Automatic Light, Temperature, and Object Detection Based Control System Using Arduino

B.Sc.(H) in Data Analytics and Artificial Intelligence

Submitted to:
Dr. Amita Sharma
Assistant Professor (SG)
Dr. Anubha Jain
Director
Department of CS&IT

Submitted by:
Ms. Teena Sharma
Enroll no.: IISU/2023/ADM/35803
Roll no.:232025

ACKNOWLEDGEMENT

I would like to express my deepest appreciation to all individuals and organizations who have **contributed** to the successful completion of our project.

First and foremost, we extend our heartfelt thanks to our project supervisor, **Dr. Amita Sharma** (Assistant Professor) and **Dr. Anubha Jain (Director)**, for her unwavering support and mentorship. **Special** gratitude to **Dr. Ruchi Nanda (Mentor)**, **Dr. Navneet Sharma (Head of Department)**, all faculty members, and lab assistants whose contribution to **for**mulating suggestions and encouragement, especially in writing this report.

Additionally, I appreciate the support and resources provided by our educational institution, **IIS (De**emed to be) **University**, throughout the duration of this project.

Furthermore, I would also like to express our gratitude to our friends and family for their unwavering support, understanding, and encouragement throughout this endeavour.

ABSTRACT

This report presents the design and implementation of two Arduino-based automation projects developed using environmental and proximity sensors. In the first project, an LDR sensor is used to detect light intensity and a DHT11 sensor is used to measure temperature. Based on the sensed light conditions, an LED is automatically controlled, while a fan is operated according to the surrounding temperature. This project demonstrates basic environmental monitoring and automatic control of electrical devices.

The second project extends the concept of automation by incorporating an ultrasonic sensor to detect the presence and distance of an object. When an object is detected within a predefined distance range under low light conditions, both the LED and the fan are automatically switched ON. This project focuses on distance-based object detection combined with light sensing to enhance intelligent control. Together, both projects highlight the practical application of sensors, Arduino microcontroller, and real-time decision-making for smart and energy-efficient automation systems.

Contents

ACKNOWLEDGEMENT 2

ABSTRACT 3

TABLE OF CONTENTS .... **Error! Bookmark not defined.**

1. INTRODUCTION 5

1.1 Application and Scope 5

1.2 Task Identification and Execution 6

2. COMPONENTS AND TOOLS 7

3. CIRCUIT DIAGRAM 16

3.2 Working of Circuit 20

4. RESULT AND OBSERVATIONS 23

5. CONCLUSION 25

REFERENCES 26

ANNEXURE 27

1. INTRODUCTION

With the rapid advancement of embedded systems and automation technologies, sensor-based monitoring and control systems have become an important part of modern electronic applications. Sensors such as Light Dependent Resistors (LDR), temperature sensors, and distance sensors are widely used to improve energy efficiency, safety, and automation in daily life. Arduino-based projects provide a simple and effective platform to understand the integration of sensors with real-time control systems.

Automatic Light and Temperature Based Control System Using LDR and DHT11 Sensor (Project 1)

It focuses on monitoring environmental conditions using an LDR and a DHT11 sensor. The system automatically controls an LED based on light intensity and a fan based on temperature. This project demonstrates the basic concept of sensor interfacing and automatic control using Arduino.

Smart Object Detection and Environment Control System Using LDR and Ultrasonic Sensor (Project 2)

It extends the concept of automation by incorporating an ultrasonic sensor along with an LDR. In this project, the LED and fan are activated when low light (Dark)conditions are detected and an object is present within a predefined distance. This project highlights the use of distance-based sensing for intelligent and responsive automation.

1.1 Application and Scope

Project 1: Automatic Light and Temperature Based Control System Using LDR and DHT11 Sensor

- This project focuses on automatic control of lighting and cooling systems based on surrounding light intensity and temperature.
- An LDR sensor is used to detect ambient light conditions and control the LED accordingly.
- A DHT11 sensor continuously monitors temperature to control the operation of the fan.
- The system helps in reducing manual effort and improves energy efficiency in indoor environments.

Project 2: Smart Object Detection and Environment Control System Using LDR and Ultrasonic Sensor

- This project is designed to control electrical devices based on light conditions and object presence.
- The LDR sensor detects darkness, while the ultrasonic sensor measures the distance of nearby objects.
- When an object is detected within a predefined range in dark conditions, the LED and fan are turned ON automatically.
- The system enhances automation, safety, and efficient use of electrical appliances.

1.2 Task Identification and Execution

1. Component Collection and Setup

Gathered Arduino Uno, LDR, DHT11, LED, DC fan, BC548Transistor, diode, resistors, Ultrasonic Sensor, jumper wires, and arranged all parts on a breadboard for circuit assembly.

2. Sensor Connections and Calibration

Connected the LDR as a voltage divider to the analog pin and attached the DHT11 to a digital pin. Checked that both sensors gave proper readings for light and temperature.

3. Output Device Wiring and Switching Setup

Connected the LED to the Arduino with a current-limiting resistor and wired the fan using the S8550 PNP transistor for switching. Added a diode across the fan terminals to protect the circuit from back EMF and ensured correct transistor orientation before testing.

4. Arduino Code Development

Wrote code to read sensor values and control the LED and fan automatically based on light and temperature levels using simple conditional logic.

5. Circuit Testing and Troubleshooting

Tested the circuit by varying light and temperature to check if the LED and fan responded correctly. Fixed wiring issues related to the transistor and verified that both outputs were working smoothly without manual switching.

6. System Verification and Observations

Observed the behaviour of the system under different conditions to confirm stable operation. Ensured the LED and fan turned ON and OFF automatically according to the programmed thresholds and checked that the circuit performed consistently.

7. Final Demonstration and Documentation

Completed the final working setup and demonstrated the output for different light and temperature levels. Prepared documentation, including circuit diagrams and observations, to support the project report and explain the working clearly

2. COMPONENTS AND TOOLS

1.Arduino Uno:

Arduino Uno was used as the main microcontroller board to read the sensor values and control the LED and fan. It provided the required input and output pins for connecting all components.

