

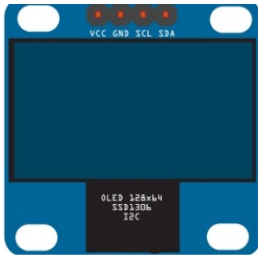


## Ziel des Projekts

Die mit einem AHT20-Sensor gemessene Temperatur und Luftfeuchtigkeit werden auf einem OLED angezeigt.

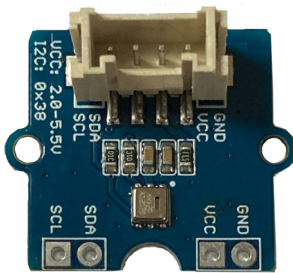
## Die Hardware

### OLED



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.

### AHT20

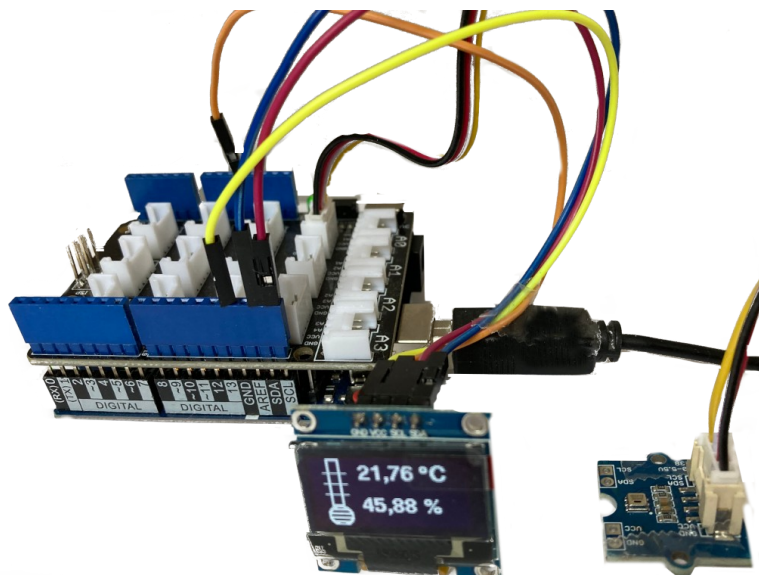


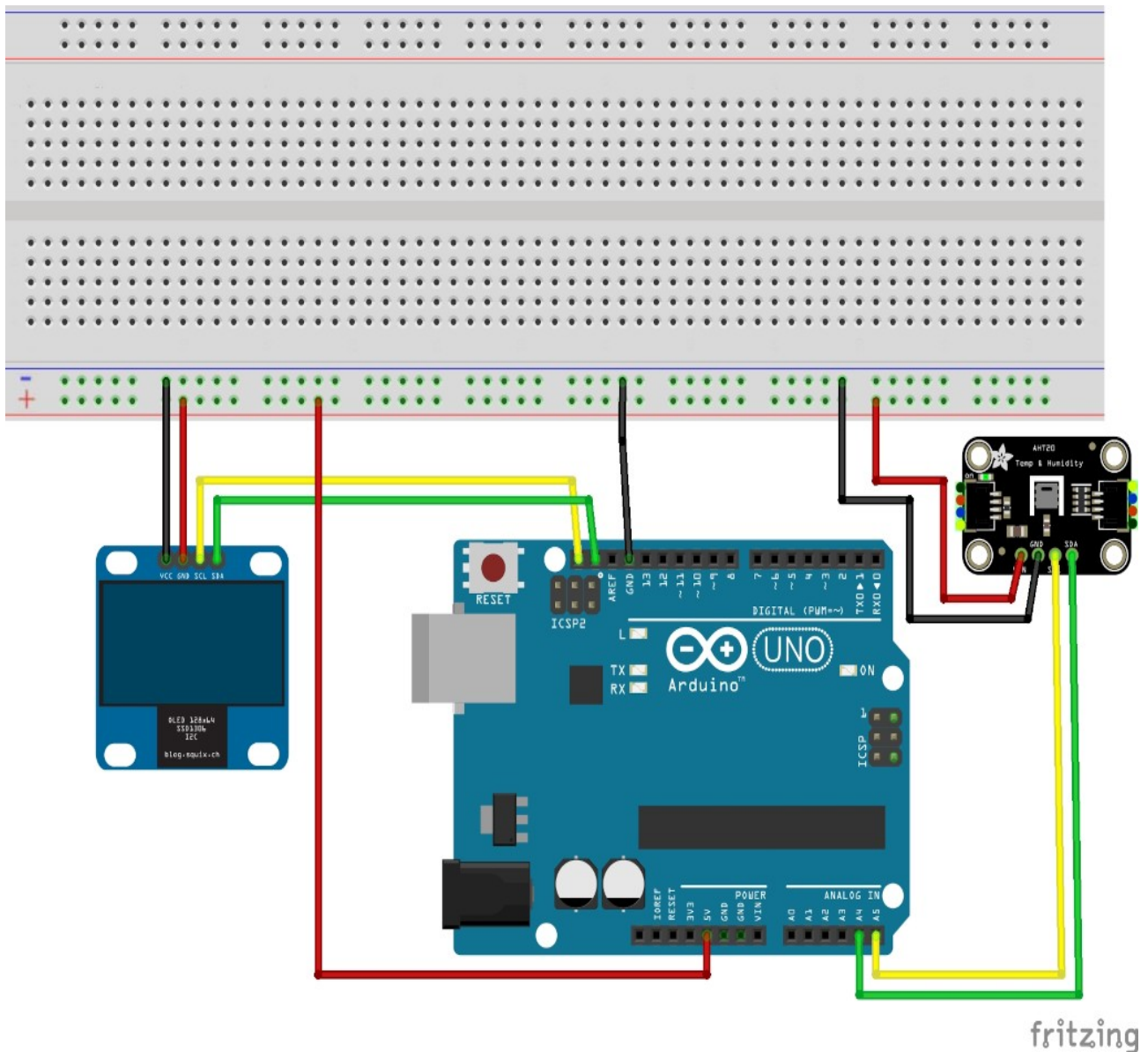
Der Sensor AHT20 misst die Temperatur und die Luftfeuchtigkeit. Er wird über I<sup>2</sup>C mit einem Grove-Stecker angeschlossen.

