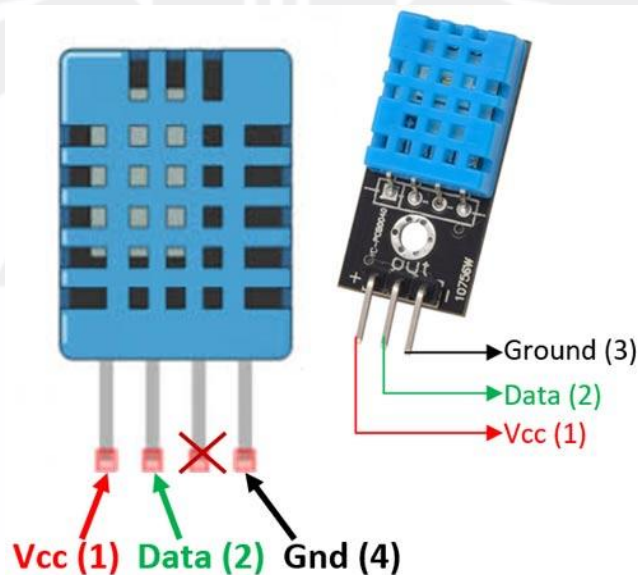


LAUNCH THE PROJECT - 09

Temperature & Humidity Measurement

(via DHT11)

On AYNOP® UNO Launchpad Kit



Author: AYNOP Enterprises | Doc Version: 1.0 | Date: 03-9-2025

Table of Contents

1. Overview.....	3
2. Components Required.....	3
3. Software Required.....	3
4. Hardware Setup	4
4.1 Wiring Diagram.....	4
5. Principle – How It Works	5
6. Procedure – Steps to Run	6
6.1 How to Use the Serial Monitor:.....	7
7. Expected Output.....	7
8. Code.....	8
8.1 Function References.....	8
9. Troubleshooting Tips.....	9
10. License.....	10
11. Support & Feedback.....	10

1. Overview

This project demonstrates how to measure **temperature and humidity** using the **DHT11 sensor** with the Arduino® UNO R4 Minima. The readings are displayed on the **Serial Monitor**.

You will learn to:

- Connect the DHT11 sensor to the Arduino.
- Use the DHT library to read temperature and humidity.
- Display the readings in the Arduino IDE's Serial Monitor.

2. Components Required

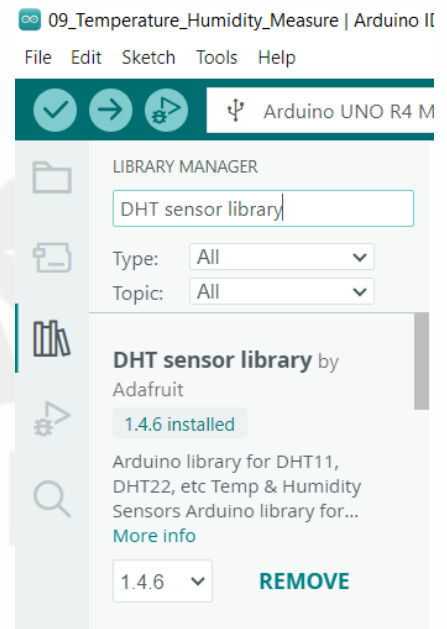
- Arduino® UNO R4 Minima board
- USB Type-C data cable
- DHT11 sensor module (3-pin version recommended)
- Breadboard
- Jumper wires

3. Software Required

- Arduino IDE (v2.3.6 or later recommended)
- Required library: **DHT sensor library (by Adafruit)**.

Installing the Library:

- Open Arduino IDE.
- Go to **Sketch → Include Library → Manage Libraries**
- Search for **DHT sensor library (by Adafruit)**.
- Install the library.



Note:

We assume the **Arduino UNO R4 Minima board package** is already installed on your machine, as explained in the 00_Getting_Started/00_GettingStarted_Arduino_R4_Minima guide. If it is not installed, please refer to that document and complete the installation before proceeding.

4. Hardware Setup

This section explains how to connect the components for the **temperature and humidity measurement** project.

4.1 Wiring Diagram

- DHT11 VCC → 5V.
- DHT11 GND → GND.
- DHT11 DATA → D2.

