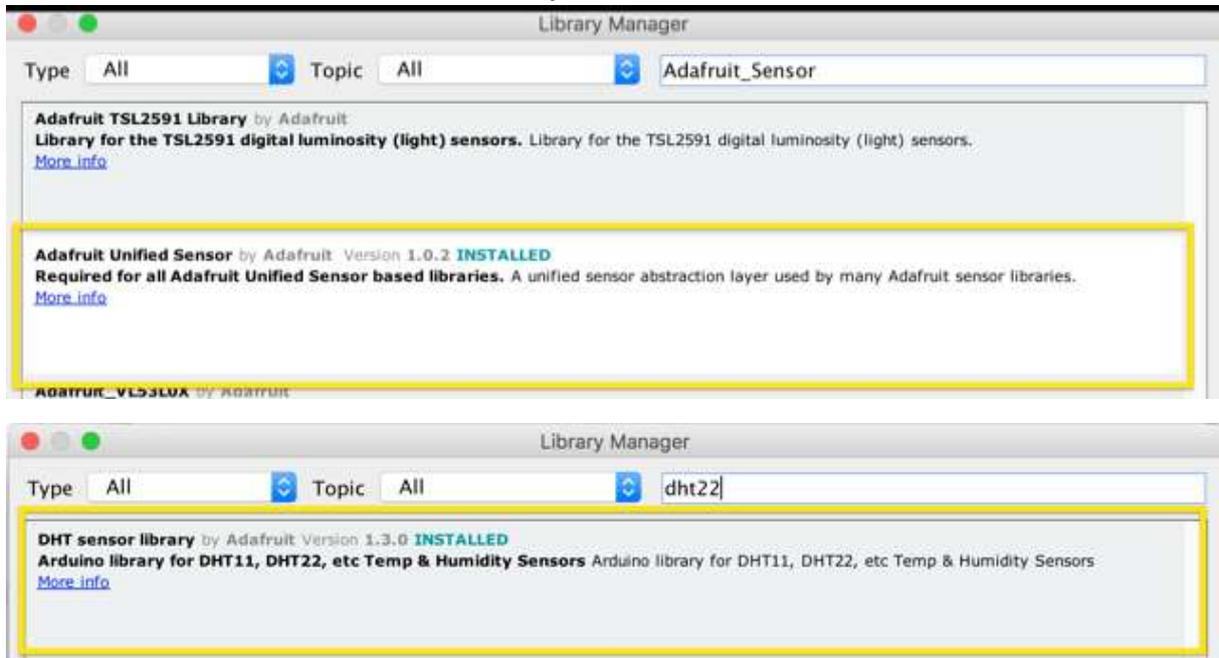
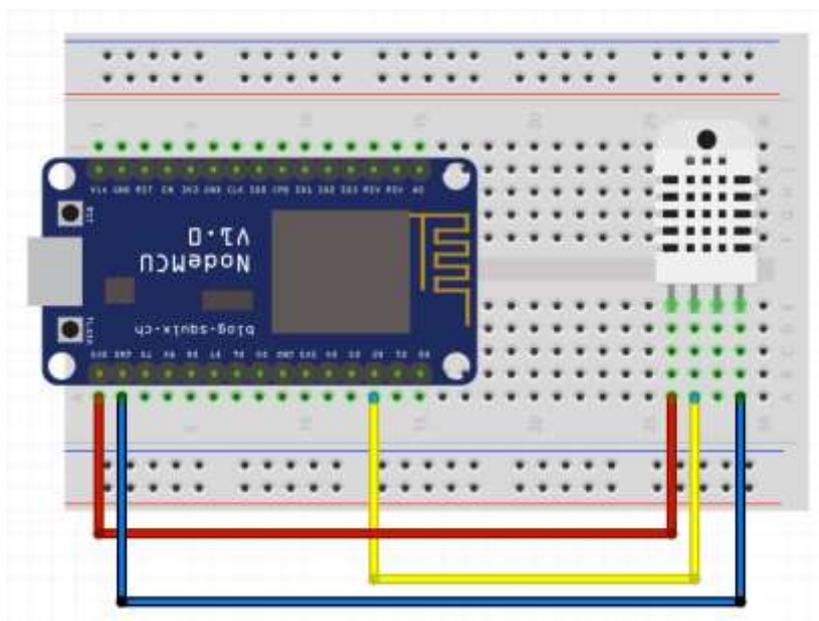


Installiere in der Arduino IDE über **Sketch – Bibliothek einbinden – Bibliothek verwalten ...** die beiden Bibliotheken **Adafruit Unified Sensor** und **DHT sensor library**.

