

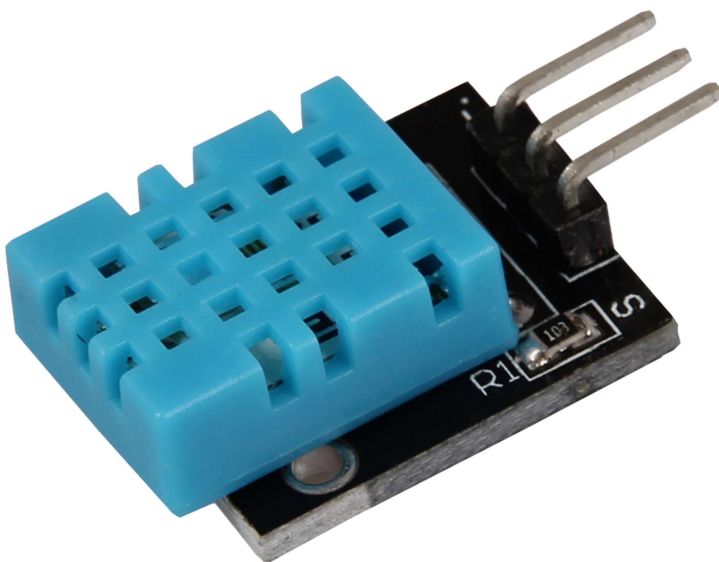
## KY-015 COMBI-SENSOR (TEMPERATURE & HUMIDITY)

This sensor is a mixture of temperature sensor and humidity sensor in a compact design.

---

**Arduino    Raspberry Pi    Micro:Bit**

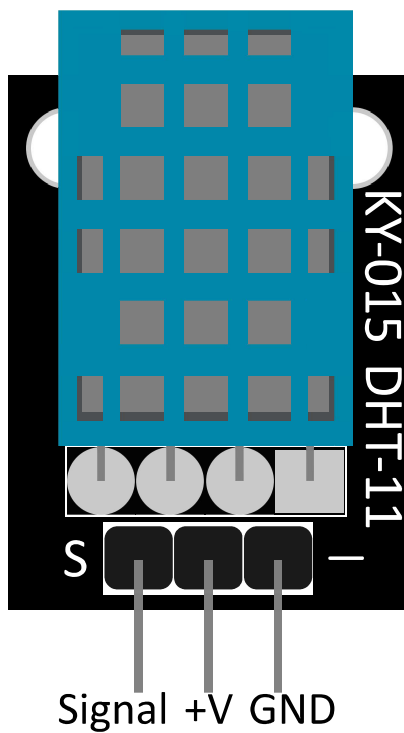
---



This sensor is a combination of temperature sensor and humidity sensor, united in a compact design. The disadvantage is the low sampling rate of the measurement, so that only every 2 seconds a new measurement result is available. This sensor is therefore particularly suitable for long-term measurements.

