

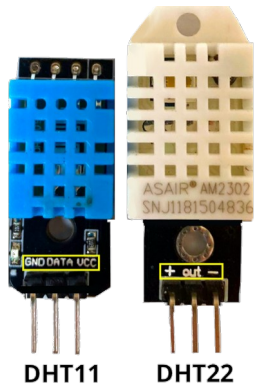
Inhaltsverzeichnis

Die Hardware.....	1
Die Schaltung.....	1
Vorbereitung.....	2
Benötigte Bibliothek.....	2
Wetterstation im Seriellen Monitor.....	3
Das Programm.....	3
Wetterstation im Browser anzeigen.....	4
Das Programm.....	4
Zusätzlich benötigte Bibliothek.....	4
Bibliotheken und Variable.....	5
Der setup-Teil.....	5
Der loop-Teil.....	6

Die Hardware

Der Sensor DHT11/DHT22 misst Temperatur und Luftfeuchtigkeit.

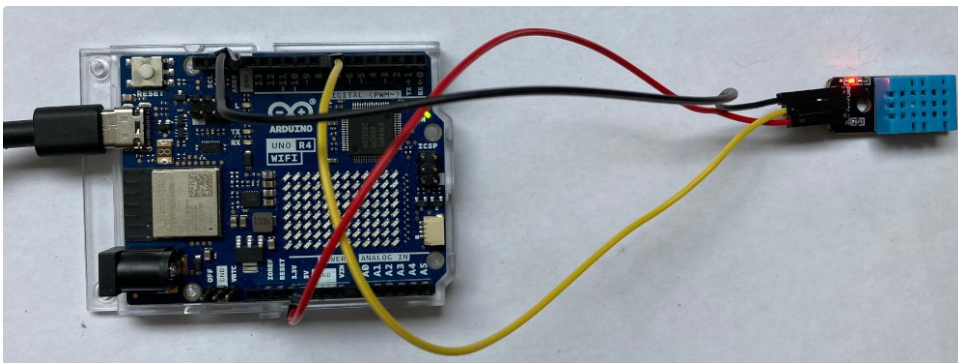
Beispiele für DHT11/DHT22 Sensoren



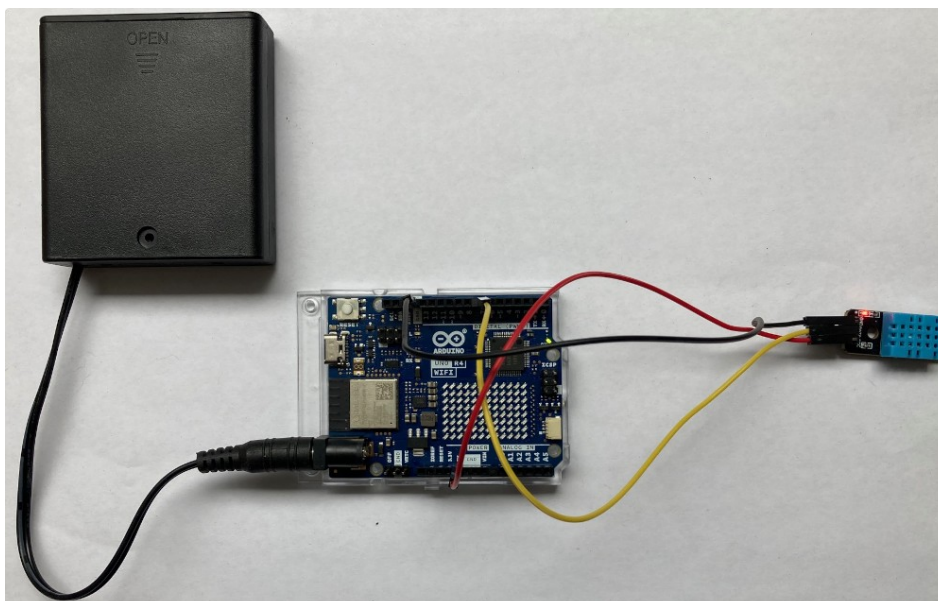
Die Pinbelegung kann sich von der hier gezeigten unterscheiden. Achte auf die Beschriftung auf dem Modul!

Schließe den Daten-Pin an den Pin 7 des Arduinos an.

Die Schaltung



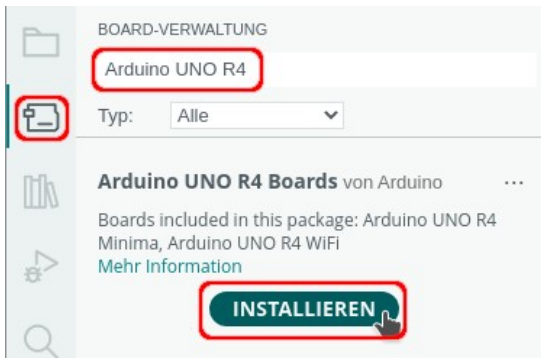
Arduino UNO R4 WiFi DHT11



Arduino UNO R4 WiFi DHT11 mit Batteriepack

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.

