**ENVIRONMENTAL MONITORING SYSTEM**

A Project Report

Submitted

by

Bhanu Sri - 21911A3501

Mybu Naik - 21911A3502

Harshitha - 21911A3503

Haripriya - 21911A3504

Department of Artificial Intelligence

**Vidya Jyothi Institute of Technology**

**Batch 2025**

**Problem Statement:**

This project involves the design and implementation of an environmental monitoring system that measures and displays temperature, humidity, and light intensity while also controlling an LED based on the ambient light level. The system integrates a DHT11 sensor for accurate measurement of temperature and humidity and an LDR (Light Dependent Resistor) to gauge light intensity. An Arduino microcontroller is employed to interface with these sensors, process the data, and control the LED. The gathered environmental data is then displayed in real-time on an SSD1306 OLED screen, providing a clear and immediate view of the current conditions.

The project aims to create an automated solution for monitoring environmental parameters, which can be useful in a variety of settings such as greenhouses, laboratories, and residential spaces where maintaining optimal environmental conditions is crucial. The system ensures that temperature, humidity, and light levels are constantly monitored and displayed, and it includes a feature where the LED is activated when the light intensity falls below a specific threshold, thus providing a visual alert for low light conditions. This project not only demonstrates the practical application of sensor integration and data visualization but also highlights the use of microcontroller programming to automate environmental control systems.

**Components Required:**

1.Arduino Board

2.DHT11 Temperature and Humidity Sensor

3.Light Dependent Resistor (LDR)

4. **10kΩ Resistor**

**5.SSD1306 OLED Display**

**6.LED**

**7.Connecting Wires**

**8.Bread Board**

**9.Power Source**

**Circuit Diagram :**

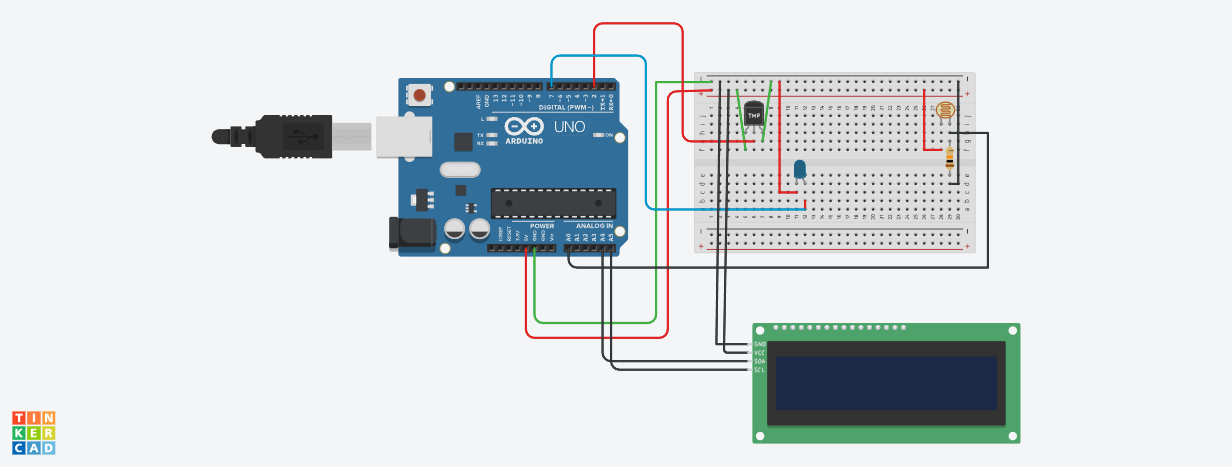


Figure :Circuit Diagram for environmental monitoring system

**Procedure :**

1. DHT11 Temperature and Humidity Sensor:

* VCC- Arduino 5V pin
* GND- Arduino GND pin
* Data Out - Arduino digital pin 2

2. LED:

* Anode (longer leg) Resistor - Arduino digital pin (e.g., pin 7)
* Cathode (shorter leg) → Arduino GND pin

3. LDR (Light Dependent Resistor):

* One leg Arduino analog pin (e.g., A0)
* Other leg - Resistor - Arduino 5V pin
* Connect a resistor between the LDR and the 5V pin to create a voltage divider. The resistor value depends on your specific LDR and the desired sensitivity.

4. OLED Display:

* VCC- Arduino 5V pin
* GND - Arduino GND pin
* SDA (Serial Data) - Arduino analog pin A4
* SCL (Serial Clock) - Arduino analog pin A5

**Code :**

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

#include <DHT.h>

#define OLED\_RESET -1

Adafruit\_SSD1306 display(OLED\_RESET);

#define DHTPIN 2

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

#define LDR\_PIN A0

#define LED\_PIN 7

void setup() {

Serial.begin(9600);

dht.begin();

pinMode(LDR\_PIN, INPUT);

pinMode(LED\_PIN, OUTPUT);

if(!display.begin(SSD1306\_SWITCHCAPVCC, 0x3C)) {

Serial.println(F("SSD1306 allocation failed"));

for(;;);

}

display.clearDisplay();

display.setTextSize(1);

display.setTextColor(SSD1306\_WHITE);

display.setCursor(0,0);

display.display();

}

void loop() {

float humidity = dht.readHumidity();

float temperature = dht.readTemperature();

int lightLevel = analogRead(LDR\_PIN);

if (isnan(humidity) || isnan(temperature)) {

Serial.println("Failed to read from DHT sensor!");

return;

}

// Control LED based on light level

if (lightLevel < 500) { // Adjust threshold as needed

digitalWrite(LED\_PIN, HIGH); // Turn on LED if light level is low

} else {

digitalWrite(LED\_PIN, LOW); // Turn off LED if light level is high

}

display.clearDisplay();

display.setCursor(0,0);

display.print("Temp: ");

display.print(temperature);

display.println(" C");

display.print("Humidity: ");

display.print(humidity);

display.println(" %");

display.print("Light: ");

display.print(lightLevel);

display.display();

delay(2000); // Delay between readings

}

**Output:**

Temperature Humidity and Light readings are displayed.

