

Inhaltsverzeichnis

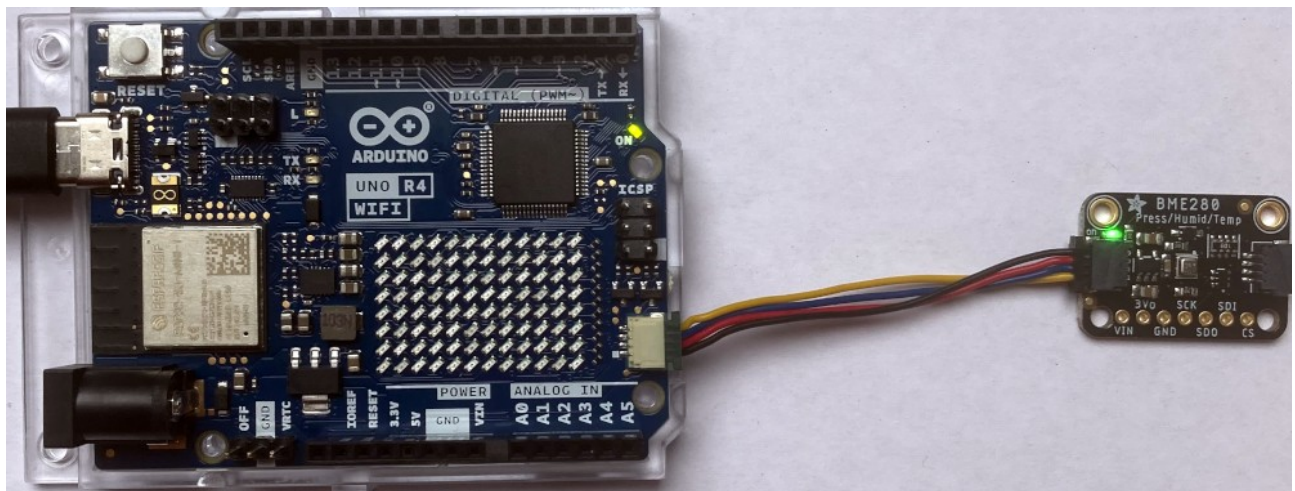
Hardware.....	1
Die Schaltung.....	1
Vorbereitung.....	1
Wetterstation im Seriellen Monitor.....	2
I2C-Bussysteme.....	2
Benötigte Bibliothek.....	3
Das Programm.....	4
Wetterstation im Browser anzeigen.....	6
Das Programm.....	6
Zusätzlich benötigte Bibliothek.....	6
Bibliotheken und Variable.....	7
Der setup-Teil.....	7
Der loop-Teil.....	9

Die Hardware

Der Sensor BME280 misst Temperatur, Luftfeuchtigkeit und Luftdruck. Der Luftdruck kann verwendet werden, um die ungefähre Höhe des Standorts zu berechnen. Der Arduino UNO R4 WiFi verfügt über ein WiFi-Modul und über einen QWIIC-Anschluss.

Dieser Anschluss wurde von Sparkfun entwickelt und fasst die Anschlüsse für 3,3 V, GND, SCL und SDA in einem Stecker zusammen. Selbstverständlich kann für diese Anleitung auch das Modul mit vier separaten Anschlüssen verwendet werden.

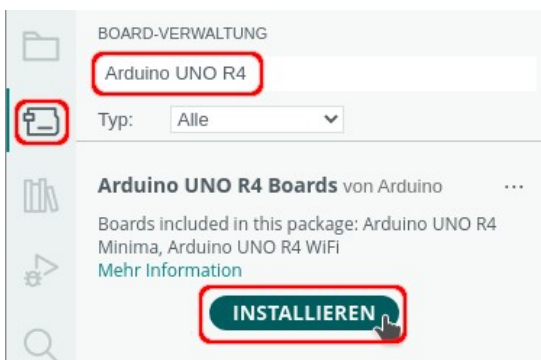
Die Schaltung



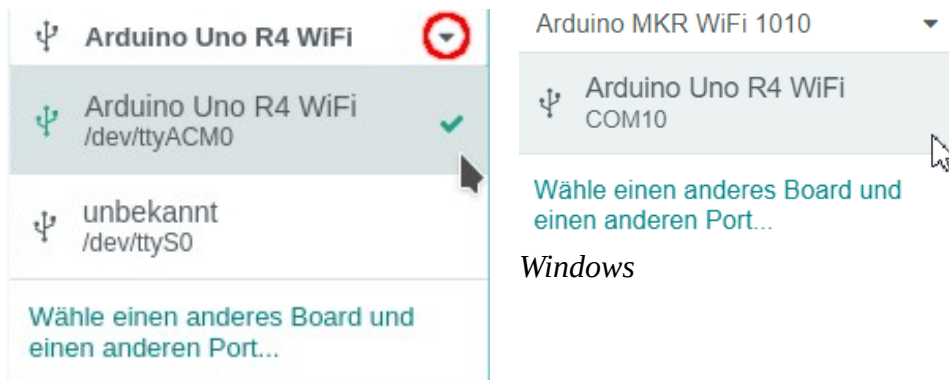
Arduino R4 WiFi BME280 QWIIC

Vorbereitung

Zunächst musst du über den Boardverwalter das Board installieren:



