

sleepCube - Programming Document

Ugur Turhal¹

¹ugur.turhal@me.com

Inhaltsverzeichnis

1	DHT22	1
1.1	Pins und deren Bedeutung	1
1.2	Verkabelung	2
2	DHT22 - Arduino	2
2.1	Arduino Code	3
2.2	Python and Arduino	4
2.3	Datenevaluation	6
3	Simples Modell	7

1 DHT22

Der Sensor ist ein DHT22, welcher wie folgt aus sieht. Mit diesem wird die Umgebungstemperatur gemessen. Der DHT22 hat 4 Pins siehe Bild 1. Dessen Erläuterung ist in der Tabelle1.

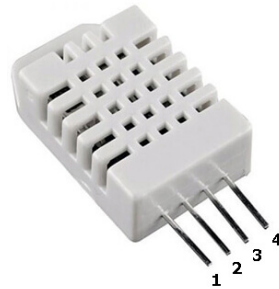


Abbildung 1. DHT22 - Pins

1.1 Pins und deren Bedeutung

Pin	Funktion
1	Betriebsspannung (3,3V - 5,5V)
2	Serial Data Line
3	Nicht belegt
4	Ground

Tabelle 1. Pins - Bedeutung

1.2 Verkabelung

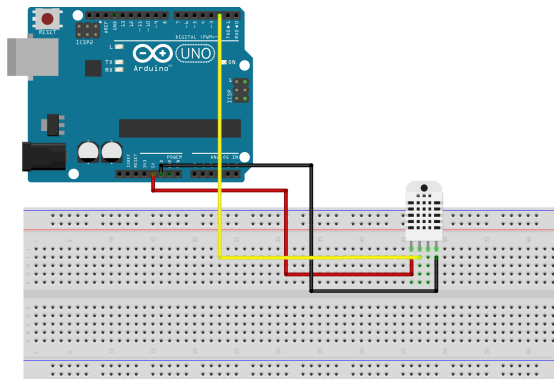


Abbildung 2. Temperatursensoren über eine bestimmte Zeit

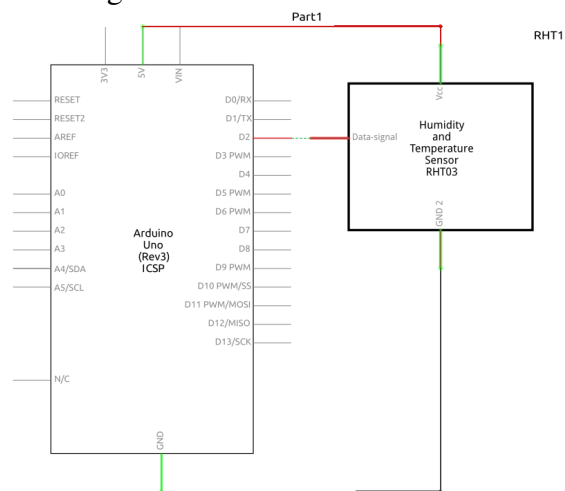
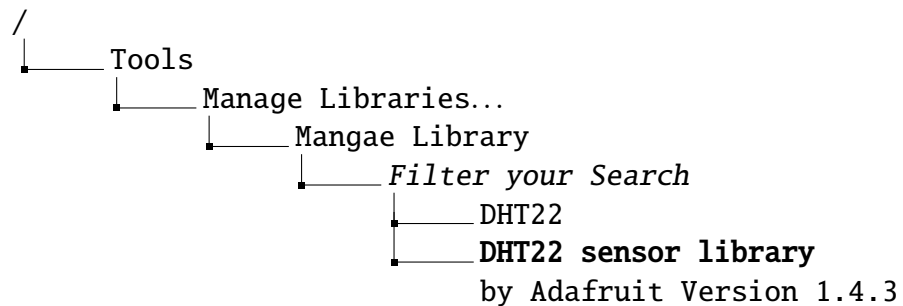


Abbildung 3. Temperatursensoren über eine bestimmte Zeit

2 DHT22 - Arduino

Installiere die DHT sensor library in die Arduino IDE:



DHT22 ist bereit zum Einsatz.

2.1 *Arduino Code*

```
1 #include "DHT.h"
3 #define DHTPIN 2 //Pin Nummer wo der Sensor angeschlossen ist
  #define DHTTYPE DHT22 //Definition was fuer ein Sensor ausgelesen wird
5
  DHT dht(DHTPIN, DHTTYPE);
7
  void setup() {
9    // put your setup code here, to run once:
    Serial.begin(9600);
11   // Serial.println("DHT22 Testprogramm");
    dht.begin();
13
  }
15
  void loop() {
17    delay(2000);

19    float h = dht.readHumidity();
    float t = dht.readTemperature();
21
23    if (isnan(h) || isnan(t)) {
      Serial.println("Fehler beim auslesen des Sensors!");
      return;
25    }

27
29    Serial.print(h);
    Serial.print(" ");
    Serial.println(t);
31  }
```

