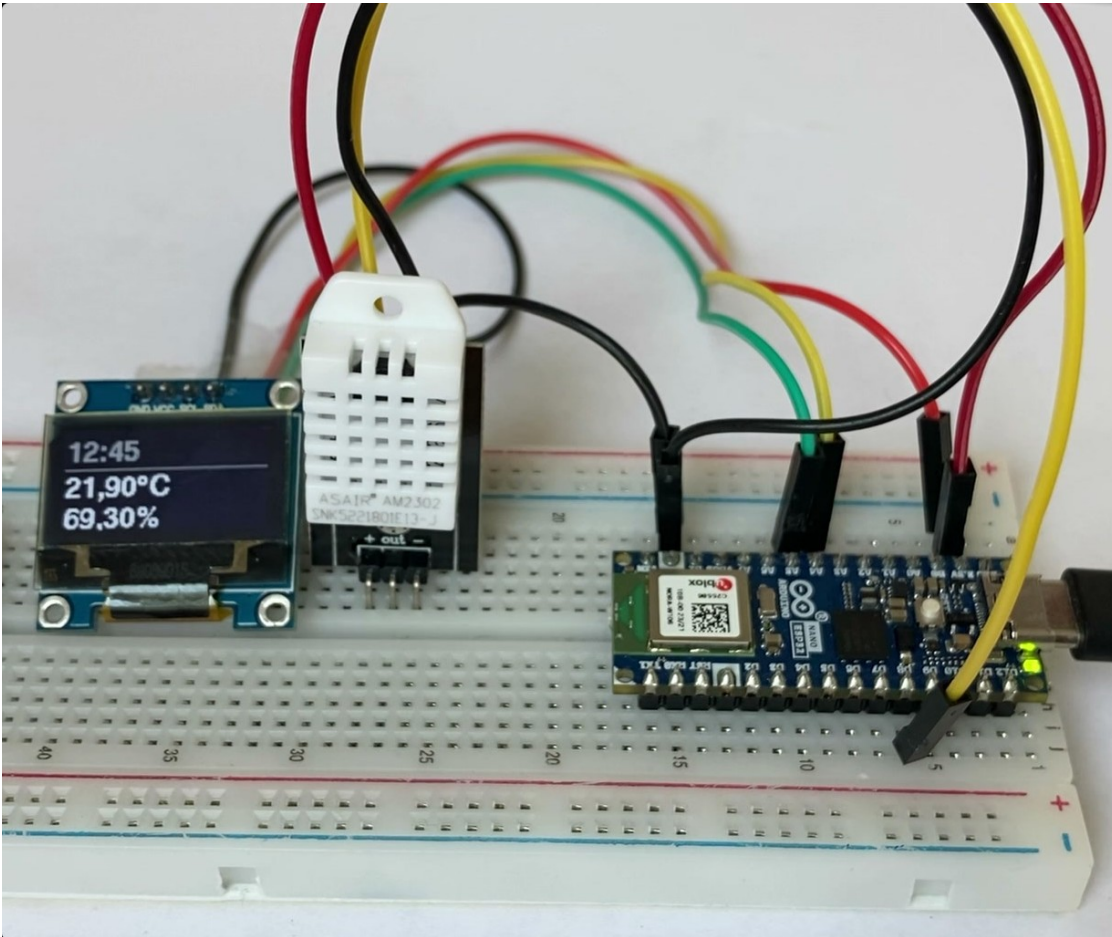
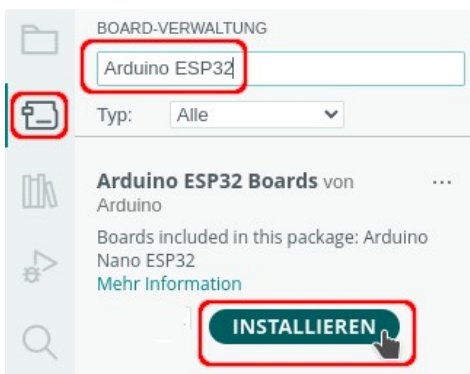


Mit dem Sensor DHT11/DHT22 werden Temperatur und Luftfeuchtigkeit gemessen, das Ergebnis wird auf einem OLED angezeigt. Die über einen Zeitserver ermittelte Zeit wird ebenfalls auf dem OLED dargestellt. Als Controller wird ein Arduino Nano ESP32 verwendet.



