**Practical 5**

**Aim:- Understand how an LDR (Light Dependent Resistor) works and how it detects light intensity**

**Theory:**

* **Light Dependent Resistor (LDR):** An LDR, also known as a photoresistor, is a type of resistor whose resistance varies with the intensity of light falling on it. LDRs are commonly used in light-sensing applications.
* **Operating Principle:**
  + **Resistance and Light Intensity:** The resistance of an LDR decreases as the light intensity increases. In bright light, the resistance is low; in low light, the resistance is high.
  + **Ohm's Law:** The relationship between voltage, current, and resistance is given by Ohm’s Law: V=IRV = IRV=IR, where VVV is voltage, III is current, and RRR is resistance.
* **LDR Characteristics:**
* **Dark Resistance:** High resistance when no light is present.
* **Light Resistance:** Low resistance when exposed to light.
* **Response Time:** Time taken for the LDR to respond to changes in light intensity.

**Materials Required**

* 1. LDR sensor
  2. Resistor (e.g., 10kΩ)
  3. Breadboard
  4. Arduino or other microcontroller
  5. Connecting wires
  6. Light source (e.g., lamp)
  7. Multimeter (for calibration)

**Procedure**

#### Part 1: System Setup

* **Circuit Assembly:**

1. Connect the LDR and resistor in series on the breadboard.
2. Connect one end of the series combination to the positive voltage supply (e.g., 5V from the Arduino).
3. Connect the other end to ground.
4. Connect the junction between the LDR and the resistor to an analog input pin on the Arduino.

#### Part 2: Calibration and Measurement

* **Calibration:**
  + Measure the resistance of the LDR in different light conditions using a multimeter.
  + Record the light intensity levels (e.g., in lux) corresponding to different resistances.
* **Data Collection:**
  + Write and upload the Arduino code to read analog values from the LDR.
  + Vary the light intensity by changing the distance or intensity of the light source.
  + Record the analog readings from the Arduino.

**Code**

// Common CATHODE RGB + LDR (AO -> A0)

// One resistor per color leg is recommended.

const int ldrPin   = A0;   // LDR analog output (AO)

const int redPin   = 9;    // RGB red

const int greenPin = 10;   // RGB green

const int bluePin  = 11;   // RGB blue

void setup() {

  Serial.begin(9600);

  pinMode(redPin, OUTPUT);

  pinMode(greenPin, OUTPUT);

  pinMode(bluePin, OUTPUT);

}

// For common cathode: HIGH turns a color ON

void setColor(bool r, bool g, bool b) {

  digitalWrite(redPin,   r ? HIGH : LOW);

  digitalWrite(greenPin, g ? HIGH : LOW);

  digitalWrite(bluePin,  b ? HIGH : LOW);

}

void loop() {

  int ldr = analogRead(ldrPin);   // 0..1023

  Serial.println(ldr);

  // Tune these for your room lighting

  const int DARK\_MAX   = 400;   // below this = dark

  const int MEDIUM\_MAX = 800;   // 400..799 = medium, 800+ = bright

  if (ldr < DARK\_MAX) {

    // Dark -> Red

    setColor(true, false, false);

  }

  else if (ldr < MEDIUM\_MAX) {

    // Medium -> Green

    setColor(false, true, false);

  }

  else {

    // Bright -> Blue

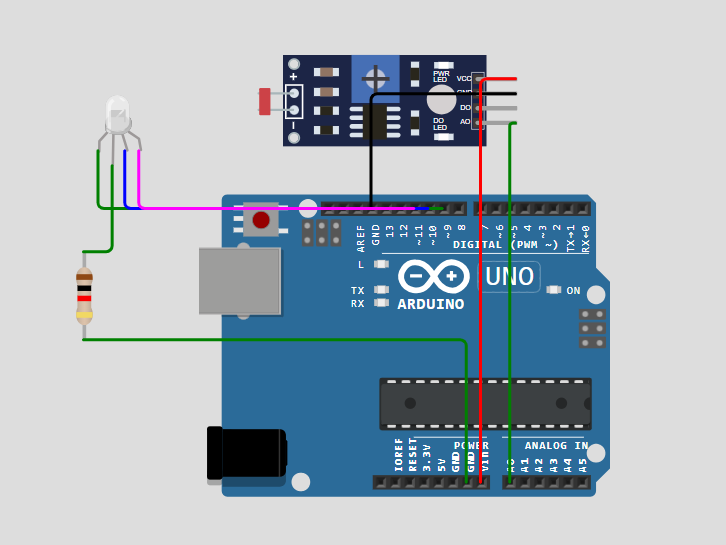
    setColor(false, false, true);

  }

  delay(200);

}

**Image**



*Fig;- Setup showing the connection of the LDR with the Arduino and light source.*

**Conclusion:**The LDR changes its resistance with light — less resistance in bright light and more in darkness. Arduino readings match the light levels. It works well in normal lighting, but is less accurate in very dark or very bright conditions. The sensor responds within a few seconds, making it suitable for real-time use.

**Results:**

* **Light Intensity vs. Analog Reading:**
  + **Bright Light:** Analog Reading ~ 100 (low resistance)
  + **Medium Light:** Analog Reading ~ 512
  + **Low Light:** Analog Reading ~ 900 (high resistance)
* **Response Time:** The LDR response time is within a few seconds, suitable for applications where real-time light monitoring is required.