ESP8266 Wetterstation

+ DATENVISUALISIERUNG IN INFLUX DB

INHALT

Mit diesem E-Book baust du dir eine ESP8266 Wetterstation, die dir die aktuelle Temperatur, Luftfeuchtigkeit und den Luftdruck anzeigt sowie deine Daten speichert und visualisiert. Die aktuellen Messdaten erscheinen auf einem kleinen OLED-Display. **Aber das ist nicht alles: Deine Messdaten speicherst du in einer Datenbank, um auf vergangene Messungen zugreifen und sie auswerten zu können.**

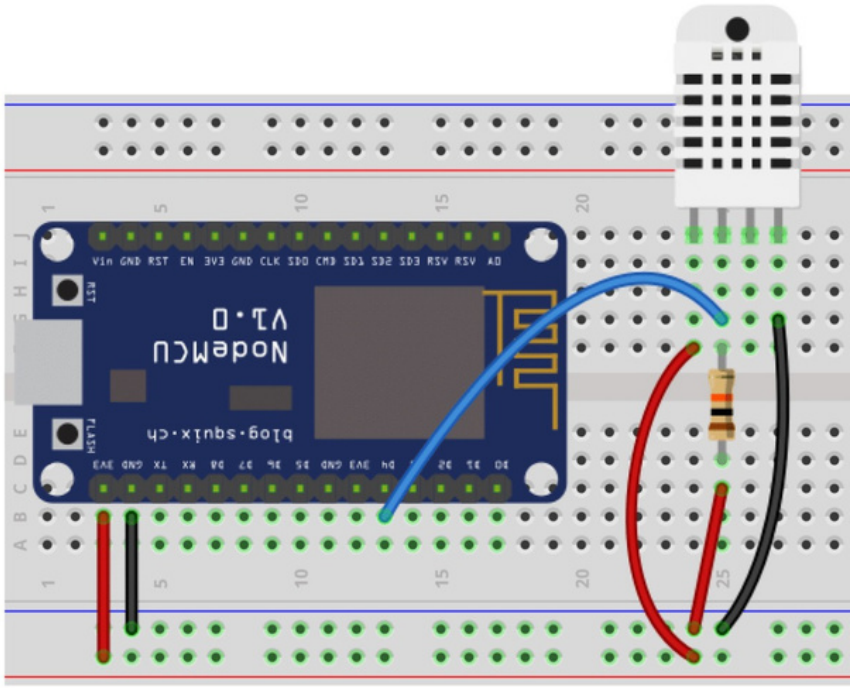
Um deine Messdaten zu speichern, verwendest du die **Datenbank InfluxDB**, die sich hervorragend dazu eignet, zeitgebundene Daten zu speichern und nebenbei auch noch die Visualisierung deiner Daten einfach und intuitiv ermöglicht. **InfluxDB 2 wird lokal auf einem Raspberry Pi laufen.** Der ESP8266 sendet die Messdaten dorthin und InfluxDB speichert und visualisiert sie. Anzeigen lassen kannst du diese dir dann in einem Browser und auf einem Gerät deiner Wahl.

Diese Bauteile benötigst du für die ESP8266 Wetterstation:

- NodeMCU ESP8266
- Sensor DHT22
- Sensor BMP180
- OLED-Display
- Raspberry Pi 4

DEN DHT22 AM ESP8266 ANSCHLIESSEN UND VERWENDEN

Der Temperatursensor DHT22 misst neben der Temperatur auch die Luftfeuchtigkeit. Um ihn an deinem ESP8266 anzuschließen, orientiere dich an der folgenden Skizze:

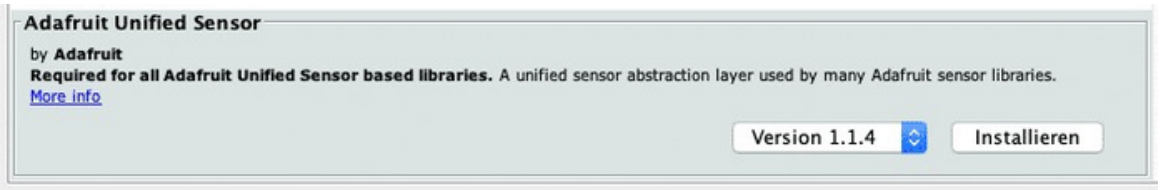


Übrigens: Falls du deinen ESP8266 noch nicht in der Arduino IDE verfügbar gemacht hast, findest du hier bei uns ein passendes Tutorial.