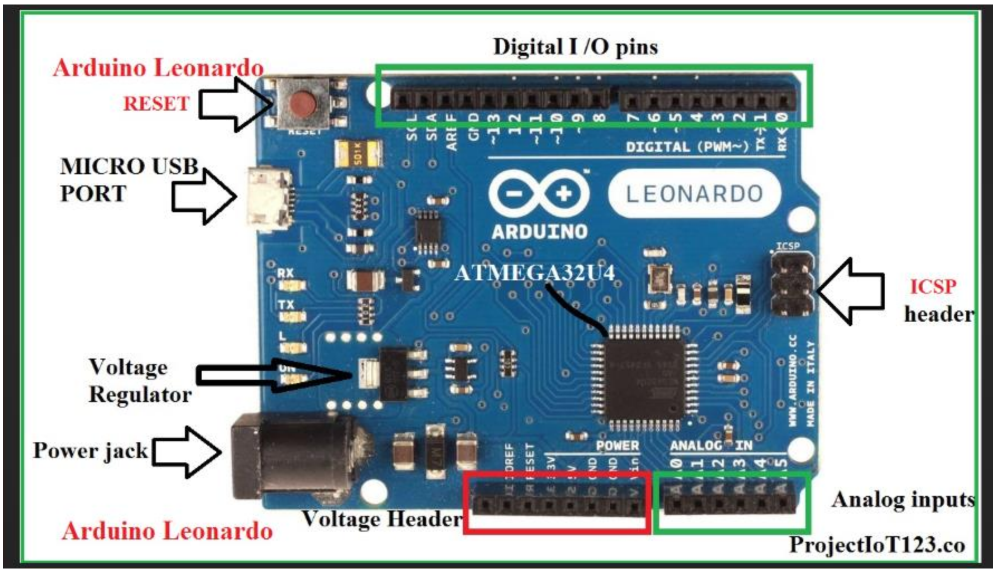


Figure 1: Arduino Uno

1. **Input**
 The Arduino Uno receives input signals from various sensors connected to its pins. In this project, inputs are obtained from sensors such as the LDR (light sensor), temperature sensor (DHT11), and other possible sensors. These inputs can be in the form of digital signals (HIGH/LOW) or analog signals (variable voltage), which are read through digital pins or analog input pins (A0–A5).

2. **Processing**
 The Arduino Uno processes the input data using the ATmega328P microcontroller. The board is programmed using the Arduino IDE, where logical conditions and control algorithms are written in embedded C/C++. Based on the programmed logic, the microcontroller analyses sensor values, compares them with predefined thresholds, and makes decisions accordingly. This processing stage is the core functioning of the Arduino Uno.

3. **Output**
 After processing, the Arduino Uno generates output signals to control external devices. Outputs are provided through digital pins or PWM (Pulse Width Modulation) pins. In this project, outputs are used to control devices such as LEDs and fans. PWM allows the Arduino to simulate variable voltage by rapidly switching the output ON and OFF, enabling efficient control of actuators.

Table 1. Components of Arduino Uno Board with their Details

S. No.	Component	Details
1	ATmega328P Microcontroller	An 8-bit microcontroller that contains flash memory, SRAM, EEPROM, timers, ADC, and communication interfaces.
2	ICSP Pins	Used for in-circuit serial programming of the microcontroller firmware.
3	Power LED	Indicates that the board is receiving power.
4	Digital I/O Pins	14 digital pins (D0–D13) used for input and output operations.
5	TX and RX LEDs	Indicate serial communication activity.
6	AREF Pin	Provides an external reference voltage for analog inputs.

7	Reset Button	Restarts the microcontroller and re-executes the program.
8	USB Port	Used for programming the board and serial communication.
9	Crystal Oscillator	Operates at 16 MHz and provides clock signals to the microcontroller.
10	Voltage Regulator	Regulates the input voltage to a stable 5V supply.
11	GND Pins	Provide ground connections for the circuit.
12	Vin Pin	Used to supply external input voltage to the board.
13	Analog Pins	Six analog input pins (A0–A5) used to read analog sensor data.

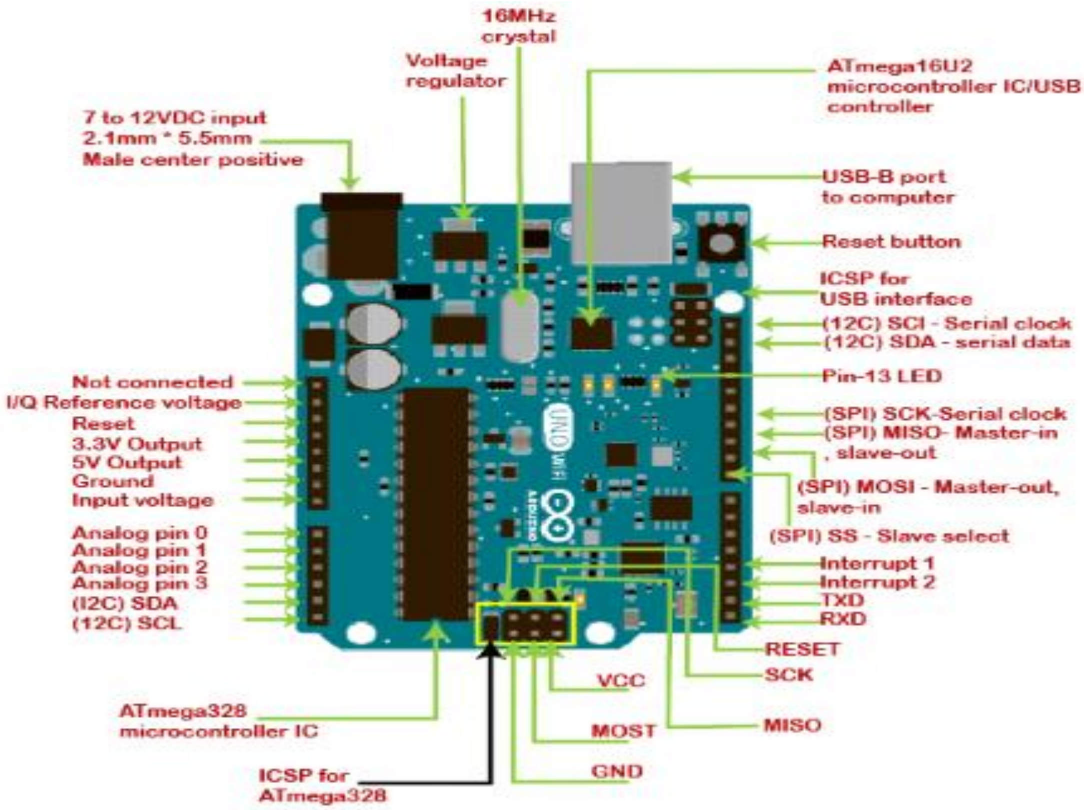


Figure 2: Arduino Pin Diagram

Technical Specifications of Arduino Uno in Details:

- **Microcontroller:** ATmega328P
- **Operating Voltage:** 5V
- **Recommended Input Voltage:** 7–12V
- **Input Voltage Limit:** 6–20V
- **Digital I/O Pins:** 14 (6 PWM pins)
- **Analog Input Pins:** 6 (A0–A5)
- **Flash Memory:** 32 KB (0.5 KB used by bootloader)
- **Clock Speed:** 16 MHz

