variable resistor to control LED blinking speed.



LDR

Detects ambient light level.

Ultrasonic Sensor

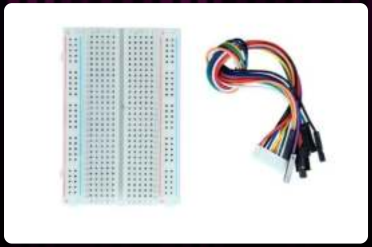
Measures distance to detect nearby objects.

PIR Sensor

Detects motion (human presence).

Buzzer

Produces sound during intruder alert.



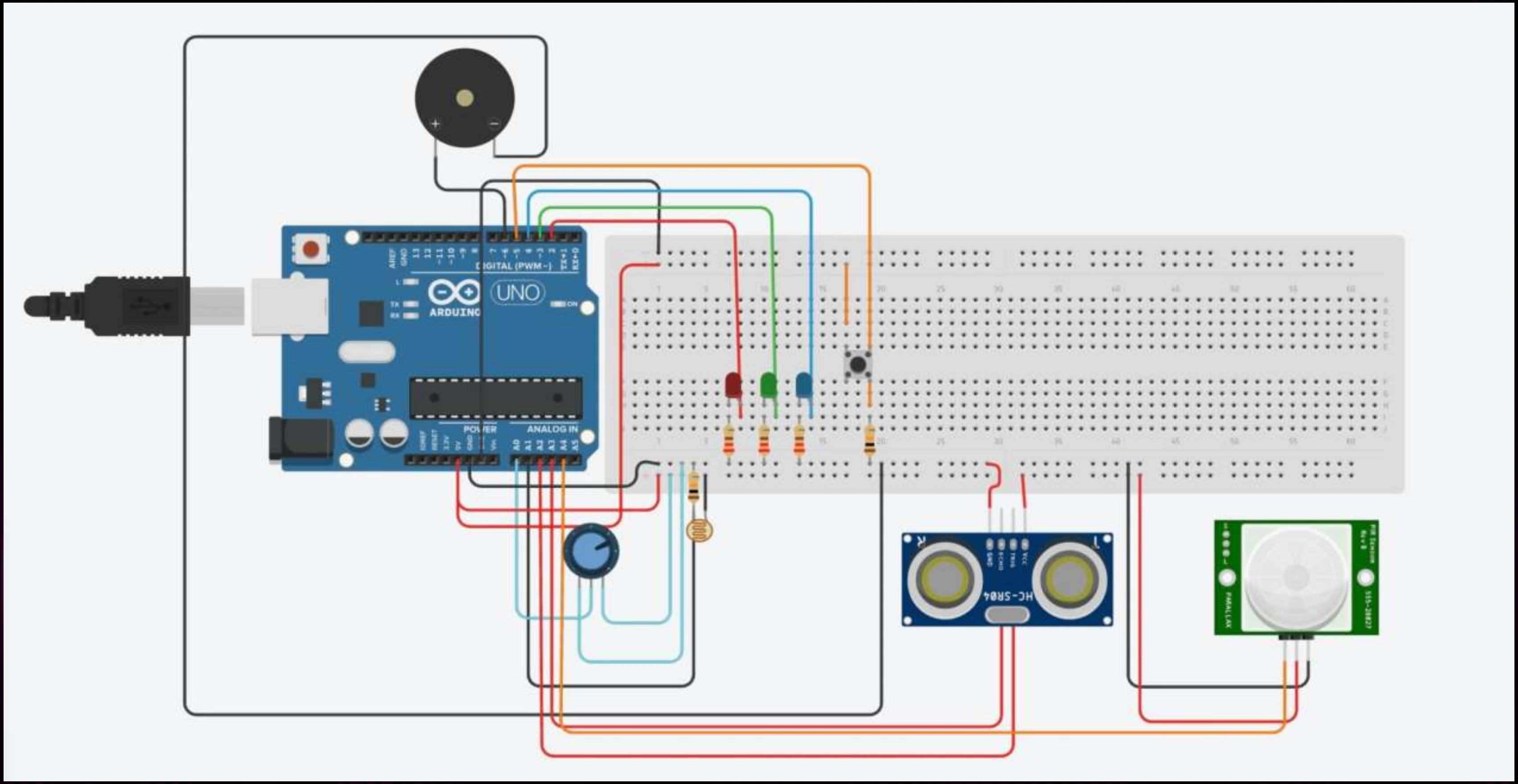
Breadboard & Jumper Wires

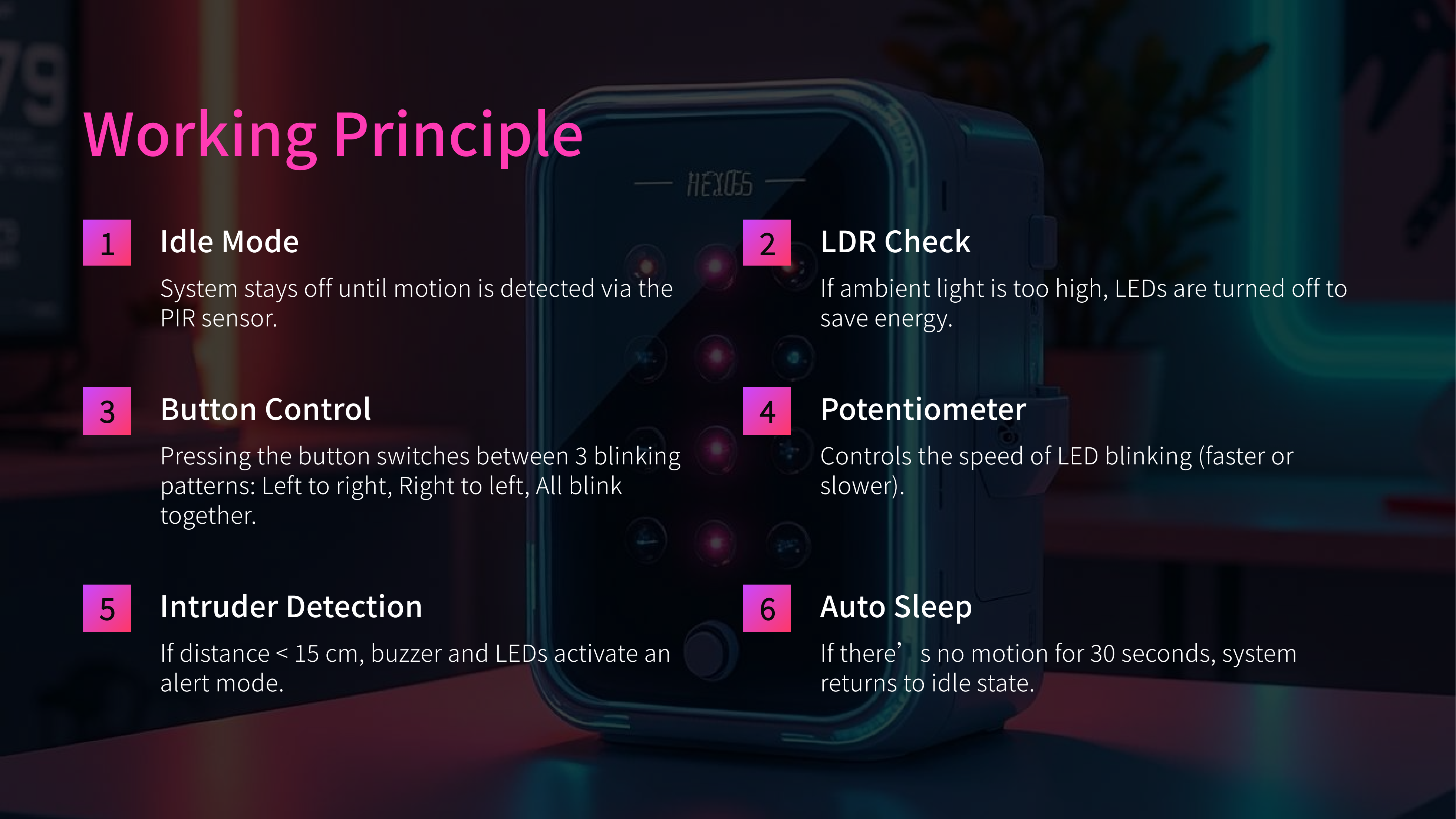
For connecting components without soldering.