Zunächst muss das Board installiert werden:



Anschließend wird das Board ausgewählt:

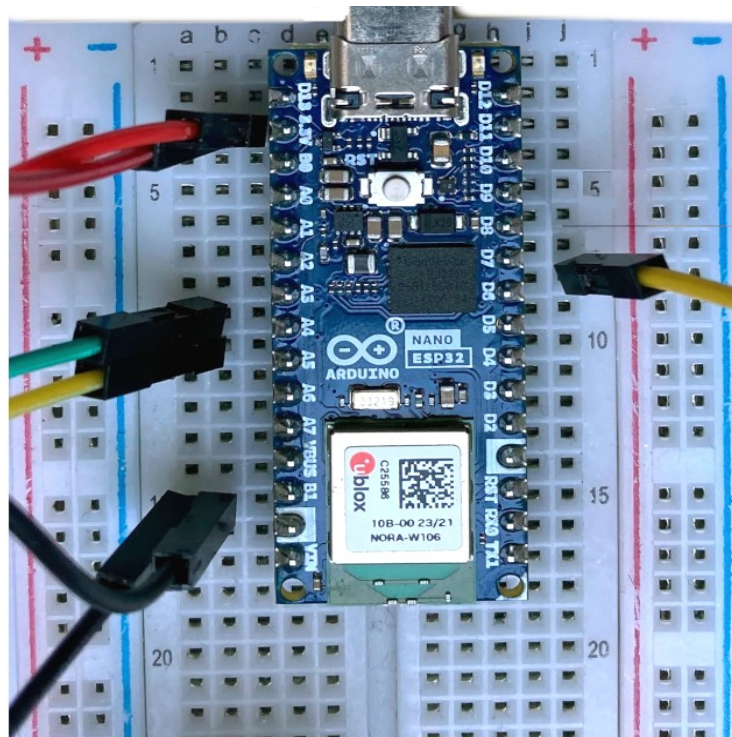


Pinbelegung:

3,3V OLED/DHT

A5 SCL OLED
A4 SDA OLED

GND OLED/DHT



D7 Datenpin DHT

Binde die benötigten Bibliotheken ein und definiere die Variablen:

```
#include "WiFi.h"
#include "time.h"
#include "U8g2lib.h"
#include "Wire.h"
#include "DHT.h"

// OLED initialisieren
U8G2_SSD1306_128X64_NONAME_1_HW_I2C oled(U8G2_R0, U8X8_PIN_NONE);

// SSID und Passwort des Routers
char Router[] = "Router_SSID";
char Passwort[] = "xxxxxxx";

// Pin des Sensors
int SENSOR_DHT = 7;

// DHT11
// # define SensorTyp DHT11

// DHT22
#define SensorTyp DHT22

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

```
// NTP-Server aus dem Pool
#define Zeitserver "de.pool.ntp.org"

/*
  Liste der Zeitzonen
  https://github.com/nayarsystems/posix_tz_db/blob/master/zones.csv
  Zeitzone CET = Central European Time -1 -> 1 Stunde zurück
  CEST = Central European Summer Time von
  M3 = März, 5.0 = Sonntag 5. Woche, 02 = 2 Uhr
  bis M10 = Oktober, 5.0 = Sonntag 5. Woche 03 = 3 Uhr
*/
#define Zeitzone "CET-1CEST,M3.5.0/02,M10.5.0/03"

// time_t enthält die Anzahl der Sekunden seit dem 1.1.1970 0 Uhr
time_t aktuelleZeit;

/*
  Struktur tm
  tm_hour -> Stunde: 0 bis 23
  tm_min -> Minuten: 0 bis 59
  tm_sec -> Sekunden 0 bis 59
  tm_mday -> Tag 1 bis 31
  tm_mon -> Monat: 0 (Januar) bis 11 (Dezember)
  tm_year -> Jahre seit 1900
  tm_yday -> vergangene Tage seit 1. Januar des Jahres
  tm_isdst -> Wert > 0 = Sommerzeit (dst = daylight saving time)
*/
tm Zeit;

// Zeit bis zur nächsten Messung
static unsigned long GesicherteStartZeit = 0;
unsigned long Startzeit;