DETAILS OF COMPONENTS

Project 1

(Automatic Light and Temperature Based Control System Using LDR and DHT11 Sensor)

S. No.	Name of Component	Specifications / Description
1	Arduino Uno	Microcontroller board based on ATmega328P
2	LDR (Light Dependent Resistor)	Light sensor used to detect light intensity
3	DHT11 Sensor	Digital temperature and humidity sensor
4	LED	5 mm indicator LED for light output
5	DC Fan	Used for temperature-based ventilation
6	BC548 Transistor	NPN transistor used for fan switching
7	Diode (1N4007)	Protects circuit from back EMF
8	Resistor (220 Ω)	Current limiting resistor for LED
9	Resistor (1 kΩ – 10 kΩ)	Base resistor for transistor
10	Breadboard	Used for circuit connections
11	Jumper Wires	Male-to-male connectors
12	Power Supply	5V supply from Arduino

Project 2

(Smart Object Detection and Environment Control System Using LDR and Ultrasonic Sensor)

S. No.	Name of Component	Specifications / Description
1	Arduino Uno	Microcontroller board based on ATmega328P
2	LDR (Light Dependent Resistor)	Detects ambient light conditions
3	Ultrasonic Sensor (HC-SR04)	Measures distance of objects using sound waves
4	LED	5 mm indicator LED
5	DC Fan	Controlled based on object detection and light
6	BC548 Transistor	NPN transistor used for fan control
7	Diode (1N4007)	Prevents reverse voltage damage
8	Resistor (220 Ω)	Current limiting resistor for LED
9	Resistor (1 kΩ – 10 kΩ)	Transistor base resistor
10	Breadboard	Platform for assembling the circuit
11	Jumper Wires	Male-to-male connecting wires
12	Power Supply	5V supply through Arduino Uno

2. LDR (Light Dependent Resistor):

The LDR was used to sense the surrounding light intensity. It helped determine when the LED should turn ON in low-light conditions by providing analog readings to the Arduino.

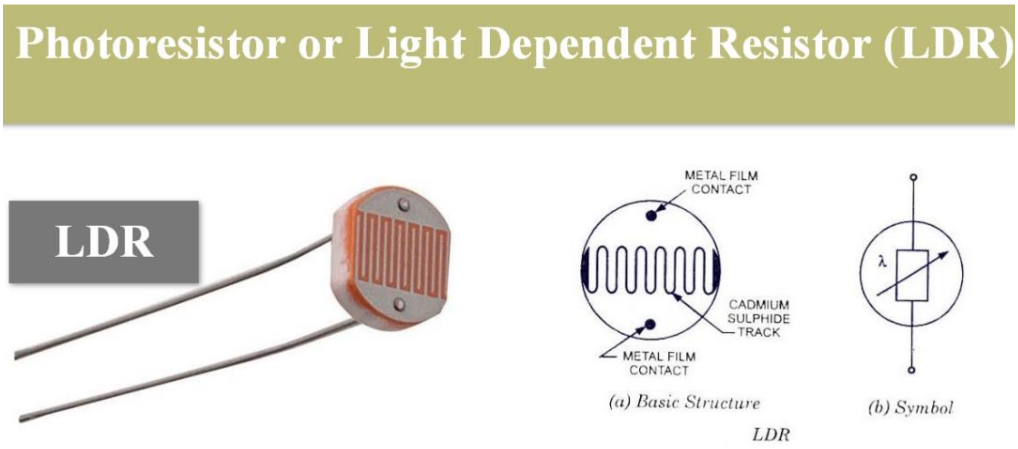


Figure 3: LDR (Light Dependent Resistor)

Table 2. General LDR Reference (for Interpretation)

Light Condition	Typical Analog Read Value (0-1023)
Dark (complete darkness)	0 - 50
Dim Light (dim room)	50 - 200
Moderate Light (normal room light)	200 - 500
Bright Light (bright room/overcast daylight)	500 - 800
Very Bright (direct sunlight/strong light)	800 - 1023

3. DHT11 Temperature Sensor:

The DHT11 sensor measured the temperature of the environment and sent the data to the Arduino. This helped in deciding when the fan needed to be switched ON automatically.

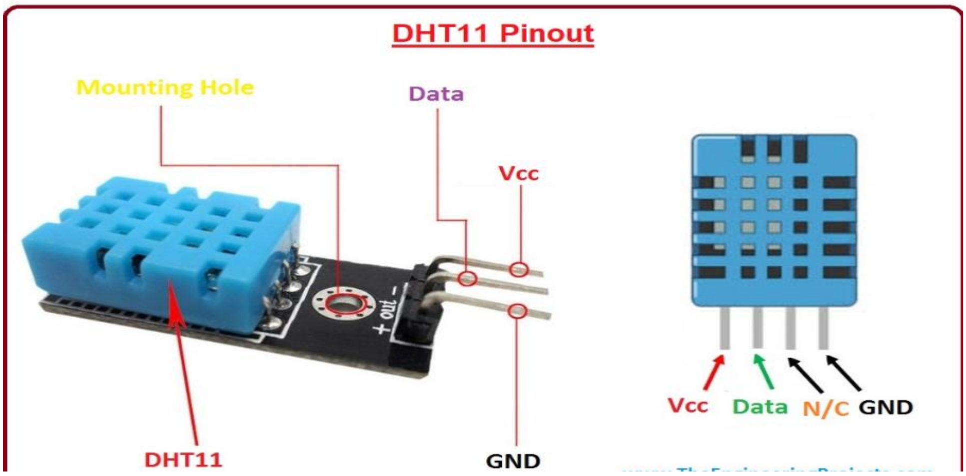


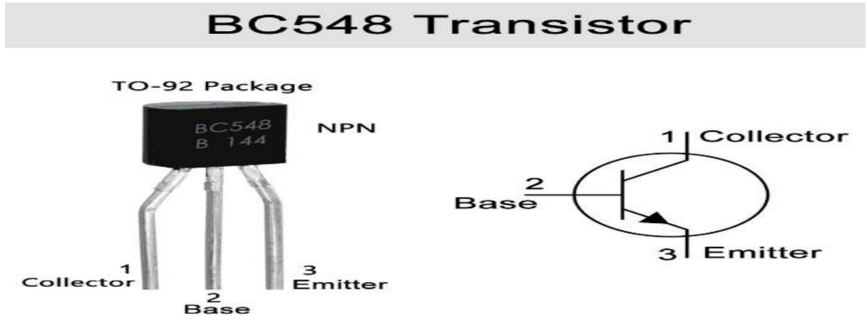
Figure 4: DHT11 Temperature Sensor

Table 3. General DHT11 Temperature Reference (°C)

Temperature (°C)	**Condition**	**Meaning**
Below 15°C	Cold	Fan remains OFF
15 – 25°C	Normal	Comfortable room temperature
25 – 35°C	Warm	Fan may turn ON
Above 35°C	Hot	Fan ON continuously

4.BC548Transistor:

The BC548 is an NPN bipolar junction transistor commonly used for switching and amplification applications in low-power electronic circuits. In this project, it is used to drive loads like the fan by allowing a small control signal from the Arduino to control a higher current device safely.



