

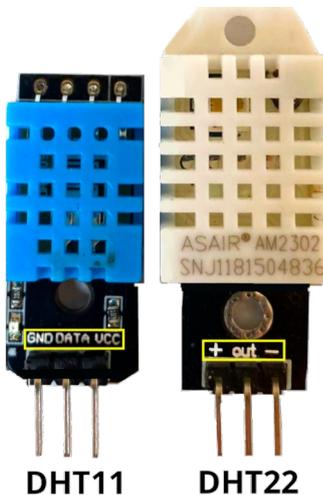
Das Programm liest Temperatur und Luftfeuchtigkeit, berechnet nach jeweils fünf Messungen einen Durchschnittswert für die Temperatur und zeigt diesen Wert ausführlich im Seriellen Monitor und in einer Kurzfassung auf einem OLED an.

So sieht es aus: (Messung an unterschiedlichen Tagen)



OLED-Displays (Organic Light Emitting Diode) benötigen im Unterschied zu LCDs keine Hintergrundbeleuchtung, sie leuchten selbst. Eine OLED besteht aus zwei Elektroden, von denen mindestens eine transparent sein muss. Im Zwischenraum befinden sich organische Halbleiterschichten aus natürlichen Farbstoffen. Die organischen Schichten leuchten, wenn sie von Gleichstrom durchflossen werden. Basis der Technik ist die Entdeckung der Elektrolumineszenz: ein Festkörper wird durch Anlegen einer elektrischen Spannung dazu angeregt Licht zu erzeugen.