## Benötigte Bauteile

- 0,96 Zoll OLED
- Temperatursensor AHT20
- Leitungsdrähte

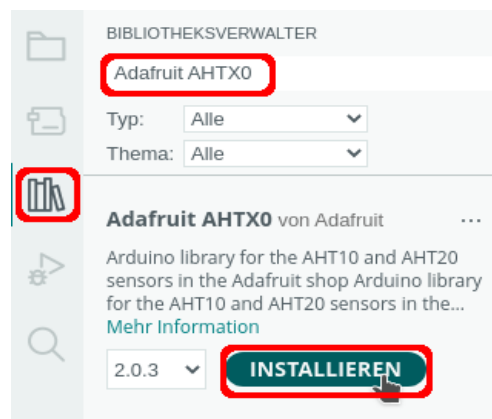
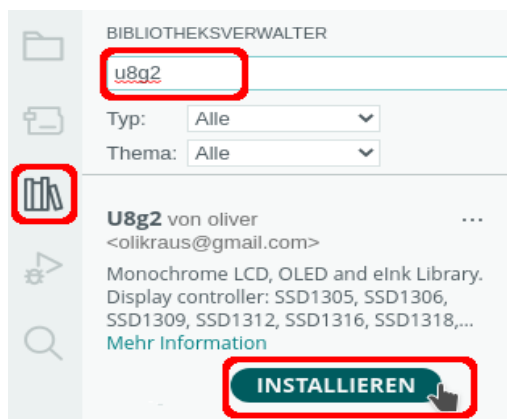
## Der Schaltplan