.

Figure 5: NPN (BC548Transistor)

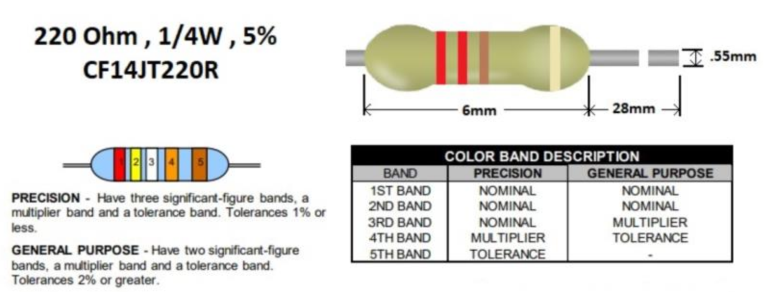
5. Diode (1N4007):

A diode was placed across the fan terminals to protect the circuit from back EMF generated by the motor. It ensured the safe operation of the transistor and the Arduino.

Figure 6: Diode (1N4007)

6. Resistors (220Ω):

The 220Ω resistor was used with the LED to limit current and prevent damage, while the 1.5kΩ resistor was used with the LDR to form a voltage divider for light measurement.



BAND	PRECISION	GENERAL PURPOSE
1ST BAND	NOMINAL	NOMINAL
2ND BAND	NOMINAL	NOMINAL
3RD BAND	NOMINAL	MULTIPLIER
4TH BAND	MULTIPLIER	TOLERANCE
5TH BAND	TOLERANCE	-

Figure 7: Resistors (220Ω)

7. Breadboard and Jumper Wires:

The breadboard and jumper wires were used to assemble the circuit without soldering. They helped in making neat, secure, and adjustable connections between all components.

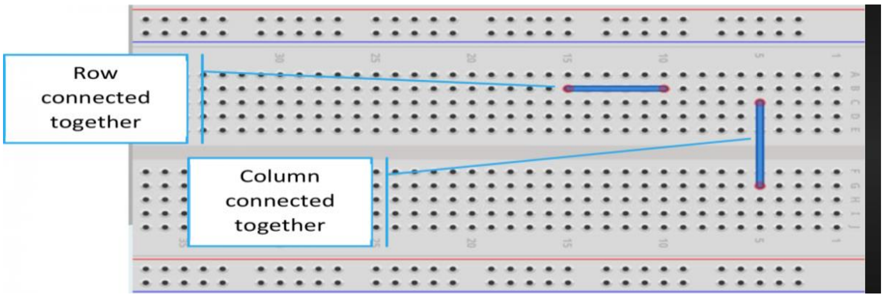


Figure 8: Breadboard

Figure 9: Jumper Wires

8. USB Cable and Arduino IDE:

A USB cable connected the Arduino to the computer for uploading the program. The Arduino IDE was used to write, edit, and upload the code that controlled the entire system.

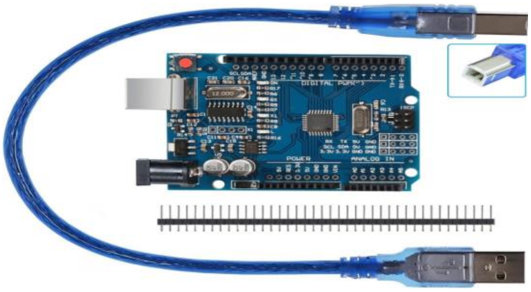


Figure 10: Jumper Wires

9. Ultrasonic Sensor (HC-SR04):

The HC-SR04 ultrasonic module measures the distance to nearby objects using sound waves and sends the distance value to the Arduino. In this project it is used to detect if a person comes within a set range so the LED or fan can turn ON automatically.

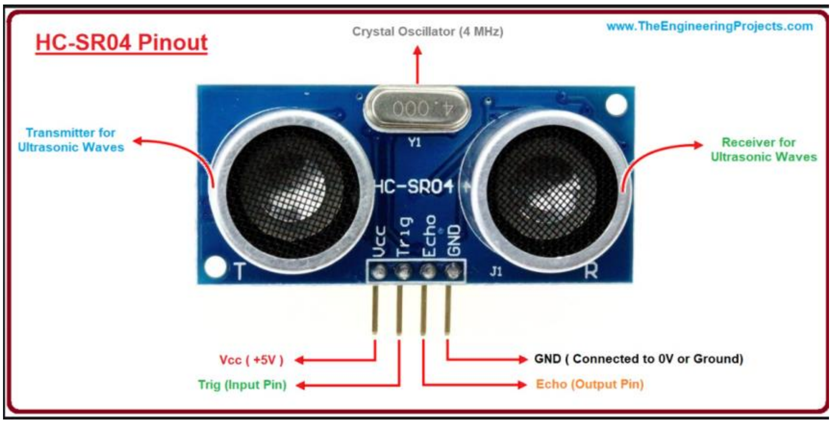


Figure 10: Ultrasonic Sensor (HC-SR04)

10. DC Fan (Cooling Fan):

The DC fan is used as an output device to provide ventilation and cooling. In this project, the fan is controlled by the Arduino and automatically turns ON when the required condition is met (such as darkness along with object detection or temperature threshold). This helps in demonstrating automatic environmental control without manual operation.

