Pakistan Institute of Engineering and Applied Sciences

Lab Report

Lab-02: Reading Temperature and Humidity from DHT11 / DHT22 Sensor

Group Members:

- M. Hammad Tahir
- Ehmaan Shafqat
- Khadija Arif
- Ali Raza

Course Name: Practical IoT

Instructor: Dr. Naveed Akhtar

Submission Date: 28/02/2025

Objectives:

- Learn how to interface the DHT11 / DHT22 sensor with WeMos D1 Mini (ESP8266).
- Read temperature and humidity data from the sensor.
- Display the readings on the Serial Monitor.

Required Components:

- WeMos D1 Mini (ESP8266)
- DHT11 / DHT22 Sensor
- 10kΩ Resistor (optional for stability)
- Breadboard & Jumper Wires
- LED (optional, for threshold alerts)
- USB cable for programming

Reading Temperature and Humidity from DHT11 / DHT22 Sensor

Circuit Diagram:

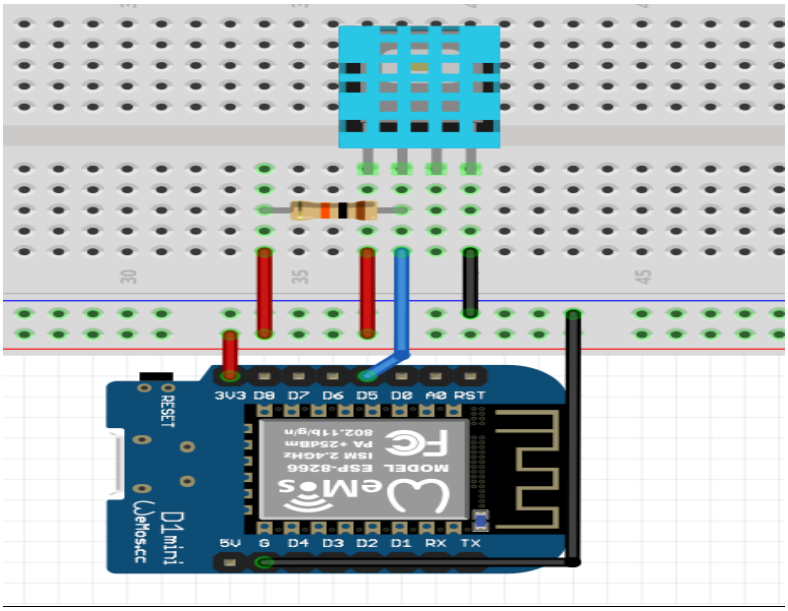
For DHT11 sensor:

• VCC: Connect to 3.3V (or 5V, based on your sensor specification) on the WeMos D1 Mini.

• GND: Connect to Ground.

• Data: Connect to a digital pin (e.g., D4) on the WeMos D1 Mini.

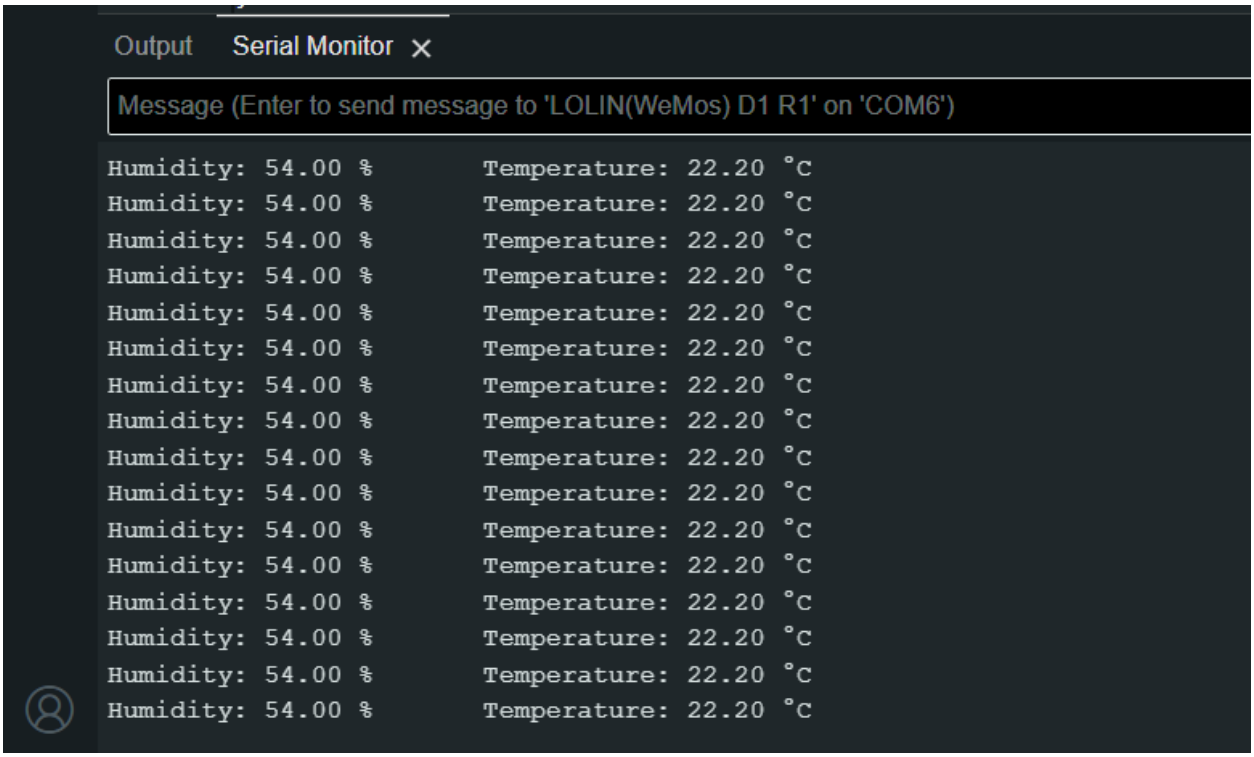
• Optionally, use a 10kΩ resistor as a pull-up between VCC and the Data line if not integrated on the sensor module.