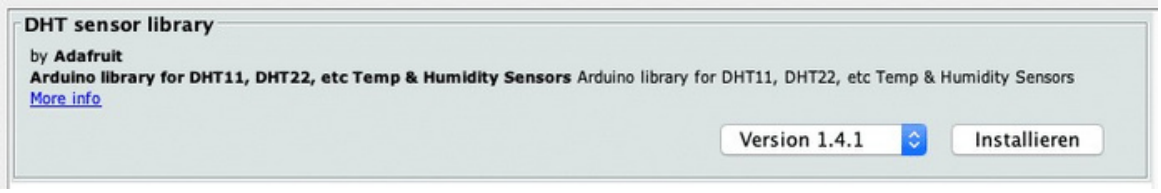
DIE PASSENDEN BIBLIOTHEKEN

Um deinen Sensor verwenden zu können, musst du zwei Bibliotheken installieren, von denen du jedoch nur eine im Sketch einbinden musst. Öffne deinen Bibliotheksmanager. Suche dort zunächst nach **Adafruit**

Unified Sensor und installiere die aktuelle Version. Die Versionsnummern in den folgenden Screenshots können abweichen.



Suche anschließend nach **DHT sensor library** und installiere die entsprechende Bibliothek.



DIE TEMPERATUR UND LUFTFEUCHTIGKEIT MESSEN

Kopiere dir den folgenden Sketch und lade ihn auf deinen Arduino hoch:

```
#include "DHT.h"

#define DHTPIN D4
#define DHTTYPE DHT22

float tempDHT22;
float humidity;

DHT dht(DHTPIN, DHTTYPE);
```

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

void loop() {

tempDHT22 = dht.readTemperature();
humidity = dht.readHumidity();

Serial.print("Temperatur: ");
Serial.print(tempDHT22);
Serial.println("*C");

  Serial.print("Luftfeuchtigkeit: ");
Serial.print(humidity);
Serial.println("%");

