Title: LDR Light Sensor Project

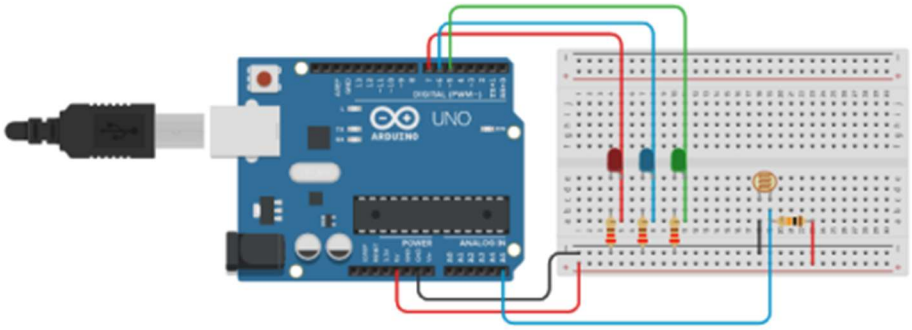
Introduction:

This project demonstrates a simple yet effective smart street lighting system using an LDR (Light Dependent Resistor) and an Arduino. The system is designed to automatically adjust the brightness of an LED based on surrounding light levels.

The LDR is a light-sensitive sensor whose resistance changes depending on the amount of ambient light—resistance decreases in bright light and increases in darkness. The Arduino reads these changes and uses Pulse Width Modulation (PWM) to control the LED's brightness. In bright conditions, the LED dims or turns off; in low light, it brightens automatically.

This method mimics real-world smart streetlights, improving energy efficiency by ensuring lights are only fully active when needed.

Circuit diagram:



Procedure:

One terminal of the LDR was connected to the 5V pin on the Arduino.

The other terminal of the LDR was connected to analog pin A0.

A 10kΩ resistor was placed between A0 and GND to form a voltage divider, allowing accurate light level readings.

The longer leg (anode) of the LED was connected to digital pin 5 through a 220Ω resistor to limit current.

The shorter leg (cathode) of the LED was connected directly to GND.

The Arduino was powered and connected to the computer using a USB cable, allowing both power supply and code uploading.

In the code, A0 was set as an input to read the LDR values.

Pin 5 was defined as a PWM output to control the LED brightness.

The analogRead(A0) function was used to get real-time light intensity data from the LDR.

These raw readings (typically between 54 and 974) were mapped to a PWM range of 255 to 0, so that brighter light results in a dimmer LED and vice versa.

The analogWrite(5, brightness) function was used to apply the calculated brightness to the LED.

Serial.print() was included to display LDR readings in the Serial Monitor for debugging or analysis.

A short 500 ms delay was added in the loop to ensure smooth brightness changes and avoid flickering.

Code:

```
#include<Arduino.h>

const int LDR = A0;

const int LED = 5;


void setup() {
  pinMode(LED, OUTPUT);
  Serial.begin(9600);
}
```

```
void loop() {
  int ldrValue = analogRead(LDR);
  int brightness = map(ldrValue, 54, 974, 255, 0);
  brightness = constrain(brightness, 0, 255);
  analogWrite(LED, brightness);
  Serial.print("LDR: ");
  Serial.print(ldrValue);
  Serial.print(" | LED Brightness: ");
  Serial.println(brightness);
  delay(500);
}
```

Discussion:

This system worked by taking advantage of a key property of the LDR: its resistance goes up when the surrounding light goes down. Because of that, the voltage read by the Arduino on the analog pin dropped in darker conditions. That lower voltage was then converted (or *mapped*) into a higher PWM value, which made the LED glow brighter in the dark and dimmer in bright light. This inverse relationship helped the LED respond smoothly and proportionally to changes in light.

The Serial Monitor displayed real-time readings, which showed that the sensor was working accurately and the system was reacting just as expected. Adding a short 500 ms delay between each reading helped keep the LED transitions smooth, preventing any flickering or jittery behavior.

In the end, this project showed that even with basic components, it's possible to create a low-cost and reliable light-sensitive system. It's a simple example, but the concept can be scaled up for practical use—like in smart streetlights or other automated lighting systems that adjust to their surroundings.