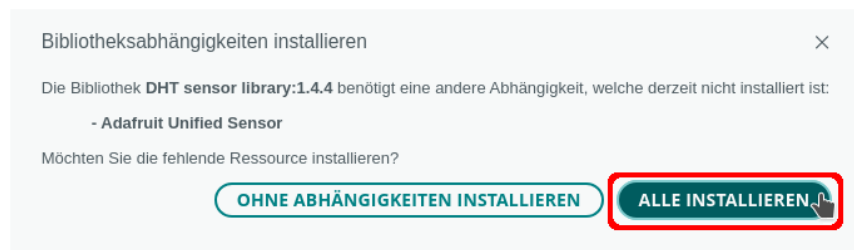
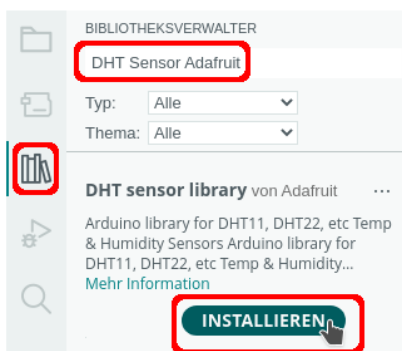


Linux



Windows

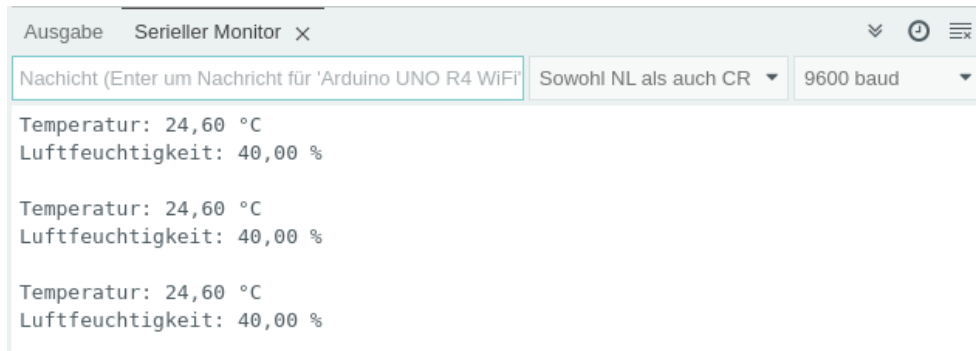
Benötigte Bibliothek



Wetterstation im Seriellen Monitor

Das Programm

So sieht es aus:



```
#include "DHT.h"

int Wartezeit = 5000;

// Pin des Sensors
int SENSOR_DHT = 7;

// DHT11
# define SensorTyp DHT11

// DHT22
// # define SensorTyp DHT22

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

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

    // Temperatursensor starten
    dht.begin();

    Serial.println();
}

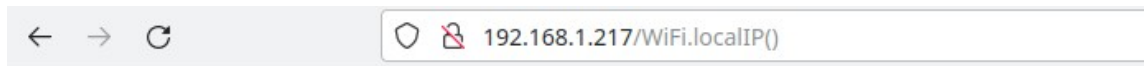
void loop()
{
    String Temperatur = String(dht.readTemperature());
    Temperatur.replace(".", ",");
    Serial.println("Temperatur: " + Temperatur + " °C");
    String Luftfeuchtigkeit = String(dht.readHumidity());
    Luftfeuchtigkeit.replace(".", ",");
    Serial.println("Luftfeuchtigkeit: " + Luftfeuchtigkeit + " %");
}
```

```
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: Freitag, 21.07.2023 Uhrzeit: 12:53:39

Temperatur:

24,70 °C

Luftfeuchtigkeit:

40,00 %

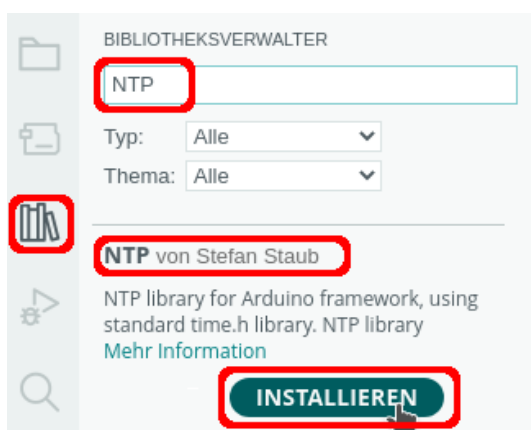
aktualisieren

Eigene IP: 192.168.1.162

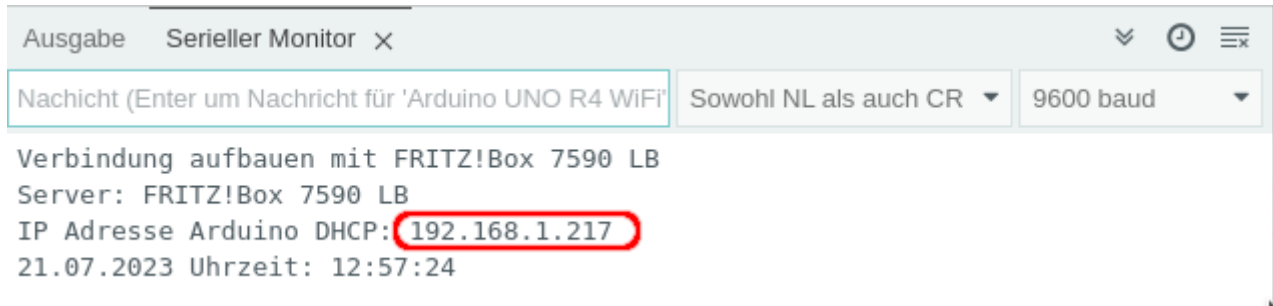
IP Klient: 192.168.1.217

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 "DHT.h"
#include "WiFiS3.h"
#include "NTP.h"

// Pin des Sensors
int SENSOR_DHT = 7;

// DHT11
# define SensorTyp DHT11

// DHT22
// # define SensorTyp DHT22

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

// 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 holen
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"));
dht.begin();
}
```

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

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

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);
WiFiClient.stop();
}
}
```