// Sekunden Intervall, kann angepasst werden
int Intervall = 5000;
```

Der setup-Teil:

```
void setup()
{
  digitalWrite(LED_RED, LOW);
  Wire.begin();

  Serial.begin(9600);

  // Parameter für die zu ermittelnde Zeit
  configTzTime(Zeitzone, Zeitserver);

  WiFi.begin(Router, Passwort);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(200);
    Serial.print(".");
  }
}
```

```
// SSID des Routers anzeigen
Serial.println();
Serial.print("Verbunden mit ");
Serial.println(WiFi.SSID());

// IP anzeigen
Serial.print("IP: ");
Serial.println(WiFi.localIP());

// OLED starten
oled.begin();

// DHT starten
dht.begin();
}
```

Im loop-Teil wird die Funktion DatenAnzeigen aufgerufen:

```
void DatenAnzeigen()
{
    // aktuelle Zeit lesen
    time(&aktuelleZeit);

    // Temperatur
    String Temperatur = String(dht.readTemperature());
    Temperatur.replace(".", ",");

    // Luftfeuchtigkeit
    String Luftfeuchtigkeit = String(dht.readHumidity());
    Luftfeuchtigkeit.replace(".", ",");

    // localtime_r -> Zeit in die lokale Zeitzone setzen
    localtime_r(&aktuelleZeit, &Zeit);
    oled.clearDisplay();

    // Zeichenfarbe weiß
    oled.setDrawColor(1);

    // horizontale Schrift
    oled.setFontDirection(0);
    oled.firstPage();

    do
    {
        // horizontale Linie
        oled.setCursor(2, 20);

        // Zeit anzeigen
        oled.setFont(u8g2_font_helvb14_tf);

        if (Zeit.tm_hour < 10) oled.print("0");
        oled.print(Zeit.tm_hour);
        oled.print(":");

        if (Zeit.tm_min < 10) oled.print("0");
```

```
oled.print(Zeit.tm_min);
oled.drawLine(1, 25, oled.getDisplayWidth());
oled.setCursor(2, 43);
oled.print(Temperatur);
oled.print((char)176);
oled.print("C");

oled.setCursor(2, 63);
oled.print(Luftfeuchtigkeit + "%");
}
while (oled.nextPage());

// Serielle Ausgabe mit Namen des Wochentages
switch (Zeit.tm_wday)
{
  case 0:
    Serial.print("Sonntag ");
    break;

  case 1:
    Serial.print("Montag ");
    break;

  case 2:
    Serial.print("Dienstag ");
    break;

  case 3:
    Serial.print("Mittwoch ");
    break;

  case 4:
    Serial.print("Donnerstag ");
    break;

  case 5:
    Serial.print("Freitag ");
    break;
  case 6:
    Serial.print("Samstag ");
    break;
}

// Tag: führende 0 ergänzen
if (Zeit.tm_mday < 10) Serial.print("0");
Serial.print(Zeit.tm_mday);
Serial.print(".");

// Monat: führende 0 ergänzen
if (Zeit.tm_mon < 10) Serial.print("0");
Serial.print(Zeit.tm_mon + 1);
Serial.print(".");

// Anzahl Jahre seit 1900
Serial.print(Zeit.tm_year + 1900);
Serial.print(" ");
```

```
if (Zeit.tm_hour < 10) Serial.print("0");
Serial.print(Zeit.tm_hour);
Serial.print(":");

if (Zeit.tm_min < 10) Serial.print("0");
Serial.print(Zeit.tm_min);
Serial.print(":");
if (Zeit.tm_sec < 10) Serial.print("0");
Serial.print(Zeit.tm_sec);
Serial.println();

Serial.print("Temperatur:");
Serial.println(Temperatur + " °C");
Serial.print("Luftfeuchtigkeit: ");
Serial.println(Luftfeuchtigkeit + " %");
}
```

Der loop-Teil:

```
void loop()
{
  // Startzeit setzen
  Startzeit = millis();

  // wenn das festgelegte Intervall erreicht ist
  if (Startzeit - GesicherteStartZeit > Intervall)
  {
    DatenAnzeigen();

    // Startzeit zurücksetzen
    GesicherteStartZeit = Startzeit;
  }
}
```

Hartmut Waller (<https://hartmut-waller.info/arduino-blog>) letzte Änderung: 04.09.23