Wenn das Board angeschlossen ist, kann der USB-Anschluss ausgewählt werden. Der Name des Anschlusses unterscheidet sich je nach verwendetem Betriebssystem.



Linux

Wetterstation im Seriellen Monitor

I2C-Bussysteme

Der UNO R4 Wifi verfügt über zwei I2C-Bussysteme:

- ➔ den ersten I2C-Bus, er wird über vier Leitungsdrähte (3,3V, GND, SCL und SDA) angeschlossen (Wire)
- ➔ den zweiten I2C-Bus, er wird über QWIIC verbunden (Wire1)

Ein Testprogramm zeigt die verwendete Adresse an.



I2C-Scanner für BME280

Das dazugehörige Programm. Wenn du den ersten I2C-Bus abfragen willst, musst du jeweils Wire1 durch Wire ersetzen.

```
#include <Wire.h>

void setup()
{
  // Wire1 → QWIIC
  Wire1.begin();
  Serial.begin(9600);
  delay(500);
  Serial.print("I2C Scanner QWIIC");
}
```

```
void loop()
{
  byte Fehler, Adresse;
  int Geraete = 0;
  Serial.println("Starte Scanvorgang");

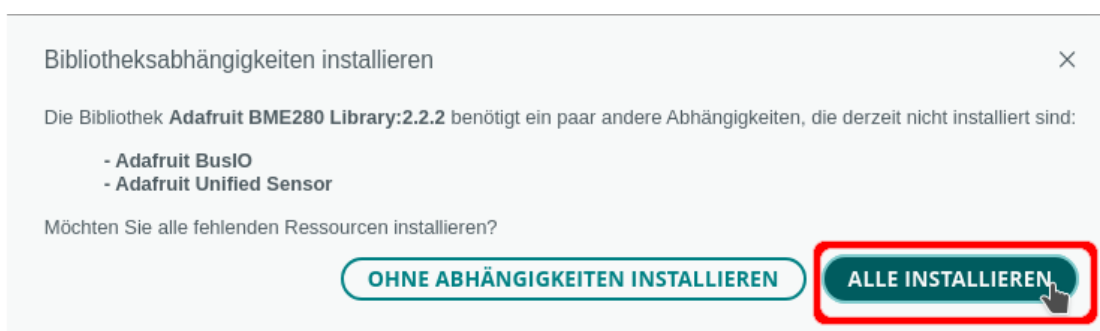
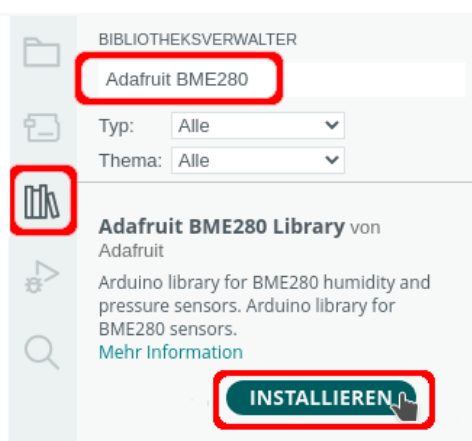
  for (Adresse = 1; Adresse < 127; Adresse++ )
  {
    // Übertragung starten
    Wire1.beginTransmission(Adresse);

    // wenn die Übertragung beendet wird
    Fehler = Wire1.endTransmission();

    if (Fehler == 0)
    {
      Serial.print("I2C Gerät gefunden - Adresse: 0x");
      if (Adresse < 16) Serial.print("0");
      Serial.print(Adresse, HEX);
      Serial.println("");
      Geraete++;
    }
  }
  if (Geraete == 0) Serial.println("Keine I2C Geräte gefunden\n");
  else Serial.println("Scanvorgang abgeschlossen");

  delay(10000);
}
```