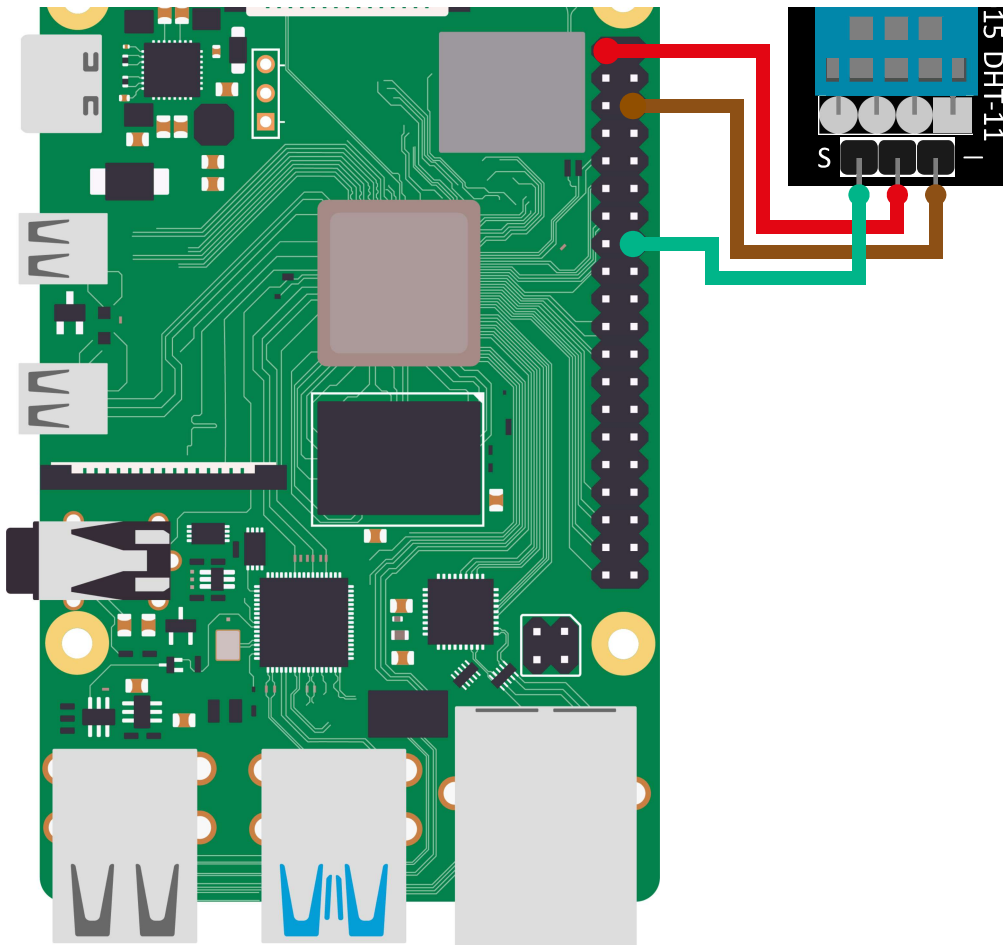
Chipset	DHT11
Communication Protocol	1-Wire
Measuring range	0 °C to 50 °C
Measurement accuracy	±2 °C
Measurement Accuracy	±5%RH
Measurable humidity	20-90%RH

## PIN ASSIGNMENT



## CODE EXAMPLE RASPBERRY PI

### PIN ASSIGNMENT RASPBERRY PI



#### RASPBERRY PI

GPIO 23 [Pin 16]

+3.3 V [Pin 1]

GND [Pin 6]

#### SENSOR

Signal

+V

GND

The program uses the corresponding **Adafruit\_CircuitPython\_DHT** library from Adafruit to control the DHT11 sensor, which is installed on this sensor module. This was released with the **MIT OpenSource license**.

This must be installed beforehand. Therefore, first update your package list and install the required dependencies using the following commands:

```
1 | sudo apt-get update
2 | sudo apt-get install build-essential python-dev
3 | sudo apt install gpio
4 |
```

Now the following Python example can be used. The program starts the measurement at the sensor and outputs the measured values for the air pressure and the temperature (in °C and °F).

```
1  import time
2  import board
3  import adafruit_dht
4
5  # Initialize the dht device with the data pin connected to pin 16 (GPIO17)
6  dhtDevice = adafruit_dht.DHT11(board.D23)
7
8  # You can pass DHT22 use_pulseio=False if you do not want to use pulseio
9  # This may be necessary on a Linux single board computer like the Raspberry Pi
10 # but it will not work in CircuitPython.
11 # dhtDevice = adafruit_dht.DHT22(board.D18, use_pulseio=False)
12
13 while True:
14     try:
15         # Print the values via the serial interface
16         temperature_c = dhtDevice.temperature
17         temperature_f = temperature_c * (9 / 5) + 32
18         humidity = dhtDevice.humidity
19         print("Temp: {:.1f} F / {:.1f} C    Humidity: {}% ".format(temperature_f,
20                               temperature_c, humidity))
21     except RuntimeError as error:
22         # Errors happen quite often, DHT's are hard to read, just move on
23         print(error.args[0])
24         time.sleep(2.0)
25         continue
26     except Exception as error:
27         dhtDevice.exit()
28         raise error
29
30     time.sleep(2.0)
```