Resistors

Limit current for LEDs and sensors

Circuit Diagram





Working Principle

1 **Idle Mode**

System stays off until motion is detected via the PIR sensor.

2 **LDR Check**

If ambient light is too high, LEDs are turned off to save energy.

3 **Button Control**

Pressing the button switches between 3 blinking patterns: Left to right, Right to left, All blink together.

4 **Potentiometer**

Controls the speed of LED blinking (faster or slower).

5 **Intruder Detection**

If distance < 15 cm, buzzer and LEDs activate an alert mode.

6 **Auto Sleep**

If there’ s no motion for 30 seconds, system returns to idle state.

Code Logic

```
void loop() {

if (digitalRead(pirPin) == HIGH) lastMotionTime = millis();

if (millis() - lastMotionTime > 30000) { allOff();

return; }



int ldrVal = analogRead(ldrPin);

if (ldrVal > 900) { allOff(); return; }
```

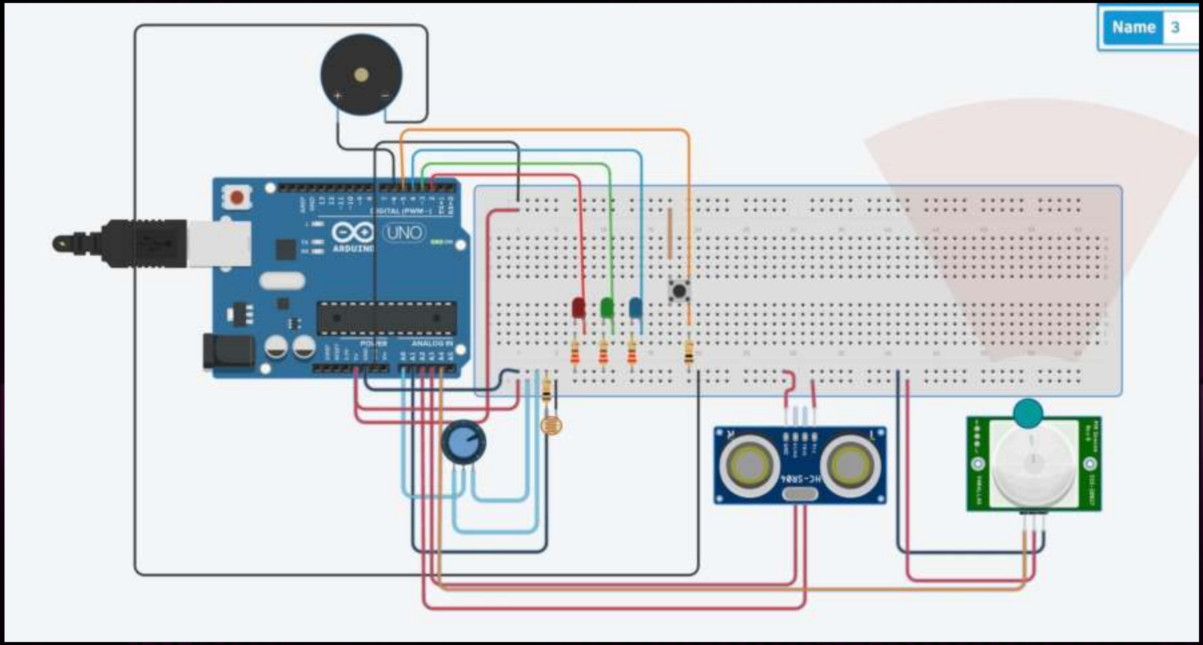
```
 int distance = getDistance();
  if (distance > 0 && distance < 15) {
    alarmMode();
    delay(3000);
    allOff();
    noTone(buzzerPin);
    return;
  }
 checkButton();
  int potVal = analogRead(potPin);
  int speed = map(potVal, 0, 1023, 100, 1000);
  updatePattern(pattern, speed);
}
```

Motion-based Idle Mode (PIR Sensor)

Uses a PIR motion sensor to detect movement.

