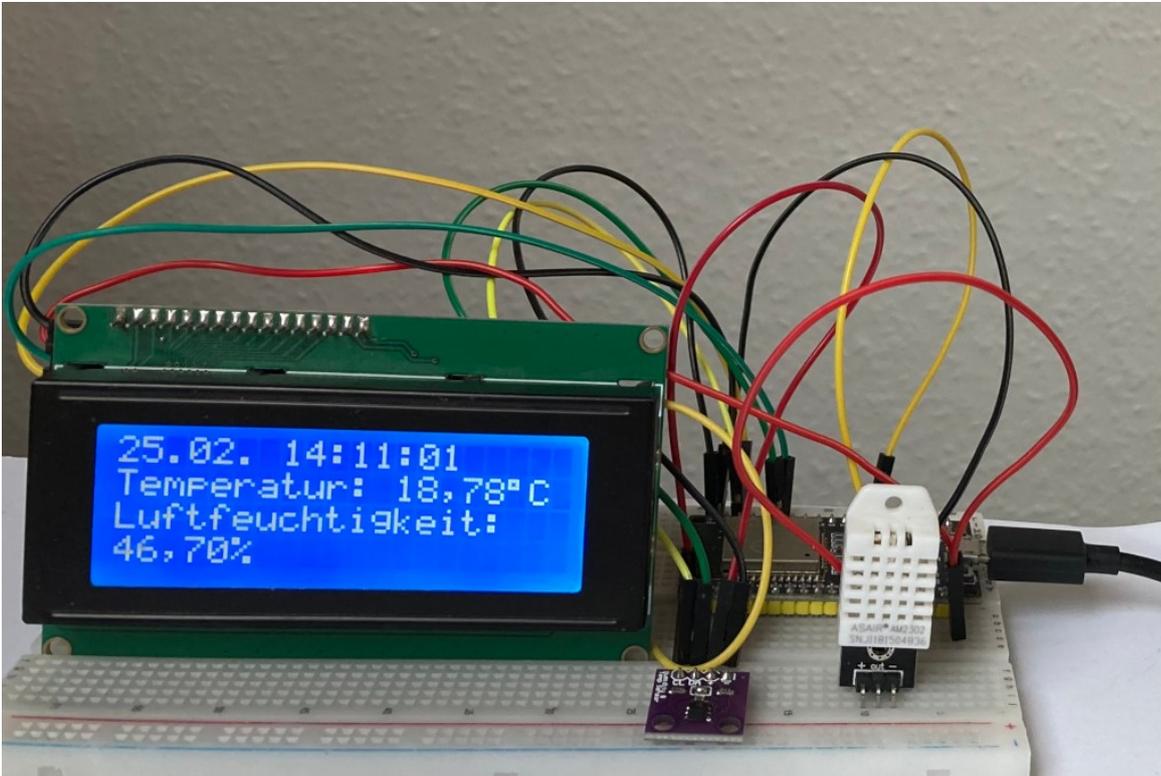


Temperatur und Luftfeuchtigkeit werden wahlweise mit einem SHT20, einem DHT11 oder einem DHT22 gemessen.

Alle Werte werden im Seriellen Monitor ausgegeben, auf dem LCD können nur die Daten eines Sensors angezeigt werden.



