



INTERNET OF THINGS

BY VANSHIKA SHARMA
STUDENT ID: CA/SE1/21214

TASK 2: *SESOR-BASED STIMULATION*

Light Intensity
Measurement
using LDR sensor
and Arduino on
TinkerCAD



CODE:

```
// Include the library for the Liquid Crystal Display (LCD)
#include <LiquidCrystal.h>

// Initialize the library with the numbers of the interface pins
// LCD RS pin connected to Arduino D12
// LCD Enable pin connected to Arduino D11
// LCD D4 pin connected to Arduino D5
// LCD D5 pin connected to Arduino D4
// LCD D6 pin connected to Arduino D3
// LCD D7 pin connected to Arduino D2
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

// Define the pin connections based on the Tinkercad diagram
const int ldrPin = A0; // LDR connected to Analog Pin A0
const int ledPin = 9;  // LED connected to Digital Pin 9 (PWM)

void setup() {
  // Set up the LED pin as an output
  pinMode(ledPin, OUTPUT);

  // Set up the LCD's number of columns and rows
  lcd.begin(16, 2);

  // Print a starting message to the LCD
  lcd.print("Light Sensor Demo");

  // Initialize Serial Monitor for debugging (optional)
  Serial.begin(9600);
}

void loop() {
  // Read the analog value from the LDR (0-1023)
  int ldrValue = analogRead(ldrPin);
```

```
// Map the LDR value to the LED brightness (0-255)
// NOTE: LDR resistance decreases as light increases.
// We want the LED to turn ON when it's DARK (low LDR value),
// so we reverse the mapping by swapping the min/max output values.
// If you want the LED to turn ON when it's BRIGHT, use: map(ldrValue, 0, 1023, 0, 255);
int ledBrightness = map(ldrValue, 0, 1023, 255, 0);

// Constrain the mapped value to ensure it stays within the PWM range
ledBrightness = constrain(ledBrightness, 0, 255);

// Set the LED brightness using PWM (Analog Write)
analogWrite(ledPin, ledBrightness);

// Clear the display for new data
lcd.clear();

// Print the LDR value on the first line
lcd.setCursor(0, 0); // Column 0, Row 0
lcd.print("LDR Value: ");
lcd.print(ldrValue);

// Print the LED brightness on the second line
lcd.setCursor(0, 1); // Column 0, Row 1
lcd.print("LED PWM: ");
lcd.print(ledBrightness);

// Print values to Serial Monitor (optional)
Serial.print("LDR: ");
Serial.print(ldrValue);
Serial.print(" | LED PWM: ");
Serial.println(ledBrightness);

// Wait for a short period before the next reading
delay(500);
}
```



CODE EXPLANATION

Line(s)	Code / Concept	Explanation
<code>#include <LiquidCrystal.h></code>	Include LCD library	This line adds the LiquidCrystal library which allows the Arduino to communicate with an LCD display .
<code>LiquidCrystal lcd(12, 11, 5, 4, 3, 2);</code>	LCD pin setup	Creates an LCD object named lcd . The numbers (12, 11, 5, 4, 3, 2) represent the Arduino pins connected to the LCD (RS, EN, D4, D5, D6, D7).
<code>const int ldrPin = A0;</code>	LDR pin	Defines the pin where the Light Dependent Resistor (LDR) is connected (Analog pin A0).
<code>const int ledPin = 9;</code>	LED pin	Defines the LED connection pin (Digital pin 9, supports PWM).

SETUP FUNCTION

Code	Explanation
<code>void setup()</code>	Runs once when the Arduino starts.
<code>pinMode(ledPin, OUTPUT);</code>	Sets pin 9 (LED) as output so Arduino can control it.
<code>lcd.begin(16, 2);</code>	Initializes LCD with 16 columns and 2 rows .
<code>lcd.print("Light Sensor Demo");</code>	Displays a welcome message on the LCD.
<code>Serial.begin(9600);</code>	Starts serial communication at 9600 baud rate — used to print data on the Serial Monitor for debugging.

