

— TEMPERATUR & LUFTFEUCHTE

AM2302 DHT22

Sensormodul mit Breakout-Board. Vom Auspacken zum ersten Messwert in rund 15 Minuten – mit UNO R3, Steckbrett und drei Kabeln.



TEMPERATUR

-40 ... +80 °C

LUFTFEUCHTE

0 ... 100 % rF

VERSORGUNG

3,3 ... 5,5 V DC

SCHNITTSTELLE

Single-Bus, 1 GPIO

01

Verkabeln

3 Leitungen zum UNO

02

Einrichten

IDE, Treiber, Bibliothek

03

Hochladen

Beispiel-Sketch

04

Messen

Serieller Monitor

01 Auf einen Blick

Der AM2302 – meist DHT22 genannt – misst Temperatur und relative Luftfeuchte in einem Bauteil. Kalibrierung und Umrechnung erledigt der Sensor selbst; dein Mikrocontroller liest über eine einzige Datenleitung fertige Messwerte.

Technische Daten

Temperaturbereich	–40 bis +80 °C
Genauigkeit Temp.	±0,5 °C
Feuchtebereich	0 bis 100 % rF
Genauigkeit Feuchte	±2 bis 5 % rF
Versorgungsspannung	3,3 bis 5,5 V DC
Schnittstelle	Single-Bus (1 GPIO)
Messintervall	min. 2 Sekunden
Pull-up-Widerstand	auf dem Breakout-Board

Lieferumfang

- 1 × **AM2302 (DHT22) auf Breakout-Board mit 3-poliger Winkel-Stiftleiste**
- 1 × **Anschlusskabel 3-adrig, Buchse–Buchse**



Nicht wasserdicht

Der Sensor ist für Innenräume und geschützte Bereiche gedacht. Vor Spritzwasser und Betauung schützen.



Gut zu wissen

Etwa alle zwei Sekunden ein neuer Wert – ideal für Raum- und Umweltklima.

Pinbelegung

PIN (PLATINE)	FUNKTION	SPANNUNG	HINWEIS
● + (VCC)	Versorgung POWER	3,3–5,5 V DC	Am UNO an 5V , am ESP32 an 3V3 .
● OUT / DATA	Datenleitung DIGITAL	wie VCC	Bidirektionaler Single-Bus. Der nötige Pull-up ist bereits bestückt – kein externer Widerstand nötig .
● - (GND)	Masse GND	0 V	Gemeinsame Masse mit dem Mikrocontroller.

Maßgeblich ist die Beschriftung neben der Stiftleiste (+ / OUT / -). Die Aderfarben des Kabels haben keine Bedeutung.

Was du zusätzlich brauchst

UNO R3 mit USB-C

ATmega328P, CH340G-USB-Wandler
[MKR-UNO-R3-013](#)

Breadboard + Jumper-Set

400 Kontakte, 40 Dupont-Kabel
[MKR-SET-BB400-JUMPER](#)

USB-C-Datenkabel

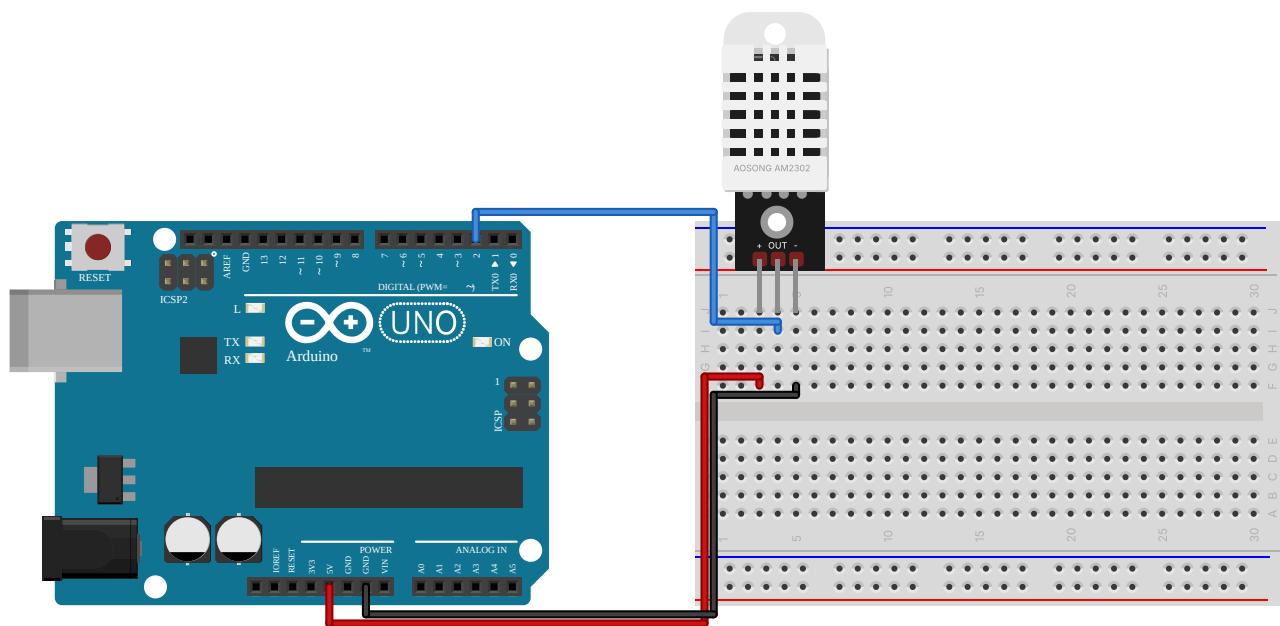
Kein reines Ladekabel – es muss Daten übertragen