### Benötigte Bauteile:

- ➔ Temperatursensor (SHT20, DHT11, DHT22)
- ➔ LCD 1602 (20x4)
- ➔ Leitungsdrähte

```

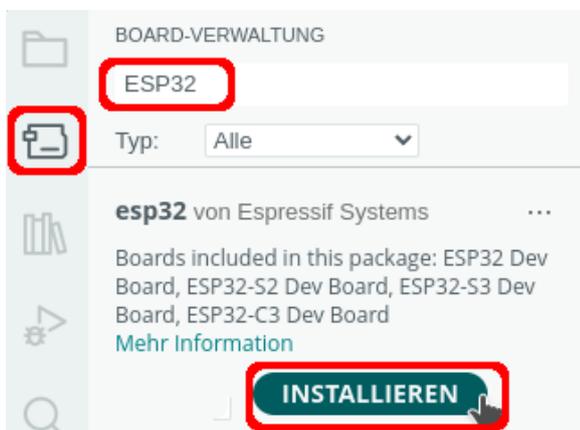
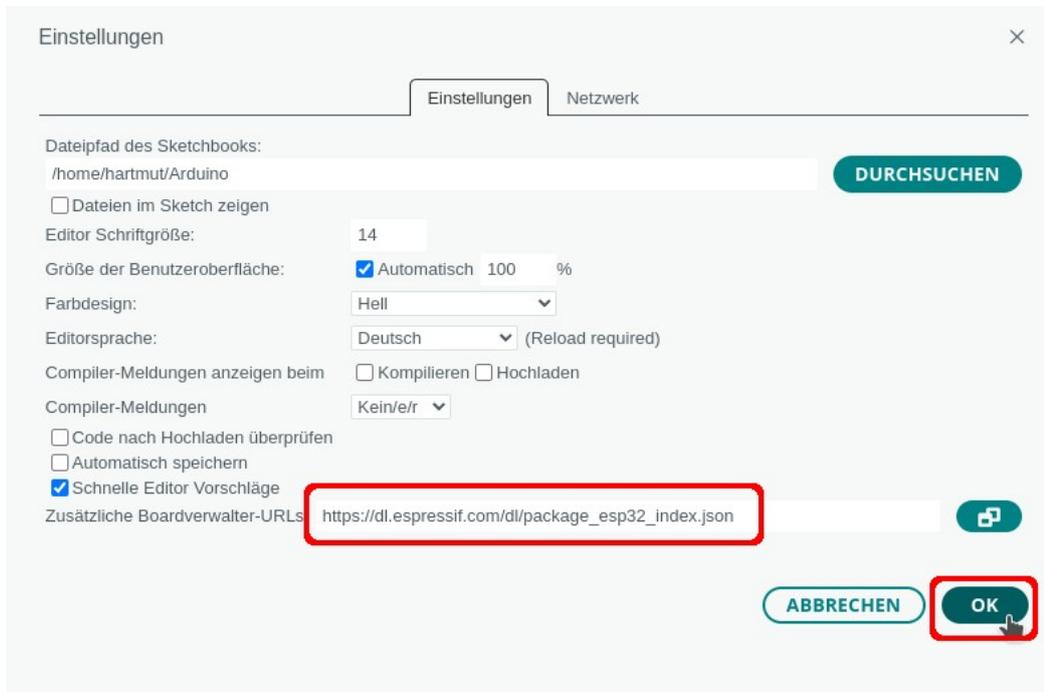
Ausgabe  Serieller Monitor x
Nachricht (Enter um Nachricht für 'ESP32-WROOM-DA | Sowohl NL als auch CR  9600 Baud
25.02.2024 18:45:44
IP: 192.168.1.200
Temperatur SHT: 19,12°C
Luftfeuchtigkeit SHT : 49,12%
Temperatur DHT: 19,80°C
Luftfeuchtigkeit DHT : 59,10%
-----
25.02.2024 18:45:50
IP: 192.168.1.200
Temperatur SHT: 19,12°C
Luftfeuchtigkeit SHT : 48,87%
Temperatur DHT: 19,80°C
Luftfeuchtigkeit DHT : 59,10%
-----
Zeile 240, Spalte 3  ESP32-WROOM-DA Module an /dev/ttyUSB0  2
    
```



### Board installieren:

Trage unter Datei -> Einstellungen eine zusätzliche Boardverwalter-URL ein:

[https://espressif.github.io/arduino-esp32/package\\_esp32\\_index.json](https://espressif.github.io/arduino-esp32/package_esp32_index.json)



→ Icon für den Boardverwalter anklicken  
oder:

→ Werkzeuge-> Board -> Boardverwalter

→ nach ESP32 suchen

→ Board installieren



Anschließend wird das Board ausgewählt. Je nach Betriebssystem wird der Port eine andere Bezeichnung haben.

