To connect a **client-server system** using **Raspberry Pi** for reading **temperature from a DHT22 sensor**, here's a step-by-step guide:

**🧠 Concept Overview**

* **DHT22 Sensor**: Reads temperature and humidity.
* **Server (on Pi)**: Reads data from DHT22 and sends it to the client.
* **Client (any machine)**: Requests data from the server and displays it.

**🔌 Wiring the DHT22 to Raspberry Pi**

| **DHT22 Pin** | **Connect To** |
| --- | --- |
| VCC | 3.3V on Pi (Pin 1) |
| DATA | GPIO 4 (Pin 7) |
| GND | GND (Pin 6) |

Also, put a **10kΩ resistor** between **VCC** and **DATA**.

**🧰 1. Install Required Libraries**

Run this on your Raspberry Pi:

bash

CopyEdit

pip install Adafruit\_DHT flask

**💻 2. Server Code (on Raspberry Pi)**

Save as server.py:

python

CopyEdit

from flask import Flask, jsonify

import Adafruit\_DHT

app = Flask(\_\_name\_\_)

SENSOR = Adafruit\_DHT.DHT22

PIN = 4 # GPIO4

@app.route("/temperature")

def get\_temperature():

humidity, temperature = Adafruit\_DHT.read\_retry(SENSOR, PIN)

if humidity is not None and temperature is not None:

return jsonify({"temperature": temperature, "humidity": humidity})

else:

return jsonify({"error": "Failed to retrieve data"}), 500

if \_\_name\_\_ == "\_\_main\_\_":

app.run(host="0.0.0.0", port=5000)

Run the server:

bash

CopyEdit

python server.py

**🌐 3. Client Code (on another device or the same Pi)**

Save as client.py:

python

CopyEdit

import requests

SERVER\_IP = "http://<raspberry\_pi\_ip>:5000"

try:

response = requests.get(f"{SERVER\_IP}/temperature")

data = response.json()

if "temperature" in data:

print(f"Temperature: {data['temperature']:.2f}°C")

print(f"Humidity: {data['humidity']:.2f}%")

else:

print("Error:", data.get("error", "Unknown error"))

except Exception as e:

print("Failed to connect to server:", e)

Replace <raspberry\_pi\_ip> with your Pi’s IP address (you can find it with hostname -I on the Pi).

**✅ Optional Enhancements**

* Add an LCD display for local display.
* Store data in a database (SQLite or Firebase).
* Create a dashboard using React or another frontend tool.

Want me to help you build a full web dashboard or add hardware components like an LCD next?

4o

**You said:**

Add an LCD display for local display.

**ChatGPT said:**

Great! Let's add an **LCD display** to your Raspberry Pi setup so it shows the temperature and humidity locally in real time.

We'll assume you're using a **16x2 I2C LCD display**, which is the most common and easy to wire.

**🧰 Parts Needed**

* Raspberry Pi (any model with GPIO)
* DHT22 sensor
* 16x2 I2C LCD display
* Jumper wires
* Optional: Breadboard

**🪛 Wiring the I2C LCD**

| **I2C LCD Pin** | **Connect To (Pi)** |
| --- | --- |
| VCC | 5V (Pin 2 or 4) |
| GND | GND (Pin 6 or 9) |
| SDA | GPIO 2 (Pin 3) |
| SCL | GPIO 3 (Pin 5) |

**🧪 Enable I2C on Raspberry Pi**

Run:

bash

CopyEdit

sudo raspi-config

* Go to **Interfacing Options** > **I2C** > Enable
* Reboot if prompted.

Then install I2C tools:

bash

CopyEdit

sudo apt-get install -y i2c-tools

i2cdetect -y 1

You should see the I2C address (e.g., 0x27 or 0x3f).

**🧩 Install LCD Library**

bash

CopyEdit

pip install RPLCD

For I2C support:

bash

CopyEdit

pip install smbus2

**🧑‍💻 Full Code with LCD + DHT22 (local display)**

Save as lcd\_display.py:

python

CopyEdit

import time

import Adafruit\_DHT

from RPLCD.i2c import CharLCD

# Sensor setup

SENSOR = Adafruit\_DHT.DHT22

DHT\_PIN = 4

# LCD setup (adjust 'address' if needed)

lcd = CharLCD('PCF8574', 0x27) # or try 0x3f

def read\_and\_display():

while True:

humidity, temperature = Adafruit\_DHT.read\_retry(SENSOR, DHT\_PIN)

if humidity is not None and temperature is not None:

lcd.clear()

lcd.write\_string(f"Temp: {temperature:.1f}C")

lcd.cursor\_pos = (1, 0)

lcd.write\_string(f"Hum: {humidity:.1f}%")

else:

lcd.clear()

lcd.write\_string("Sensor error")

time.sleep(2)

try:

read\_and\_display()

except KeyboardInterrupt:

lcd.clear()

lcd.write\_string("Goodbye!")

time.sleep(1)

lcd.clear()

**✅ Result**

Your Pi will now show:

makefile

CopyEdit

Temp: 25.1C

Hum: 52.3%

And refresh every 2 seconds.