Benötigte Bibliothek



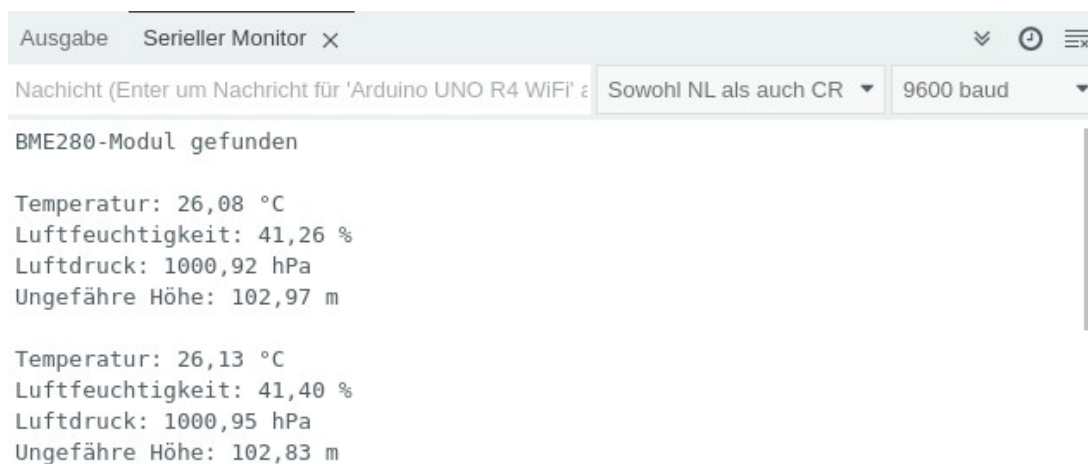
Funktionen der Bibliothek [Adafruit_BME280](#)

Schlüsselwort	Aktion
readTemperature()	Temperatur messen
readHumidity()	Luftfeuchtigkeit messen
readPressure()	Luftdruck messen
readAltitude(LuftdruckMeeresHoehe)	Höhe ermitteln

Der Luftdruck auf Meereshöhe (0 m) beträgt 1013,25 hPa (Hekto-Pascal)

Das Programm

So sieht es aus:



The screenshot shows the Arduino IDE Serial Monitor window. The title bar indicates 'Ausgabe' and 'Serieller Monitor'. The dropdown menu shows 'Nachricht (Enter um Nachricht für 'Arduino UNO R4 WiFi' zu senden)' and 'Sowohl NL als auch CR'. The baud rate is set to '9600 baud'. The output shows the following data:

```
BME280-Modul gefunden

Temperatur: 26,08 °C
Luftfeuchtigkeit: 41,26 %
Luftdruck: 1000,92 hPa
Ungefähre Höhe: 102,97 m

Temperatur: 26,13 °C
Luftfeuchtigkeit: 41,40 %
Luftdruck: 1000,95 hPa
Ungefähre Höhe: 102,83 m
```

```
#include "Wire.h"
#include "SPI.h"
#include "Adafruit_Sensor.h"
#include "Adafruit_BME280.h"

// Luftdruck auf Meereshöhe
#define LuftdruckMeeresHoehe (1013.25)

// Name des Sensors
Adafruit_BME280 bme;

int Wartezeit = 5000;

void setup()
{
  Serial.begin(9600);
  while (!Serial);
  delay(1000);

  // I2C über SCL/SDA oder A5/A4
  // if (!bme.begin())
```

```
// I2C über QWIIC-Verbindung
// Adresse und Wire1-Bus
if (!bme.begin(0x77, &Wire1))
{
    Serial.println("Kein BME280-Modul gefunden");
    while (true);
}
else Serial.println("BME280-Modul gefunden");

Serial.println();
}

void loop()
{
    // Temperatur messen
    String Temperatur = String(bme.readTemperature());
    // . durch , ersetzen
    Temperatur.replace(".", ",");
    Serial.println("Temperatur: " + Temperatur + " °C");

    // Luftfeuchtigkeit messen
    String Luftfeuchtigkeit = String(bme.readHumidity());
    // . durch , ersetzen
    Luftfeuchtigkeit.replace(".", ",");
    Serial.println("Luftfeuchtigkeit: " + Luftfeuchtigkeit + " %");

    // Luftdruck messen
    String Luftdruck = String(bme.readPressure() / 100.0);
    // . durch , ersetzen
    Luftdruck.replace(".", ",");
    Serial.println("Luftdruck: " + Luftdruck + " hPa");

    // aus dem Luftdruck die ungefähre Höhe bestimmen
    String Hoehe = String(bme.readAltitude(LuftdruckMeeresHoehe));
    // . durch , ersetzen
    Hoehe.replace(".", ",");
    Serial.println("Ungefähre Höhe: " + Hoehe + " m");
    Serial.println();
    delay(Wartezeit);
}
```

Wetterstation im Browser anzeigen

Jetzt sollen die gemessenen Daten, Datum und Zeit in einem Browser angezeigt werden.