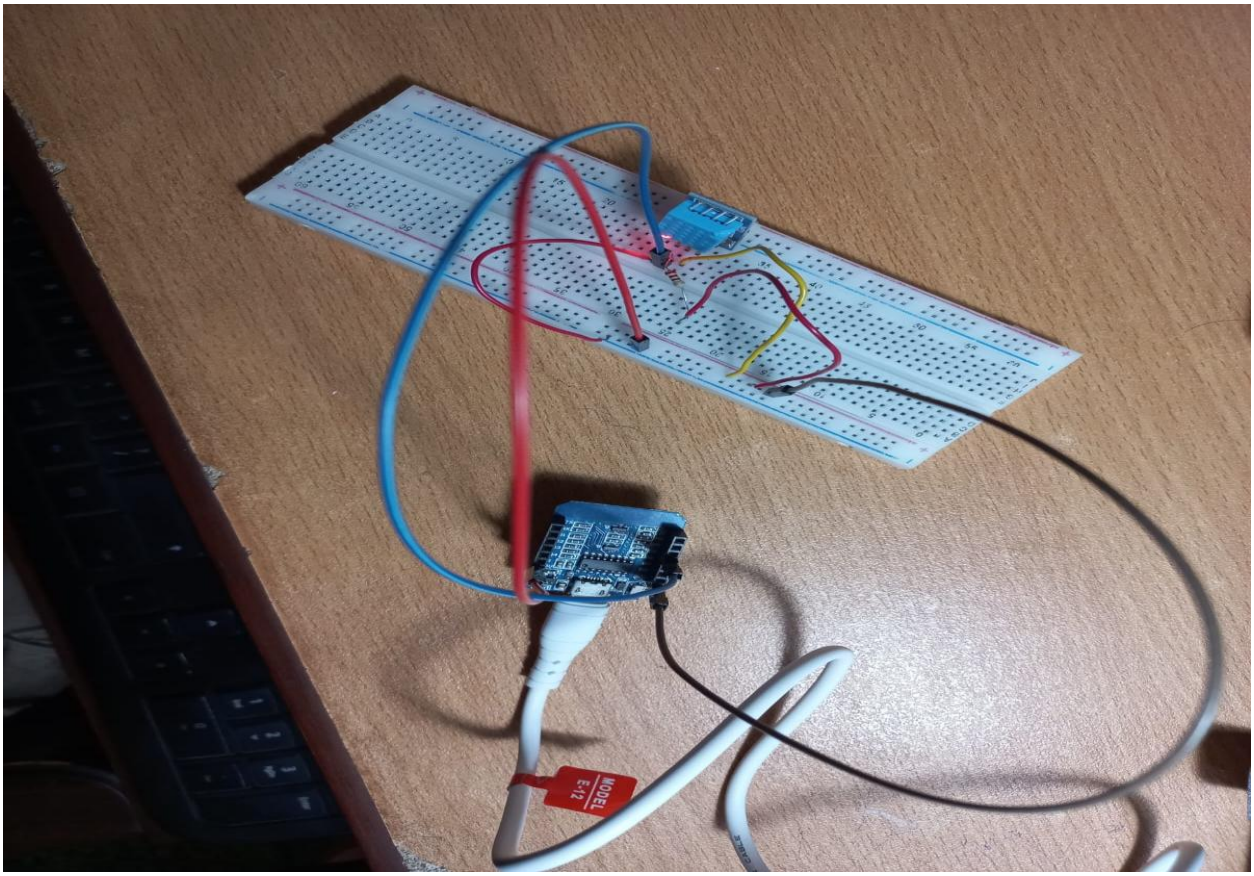


Code:

```ino
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>

#define DHTPIN D5           //  Data pin connected to the sensor (GPIO2)
#define DHTTYPE DHT11       //  As we are using DHT22 sensor

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(115200);
  Serial.println("DHT Sensor Reading");
  dht.begin();
}

void loop() {
  // wait for 2 seconds between measurments.
  delay(2000);

  // Reading Temperature and Humidity  values.
  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();  //  Temperature is in Celsius

  //  Check if any read failed and exit early (to try again).
  if (isnan(humidity) || isnan(temperature)){
    Serial.println("Failed to read form DHT sensor!");
    return;
  }

  //  Print the readings to the serial Monitor.
  Serial.print("Humidity: ");
  Serial.print(humidity);
  Serial.print(" %\t");
  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.print(" °C");
  Serial.print("\n");
}
```