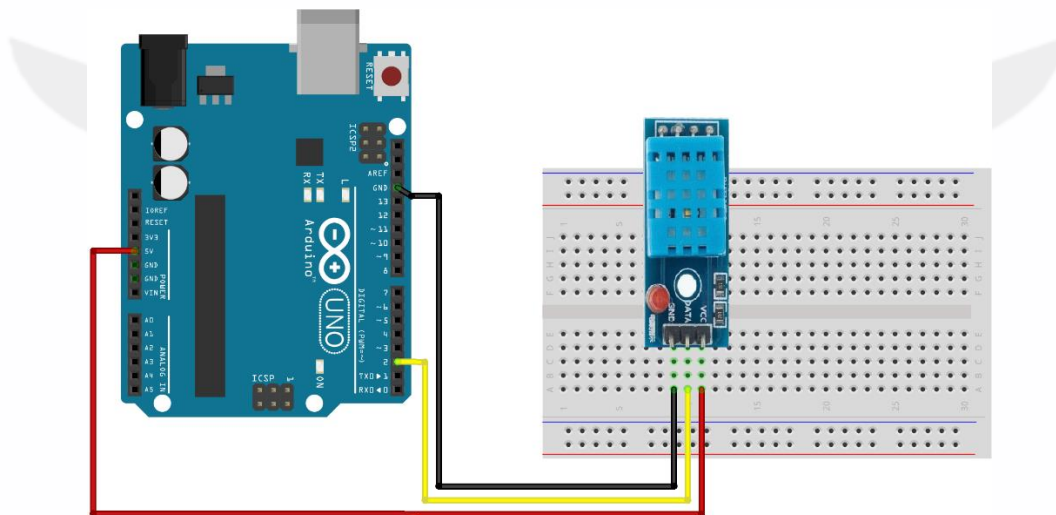


Figure 4.1 – Wiring diagram for DHT11 sensor with Arduino UNO R4 Minima

💡 Tip:

- Always disconnect the USB cable before making or changing hardware connections.
- If you are using a bare DHT11 sensor (not a module), connect a **10 kΩ resistor** between DATA and VCC.

5. Principle – How It Works

The **DHT11 sensor** contains:

- A **humidity sensing capacitor** whose resistance changes with humidity.
- A **thermistor** for temperature measurement.
- An internal **chip** that converts the readings into digital signals.

The Arduino communicates with the DHT11 via a **single-wire data bus**. The **DHT library** simplifies this communication by handling the sensor protocol and providing easy functions:

- *readTemperature()* → returns temperature in Celsius.
- *readHumidity()* → returns relative humidity in %.

The measured values are displayed on the **Serial Monitor**, which acts as a debugging/output tool for the user.



6. Procedure – Steps to Run

1. Build the Circuit

- a. Connect the DHT11 sensor as shown in the **Wiring Diagram (Figure 4.1)**.

2. Connect the Board

- a. Use a USB Type-C data cable to connect your UNO R4 Minima to your computer.

3. Open the Project Code

- a. Simply **double-click** the file *09_Temperature_Humidity_Measure.ino* in the project folder, and it will open directly in the Arduino IDE (if installed).

4. Confirm Board Selection

- a. The IDE usually auto-detects the UNO R4 Minima if the package is installed.
- b. If not installed, refer to the *00_Getting_Started/00_GettingStarted_Arduino_R4_Minima* document to install the necessary board package.
- c. Verify that *Arduino UNO R4 Minima* is selected in the IDE's board selector (top toolbar).

5. Upload the Code

- a. Click the **Upload** button (arrow icon) in the top-left corner of the IDE.
- b. Wait until the console displays **"Done uploading."**

6. Observe the Behaviour

- a. The serial monitor will show the real time **Temperature (°C)** and **Humidity (%)** value.
- b. To test:
 - i. **Touch the sensor with your finger** → temperature will slowly increase.
 - ii. **Breathe gently on the sensor** → humidity will rise quickly.

6.1 How to Use the Serial Monitor:

The Serial Monitor is a **tool in the Arduino IDE** that lets you see text/data sent by the microcontroller (MCU) over the USB connection.

- In Arduino IDE, go to **Tools → Serial Monitor**, or press **Ctrl + Shift + M**.
- A new window will open showing real-time data from your UNO R4 Minima.
- At the bottom-right of the Serial Monitor window, set the baud rate to **9600**. This must match the *Serial.begin(9600)* in the code.