02

Verkabeln

Drei Leitungen genügen. Zieh vorher das USB-Kabel ab – verkabelt wird immer stromlos.

STECKPLATTEN-ANSICHT



Verbindungen

SENSOR		UNO R3	LEITUNG
● + (VCC)	→	5V	Versorgung
● OUT	→	D2	Messdaten
● - (GND)	→	GND	Masse

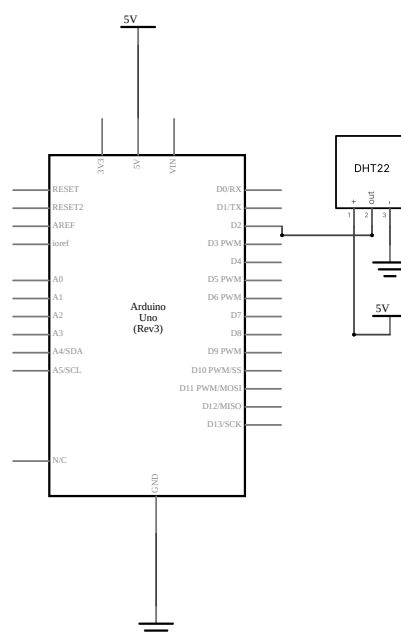
i Zwei Wege

Wie im Bild: Modul ins Steckbrett, dann Stecker-Stecker-Jumper zum UNO. Ohne Steckbrett: beiliegendes Kabel aufs Modul und je einen Stecker-Stecker-Jumper in die Buchsen-Enden.

! Erst prüfen, dann einstecken

Vertauschte + und – können den Sensor beschädigen. Erst alle Leitungen prüfen, dann USB anstecken.

SCHALTPLAN



Verdrahtung erstellt mit Fritzing

03 Einrichten & Hochladen

Schritte 1 bis 3 machst du nur ein einziges Mal – danach ist dein UNO R3 für alle weiteren Projekte startklar.

1 Arduino IDE installieren

Lade die aktuelle Arduino IDE 2 von arduino.cc/en/software herunter und installiere sie.

2 Treiber für den CH340G prüfen

Der UNO R3 nutzt den CH340G als USB-Seriell-Wandler. Aktuelle Systeme erkennen ihn meist automatisch. Taucht in Schritt 3 kein Port auf, installiere den Treiber „CH341SER“ des Chipherstellers WCH (wch-ic.com).

3 Board und Port wählen

UNO per USB-C anschließen. Dann **Werkzeuge** → **Board** → **Arduino AVR Boards** → **Arduino Uno** wählen und unter **Werkzeuge** → **Port** den neu aufgetauchten COM-Port (Windows) bzw. `/dev/ttyUSB0` (Linux) / `/dev/cu.wchusbserial...` (macOS).

4 Bibliothek installieren

Bibliotheksverwalter öffnen (`Strg` + `Umschalt` + `I`), nach **DHT sensor library** von Adafruit suchen und installieren. Die Frage nach Abhängigkeiten (u. a. *Adafruit Unified Sensor*) mit **Alle installieren** beantworten.

5 Sketch hochladen

Neuen Sketch anlegen, den Code von Seite 5 einfügen und mit dem Pfeil-Symbol hochladen (`Strg` + `U`). Unten erscheint „Hochladen abgeschlossen“.

6 Seriellen Monitor öffnen

`Strg` + `Umschalt` + `M` , unten rechts **9600 Baud** einstellen. Alle zwei Sekunden erscheint eine neue Messzeile.

BEISPIELAUSGABE · SERIELLER MONITOR · 9600 BAUD

MAKER00 DHT22-Test startet ...

Temperatur: 21.4 °C | Feuchte: 48.2 % | Taupunkt: 10.0 °C

Temperatur: 21.4 °C | Feuchte: 48.0 % | Taupunkt: 9.9 °C

Temperatur: 21.5 °C | Feuchte: 47.9 % | Taupunkt: 9.9 °C

i Schnelltest

Hauch den Sensor kurz an: Die Feuchte muss innerhalb weniger Sekunden deutlich steigen und danach langsam wieder fallen. Dann funktioniert alles.

04 Der erste Sketch

Liest alle zwei Sekunden Temperatur und Luftfeuchte und berechnet daraus den Taupunkt – die Größe, auf die es bei Schimmel und Kondenswasser ankommt.