Output:


Output
Serial Monitor
Message (Enter to send message to 'LOLIN(WeMos) D1 R1' on 'COM6')
Humidity: 54.00 % Temperature: 22.20 °C
Humidity: 54.00 % Temperature: 22.20 °C
Humidity: 54.00 % Temperature: 22.20 °C
Humidity: 54.00 % Temperature: 22.20 °C
Humidity: 54.00 % Temperature: 22.20 °C
Humidity: 54.00 % Temperature: 22.20 °C
Humidity: 54.00 % Temperature: 22.20 °C
Humidity: 54.00 % Temperature: 22.20 °C
Humidity: 54.00 % Temperature: 22.20 °C
Humidity: 54.00 % Temperature: 22.20 °C
Humidity: 54.00 % Temperature: 22.20 °C
Humidity: 54.00 % Temperature: 22.20 °C
Humidity: 54.00 % Temperature: 22.20 °C
Humidity: 54.00 % Temperature: 22.20 °C
Humidity: 54.00 % Temperature: 22.20 °C
Humidity: 54.00 % Temperature: 22.20 °C

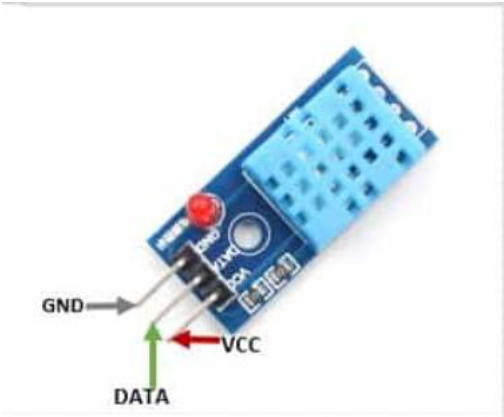


Documentation and report

Design decisions:

	DHT11 Sensor	DHT22 Sensor
Temperature Range:	0–50°C	-40–80°C
Humidity Range:	20–90% RH	0–100% RH
Accuracy:	±2°C for temperature, ±5% RH for humidity	±0.5°C for temperature, ±2–5% RH for humidity
Suitable for:	Lower resolution and accuracy; suitable for basic applications.	Higher resolution and wider range; preferred for more precise applications.
Data Communication:	1-Wire Protocol (single data pin for both temperature and humidity)	1-Wire Protocol (single data pin for both temperature and humidity)

For this lab, we choose VMOS d1 mini, DHT11 and carefully interface its pins according to the following picture.



- For **Task 1**, we built a circuit by connecting the VCC terminal of the sensor to the 3.3V terminal of the microcontroller and the ground (GND) terminal of the sensor to the ground terminal of the microcontroller. The data pin (middle pin) of the sensor is connected to D5. Additionally, a pull-up resistor is used between D5 and the data pin of the sensor because the DHT sensor operates using a **single-wire communication protocol.** This requires a stable **HIGH** state when the sensor is not actively transmitting data. The pull-up resistor ensures the data line remains in a **default HIGH state**, preventing noise and false signals.

Code Explanation:

- **Library Initialization:** The code begins by including the necessary libraries and creating a DHT object.
- **Sensor Selection:** #define DHTTYPE DHT11 configures the code for the DHT11 sensor.
- **Serial Monitor**: Serial.begin(115200); sets the baud rate in the serial monitor and defines the speed of data transmission between the microcontroller and the computer. Serial.print() uses to print the values that read from the sensor.
- **Data Reading:** The loop() function reads temperature and humidity every 2 seconds, ensuring reliable sensor operation.
- **Error Handling:** The code checks for failed readings (resulting in NaN values) and prints an error message.

Challenges:

- In the lab manual, D4 was set as the **DHTPIN**, but when I built the circuit, D4 did not work properly, and the temperature or humidity readings failed to be read. So, I modified the code and set D5 as the **DHTPIN**, where the controller was able to read the temperature and humidity correctly.
- In the lab, we received the **DHT11** sensor, so we modified **DHTTYPE** to **DHT11** accordingly.

Learning outcomes:

Read Temperature & Humidity on Serial Monitor:

- **Read** temperature and humidity using the **DHT11** sensor and display the values on the screen through the **serial monitor**.
- **Understand** the interfacing of **DHT11** with **VMOS**, which is fundamental to **IoT** applications.