# ESP32 with LDR, DHT and OLED

## 1. Introduction

This project uses an ESP32 microcontroller to read environmental data from two sensors: an LDR (Light Dependent Resistor) for measuring light intensity, and a DHT11 sensor for measuring temperature and humidity. The collected data is displayed on an OLED screen and also printed on the Serial Monitor for debugging or observation.

## 2. Libraries Used

- Arduino.h: Basic library for Arduino functions.  
- Wire.h: Used for I2C communication between ESP32 and OLED.  
- Adafruit\_GFX.h: Core graphics library for display handling.  
- Adafruit\_SSD1306.h: Library to control the OLED display.  
- DHT.h: Library to interface with the DHT11/DHT22 sensor.

## 3. Pin Configuration

The following GPIO pins of the ESP32 are used:  
- LDR\_PIN (GPIO36): Analog input for the LDR sensor.  
- DHTPIN (GPIO14): Data pin for DHT11 sensor.  
- SDA\_PIN (GPIO21): I2C data line for OLED.  
- SCL\_PIN (GPIO22): I2C clock line for OLED.

## 4. OLED Display Setup

The OLED display has a resolution of 128x64 pixels. The Adafruit\_SSD1306 object is created with these dimensions and connected via the I2C bus using pins 21 (SDA) and 22 (SCL). The display is initialized at address 0x3C, which is the standard I2C address for most 0.96-inch OLED screens.

## 5. DHT Sensor Setup

The DHT11 sensor is initialized using the DHT library. It measures both temperature and humidity. If using DHT22, the DHTTYPE constant should be changed to DHT22.

## 6. setup() Function

The setup function runs once when the ESP32 starts. It performs these tasks:  
1. Starts the Serial Monitor at 115200 baud rate.  
2. Initializes I2C communication using SDA and SCL pins.  
3. Initializes the OLED display and shows 'Initializing...'.  
4. Starts the DHT sensor.

## 7. loop() Function

The loop function runs repeatedly. It performs these steps:  
1. Reads analog voltage from the LDR using analogRead().  
2. Converts the ADC value to actual voltage using (adcValue / 4095.0) \* 3.3.  
3. Reads temperature and humidity from the DHT sensor.  
4. If sensor readings are invalid, it displays an error message on OLED.  
5. Otherwise, it prints all values on the Serial Monitor and displays them on the OLED screen.  
6. Updates every 2 seconds using delay(2000).

## 8. Code

#include <Arduino.h>

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

#include <DHT.h>

// --- Pin Configuration ---

#define LDR\_PIN 36       // Analog input for LDR

#define DHTPIN 14         // DHT data pin

#define DHTTYPE DHT11     // Change to DHT22 if using that model

#define SDA\_PIN 21        // OLED SDA

#define SCL\_PIN 22        // OLED SCL

// --- OLED Display Setup ---

#define SCREEN\_WIDTH 128

#define SCREEN\_HEIGHT 64

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, -1);

// --- DHT Sensor Setup ---

DHT dht(DHTPIN, DHTTYPE);

// --- Setup Function ---

void setup() {

  Serial.begin(115200);

  Serial.println("ESP32 with LDR + DHT + OLED");

  // Initialize I2C

  Wire.begin(SDA\_PIN, SCL\_PIN);

  // Initialize OLED

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

    Serial.println("SSD1306 allocation failed");

    for (;;);

  }

  display.clearDisplay();

  display.setTextColor(SSD1306\_WHITE);

  display.setTextSize(2);

  display.setCursor(0, 0);

  display.println("Initializing...");

  display.display();

  // Initialize DHT

  dht.begin();

  delay(1000);

}

// --- Main Loop ---

void loop() {

  // --- LDR Reading ---

  int adcValue = analogRead(LDR\_PIN);

  float voltage = (adcValue / 4095.0) \* 3.3;

  // --- DHT Readings ---

  float temperature = dht.readTemperature();

  float humidity = dht.readHumidity();

  // --- Check Sensor Errors ---

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

    Serial.println("Error reading DHT sensor!");

    display.clearDisplay();

    display.setTextSize(1);

    display.setCursor(0, 0);

    display.println("DHT Error!");

    display.display();

    delay(2000);

    return;

  }

  // --- Print to Serial Monitor ---

  Serial.printf("LDR ADC: %d | Voltage: %.2f V | Temp: %.1f C | Humidity: %.1f %%\n",

                adcValue, voltage, temperature, humidity);

  // --- Display on OLED ---

  display.clearDisplay();

  display.setTextSize(1);

  display.setCursor(0, 0);

  display.println("Sensor Readings");

  display.setCursor(0, 14);

  display.print("LDR ADC: ");

  display.println(adcValue);

  display.print("Volt: ");

  display.print(voltage, 2);

  display.println("V");

  display.setCursor(0, 34);

  display.print("Temp: ");

  display.print(temperature, 1);

  display.println(" C");

  display.setCursor(0, 48);

  display.print("Hum:  ");

  display.print(humidity, 1);