Binde die benötigten Bibliotheken ein und definiere die Variablen.

```

/*
WiFi.h    -> WiFi-Verbindungen herstellen
time.h    -> Zeitfunktionen bereitstellen
SHT2x.h   -> Messfunktionen für den Sensor SHT20
LCDIC2.h  -> Anzeige auf LCD1602 mit I2C
DHT.h     -> Messfunktionen für die Sensoren DHT11/DHT22

```

```
*/
#include "WiFi.h"
#include "time.h"
#include "SHT2x.h"
#include "LCDIC2.h"
#include "DHT.h"
// Pin des DHT-Sensors
int SENSOR_DHT = 4;

// Sensortyp festlegen
// DHT22
# define SensorTyp DHT22

// DHT11
// # define SensorTyp DHT11

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

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

// NTP-Server aus dem Pool für Deutschland
#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)
*/
// Zeit -> Name der Struktur
tm Zeit;

// Kommunikation des Servers über den Standardport 80
WiFiServer Server(80);
```

```
// Name des Klienten
WiFiClient Client;

// 4-zeiliges LCD
LCDIC2 lcd(0x27, 20, 4);

// ESP32 I2C-Pins
#define SDA 21
#define SCL 22

// sht Wire zuordnen
SHT2x sht(&Wire);

// statischeIP = false -> IP-Adresse über DHCP vergeben
// statischeIP = true -> statische IP festlegen
// ip und gateway müssen an das lokale Netz angepasst werden
bool statischeIP = false;
IPAddress ip(192, 168, 1, 200);
IPAddress gateway(192, 168, 1, 1);
IPAddress subnet(255, 255, 255, 0);

/*
  öffentliche DNS-Server
  -----
  OpenDNS 208, 67, 222, 222 (USA)
  Google 8, 8, 8, 8 (USA)
  Cloudflare 1, 1, 1, 1 (USA)
  DNSWatch 84.200.69.80 (Deutschland)
  Quad9 9, 9, 9, 9 (Schweiz)
  Neustar UltraDNS 56, 154, 70, 3 (USA, gefiltert)
  Deutsche Telekom 217, 5,100,185
  -----
  oder die im Router eingetragene IP
  im Beispiel: 192, 168, 1, 20
*/
IPAddress primaryDNS(192, 168, 1, 20);
IPAddress secondaryDNS(9, 9, 9, 9);
```

Der setup-Teil. Beachte die Kommentare.

```
void setup()
{
  // Zeitzone: Parameter für die zu ermittelnde Zeit
  configTzTime(Zeitzone, Zeitserver);

  Serial.begin(9600);

  // auf serielle Verbindung warten
  while (!Serial);
  delay(500);

  // WiFi starten
  WiFi.begin(Router, Passwort);
```

```

// statische IP vergeben
if (statischeIP) WiFi.config(ip, gateway, subnet, primaryDNS, secondaryDNS);

// IP über DHCP ermitteln
else
{
    while (WiFi.status() != WL_CONNECTED)
    {
        delay(200);
        Serial.print(".");
    }
}

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

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

// Wire mit den Parametern für I2C starten
Wire.begin(SDA, SCL);

// sht-Sensor starten
sht.begin();

// dht-Sensor starten
dht.begin();

// LCD starten
lcd.begin();

// Cursor "verstecken"
lcd.setCursor(false);
}
    
```