2.2 Python and Arduino

¹ Zwei kleine Python Programme wurden geschrieben um Daten aufzunehmen und zu visualisieren.

```
import serial
import time
import csv
import matplotlib
matplotlib.use("tkAgg")
import matplotlib.pyplot as plt
import numpy as np

ser = serial.Serial('/dev/ttyACM0')
ser.flushInput()

plot_window = 360
y_var = np.array(np.zeros([plot_window]))

plt.ion()
fig, ax = plt.subplots()
ax.grid(linestyle="--")
ax.set_ylabel("Celsius")
ax.set_xlabel("x")
ax.set(xlabel='x', ylabel='Celsius',
       title='Increase & Decrease of \n Temperature')
line, = ax.plot(y_var)

while True:
    try:
        ser_bytes = ser.readline()
        try:
            decoded_bytes = float(ser_bytes[5:len(ser_bytes)-2].
                                  decode("utf-8"))

            print(decoded_bytes)
        except:
            continue
        with open("test_data.csv", "a") as f:
            writer = csv.writer(f, delimiter=",")
            writer.writerow([time.time(), decoded_bytes])
        y_var = np.append(y_var, decoded_bytes)
        y_var = y_var[1:plot_window+1]
        line.set_ydata(y_var)
        ax.relim()
        ax.autoscale_view()
        fig.canvas.draw()
        fig.canvas.flush_events()
        fig.savefig('full_figure.png')
    except:
        print("Keyboard Interrupt")
        break
```

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

plt.style.use('seaborn')

def main():
    df = pd.read_csv("groupeddata.csv", usecols=['Measurement', '
        Temperature']).
        drop_duplicates(keep='first')
        .reset_index()

    measurement = df['Measurement']
    temperature = df['Temperature']
    df.to_csv("cleanData.csv");
    colors = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
        16, 17, 18, 19, 20, 21, 22,
        23, 24, 25, 26, 27, 28, 29,
        30, 31, 32, 33, 34, 35, 36,
        37, 38, 39, 40, 41, 42, 43,
        44, 45, 46, 47, 48, 49, 50,
        51, 52, 53, 54, 55, 56, 57,
        58, 59, 60, 61, 62, 63, 64,
        65, 66, 67, 68, 69, 70, 71,
        72, 73, 74, 75, 76, 77, 78,
        79, 80, 81, 82, 83, 84, 85,
        86, 87, 88, 89, 90]

    plt.scatter(x=measurement, y=temperature, s=25, c=colors,
        alpha=0.6, edgecolor='black',
        linewidth=1,
        cmap='Blues_r')

    cbar = plt.colorbar()
    cbar.set_label('Measurements with Temperature')
    plt.xscale('log')
    plt.yscale('log')
    plt.title('Scatter of Temperature')
    plt.xlabel('Measurement')
    plt.ylabel('Temperature')

    #plt.tight_layout()
    plt.savefig("data.png")
    plt.show()

if __name__ == "__main__":
    print(main())

```

2.3 Datenevaluation

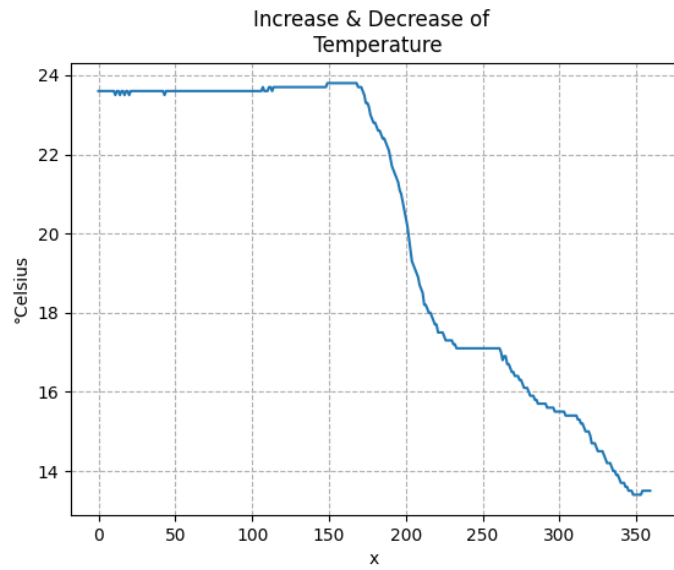


Abbildung 4. Temperatursausgaben über eine bestimmte Zeit

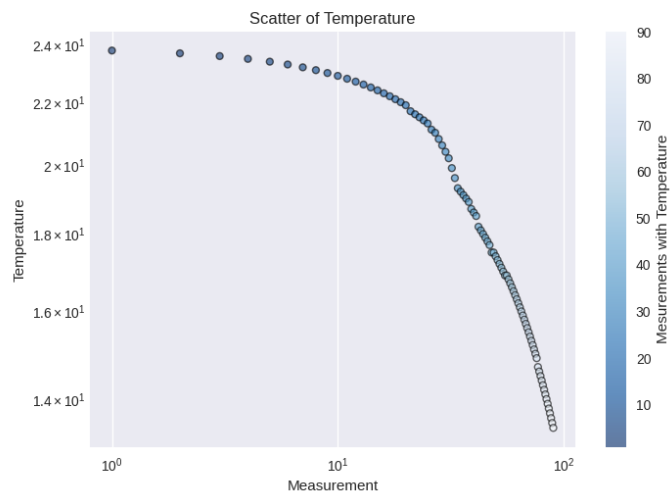


Abbildung 5. Einzelne Messpunkte des DHT22

3 Simples Modell

