QUICK REFERENCE · TK38

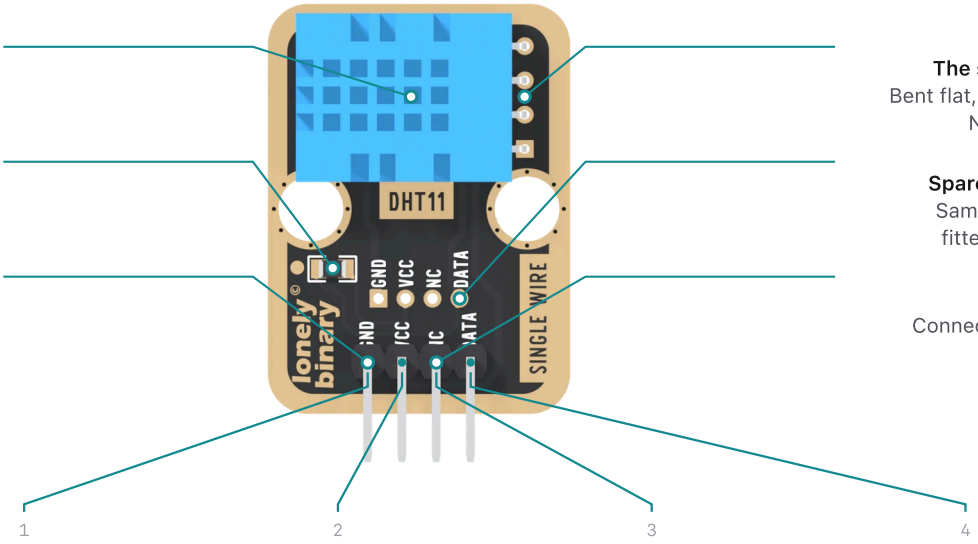
DHT11 Temperature and Humidity

Two numbers down one wire, in whole degrees.

VCC sets the data line The 10 kΩ pull-up ties DATA to VCC. **On an ESP32, ESP32-S3 or Pico, feed VCC from 3V3, never 5V**: 5 V would idle on a 3.3 V input. On an Uno, 5V.

Pins, and which to wire

Sensor up, header at the bottom. Wire GND, VCC and DATA; NC is connected to nothing on the board.



DHT11, blue cage
0 to 50 °C, ±2 °C.
Nothing below freezing.

10 kΩ pull-up
DATA to VCC. Already fitted: add nothing.

GND is the square pad
Count from it: GND, VCC, NC, DATA.

The sensor's leads
Bent flat, soldered here.
Nothing to wire.

Spare 2.0 mm pads
Same nets, nothing fitted. Leave them.

NC
Connected to nothing on the board.

Front	GND	VCC	NC	DATA
Single wire	GND	VCC	●	DATA

nan means nobody answered It is the only error the protocol has: no reply, or a checksum that failed. Check the wires before the code.

What the DHT11 can tell you

Temperature

±2 °C, whole degrees

0 to 50 °C. Below 0 °C it is outside its range and does not warn you.

Humidity

±5 % RH, whole per cent

20 to 80 % on the board's own silkscreen.

How often

2 s between reads

The datasheet allows 1 s; the library caches for 2 s.

Good for learning the wire and for a display you glance at. For anything that logs, alarms or goes outdoors, the TK39's DHT22 is the same wiring and one word in the sketch.

From Aosong's DHT11 datasheet and Adafruit's DHT sensor library 1.4.7. Accuracies are the datasheet's; nothing on this sheet was measured.

The first reading, on a timer

Arduino IDE · Library: DHT sensor library (Adafruit), with Adafruit Unified Sensor · ESP32-S3: USB CDC On Boot: Enabled

```
#include <DHT.h>   // DHT sensor library by Adafruit

// DATA's pin. Uno: D2. ESP32: 18. ESP32-S3: 4. Pico: GP2.
#define DHT_PIN  4
#define DHT_TYPE DHT11   // must match the cage: blue

DHT dht(DHT_PIN, DHT_TYPE);
unsigned long last = 0;

void setup() {
  Serial.begin(115200);
  dht.begin();
}

void loop() {
  if (millis() - last < 2500) return;   // over the library's 2 s ca
  last = millis();

  float h = dht.readHumidity();
  float t = dht.readTemperature();
  if (isnan(h) || isnan(t)) {        // nobody answered
    Serial.println("read failed");
    return;
  }
  Serial.print(t, 1); Serial.print(" C  ");
  Serial.print(h, 1); Serial.println(" %");
}
```

```
23.0 C 51.0 %
23.0 C 52.0 %
```

Serial Monitor at 115200, one line every 2.5 seconds. Usually whole numbers: most DHT11s send zero tenths.

Wiring

	FRONT	UNO	ESP32	S3	PICO
1	GND	GND	GND	GND	GND
2	VCC	5V	3V3	3V3	3V3
3	NC				
4	DATA	D2	GPIO 18	GPIO 4	GP2

Hole 1 is the square pad. VCC is your board's logic voltage, because DATA idles at VCC. The second row of pads above the header is an unfitted 2.0 mm connector on the same nets: leave it.

What happens on the wire

The sketch pulls DATA low for 20 ms to wake the sensor (the DHT11 needs 18 ms, the DHT22 only 1 ms). The sensor answers 80 µs low, 80 µs high, then sends 40 bits: each a 50 µs low and a high whose length is the bit, 26 to 28 µs for 0, 70 µs for 1. About 4 ms in all.

When it does not work

Every reading is nan

Nobody answered. Check GND, then VCC, then DATA on the pin the sketch names, counting from the square pad: GND, VCC, NC, DATA.

The reading never changes

Reads closer than 2 s apart return the last answer without touching the wire. Time reads with millis().

Works on an Uno, not an ESP32

Check VCC is 3V3. The pull-up takes DATA to VCC, so 5 V there sits on a 3.3 V pin.

Numbers, but silly ones

The type in the sketch does not match the cage. Blue is DHT11, white is DHT22; the checksum passes either way.

Reads warm, and dry

It is measuring the board it sits on. Move it off the breadboard on jumper wires, away from regulators, and wait a few minutes.

Two sensors on one pin

There are no addresses: both answer at once and neither survives. One DATA pin per sensor.

Specifications

Sensor	DHT11, factory calibrated	**Temperature**	0 to 50 °C, ±2 °C
Humidity	20 to 80 % RH, ±5 %	**Steps**	1 °C, 1 %
Supply	3 to 5.5 V; use your logic voltage	**Current**	0.5 to 2.5 mA measuring
Pull-up	10 kΩ, DATA to VCC, fitted	**Board**	22.4 × 30.4 mm, 4-pin 2.54 mm header