If no motion for 30 seconds:

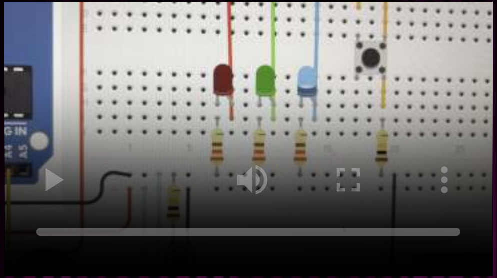
Enters "idle mode": turns off LEDs and stops patterns anResumes when motion is detected again.



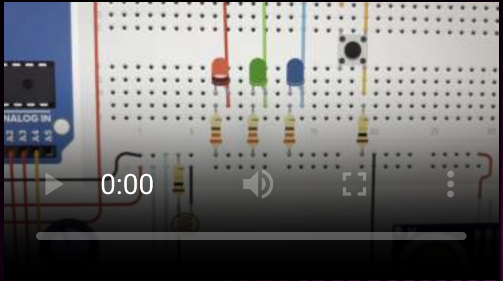
LED Pattern Modes (Button-controlled)

Pressing a button cycles through 3 patterns:

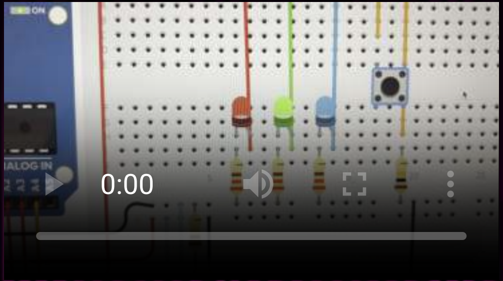
Pattern 0: Left to right.



Pattern 1: Right to left.



Pattern 2: All leds blink together.

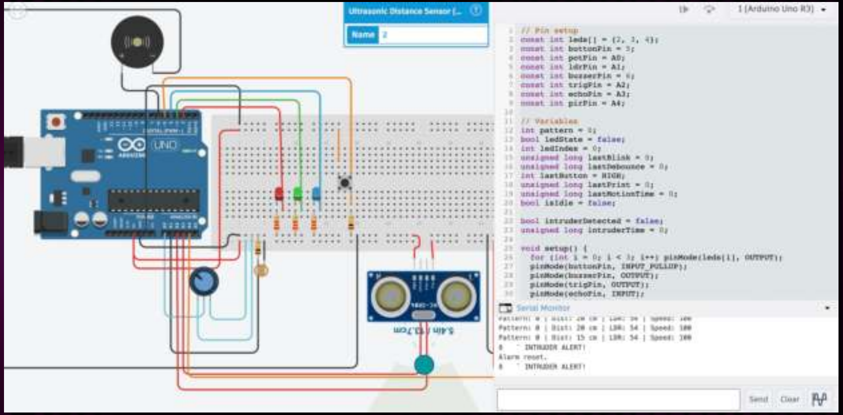


● Intruder Detection & Alarm

Ultrasonic sensor (HC-SR04) detects if something is closer than 15 cm.