```
1 #include "DHT.h"

3 #define DHTPIN 2 //Pin Nummer wo der Sensor angeschlossen ist
  #define DHTTYPE DHT22 //Definition was fuer ein Sensor ausgelesen wird

5 #define pinRed 4 // LEDPin4
  #define pinGreen 5 //LEDPin5

7
DHT dht(DHTPIN, DHTTYPE);

9
void setup() {
11     // put your setup code here, to run once:
    Serial.begin(9600);
13     // Serial.println("DHT22 Testprogramm");
    pinMode(pinRed, OUTPUT);
15     pinMode(pinGreen, OUTPUT);

17     dht.begin();

19 }

21 void loop() {
    delay(2000);

23
    float h = dht.readHumidity();
25     float t = dht.readTemperature();

27     if (isnan(h) || isnan(t)) {
        Serial.println("Fehler beim auslesen des Sensors!");
29         return;
    }

31
    if (t >= 15.00 && t <= 18.00) {
33         digitalWrite(pinRed, LOW);
        digitalWrite(pinGreen, HIGH);
35     }
    else {
37         digitalWrite(pinGreen, LOW);
        digitalWrite(pinRed, HIGH);
39     }

41
    Serial.print(digitalRead(pinRed));
43     Serial.print(" ");
    Serial.print(digitalRead(pinGreen));
45     Serial.print(" ");
    Serial.println(t);
47 }
}
```

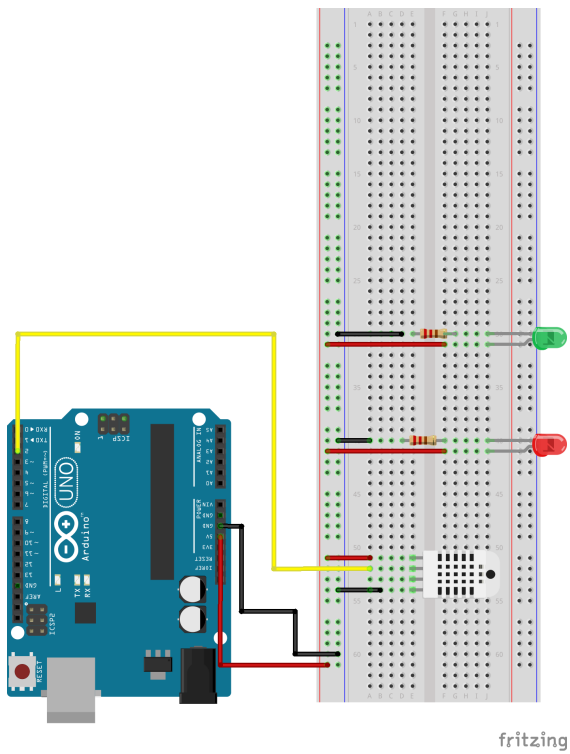



Abbildung 6. Einzelne Messpunkte des DHT22

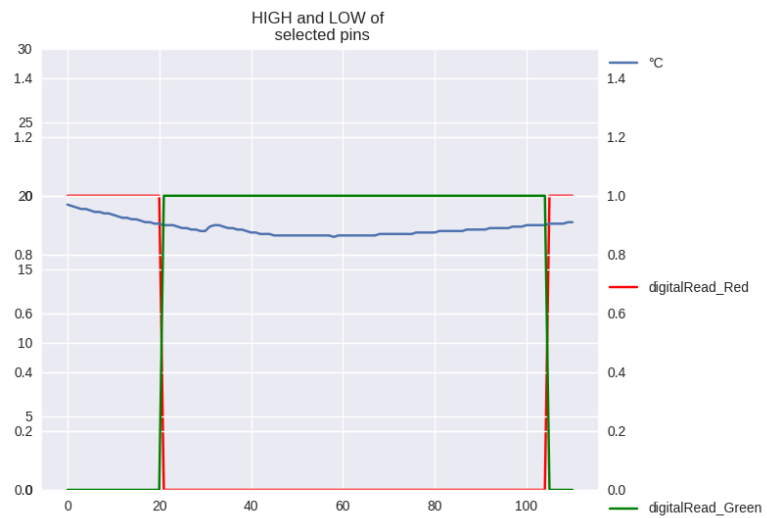


Abbildung 7. Einzelne Messpunkte des DHT22