Serial.println();
delay(2000);
}
```

So funktioniert der Sketch: Nachdem du die Bibliothek eingebunden hast, legst du den Pin fest, an dem der Sensor angeschlossen ist. In unserem Sketch ist das der Pin **D4**.

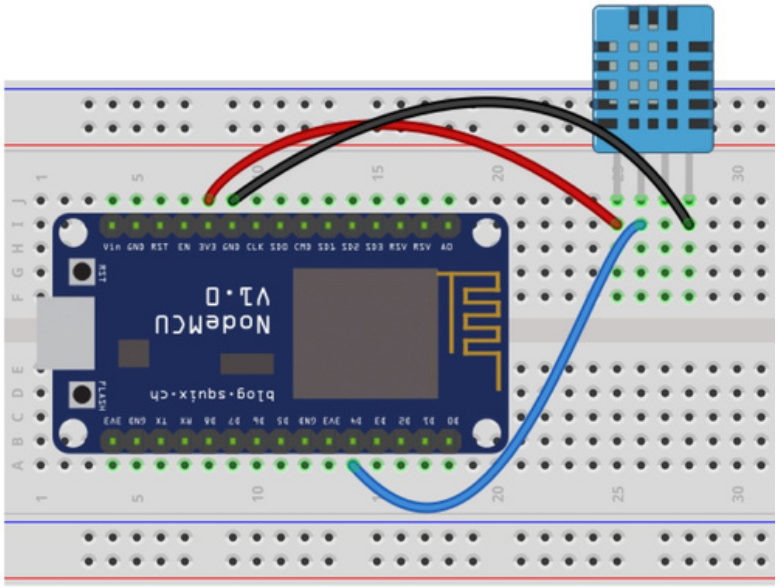
In der nächsten Zeile legst du das Modell des Sensors fest – in unserem Fall also ein DHT22. Anschließend erstellst du ein Objekt der Bibliothek names **dht**, das später bei der Messung mit den Funktionen **dht.readTemperature()** und **dht.readHumidity()** zum Einsatz kommt.

Der Rest des Sketchs dürfte für dich kein Problem sein. **Achte jedoch darauf, dass die Baudrate von Sketch und Seriellem Monitor übereinstimmt.**

Hinweis: Wenn du doch lieber den Sensor DHT11 verwenden möchtest, musst du nur eine Stelle im Sketch anpassen:

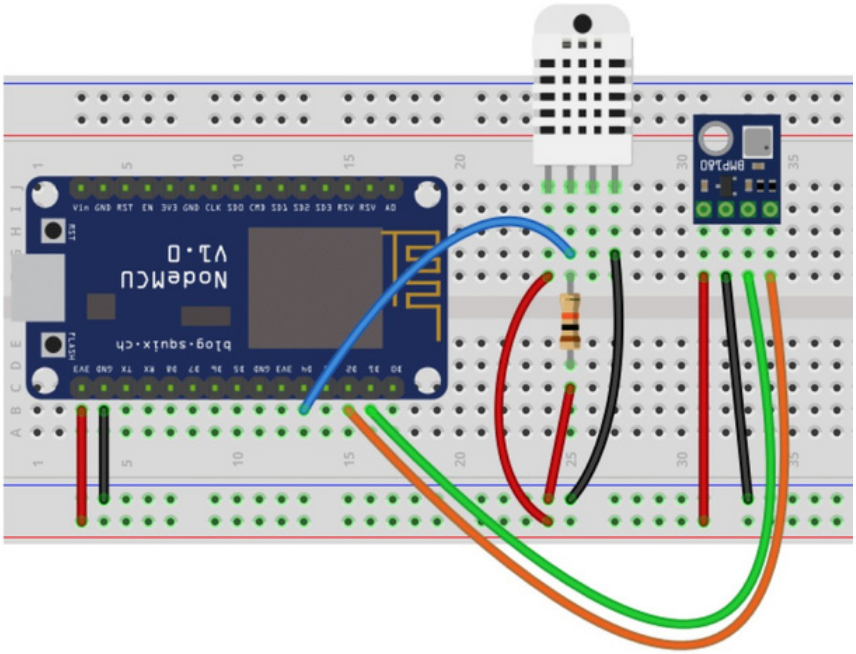
```
#define DHTTYPE DHT11
```

Wie du den “kleinen Bruder” des DHT22 – also den DHT11 – anschließt, erfährst du in der folgenden Skizze. Im weiteren Verlauf des Projekts verwenden wir jedoch weiterhin den DHT22.



DEN BMP180 AM ESP8266 ANSCHLIESSEN UND VERWENDEN

Neben der Luftfeuchtigkeit und der Temperatur soll die ESP8266 Wetterstation auch den aktuellen Luftdruck messen. Hierfür eignet sich der Sensor BMP180. Da dieser auch die Temperatur messen kann, schauen wir uns gleich auch noch die Messunterschiede zwischen DHT11 und BMP180 an. Doch zunächst zum Anschluss – orientiere dich hierbei an der folgenden Skizze:



DIE BENÖTIGTE BIBLIOTHEK

Neben der bereits vorinstallierten Bibliothek **Wire** (für die Kommunikation per I²C), benötigst du noch eine weitere, um die Daten des Sensors problemlos auslesen zu können.

Öffne also den Bibliotheksmanager in der Arduino IDE und suche nach **BMP180**. Du findest nun eine Bibliothek namens **Adafruit BMP085 Library** – das ist die richtige, auch wenn sie ein anderes Modell im Namen trägt. Der BMP085 war das Vorgängermodell des BMP180, was die Kommunikation angeht, jedoch mehr oder weniger baugleich.

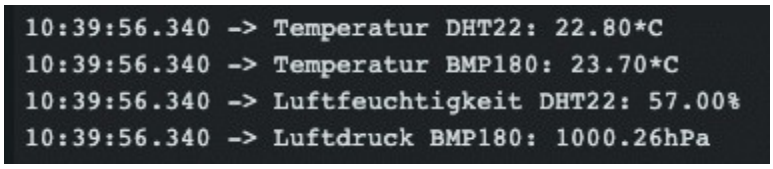
DIE TEMPERATUR UND DEN LUFTDRUCK MESSEN

Nun erweiterst du den obigen Sketch um den Code für den BMP180:

```
#include "DHT.h"
#include "Wire.h"
#include "Adafruit_BMP085.h"

#define DHTPIN D4
#define DHTTYPE DHT22

Adafruit_BMP085 bmp;

float tempDHT22;
float tempBMP180;
float humidity;
float pressure;

DHT dht(DHTPIN, DHTTYPE);

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

if (!bmp.begin()) {
    Serial.println("Sensor BMP180 nicht gefunden!");
while (1) {}
}
}

void loop() {

tempDHT22 = dht.readTemperature();
tempBMP180 = bmp.readTemperature();
humidity = dht.readHumidity();
pressure = bmp.readPressure();

Serial.print("Temperatur DHT22: ");
Serial.print(tempDHT22);
Serial.println("*C");

Serial.print("Temperatur BMP180: ");
Serial.print(tempBMP180);
Serial.println("*C");

Serial.print("Luftfeuchtigkeit DHT22: ");
Serial.print(humidity);
Serial.println("%");
```

```
  Serial.print("Luftdruck BMP180: ");
Serial.print(pressure/100);
Serial.println("hPa");