If triggered:

- LEDs flash
- Buzzer sounds
- Message: "🚨 INTRUDER ALERT!" is printed to Serial Monitor.



```
// Pin setup
const int leds[] = {2, 3, 4};
const int buttonPin = 5;
const int potPin = A0;
const int ldrPin = A1;
const int buzzerPin = 6;
const int trigPin = A2;
const int echoPin = A3;
const int pirPin = A4;

// Variables
int pattern = 0;
bool ledState = false;
int ledIndex = 0;
unsigned long lastBlink = 0;
unsigned long lastDebounce = 0;
int lastButton = HIGH;
unsigned long lastPrint = 0;
unsigned long lastMotionTime = 0;
bool isIdle = false;

bool intruderDetected = false;
unsigned long intruderTime = 0;

void setup() {
  for (int i = 0; i < 3; i++) pinMode(leds[i], OUTPUT);
  pinMode(buttonPin, INPUT_PULLUP);
  pinMode(buzzerPin, OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
```

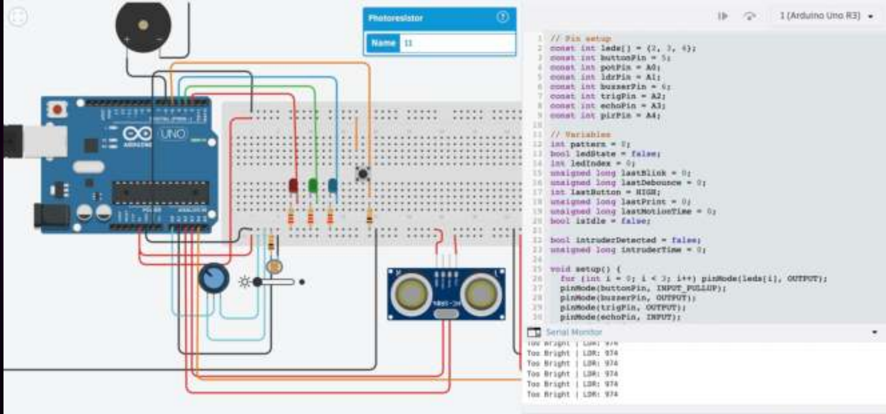
```
Pattern: 0 | Dist: 20 cm | LDR: 54 | Speed: 100
Pattern: 0 | Dist: 20 cm | LDR: 54 | Speed: 100
Pattern: 0 | Dist: 15 cm | LDR: 54 | Speed: 100
8   ` INTRUDER ALERT!
Alarm reset.
8   ` INTRUDER ALERT!
```

Auto-disable in Bright Light (LDR)

Reads ambient light with an LDR (light-dependent resistor).

If it's too bright (LDR > 900):

- Turns off all LEDs.
- Pauses other behavior.
- Prints: "Too Bright | LDR: ..." to Serial Monitor.