MAKER00_DHT22_Start.ino

```
1 // MAKER00 Getting Started - AM2302 (DHT22) am UNO R3
2 // Misst Temperatur und Luftfeuchte alle 2 Sekunden
3 // und berechnet zusätzlich den Taupunkt.
4
5 #include <DHT.h>
6
7 #define DHTPIN 2 // DATA-Pin des Sensors an D2
8 #define DHTTYPE DHT22 // AM2302 = DHT22
9
10 DHT dht(DHTPIN, DHTTYPE);
11
12 void setup() {
13   Serial.begin(9600);
14   Serial.println(F("MAKER00 DHT22-Test startet ..."));
15   dht.begin();
16 }
17
18 void loop() {
19   delay(2000); // DHT22 liefert max. alle 2 s einen neuen Wert
20
21   float feuchte = dht.readHumidity(); // % rel. Feuchte
22   float temp = dht.readTemperature(); // °C
23
24   if (isnan(feuchte) || isnan(temp)) {
25     Serial.println(F("Fehler: Sensor antwortet nicht!"));
26     return;
27   }
28
29   // Taupunkt nach der Magnus-Formel
30   float g = log(feuchte / 100.0) + (17.62 * temp) / (243.12 + temp);
31   float tp = 243.12 * g / (17.62 - g);
32
33   Serial.print(F("Temperatur: ")); Serial.print(temp, 1);
34   Serial.print(F(" °C | Feuchte: ")); Serial.print(feuchte, 1);
35   Serial.print(F(" % | Taupunkt: ")); Serial.print(tp, 1);
36   Serial.println(F(" °C"));
37 }
```

Zeile 7-8

Pin & Typ

Hier legst du fest, an welchem Pin DATA hängt und welcher Sensor es ist. Andere Pins? Nur diese Zahl ändern.

Zeile 24

Plausibilität

`isnan()` fängt fehlgeschlagene Messungen ab, statt Unsinn auszugeben.

Zeile 30-31

Taupunkt

Magnus-Formel: Unter dieser Temperatur kondensiert Wasser an Oberflächen.

Geprüft: kompiliert für Arduino Uno mit DHT sensor library 1.4.7 (5,9 kB Flash = 18 %, 219 Byte RAM).

05

Wenn es nicht klappt

SYMPTOM

LÖSUNG

„Fehler: Sensor antwortet nicht!“

DATA wirklich an D2? Steht im Code DHTTYPE DHT22 (nicht DHT11)?
+ und – auf festen Sitz im Steckbrett prüfen.

Kein Port in der IDE

Anderes USB-C-Kabel probieren – viele Kabel laden nur. Danach
CH340-Treiber installieren (Schritt 2).

Nur wirre Zeichen im Monitor

Baudrate im seriellen Monitor auf 9600 stellen.

Temperatur deutlich zu hoch

Sensor sitzt zu nah an Wärmequellen (Board, Netzteil, Hand). Mit
etwas Abstand montieren.

Werte springen oder fehlen

Nicht häufiger als alle 2 Sekunden abfragen. Bei langen Leitungen
mit 5 V statt 3,3 V versorgen.

06

Wie geht's weiter?

Mit ESP32 & ESPHome

VCC an **3V3**, DATA z. B. an **GPIO4**; im Sketch
#define DHTPIN 4. Für Home Assistant genügt in
ESPHome:

```
sensor:
  - platform: dht
    pin: GPIO4
    model: AM2302
    temperature:
      name: "Temperatur"
    humidity:
      name: "Luftfeuchte"
    update_interval: 30s
```

Projektideen

Raumklima-Anzeige

Werte auf einem OLED-Display zeigen

Schimmel-Wächter

Warnen, wenn die Wand nahe am Taupunkt ist

Gewächshaus

Lüfter per Relais nach Feuchte schalten

Keller-Logger

Messreihen auf MicroSD-Karte speichern

Passt dazu

0,96" OLED-Display

Messwerte direkt anzeigen

[MKR-0.96-OLED-015](#)

ESP32 DevKit USB-C

WLAN, ESPHome, Home

Assistant

[MKR-ESP32-005](#)

Relais KY-019

Lüfter oder Befeuchter

schalten

[MKR-KY019-034](#)

MicroSD-Modul

Messreihen speichern

[MKR-SDREADER-014](#)

Sicherheitshinweise. Elektronikmodul für Ausbildungs-, Entwicklungs- und Experimentierzwecke. Nur mit Kleinspannung betreiben, niemals an Netzspannung. Nicht geeignet für Kinder unter 14 Jahren; Nutzung nur unter Aufsicht. Elektroaltgeräte gehören nicht in den Hausmüll – Rückgabe kostenlos bei kommunalen Sammelstellen.

Verantwortlicher Wirtschaftsakteur (GPSR): Alexander Prez · makeroo · Hinterm Bahnhof 4 · 90513 Zirndorf · support@makeroo.de
Fragen? Schreib uns mit einem Foto deiner Verkabelung. Arduino ist eine Marke der Arduino SA und dient nur der Kompatibilitätsangabe.