### Beispiele für DHT11/DHT22 Sensoren

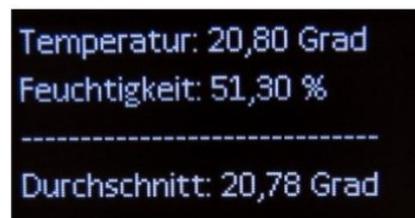


Die Pinbelegung kann sich von der hier gezeigten unterscheiden. Achte auf die Beschriftung auf dem Modul!

```

Ausgabe  Serieller Monitor x
Nachricht (Enter um Nachricht für 'Arduino Uno' auf '/dev/ttyACM0' zu senden)  Neue Zeile  9600 baud
-----
Messung 1: 16,30
Messung 2: 16,30
Messung 3: 16,20
Messung 4: 16,20
Messung 5: 16,30
Durchschnitt der letzten 5 Messungen: 16,26 Grad
-----
Messung: 1
Temperatur:      16,70 Grad
Luftfeuchtigkeit: 67,30 %
-----
Messung: 2
Temperatur:      16,50 Grad
Luftfeuchtigkeit: 67,20 %
-----
Messung: 3
Temperatur:      16,40 Grad
Luftfeuchtigkeit: 67,10 %
-----
Messung: 4
Temperatur:      16,20 Grad
Luftfeuchtigkeit: 66,90 %
-----
Messung: 5
Temperatur:      16,40 Grad
Luftfeuchtigkeit: 67,00 %
-----
Messung 1: 16,70
Messung 2: 16,50
Messung 3: 16,40
Messung 4: 16,20
Messung 5: 16,40
Durchschnitt der letzten 5 Messungen: 16,44 Grad
-----

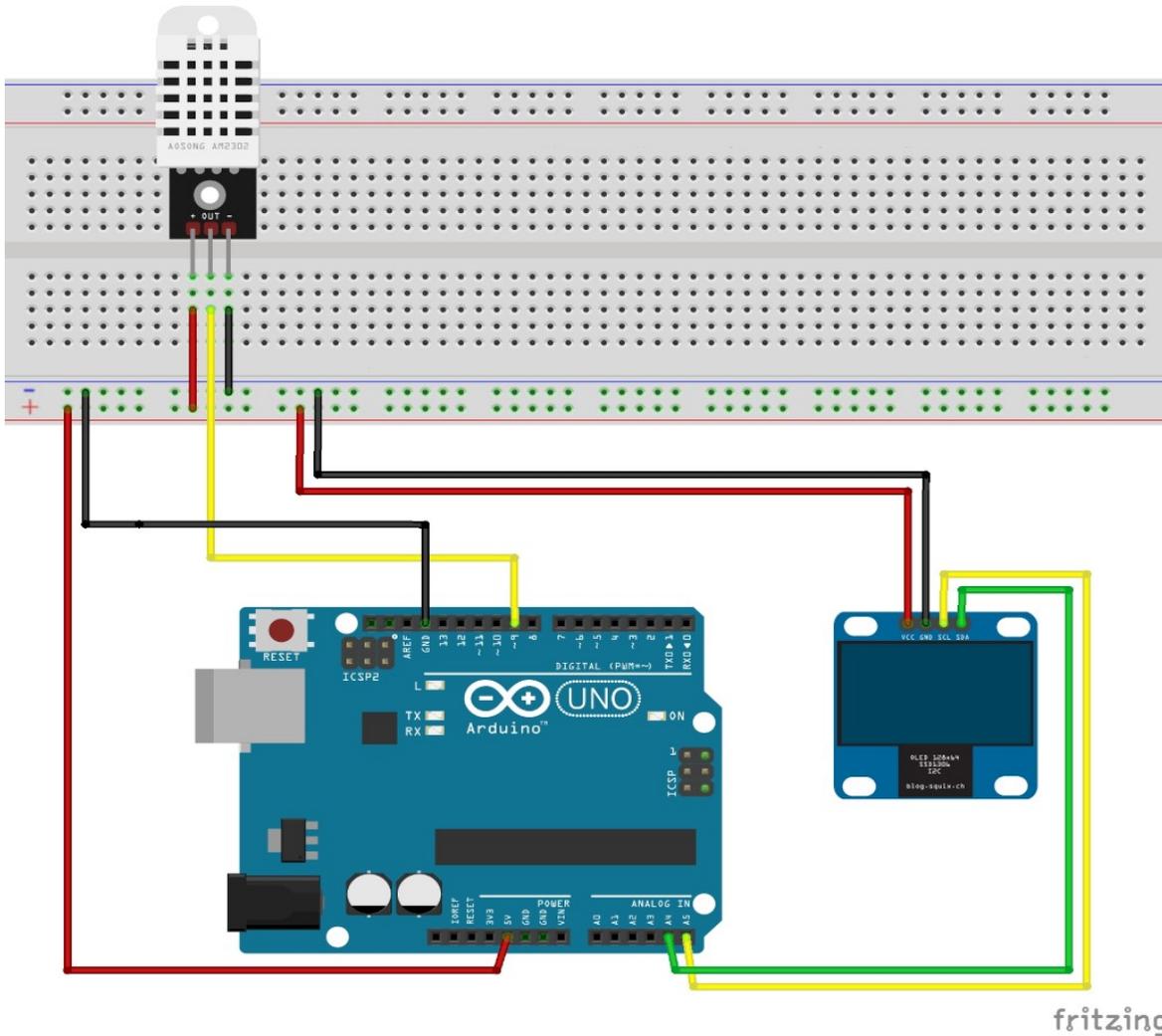
```



**Benötigte Bauteile:**

- ➔ LED
- ➔ 0,96 Zoll OLED
- ➔ Temperatursensor DHT22
- ➔ Leitungsdrähte

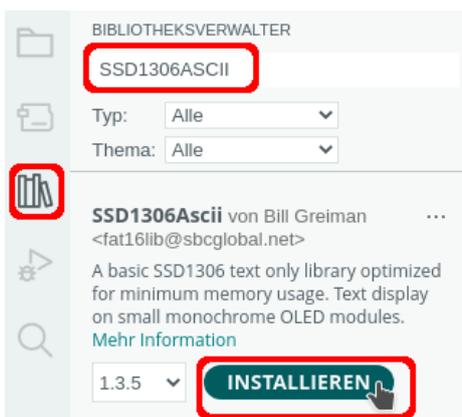
Baue die Schaltung auf.



fritzing

Benötigte Bibliotheken:

Sketch → **Bibliothek einbinden** → **Bibliotheken verwalten**





Binde die benötigten Bibliotheken ein und definiere die Variablen.

```
# include <DHT.h>
# include <SSD1306Ascii.h>
# include <SSD1306AsciiWire.h>

// Name des OLED festlegen (oled)
SSD1306AsciiWire oled;

// Pin des DHT-Sensors
int SENSOR_DHT = 9;

// Sensortyp festlegen
// DHT22 oder DHT11
# define SensorTyp DHT22

// Sensor DHT einen Namen zuweisen
DHT dht(SENSOR_DHT, SensorTyp);

int AnzahlMessungen = 0;

// Array für die Anzahl der Messungen
float Messung[5];

// Variable für die Berechnung der Durchschnittstemperatur
float Durchschnitt;

// Länge des Arrays Messung
int Anzahl = sizeof(Messung) / sizeof(Messung[0]);
```