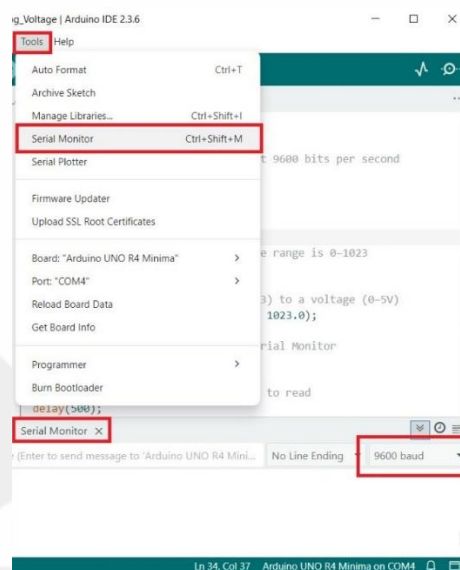


Figure 6.1 – Arduino IDE Serial Monitor set to 9600 baud

Tip: If the Serial Monitor shows nothing, check that:

- The correct **COM port** is selected.
- The baud rate is set to **9600**

7. Expected Output

- The Serial Monitor will show **Temperature (°C)** and **Humidity (%)** values like this:

```


AYNOP: Temperature & Humidity Measurement
Temperature: 28.10 *C
Humidity: 55.50 %
-----
Temperature: 28.20 *C
Humidity: 60.20 %
-----
    
```

8. Code

The source code for this project is included in the downloaded folder:

 `uno-launchpad-kit/01_Basic_Projects/09_Temperature_Humidity_Measure/09_Temperature_Humidity_Measure.ino`



Tip:

- To open the project, simply **double-click the .ino file**. If the Arduino IDE is installed, it will launch automatically and load the code.
- If you **haven't installed the Arduino IDE yet**, please refer to:
 `uno-launchpad-kit/00_Getting_Started/00_GettingStarted_Arduino_R4_Minima` to **download and install it**.


8.1 Function References


- `setup()` – runs once when the board is powered on or reset.
- `loop()` – runs continuously after `setup()` finishes.
- `dht.begin()` – initializes the sensor.
- `dht.readTemperature()` – returns temperature in Celsius.
- `dht.readHumidity()` – returns humidity in %.
- `Serial.begin(baudrate)` – starts serial communication.
- `Serial.print()` / `Serial.println()` – prints data to Serial Monitor.
- `delay(ms)` – pauses program execution.

 For more details and advanced usage, visit:

-  [DHT sensor library](#) — The GitHub repo of DHT sensor library.
-  [Arduino Language Reference](#) — The official guide for all Arduino functions.

9. Troubleshooting Tips

- **No output on Serial Monitor?**
 - Check the baud rate (must be 9600).
 - Ensure Serial Monitor is open after upload.
- **Reading shows NaN (Not a Number)?**
 - Sensor not connected properly.
 - Wrong pin defined in code.
 - DHT11 needs ~2 seconds after startup to give stable readings.
- **Upload error in Arduino IDE?**
 - Verify that the correct board (**Arduino UNO R4 Minima**) is selected in the IDE.
 - Check that the correct **COM port** is chosen.
- **Board not detected via USB?**
 - Ensure you are using a **data-capable USB Type-C cable** (some cables only provide charging).
 - Try reconnecting the cable or using a different USB port.
- **Board not listed in Arduino IDE?**
 - If you don't see **Arduino UNO R4 Minima** in the board selector, the **board package is not installed**.
 - To fix this, follow the installation steps in:
 *uno-launchpad-kit/00_Getting_Started/00_GettingStarted_Arduino_R4_Minima*

 **Tip:** If nothing works, press the **RESET** button on the UNO R4 Minima and try uploading the code again.

10. License

This content (source code and documentation) is licensed under **Creative Commons BY-NC-SA 4.0**.
© 2025 AYNOP. You may use, modify, and share for personal and educational purposes only.
Commercial use or redistribution without prior written permission is strictly prohibited.
Refer to the **LICENSE** file for complete details.

11. Support & Feedback

We value your feedback and are happy to assist with any questions, troubleshooting, or suggestions you may have.

✉ Email: support@aynop.com

When sending an email, please include your kit name (AYNOP® UNO Launchpad Kit) and, if applicable, the project name in the subject line. This will help our team respond faster and more accurately. We aim to respond to all queries within 2–3 business days. Your feedback helps us improve our products and create even better learning experiences for all Arduino beginners.