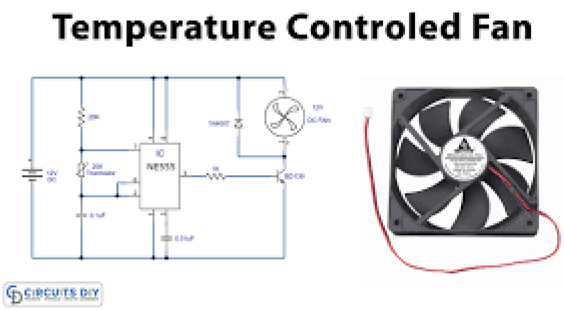


Figure 11: DC Fan (Cooling Fan)

3. CIRCUIT DIAGRAM

3.1 Simulation

Project 1 (Automatic Light and Temperature Based Control System Using LDR and DHT11 Sensor)

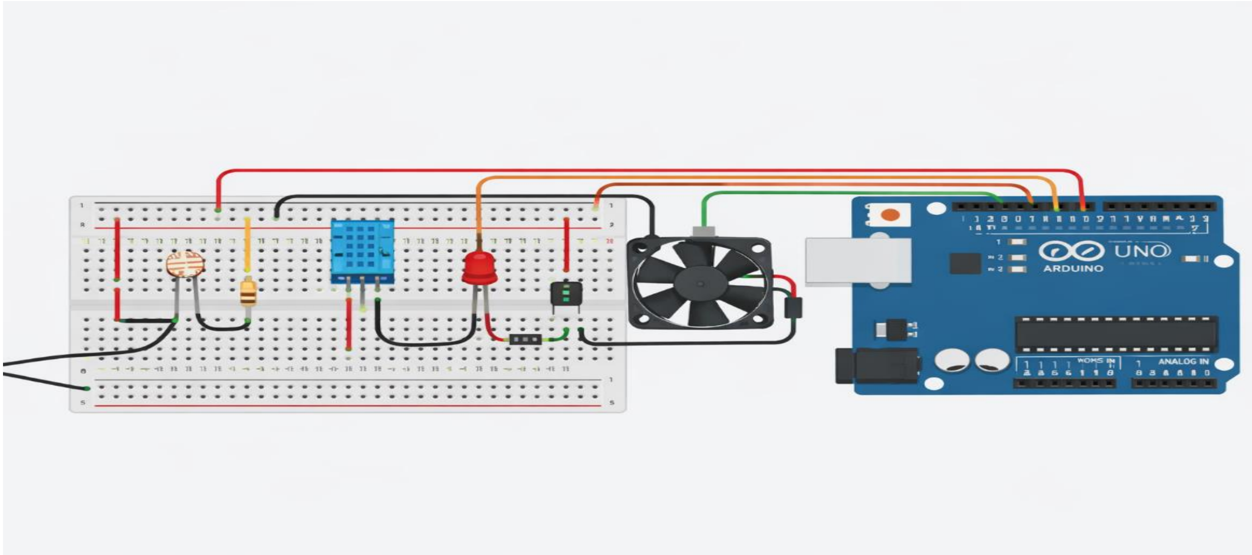


Figure 1: Automatic Light and Temperature Based Control System Using LDR and DHT11 Sensor

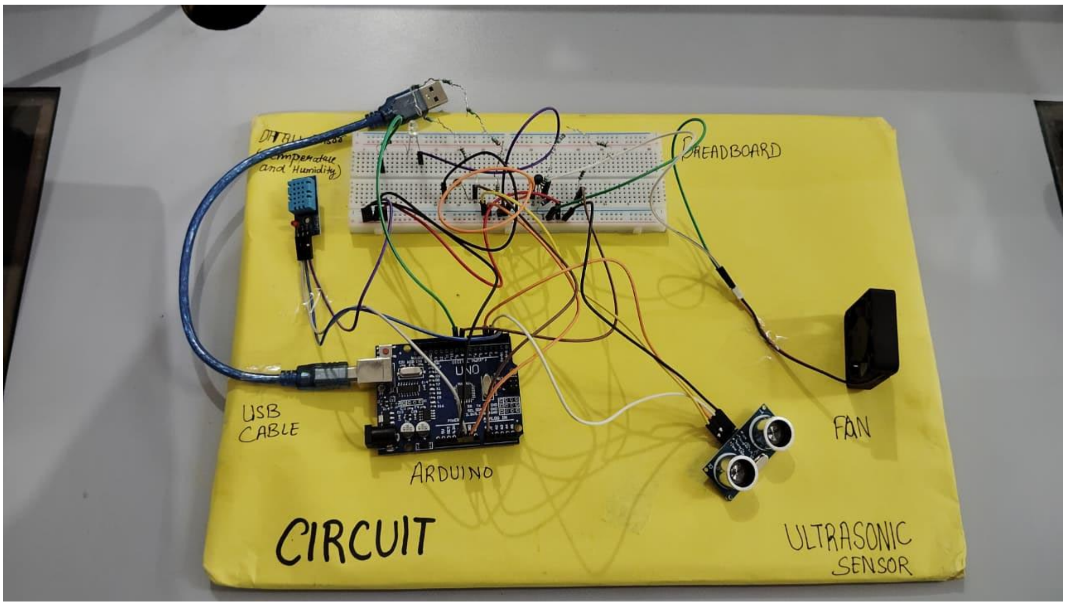


Figure 2: Automatic Light and Temperature Based Control System Using LDR and DHT11 Sensor

Simulation on Tinkercard

1. **Arduino Uno Board** is selected as the main controlling unit of the system to process sensor data and control output devices.

2. **A breadboard** is used to assemble the circuit, allowing easy and flexible connections between different electronic components.

3. **LDR (Light Dependent Resistor)** is connected in a voltage divider configuration:

- One terminal of the LDR is connected to **5V** supply.
- The other terminal of the LDR is connected to **analog pin A0** of the Arduino.
- A fixed resistor is connected between A0 and **GND** to complete the voltage divider.
- This arrangement allows the Arduino to read changing light intensity as analog values.

4. **DHT11 Temperature Sensor** is connected to measure ambient temperature:

- VCC pin of DHT11 is connected to **5V**.
- GND pin is connected to **ground**.
- Data pin is connected to **digital pin D2** of the Arduino.

5. **LED (Lighting Indicator)** is connected to represent automatic lighting control:

- The positive terminal of the LED is connected to **digital pin D9** through a current-limiting resistor.
- The negative terminal of the LED is connected to **GND**.

6. **DC Fan** is connected to the Arduino using a **BC548 transistor** as a switching device:

- The base of the transistor is connected to **digital pin D8** through a resistor.
- The emitter is connected to **GND**.
- The collector is connected to one terminal of the DC fan.
- The other terminal of the fan is connected to **5V** supply.

7. Common Ground **is provided to all components to ensure proper and stable circuit operation.**

8. The Arduino is programmed to continuously read light intensity from the LDR **and** temperature from the DHT11 sensor**.**

9. When the light intensity falls below the set threshold (dark condition)**, the LED automatically turns ON.**

10. When the temperature exceeds the predefined limit**, the DC fan automatically turns ON.**

11. The complete system is simulated using Tinkercad**, and real-time sensor values and output status are monitored through the** Serial Monitor**.**

Expected Readings and Output

1. LDR Sensor Output

- When surrounding light is **bright**, LDR value is **high** (around 700–1023).
- When surrounding light is **dark**, LDR value is **low** (around 0–300).
- LED turns **ON** only when light intensity falls below the set threshold.