Der loop-Teil. Beachte die Kommentare.

```

void loop()
{
    // aktuelle Zeit holen
    time(&aktuelleZeit);

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

    lcd.setCursor(0, 0);
    
```

```
// es kann bis zu 60 Sekunden dauern
// bis die Zeit ermittelt wird
// Tag: führende 0 ergänzen
if (Zeit.tm_mday < 10)
{
  Serial.print("0");
  lcd.print("0");
}
Serial.print(Zeit.tm_mday);
lcd.print(String(Zeit.tm_mday));

Serial.print(".");
lcd.print(".");

// Monat: führende 0 ergänzen
if (Zeit.tm_mon < 10)
{
  Serial.print("0");
  lcd.print("0");
}

// Zählung beginnt mit Januar = 0 -> Monat +1
Serial.print(Zeit.tm_mon + 1);
lcd.print(String(Zeit.tm_mon + 1));
Serial.print(".");
lcd.print(". ");

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

// Stunde: wenn Stunde < 10 -> 0 davor setzen
if (Zeit.tm_hour < 10)
{
  Serial.print("0");
  lcd.print("0");
}
Serial.print(Zeit.tm_hour);
Serial.print(":");
lcd.print(String(Zeit.tm_hour));
lcd.print(":");

// Minuten
if (Zeit.tm_min < 10)
{
  Serial.print("0");
  lcd.print("0");
}
Serial.print(Zeit.tm_min);
lcd.print(String(Zeit.tm_min));

Serial.print(":");
lcd.print(":");
```

```
// Sekunden
if (Zeit.tm_sec < 10)
{
  Serial.print("0");
  lcd.print("0");
}
Serial.print(Zeit.tm_sec);
lcd.print(String(Zeit.tm_sec));

Serial.println();
// sht-Sensor lesen
sht.read();

// gelesene Temperatur sht in String mit 2 Nachkommastellen umwandeln
String gemesseneTemperaturSHT = String(sht.getTemperature(), 2);

// . durch , ersetzen
gemesseneTemperaturSHT.replace(".", ",");

// gemessene Luftfeuchtigkeit sht in String mit 2 Nachkommastellen umwandeln
String gemesseneLuftfeuchtigkeitSHT = String(sht.getHumidity(), 2);

// . durch , ersetzen
gemesseneLuftfeuchtigkeitSHT.replace(".", ",");

// gemessene Temperatur dht in String umwandeln
String gemesseneTemperaturDHT = String(dht.readTemperature());

// . durch , ersetzen
gemesseneTemperaturDHT.replace(".", ",");

// gemessene Luftfeuchtigkeit dht in String umwandeln
String gemesseneLuftfeuchtigkeitDHT = String(dht.readHumidity());

// . durch , ersetzen
gemesseneLuftfeuchtigkeitDHT.replace(".", ",");

// Temperatur/Luftfeuchtigkeit sht anzeigen
Serial.println("Temperatur SHT: " + gemesseneTemperaturSHT + "°C");
Serial.println("Luftfeuchtigkeit SHT : " + gemesseneLuftfeuchtigkeitSHT + "%");

// Temperatur/Luftfeuchtigkeit dht anzeigen
Serial.println("Temperatur DHT: " + gemesseneTemperaturDHT + "°C");
Serial.println("Luftfeuchtigkeit DHT : " + gemesseneLuftfeuchtigkeitDHT + "%");
Serial.println("-----");

// Ausgabe auf dem LCD sht
lcd.setCursor(0, 1);
lcd.print("Temperatur: ");
lcd.print(gemesseneTemperaturSHT + "\337C");
lcd.setCursor(0, 2);
lcd.print("Luftfeuchtigkeit:");
lcd.setCursor(0, 3);
lcd.print(gemesseneLuftfeuchtigkeitSHT + "%");
```

```
/*
  Ausgabe auf dem LCD dht
  lcd.setCursor(0, 1);
  lcd.print("Temperatur: ");
  lcd.print(gemesseneTemperaturDHT + "\337C");
  lcd.setCursor(0, 2);
  lcd.print("Luftfeuchtigkeit:");
  lcd.setCursor(0, 3);
  lcd.print(gemesseneLuftfeuchtigkeitDHT + "%");
*/
delay(5000);
}
```