So sieht es aus:

Temperatur und Luftfeuchtigkeit messen

Letzte Messung: Dienstag, 18.07.2023 Uhrzeit: 20:45:33

Temperatur:

26,28 °C

Luftfeuchtigkeit:

40,58 %

Luftdruck:

1000,94 hPa

Ungefähre Höhe:

103,33 m

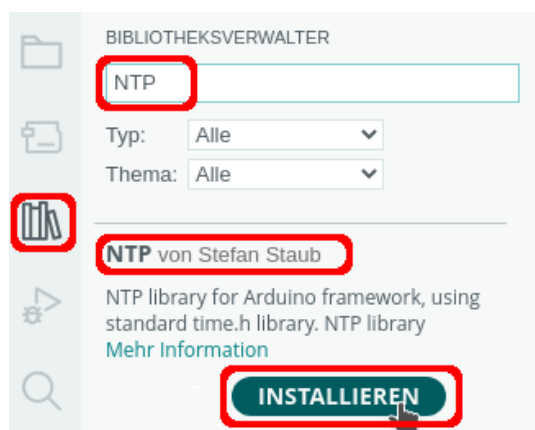
aktualisieren

Eigene IP: 192.168.1.162

IP Klient: 192.168.1.217

Das Programm

Zusätzlich benötigte Bibliothek



Im Seriellen Monitor wird die verwendete IP-Adresse angezeigt. Diese Adresse musst du in einem Browser deiner Wahl eingeben.



Bibliotheken und Variable

```
#include "Wire.h"
#include "SPI.h"
#include "Adafruit_Sensor.h"
#include "Adafruit_BME280.h"
#include "WiFiS3.h"
#include "NTP.h"

// Luftdruck auf Meereshöhe
#define LuftdruckMeeresHoehe (1013.25)

// Name des BME280
Adafruit_BME280 bme;

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

int Status = WL_IDLE_STATUS;
WiFiServer WiFiServer(80);

WiFiClient WiFiClient;
WiFiUDP wifiUdp;
NTP ntp(wifiUdp);
```

Der setup-Teil

```
void setup()
{
  Serial.begin(9600);
  while (!Serial);
  delay(1000);

  // Verbindung aufbauen
  if (WiFi.status() == WL_NO_MODULE)
  {
    Serial.println(F("Verbindungsaufbau gescheitert!"));
  }

  Serial.print("Verbindung aufbauen mit ");
  Serial.println(Router);
```

```
while (Status != WL_CONNECTED)
{
    Status = WiFi.begin(Router, Passwort);

    // Zeit für den Verbindungsaufbau
    // wenn die Verbindung nicht zustandekommt -> Zeit vergrößern
    delay(500);
}

// Webserver starten
WiFiServer.begin();

// IP des Servers/des verbundenen Computers anzeigen
Serial.print("Server: ");
Serial.println(WiFi.SSID());

// IP des Arduinos anzeigen
Serial.print("IP Adresse Arduino DHCP: ");
Serial.println(WiFi.localIP());

/*
    Zeitzone
    CEST: Central European Summertime
    Beginn europäische Sommerzeit letzter Sonntag im März 2 Uhr GMT + 2 Stunden
*/
ntp.ruleDST("CEST", Last, Sun, Mar, 2, 120);

// CET: Central European Time
// Beginn Normalzeit letzter Sonntag im Oktober 3 Uhr GMT + 1 Stunde
ntp.ruleSTD("CET", Last, Sun, Oct, 3, 60);

// ntp starten
ntp.begin();

// Zeit aktualisieren
ntp.update();

// Zeit mit formattedTime() anzeigen:
// %d = Tag, %m = Monat, %Y = Jahr, %T = Zeit in Stunden, Minuten, Sekunden
Serial.println(ntp.formattedTime("%d.%m.%Y Uhrzeit: %T"));

// I2C über SCL/SDA oder A5/A4
// if (!bme.begin())

// I2C über QWIIC-Verbindung
// Adresse und Wire1-Bus
if (!bme.begin(0x77, &Wire1))
{
    Serial.println("Kein BME280-Modul gefunden");
    while (true);
}
else Serial.println("BME280-Modul gefunden");

Serial.println();
}
```