2. DHT11 Temperature Output

- Temperature is displayed in **degrees Celsius (°C)** on the Serial Monitor.
- When temperature is **below 20°C**, the fan remains **OFF**.
- When temperature rises **above 20°C**, the fan turns **ON** automatically.

3. Serial Monitor Display

- LDR = 250 | Temp = 27°C | LED = ON | FAN = ON
- LDR = 850 | Temp = 23°C | LED = OFF | FAN = ON

Project 2 (Smart Object Detection and Environment Control System Using LDR and Ultrasonic Sensor)

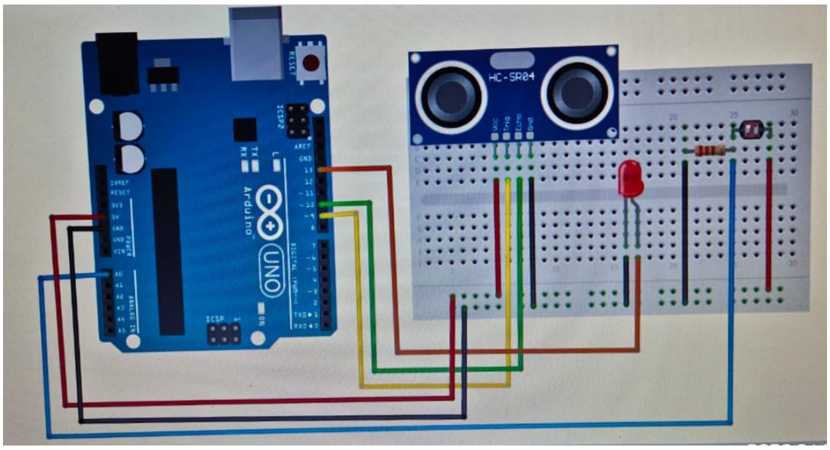


Figure 3: Project 2 (Smart Object Detection and Environment Control System Using LDR and Ultrasonic Sensor)

Simulation on Tinkercad

1. Arduino Uno is used as the central controller to process light and distance data.
2. The circuit is constructed on a breadboard for ease of connection and simulation.
3. LDR Sensor is connected in a voltage divider circuit:

- One terminal is connected to 5V.
- The second terminal is connected to analog pin A0.
- A resistor is connected between A0 and GND to sense light intensity accurately.

4. Ultrasonic Sensor (HC-SR04) is connected for object distance measurement:

- VCC pin is connected to 5V.
- GND pin is connected to ground.
- Trigger (TRIG) pin is connected to a digital output pin of Arduino.
- Echo (ECHO) pin is connected to a digital input pin of Arduino.

5. LED is connected to a digital output pin through a resistor to indicate automatic lighting operation.
6. DC Fan is connected using a transistor to safely control motor operation from the Arduino.
7. All components share a common ground to maintain proper electrical reference.
8. The Arduino program calculates the distance of an object using ultrasonic sensor echo timing.
9. The system continuously checks both light intensity and object distance.
10. When the surrounding environment is dark and an object is detected within the defined distance range, both the LED and fan turn ON automatically.
11. If sufficient light is available or no object is detected, the LED and fan remain OFF.
12. The complete setup is simulated in Tinkercad, and system behaviour is verified using the Serial Monitor.

Expected Readings and Output

1. Ultrasonic Sensor Output

- Distance is measured in **centimetres (cm)**.
- When no object is present, distance is **greater than threshold** (e.g., >50 cm).
- When an object comes **within threshold distance** (e.g., ≤30 cm), detection occurs.

2. Combined LDR and Ultrasonic Sensor Logic

- If the environment is **dark** AND

- If an object is detected **within the specified distance**.
- Then the **LED and fan turn ON automatically**.

3. Serial Monitor Display

- LDR = 280 | Distance = 25 cm | LED = ON | FAN = ON
- LDR = 900 | Distance = 20 cm | LED = OFF | FAN = OFF

3.2 Working of Circuit

The working of the circuit explains how different sensors and components interact with the Arduino Uno to control the LED and fan automatically. The system continuously monitors environmental conditions such as light intensity, temperature, and object presence. Based on the sensor readings and predefined conditions, the Arduino processes the data and activates or deactivates the output devices. This makes the system automatic, efficient, and suitable for real-time applications.

Project 1: Automatic Light and Temperature Based Control System Using LDR and DHT11 Sensor

Working:

1. **Power Supply and Initialization**
    - The Arduino Uno is powered using a USB cable or external power supply.
    - All sensors (LDR and DHT11) and output devices (LED and DC fan) are initialized when the system starts.
2. **LDR-Based Light Detection**
    - The LDR is connected in a voltage divider circuit with a fixed resistor.
    - One end of the LDR is connected to 5V, and the other end is connected to a resistor whose opposite end is connected to GND.
    - The junction points between the LDR and resistor is connected to analog pin A0 of the Arduino.
    - The Arduino reads the analog value from the LDR to determine light intensity.
    - When the surrounding light decreases (dark condition), the LDR resistance increases, changing the voltage level.

- If the light intensity falls below the threshold value, the LED turns ON automatically.
- When sufficient light is present, the LED remains OFF.

3. **DHT11-Based Temperature Monitoring**
    - The DHT11 sensor continuously measures the ambient temperature.
    - The sensor sends digital temperature data to the Arduino through a data pin.
    - The Arduino compares the measured temperature with a predefined threshold value.
4. **Fan Control Based on Temperature**
    - When the temperature rises above the set threshold, the Arduino sends a signal to the transistor.
    - The transistor acts as a switch and turns the DC fan ON.
    - When the temperature drops below the threshold, the fan is turned OFF automatically.
5. **Real-Time Monitoring**
    - Sensor readings such as LDR values, temperature, LED status, and fan status are displayed on the Serial Monitor.
    - This allows real-time observation and verification of system performance.

Project 2: Smart Object Detection and Environment Control System Using LDR and Ultrasonic Sensor

Working:

1. **System Initialization**
    - The Arduino Uno initializes the LDR sensor, ultrasonic sensor, LED, and DC fan.
    - Proper pin configuration ensures accurate input and output operation.
2. **LDR-Based Light Detection**
    - Similar to Project 1, the LDR detects ambient light levels using a voltage divider circuit.
    - The Arduino continuously reads the LDR values from the analog input pin.
    - The system checks whether the environment is dark or bright.