Der setup-Teil. Beachte die Kommentare.

```
void setup()
{
  // Sensor DHT starten
  dht.begin();

  Serial.begin(9600);

  // OLED starten
  Wire.begin();
  Wire.setClock(400000L);
  oled.begin(&Adafruit128x64, I2C_ADDRESS);
```

```
// Schriftart bestimmen
oled.setFont(Callibri11);
oled.clear();
}
```

Im loop-Teil zwei Funktionen aufgerufen. Die Funktion DurchschnittBerechnen berechnet den Durchschnitt der letzten Messungen. Die Anzahl wird durch die im Kopf definierte Größe des Arrays Messung festgelegt. Als Variable werden ihr das Array Messung und die Größe des Arrays Messung übergeben.

```
float DurchschnittBerechnen(float Messung[], int Anzahl)
{
    float Durchschnitt = 0;
    for (int i = 0; i < Anzahl; i++)
    {
        Durchschnitt = Durchschnitt + Messung[i];
        // Funktion PunktErsetzen -> , durch . ersetzen
        String AnzeigeDurchschnitt = PunktErsetzen(String(Messung[i]));
        // aktuelle Messung anzeigen
        Serial.println("Messung " + String(i + 1) + ": " + AnzeigeDurchschnitt);
    }
    // Durchschnitt berechnen
    Durchschnitt = Durchschnitt / Anzahl;
    return Durchschnitt;
}
```

Die Funktion PunktErsetzen ersetzt in den gemessenen Werten den Punkt durch ein Komma.

```
String PunktErsetzen(String ErsatzString)
{
    ErsatzString.replace(".", ",");
    return ErsatzString;
}
```

Der loop-Teil. Beachte die Kommentare.

```
void loop()
{
    float Temperatur;
    float Luftfeuchtigkeit;

    // Daten lesen
    Temperatur = dht.readTemperature();
    Luftfeuchtigkeit = dht.readHumidity();

    Serial.println("Messung: " + String(AnzahlMessungen + 1));

    // . durch , ersetzen
    String AnzeigeTemperatur = PunktErsetzen(String(Temperatur));
    String AnzeigeLuftfeuchtigkeit = PunktErsetzen(String(Luftfeuchtigkeit));

    // Ausgabe im Seriellen Monitor mit Tabstopps formatieren
    Serial.print("Temperatur: \t\t");
    Serial.println(AnzeigeTemperatur + " °C");
    Serial.print("Luftfeuchtigkeit: \t");
    Serial.println(AnzeigeLuftfeuchtigkeit + " %");
}
```

```
Serial.println("-----");

// Ausgabe auf dem OLED
oled.setCursor(0, 0);
oled.print("Temperatur: ");
oled.print(AnzeigeTemperatur + " Grad");
oled.println();
oled.print("Feuchtigkeit: ");
oled.println(AnzeigeLuftfeuchtigkeit + " %");

// aktuelle Temperatur in das Array Messung schreiben
Messung[AnzahlMessungen] = Temperatur;
AnzahlMessungen ++;

// wenn jeweils Anzahl Messungen erfolgt sind
// AnzahlMessungen lässt sich ohne Rest durch Anzahl teilen
if (AnzahlMessungen % Anzahl == 0)
{
    /*
    Durchschnitt berechnen
    der Funktion DurchschnittBerechnen wird das Array Messung
    und die Länge des Arrays Messung (Anzahl übergeben
    */
    Durchschnitt = DurchschnittBerechnen(Messung, Anzahl);

    // . durch , ersetzen
    String AnzeigeDurchschnitt = PunktErsetzen(String(Durchschnitt));

    // Ausgabe der Durchschnittstemperatur Serieller Monitor
    Serial.print("Durchschnitt der letzten " + String(Anzahl));
    Serial.println(" Messungen: " + AnzeigeDurchschnitt + " °C");
    Serial.println("-----");

    // Ausgabe der Durchschnittstemperatur OLED
    oled.println("-----");
    oled.print("Durchschnitt: " + AnzeigeDurchschnitt + " Grad");
    AnzahlMessungen = 0;
}

// mindestens 2 Sekunden Pause
delay(2000);
}
```