Der loop-Teil

```
void loop()
{
    WiFiClient = WiFiServer.available();

    if (WiFiClient) {
        // Seite aufbauen wenn SeiteAufbauen true ist
        boolean SeiteAufbauen = true;

        while (WiFiClient.connected())
        {
            if (WiFiClient.available())
            {
                char Zeichen = WiFiClient.read();
                if (Zeichen == '\n' && SeiteAufbauen)
                {
                    // HTTP-Anforderung senden
                    WiFiClient.println("HTTP/1.1 200 OK");
                    WiFiClient.println("Content-Type: text/html");

                    // Leerzeile zwingend erforderlich
                    WiFiClient.println();

                    /*
                     HTML-Seite aufbauen
                     die folgenden Anweisungen müssen mit print oder println gesendet werden
                     println "verschönert" den Quelltext
                     " muss mit \" maskiert werden
                    */
                    WiFiClient.println("<!doctype html>");
                    WiFiClient.println("<html>");
                    WiFiClient.println("<body>");

                    // alle 60 Sekunden aktualisieren mit meta-Tag
                    WiFiClient.println("<meta http-equiv=\"refresh\" content=\"60\\>");
                    WiFiClient.println("<h1> Temperatur und Luftfeuchtigkeit messen</h1>");
                    WiFiClient.println("<hr />");
                    WiFiClient.print("<h2>Letzte Messung: ");

                    // formattedTime() zeigt Wochentage in englischer Sprache
                    // -> müssen einzeln abgefragt werden
                    switch (ntp.weekDay())
                    {
                        case 0:
                            WiFiClient.print("Sonntag");
                            break;
                        case 1:
                            WiFiClient.print("Montag");
                            break;
                        case 2:
                            WiFiClient.print("Dienstag");
                            break;
```

```

        case 3:
            WiFiClient.print("Mittwoch");
            break;
        case 4:
            WiFiClient.print("Donnerstag");
            break;
        case 5:
            WiFiClient.print("Freitag");
            break;
        case 6:
            WiFiClient.print("Samstag");
            break;
    }
    WiFiClient.print(", ");
    WiFiClient.print(ntp.formattedTime("%d.%m.%Y Uhrzeit: %T"));

    WiFiClient.println("</h2>");
    WiFiClient.println("<hr />");

    // Temperatur
    String Temperatur = String(bme.readTemperature());
    Temperatur.replace(".", ",");
    WiFiClient.print("<b>Temperatur:</b><blockquote>");
    WiFiClient.println(Temperatur + " &deg;C</blockquote>");
    WiFiClient.println("<br>");

    // Luftfeuchtigkeit
    String Luftfeuchtigkeit = String(bme.readHumidity());
    Luftfeuchtigkeit.replace(".", ",");
    WiFiClient.print("<b>Luftfeuchtigkeit:</b><blockquote>");
    WiFiClient.println(Luftfeuchtigkeit + " %</blockquote>");
    WiFiClient.println("<br>");

    // Luftdruck
    String Luftdruck = String(bme.readPressure() / 100);
    Luftdruck.replace(".", ",");
    WiFiClient.print("<b>Luftdruck:</b><blockquote>");
    WiFiClient.println(Luftdruck + " hPa</blockquote>");
    WiFiClient.println("<br>");

    // Höhe
    String Hoehe = String(bme.readAltitude(LuftdruckMeeresHoehe));
    Hoehe.replace(".", ",");
    WiFiClient.println("<b>Ungef&auml;hre H&ouml;he:</b><blockquote>");
    WiFiClient.println(Hoehe + " m</blockquote>");
    WiFiClient.println("<hr>");

    WiFiClient.println("<form>");

    // Button formatieren
    WiFiClient.print("<input style=\"font-size:16pt; font-weight:bold;");
    WiFiClient.print("background-color:#55A96B;");
    WiFiClient.print("display:block; cursor:pointer;\"type=\"button\"");
    WiFiClient.println(" onClick=\"location.href='WiFi.localIP()'\" value=\"aktualisieren\">");
    WiFiClient.println("</form>");

```

```
WiFiClient.println("<hr />");

// IPs anzeigen
WiFiClient.print(F("<b>Eigene IP: "));
WiFiClient.print(WiFiClient.remoteIP());
WiFiClient.print(F("</b>"));
WiFiClient.print(F("<br><b>IP Klient: "));
WiFiClient.print(WiFi.localIP());
WiFiClient.print(F("</b>"));
WiFiClient.println("</b>");
WiFiClient.println("</body>");
WiFiClient.print("</html>");

// HTTP-Antwort endet mit neuer Zeile
WiFiClient.println();

// Seite vollständig geladen -> loop verlassen
break;
}

if (Zeichen == '\n') SeiteAufbauen = true;
else if (Zeichen != '\r') SeiteAufbauen = false;
}
}
delay(1);

// Seitenaufbau stoppen
WiFiClient.stop();
}
}
```