Serial.println();
delay(2000);
}
```

Sobald du den Sketch auf deinen ESP8266 geladen hast, sollten im seriellen Monitor die vier Messdaten erscheinen.



In den allermeisten Fällen dürfte die Temperaturmessung des DHT22 und des BMP180 etwas auseinander liegen. Ein Blick in die jeweiligen Datenblätter verrät, dass beide Sensoren eine Genauigkeit von ±0,5°C haben – das ist eigentlich schon recht genau und für eine Wetterstation sicherlich genau genug. Für welchen Messwert du dich entscheidest, liegt nun bei dir – vielleicht hast du noch ein weiteres Thermometer zur Hand, das du als Referenz einsetzen kannst.

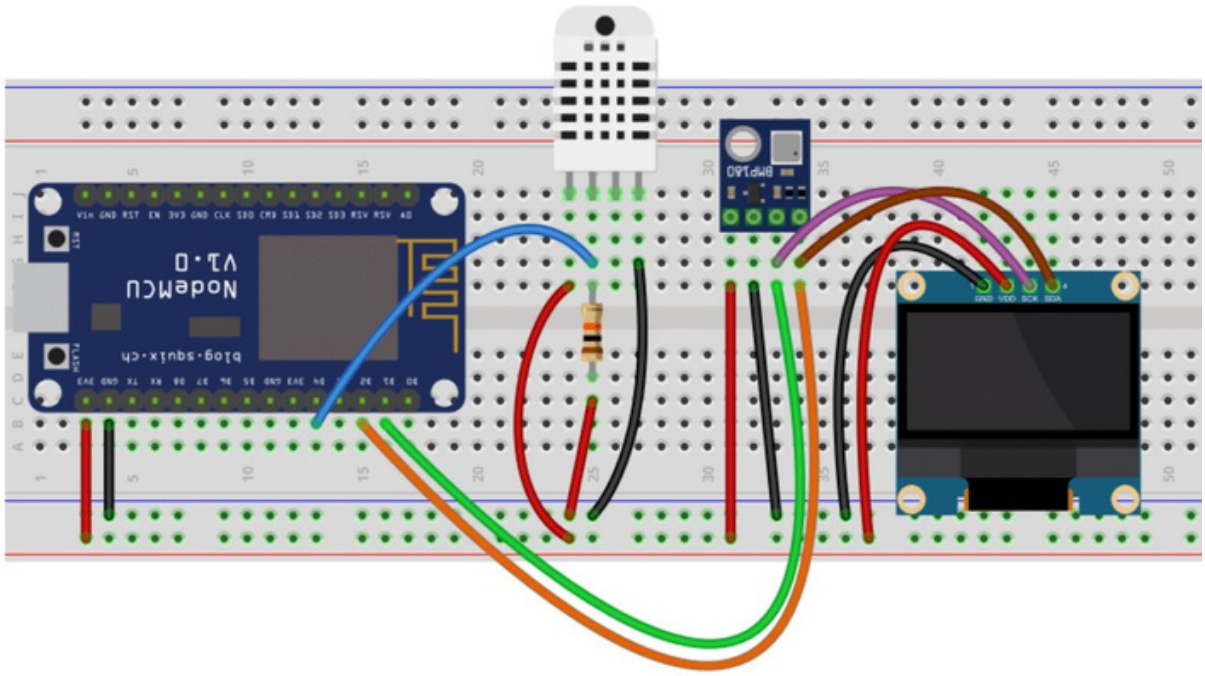
Im weiteren Verlauf dieses Projekts verwenden wir die Temperaturdaten des BMP180.

DAS OLED-DISPLAY ANSCHLIESSEN

Aktuell siehst du deine Messdaten nur im Seriellen Monitor. Deshalb kommt nun ein Display zum Einsatz, auf dem du sie bequemer ablesen

kannst. In diesem Projekt verwenden wir das handelsübliche **OLED-Display Adafruit SSD1306 mit einer Größe von 128×64 px.**

Erweitere also den Aufbau auf deinem Breadboard wie folgt:



Wie du siehst, sind sowohl der BMP180 als auch das OLED-Display per I²C (also an den Pins D1 und D2) am ESP8266 angeschlossen. Damit der Microcontroller beide Bauteile ansprechen kann, besitzen sie unterschiedliche Adressen – das OLED-Display mit 128×64 px die Adresse **0x3C** und der BMP180 die Adresse **0x77**.

DIE BENÖTIGTEN BIBLIOTHEKEN

Nun ist deine ESP8266 Wetterstation vollständig. Allerdings fehlt noch der Sketch, mit dem du deine Messdaten auf dem OLED-Display anzeigst. Auch für das Display benötigst du die Unterstützung von