3. **Ultrasonic Sensor-Based Object Detection**
    - The ultrasonic sensor emits ultrasonic waves through the trigger pin.
    - These waves reflect back when they hit an object and are received by the echo pin.
    - The Arduino calculates the distance of the object using the time taken for the echo to return.
    - If an object is detected within a predefined distance range, it is considered valid motion or presence.
4. **Combined Decision Logic**
    - The Arduino processes both sensor inputs together.
    - The LED and fan are activated only when:
        - The environment is dark (LDR condition satisfied), **and**
        - An object is detected within the set distance (ultrasonic sensor condition satisfied).
5. **Automatic Control of LED and Fan**
    - When both conditions are met, the Arduino turns ON the LED and DC fan automatically.
    - If either the light condition or object detection condition fails, the LED and fan are turned OFF.
    - This ensures energy-efficient and intelligent operation.
6. **Observation Through Serial Monitor**
    - LDR readings, distance values from the ultrasonic sensor, and output status are displayed on the Serial Monitor.
    - This helps in analysing system behaviour and verifying correct operation.

4. RESULT AND OBSERVATIONS

Interpretation of Results

Project 1: Automatic Light and Temperature Based Control System Using LDR and DHT11 Sensor

- When the LDR (smoothed) value is less than the threshold, it indicates a dark environment.
  → The LED turns ON automatically.
- When the LDR value becomes equal to or greater than the threshold, it indicates sufficient light.
  → The LED turns OFF, preventing unnecessary power usage.
- The temperature recorded by the DHT11 sensor remains between 24°C to 27°C.
  → Since the temperature is above the predefined limit, the fan remains ON continuously.
- The system responds accurately and in real time, as seen in the Serial Monitor output.

Observations

- The LDR sensor correctly detects changes in ambient light conditions.
- The LED operates only during dark conditions, ensuring energy efficiency.
- The DHT11 sensor provides stable and accurate temperature readings.
- The fan automatically turns ON when the temperature exceeds the set limit.
- The overall system works reliably without manual intervention.

Table 3. This is section discuss the results which obtained from circuit (Automatic Light and Temperature Based Control System Using LDR and DHT11 Sensor)

S. No.	LDR Value (Smoothed)	Threshold	Temperature (°C)	LED Status	Fan Status	Observation
1	10 – 20	42	24.20	ON	ON	Dark condition detected
2	21 – 29	42	24.20	ON	ON	Low light, LED remains ON
3	30 – 31	42	24.80	ON	ON	Still below threshold
4	85	85	27.70	OFF	ON	Bright environment
5	86 – 87	85	27.60	OFF	ON	High light intensity

Interpretation of Results

Project 2: Smart Object Detection and Environment Control System Using LDR and Ultrasonic Sensor

- The LDR sensor continuously measures the surrounding light intensity and sends analog values to the Arduino.
- When the LDR value is lower than the threshold, it indicates a dark condition, due to which the LED turns ON automatically.
- When the LDR value exceeds the threshold, it indicates a bright environment, and the LED turns OFF automatically.
- The DHT11 sensor measures the surrounding temperature in real time.
- When the temperature rises above the set temperature limit, the fan turns ON automatically to provide cooling.
- When the temperature is below the limit, the fan remains OFF.
- The use of smoothing and hysteresis prevents unnecessary flickering of the LED and ensures stable operation.

Observations

- The system successfully detects light intensity changes using the LDR sensor.
- Automatic switching of the LED based on darkness and brightness is observed.
- The fan operates correctly based on real-time temperature values.
- The system remains stable and does not respond to minor fluctuations in sensor readings.
- Real-time data displayed on the Serial Monitor helps in easy monitoring and debugging of the system.

Table 4. This is section discuss the results which obtained from circuit (LDR and Temperature Based Control)

S. No.	**LDR Value (Smoothed)**	**Threshold Value**	**Temperature (°C)**	**LED Status**	**Fan Status**	**Observation**
1	20	42	24.20	ON	ON	Dark condition detected, LED turned ON and fan activated
2	24	42	24.20	ON	ON	Stable dark condition, system working normally
3	31	42	24.80	ON	ON	Light still below threshold, LED remains ON
4	16	42	24.70	ON	ON	Increased darkness, LED stays ON
5	10	42	24.70	ON	ON	Very dark condition, LED and fan continuously ON
6	85	85	27.70	OFF	ON	Bright condition detected, LED turned OFF
7	86	85	27.60	OFF	ON	Light intensity above threshold, LED remains OFF
8	87	85	27.30	OFF	ON	Bright environment, system stable

5. CONCLUSION

In Project 1: Automatic Light and Temperature Based Control System Using LDR and DHT11 Sensor, the system successfully demonstrated how environmental parameters such as light intensity and temperature can be monitored and used for automatic control. The LDR sensor continuously sensed ambient light levels to control the LED, ensuring that the light turned ON only in dark conditions and OFF in sufficient light. Simultaneously, the DHT11 sensor measured real-time temperature to control the operation of the DC fan. This project highlights the effective use of sensors and Arduino Uno to reduce manual intervention, improve energy efficiency, and provide a simple yet reliable automation solution suitable for homes and smart environments.

In Project 2: Smart Object Detection and Environment Control System Using LDR and Ultrasonic Sensor, the system was designed to enhance automation by combining light detection with object detection. The LDR was used to identify dark conditions, while the ultrasonic sensor measured the distance of nearby objects. When an object was detected within a predefined distance under dark conditions, both the LED and the fan were activated automatically.

This project demonstrates a smarter control approach by integrating multiple sensor conditions, making the system more responsive and practical for real-world applications such as automatic lighting in corridors, security zones, and energy-saving systems. Together, both projects showcase the practical implementation of Arduino-based automation and sensor integration.