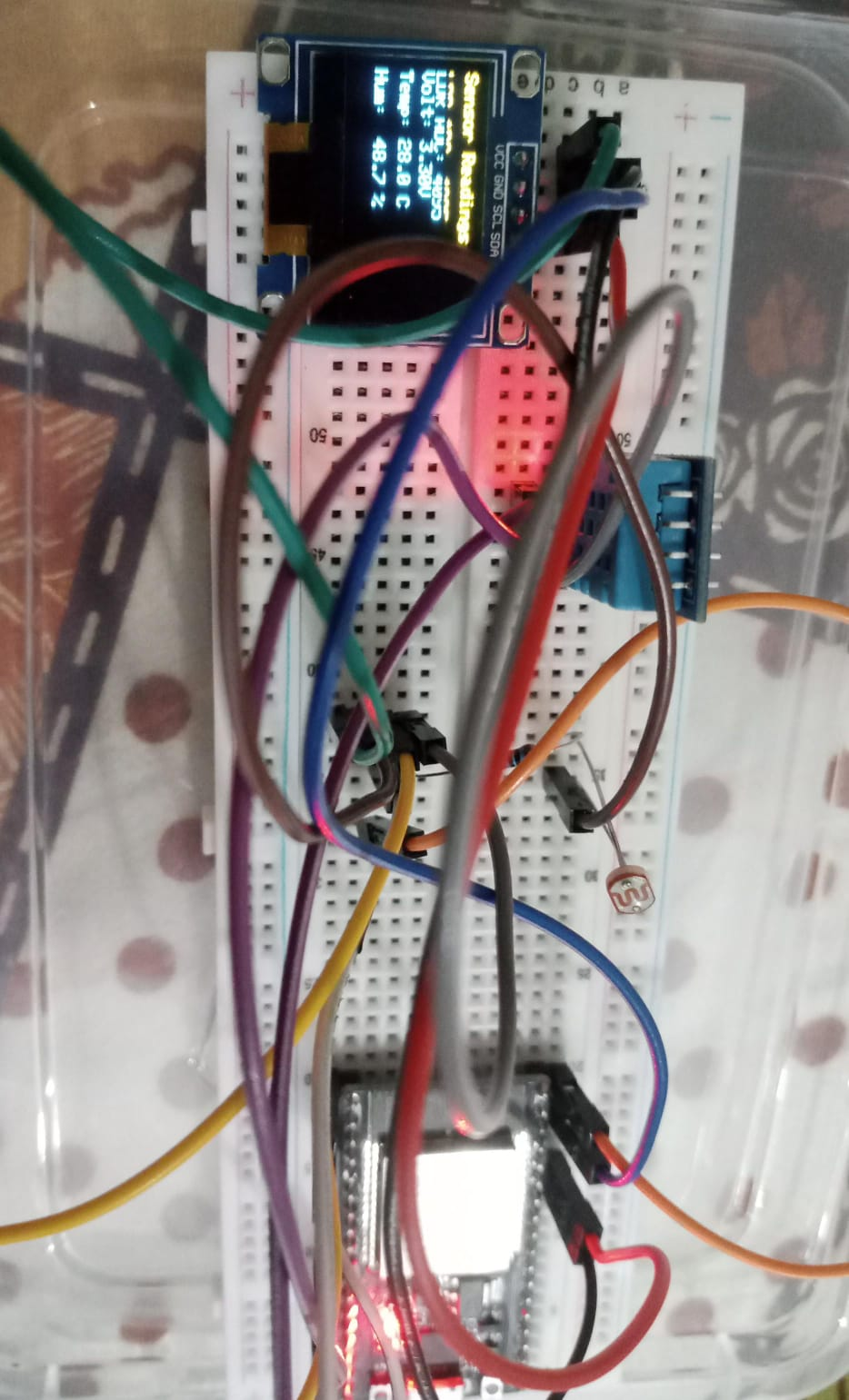
  display.println(" %");

  display.display();

  delay(2000); // Update every 2 seconds

}

## 9.Display



## 10. ADC Explanation

ADC (Analog to Digital Converter) converts the analog voltage from the LDR into a digital value ranging from 0 to 4095. The voltage can be calculated using (adcValue / 4095.0) \* 3.3. This helps the ESP32 understand light intensity levels as numerical data.

## 11. Serial Monitor vs OLED Display

The Serial Monitor is used for debugging or monitoring sensor data on a computer through USB. The OLED display shows the same readings directly on the device, allowing standalone operation without needing a computer connection.

## 12. Conclusion

This ESP32 project demonstrates how to interface multiple sensors and an OLED display. It reads light, temperature, and humidity, processes the analog data through ADC, and presents results both on-screen and via serial output. It can be extended to data logging or IoT-based monitoring applications.