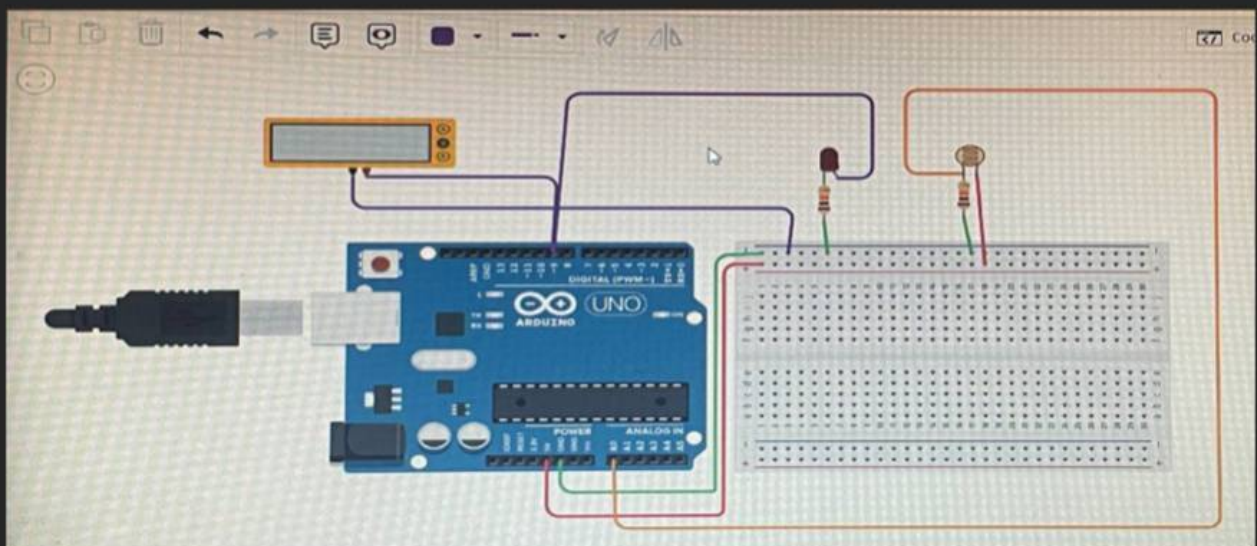
LOOP FUNCTION

Code / Concept	Explanation		
<pre>int ldrValue = analogRead(ldrPin);</pre>	Reads the analog input from the LDR (value between 0–1023).  Low value = dark ,  High value = bright .	<pre>lcd.setCursor(0, 0);</pre>	Sets the cursor to the first row, first column .
<pre>int ledBrightness = map(ldrValue, 0, 1023, 255, 0);</pre>	Converts (maps) the LDR reading to a PWM range (0–255) . Reversed mapping so that LED brightness increases when it's dark .	<pre>lcd.print("LDR Value: "); lcd.print(ldrValue) ;</pre>	Displays the current LDR sensor reading .
<pre>ledBrightness = constrain(ledBright ness, 0, 255);</pre>	Ensures the brightness value stays between 0 and 255 .	<pre>lcd.setCursor(0, 1);</pre>	Moves to the second row .
<pre>analogWrite(ledPin, ledBrightness);</pre>	Sends a PWM signal to the LED pin — controls brightness.	<pre>lcd.print("LED PWM: "); lcd.print(ledBright ness);</pre>	Displays the LED brightness level (PWM value).
<pre>lcd.clear();</pre>	Clears the LCD screen before printing new data.	<pre>Serial.print(...)/ Serial.println(...)</pre>	Prints both LDR and LED values on the Serial Monitor for debugging.
		<pre>delay(500);</pre>	Waits for 0.5 seconds before repeating the loop.

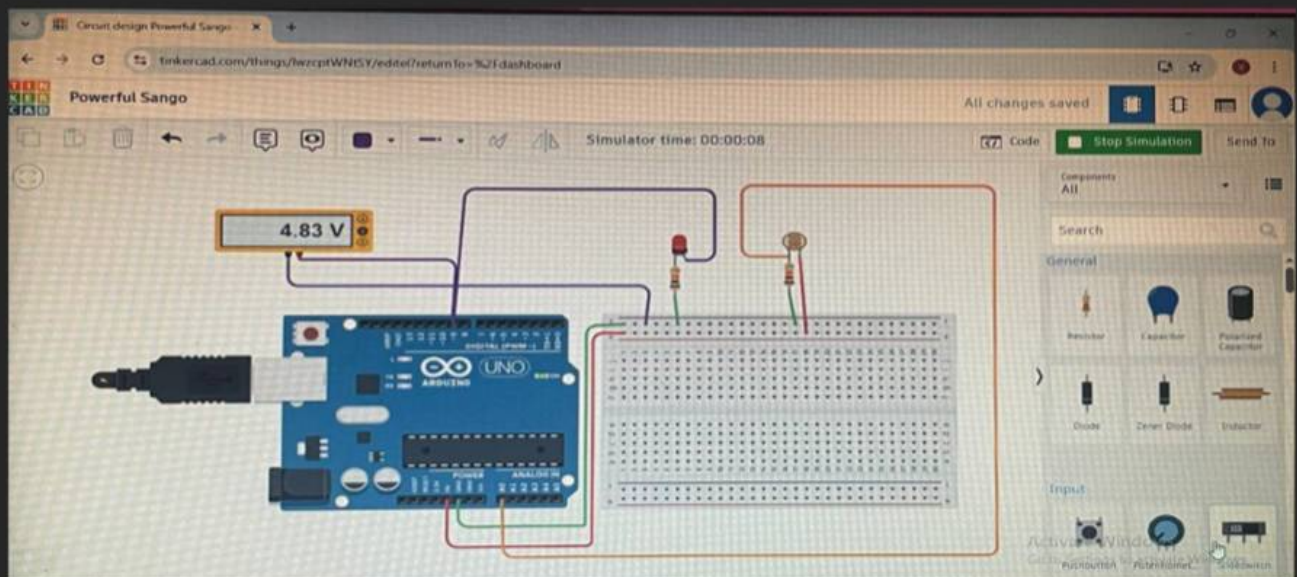
SUMMARY OF WORKING

1. The LDR senses ambient light intensity.
 2. The Arduino reads this analog value.
 3. Using `map()`, it converts that light value to a brightness level for the LED.
 4. When it's dark, the LED gets brighter.
 5. The LCD and Serial Monitor show real-time sensor data.
-

BEFORE SIMULATION :



AFTER SIMULATION :





Thank you