## Benötigte Bibliotheken

Sketch → Bibliothek einbinden → Bibliotheken verwalten





## Das Programm

Binde die benötigten Bibliotheken ein und definiere die Variablen.

```
#include "U8g2lib.h"
#include "Adafruit_AHTX0.h"

// Namen des Sensors (aht) festlegen
Adafruit_AHTX0 aht;

// 1,3 Zoll SH1106
// U8G2_SH1106_128X64_NONAME_1_HW_I2C oled(U8G2_R0, U8X8_PIN_NONE);
// 0,96 Zoll SSD1306
U8G2_SSD1306_128X64_NONAME_1_HW_I2C oled(U8G2_R0, U8X8_PIN_NONE);

// Bild Thermometer: Breite und Höhe definieren
# define ThermometerBreite 21
# define ThermometerHoehe 64

// Array Bild Thermometer
static unsigned char Thermometer[] =
{
  0xf0, 0x7f, 0x00, 0xf0, 0x7f, 0x00, 0x30, 0x60, 0x00, 0x30, 0x60, 0x00,
  0x30, 0x60, 0x00, 0x30, 0x60, 0x00, 0x30, 0x60, 0x00, 0x30, 0x60, 0x00,
  0x30, 0x60, 0x00, 0x30, 0x60, 0x00, 0x30, 0x60, 0x00, 0x30, 0x60, 0x00,
  0x30, 0x60, 0x00, 0xff, 0xff, 0x0f, 0x30, 0x60, 0x00, 0x30, 0x60, 0x00,
  0x30, 0x60, 0x00, 0x30, 0x60, 0x00, 0x30, 0x60, 0x00, 0x30, 0x60, 0x00,
  0x30, 0x60, 0x00, 0x30, 0x60, 0x00, 0x30, 0x60, 0x00, 0xff, 0xff, 0x0f,
  0x30, 0x60, 0x00, 0x30, 0x60, 0x00, 0x30, 0x60, 0x00, 0x30, 0x60, 0x00,
  0x30, 0x60, 0x00, 0x30, 0x60, 0x00, 0x30, 0x60, 0x00, 0x30, 0x60, 0x00,
  0x30, 0x60, 0x00, 0xff, 0xff, 0x0f, 0x30, 0x60, 0x00, 0x30, 0x60, 0x00,
  0x30, 0x60, 0x00, 0x30, 0x60, 0x00, 0x30, 0x60, 0x00, 0x30, 0x60, 0x00,
  0x30, 0x60, 0x00, 0x30, 0x60, 0x00, 0x30, 0x60, 0x00, 0xf0, 0x7f, 0x00,
  0xf0, 0x7f, 0x00, 0xf0, 0xff, 0x01, 0xf8, 0xe0, 0x03, 0xfc, 0xff, 0x07,
  0xfe, 0xff, 0x0f, 0x0e, 0x00, 0x0e, 0x07, 0x00, 0x1c, 0xff, 0xff, 0x1f,
  0xff, 0xff, 0x1f, 0x03, 0x00, 0x18, 0x03, 0x00, 0x18, 0xff, 0xff, 0x1f,
  0xff, 0xff, 0x1f, 0x07, 0x00, 0x1c, 0x0e, 0x00, 0x0e, 0xfe, 0xff, 0x0f,
  0xfc, 0xff, 0x07, 0xf8, 0xe0, 0x03, 0xf0, 0xff, 0x01, 0xc0, 0x7f, 0x00
};
```

Der setup-Teil. Beachte die Kommentare.

```
void setup()
{
  // Display starten
  oled.begin();

  // AHT starten
  aht.begin();

  // Kontrast maximal 255
  oled.setContrast(200);

  // Farbe weiß setzen
  oled.setDrawColor(1);
```



```
// Schriftart festlegen
oled.setFont(u8g2_font_helvB14_tf);
}
```

Im loop-Teil wird die Funktion PunktErsetzen aufgerufen. Sie ersetzt mit dem Befehl replace den Punkt durch ein Komma.

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

    // neuen String an das Hauptprogramm zurückgeben
    return ErsatzString;
}
```

Der loop-Teil liest die Werte des Sensors und zeigt sie auf dem OLED-Display an.

```
void loop()
{
    // Variablen für Temperatur und Luftfeuchtigkeit definieren
    sensors_event_t Feuchtigkeit, Temperatur;

    // Werte Temperatur und Luftfeuchtigkeit erfassen
    aht.getEvent(&Feuchtigkeit, &Temperatur);

    oled.firstPage();
    do
    {
        // Thermometer anzeigen
        oled.drawXBM(10, 1, ThermometerBreite, ThermometerHoehe, Thermometer);

        // . durch , ersetzen
        String AnzeigeTemperatur = PunktErsetzen(String(Temperatur.temperature));
        String AnzeigeLuftfeuchtigkeit = PunktErsetzen(String(Feuchtigkeit.relative_humidity));

        // Werte auf dem OLED anzeigen
        oled.setCursor(40, 20);
        oled.print(AnzeigeTemperatur + " " +char(176) + "C");

        oled.setCursor(40, 50);
        oled.print(AnzeigeLuftfeuchtigkeit + " %");
    }
    while (oled.nextPage());

    delay(2000);
}
```



```
float DurchschnittBerechnen(float Messung[], int Anzahl)
{
    float Durchschnitt = 0;
    for (int i = 0; i < Anzahl; i++)
    {
        Durchschnitt = Durchschnitt + Messung[i];

        // Methode 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:

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

    // Daten lesen
    dht22.read2(&Temperatur, &Luftfeuchtigkeit, NULL);

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

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

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