Die Bibliothek **Adafruit Unified Sensor** ist die Grundlage für eine Reihe von weiteren Bibliotheken, unter anderem auch für die **DHT sensor library**.



Schließe den DHT22-Sensor an dein ESP-Modul an.



Lade das nachfolgende Programm.

```
#include <DHT.h>           // DHT Bibliothek laden
#define DHTPIN 4           // Der Sensor wird an PIN D2/GPIO4 angeschlossen
#define DHTTYPE DHT22     // Es handelt sich um den DHT22 Sensor

DHT dht(DHTPIN, DHTTYPE); // Der Sensor wird ab jetzt mit „dht“ angesprochen

void setup() {
  Serial.begin(115200);    // Verbindung zum seriellen Monitor starten
  dht.begin();            // DHT22 Sensor starten
}

void loop() {
  delay(2000);           // Warten bis zur nächsten Messung
  float Luftfeuchtigkeit = dht.readHumidity(); // Luftfeuchtigkeit auslesen
  float Temperatur = dht.readTemperature();    // die Temperatur auslesen
  float Hitzeindex = dht.computeHeatIndex(Temperatur, Luftfeuchtigkeit);
  Serial.print("Luftfeuchtigkeit: " + String(Luftfeuchtigkeit)+" % \t");
  Serial.print("Temperatur: " + String(Temperatur) + " °C\t");
  Serial.println("Hitzeindex: " + String(Hitzeindex) + " °C");
}
```

Statt die Daten auf den Seriellen Monitor der Arduino IDE auszugeben, können wir auch den Seriellen Plotter der IDE nutzen, um die Daten in einem Diagramm zu visualisieren.

Dieser ist, wie der Serielle Monitor, im Menü Werkzeuge zu finden.

Der Seriellen Plotter stellt alle numerischen Daten einer Ausgabezeile als Messpunkte im Diagramm dar. Entsprechend der Reihenfolge wird jeweils eine andere Farbe gewählt. Die Ordinatenachse skaliert sich automatisch und auf der Abszissenachse werden nur die letzten 500 Messpunkte dargestellt.

Für die Ausgabe der Messwerte lernen wir noch einige weitere Versionen kennen.

```
char text[44];
// sprintf(text, "%6.1f,%6.1f,%6.1f,%7d", Luftfeuchtigkeit, Temperatur, Hitzeindex);
sprintf(text, "%f,%f,%f,%d", Luftfeuchtigkeit, Temperatur, Hitzeindex);
Serial.println(text);
```

Diese veränderte Darstellung brauchen wir für die Nutzung des TelemetryViewers.

## Senden der Daten zu einem Server bzw. auf eine IoT-Plattform/Cloud

Du kannst dir ein das Programmpaket XAMPP auf deinen Rechner verfügbar machen und die Daten von deinem ESP-Modul zum Webserver senden. Dort werden die Daten gespeichert und können weiter aufbereitet werden.

Eine weitere Möglichkeit besteht in der Nutzung von verschiedenen Internetplattformen, die für diese Zwecke zur Verfügung stehen.

Wir probieren es mit der IoT-Plattform [DEVELOPER SANDBOX von Losant](#).

1. Erstelle dir einen kostenfreien Account für Losant.
2. Erstelle ein standalone Device und sichere die Device ID

<https://www.losant.com/blog/getting-started-with-the-esp8266-and-dht22-sensor>

[How to Use Adafruit IO with an ESP8266 and the Arduino IDE](#)