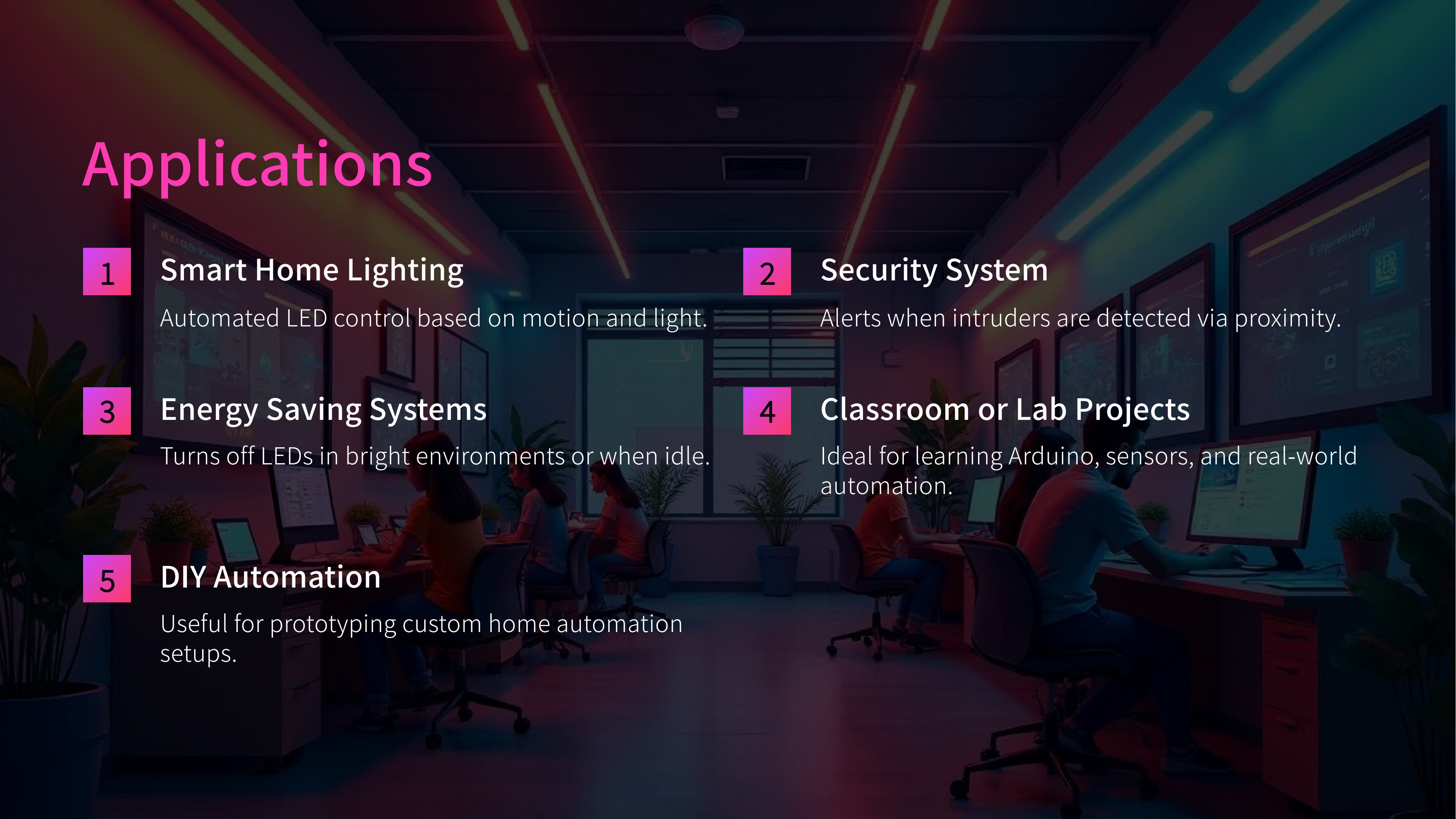
```
// Pin setup
const int leds[] = {2, 3, 4};
const int buttonPin = 5;
const int potPin = A0;
const int ldrPin = A1;
const int buzzerPin = 6;
const int trigPin = A2;
const int echoPin = A3;
const int pirPin = A4;

// Variables
int pattern = 0;
bool ledState = false;
int ledIndex = 0;
unsigned long lastBlink = 0;
unsigned long lastDebounce = 0;
int lastButton = HIGH;
unsigned long lastPrint = 0;
unsigned long lastMotionTime = 0;
bool isIdle = false;

bool intruderDetected = false;
unsigned long intruderTime = 0;

void setup() {
  for (int i = 0; i < 3; i++) pinMode(leds[i], OUTPUT);
  pinMode(buttonPin, INPUT_PULLUP);
  pinMode(buzzerPin, OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
```

```
Too Bright | LDR: 974
Too Bright | LDR: 974
Too Bright | LDR: 974
Too Bright | LDR: 974
Too Bright | LDR: 974
Too Bright | LDR: 974
```



Applications

1 Smart Home Lighting

Automated LED control based on motion and light.

2 Security System

Alerts when intruders are detected via proximity.

3 Energy Saving Systems

Turns off LEDs in bright environments or when idle.

4 Classroom or Lab Projects

Ideal for learning Arduino, sensors, and real-world automation.

5 DIY Automation

Useful for prototyping custom home automation setups.

Project Conclusion

1

Successful Integration

The project successfully integrates sensors, LEDs, and user input using Arduino.

2

Smart Automation

Demonstrates smart automation with energy saving and security features.

3

Real-World Applications

Highlights the use of motion, light, and proximity sensors in real-world applications.

4

Intelligent Embedded System

A great example of how simple components can create an intelligent embedded system.

THANK YOU