REFERENCES

1. **https://docs.arduino.cc/**

2. **https://lastminuteengineers.com/**

3. **https://www.electronics-tutorials.ws/transistor/tran_3.html**

4. **https://learn.adafruit.com/lesson-0-getting-started/overview**

ANNEXURE

Project 1: Automatic Light and Temperature Based Control System Using LDR and DHT11 Sensor

```
// Fan + LED control with LDR auto-calibration (for R_fixed ≈ 1.5kΩ)
// Pins:
//  A0  -> LDR voltage divider midpoint
//  D2  -> DHT11 data
//  D8  -> Fan control (via transistor base + base resistor)
//  D9  -> LED
#include <DHT.h>
#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
const int ldrPin = A0;
const int ledPin = 9;
const int fanPin = 8;
const unsigned long calibMillis = 5000UL; // 5 seconds calibration
int minLdr = 1023;
int maxLdr = 0;
int threshold = 300; // fallback threshold if calibration fails
// smoothing (moving average)
const int SMOOTH_N = 6;
int smoothBuf[SMOOTH_N];
int smoothIdx = 0;
long smoothSum = 0;
void setup() {
  Serial.begin(9600);
  dht.begin();
```

```
pinMode(ledPin, OUTPUT);
pinMode(fanPin, OUTPUT);
digitalWrite(ledPin, LOW);
digitalWrite(fanPin, LOW);
// initialize smoothing buffer
for (int i = 0; i < SMOOTH_N; ++i) {
  smoothBuf[i] = analogRead(ldrPin);
  smoothSum += smoothBuf[i];
}
// Auto-calibration
unsigned long start = millis();
Serial.println(F("Auto-calibrating LDR for 5 seconds..."));
while (millis() - start < calibMillis) {
  int v = analogRead(ldrPin);
  if (v < minLdr) minLdr = v;
  if (v > maxLdr) maxLdr = v;
  Serial.print(F("Calib read: "));
  Serial.println(v);
  delay(200);
}
// Safety: if max/min still default, set reasonable defaults
if (minLdr == 1023) minLdr = 0;
if (maxLdr == 0) maxLdr = 1023;
// Choose midpoint as threshold, but bias slightly toward 'dark' if needed
threshold = (minLdr + maxLdr) / 2;

Serial.println(F("Calibration complete."));
Serial.print(F("minLdr = ")); Serial.print(minLdr);
Serial.print(F("   maxLdr = ")); Serial.print(maxLdr);
Serial.print(F("   threshold = ")); Serial.println(threshold);
```

```
  Serial.println(F("If LED logic seems inverted, either flip threshold logic or wiring."));
  delay(500);
}
void loop() {
  // Read & smooth LDR
  int raw = analogRead(ldrPin);
  smoothSum -= smoothBuf[smoothIdx];
  smoothBuf[smoothIdx] = raw;
  smoothSum += smoothBuf[smoothIdx];
  smoothIdx = (smoothIdx + 1) % SMOOTH_N;
  int ldrValue = smoothSum / SMOOTH_N;

  // Read temperature (DHT)
  float temp = dht.readTemperature();
  bool tempValid = !isnan(temp);

  // LED control (dark -> LED ON)
  // If your wiring inverts (LED wired to Vcc side), reverse logic
  if (ldrValue < threshold) {
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, LOW);
  }
 // Fan control based on temperature
  if (tempValid && temp > 15.0) {
    digitalWrite(fanPin, HIGH); // transistor will drive fan
  } else {
digitalWrite(fanPin, LOW);
  }
  // Serial output for debugging
  Serial.print(F("LDR(smoothed)=")); Serial.print(ldrValue);
```

```
  Serial.print(F("  thr=")); Serial.print(threshold);
  Serial.print(F("  | temp="));
  if (tempValid) Serial.print(temp);
  else Serial.print(F("N/A"));
  Serial.print(F("  | LED=")); Serial.print(digitalRead(ledPin));
  Serial.print(F("  | FAN=")); Serial.println(digitalRead(fanPin));

  delay(500);
}
```

Project 2: Smart Object Detection and Environment Control System Using LDR and Ultrasonic Sensor

```
// LED + Fan control using LDR (auto-calibration) + Ultrasonic Sensor
// Conditions:
// DARK + Object detected → LED ON + FAN ON
// Pins:
// A0  -> LDR voltage divider
// D3  -> Ultrasonic TRIG
// D4  -> Ultrasonic ECHO
// D8  -> Fan control (via transistor)
// D9  -> LED

const int ldrPin = A0;
const int ledPin = 9;
const int fanPin = 8;

// Ultrasonic pins
const int trigPin = 3;
const int echoPin = 4;
// ---------- LDR AUTO CALIBRATION ----------
const unsigned long calibMillis = 5000UL;
int minLdr = 1023;
```

```
int maxLdr = 0;
int threshold = 300;
// ---------- SMOOTHING ----------
const int SMOOTH_N = 6;
int smoothBuf[SMOOTH_N];
int smoothIdx = 0;
long smoothSum = 0;
// ---------- ULTRASONIC ----------
const int DIST_THRESHOLD = 10; // cm (object detected within 30cm)

void setup() {
 Serial.begin(9600);

 pinMode(ledPin, OUTPUT);
 pinMode(fanPin, OUTPUT);
 pinMode(trigPin, OUTPUT);
 pinMode(echoPin, INPUT);

 digitalWrite(ledPin, LOW);
 digitalWrite(fanPin, LOW);
// Initialize smoothing buffer
 for (int i = 0; i < SMOOTH_N; i++) {
  smoothBuf[i] = analogRead(ldrPin);
  smoothSum += smoothBuf[i];
 }
 // -------- AUTO CALIBRATION --------
 Serial.println("Auto-calibrating LDR for 5 seconds...");
 unsigned long start = millis();
 while (millis() - start < calibMillis) {
  int v = analogRead(ldrPin);
  if (v < minLdr) minLdr = v;
```

```
    if (v > maxLdr) maxLdr = v;
    delay(200);
  }
  threshold = (minLdr + maxLdr) / 2;

  Serial.println("Calibration Done");
  Serial.print("LDR Threshold = ");
  Serial.println(threshold);
}
void loop() {
  // -------- LDR READ & SMOOTH --------
  int raw = analogRead(ldrPin);
  smoothSum -= smoothBuf[smoothIdx];
  smoothBuf[smoothIdx] = raw;
  smoothSum += raw;
  smoothIdx = (smoothIdx + 1) % SMOOTH_N;
  int ldrValue = smoothSum / SMOOTH_N;
 bool isDark = (ldrValue < threshold);

  // -------- ULTRASONIC READ --------
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
 long duration = pulseIn(echoPin, HIGH, 25000); // timeout
  int distance = duration * 0.034 / 2;
bool objectDetected = (distance > 0 && distance <= DIST_THRESHOLD);
// -------- CONTROL LOGIC --------
  if (isDark && objectDetected) {
    digitalWrite(ledPin, HIGH);   // LED ON
```

```
    digitalWrite(fanPin, HIGH);   // FAN ON
  } else {
    digitalWrite(ledPin, LOW);    // LED OFF
    digitalWrite(fanPin, LOW);    // FAN OFF
  }
 // -------- SERIAL MONITOR --------
  Serial.print("LDR=");
  Serial.print(ldrValue);
  Serial.print(" | Dark=");
  Serial.print(isDark ? "YES" : "NO");
  Serial.print(" | Distance=");
  Serial.print(distance);
  Serial.print("cm");
  Serial.print(" | Object=");
  Serial.print(objectDetected ? "YES" : "NO");
  Serial.print(" | LED=");
  Serial.print(digitalRead(ledPin) ? "ON" : "OFF");
  Serial.print(" | FAN=");
  Serial.println(digitalRead(fanPin) ? "ON" : "OFF");

  delay(500);
}
```