

DHT22 – Wetterstation Anzeige im Internetbrowser

Die mit dem Temperatursensor DHT22 gemessenen Werte sollen in einem Internetbrowser angezeigt werden.

So sieht es aus:

Temperatur und Luftfeuchtigkeit messen

Letzte Messung: Montag, 22.02.2021 15:48:52 Uhr

aktualisieren

Temperatur:

24,10 °C

Gespeicherte Werte:

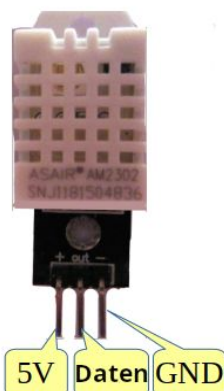
- 1: 15:48:26 Uhr ▶ 23,60 °C
- 2: 15:48:29 Uhr ▶ 23,80 °C
- 3: 15:48:32 Uhr ▶ 23,80 °C
- 4: 15:48:35 Uhr ▶ 23,80 °C
- 5: 15:48:38 Uhr ▶ 23,90 °C
- 6: 15:48:41 Uhr ▶ 23,90 °C
- 7: 15:48:43 Uhr ▶ 24,00 °C
- 8: 15:48:46 Uhr ▶ 24,00 °C
- 9: 15:48:49 Uhr ▶ 24,10 °C
- 10: 15:48:52 Uhr ▶ 24,10 °C

Luftfeuchtigkeit:

63,30 %

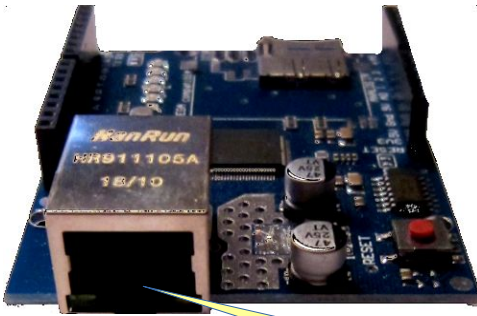
Eigene IP: 192.168.1.12

IP des Ethernet-Shields: 192.168.1.36



Der Sensor DHT22 misst die Temperatur und die Luftfeuchtigkeit. Das hier verwendete Bauteil besitzt drei Anschlüsse.

Die Version mit vier Pins muss in der Reihenfolge 5V → Daten (Pin) → leer → GND geschaltet werden.



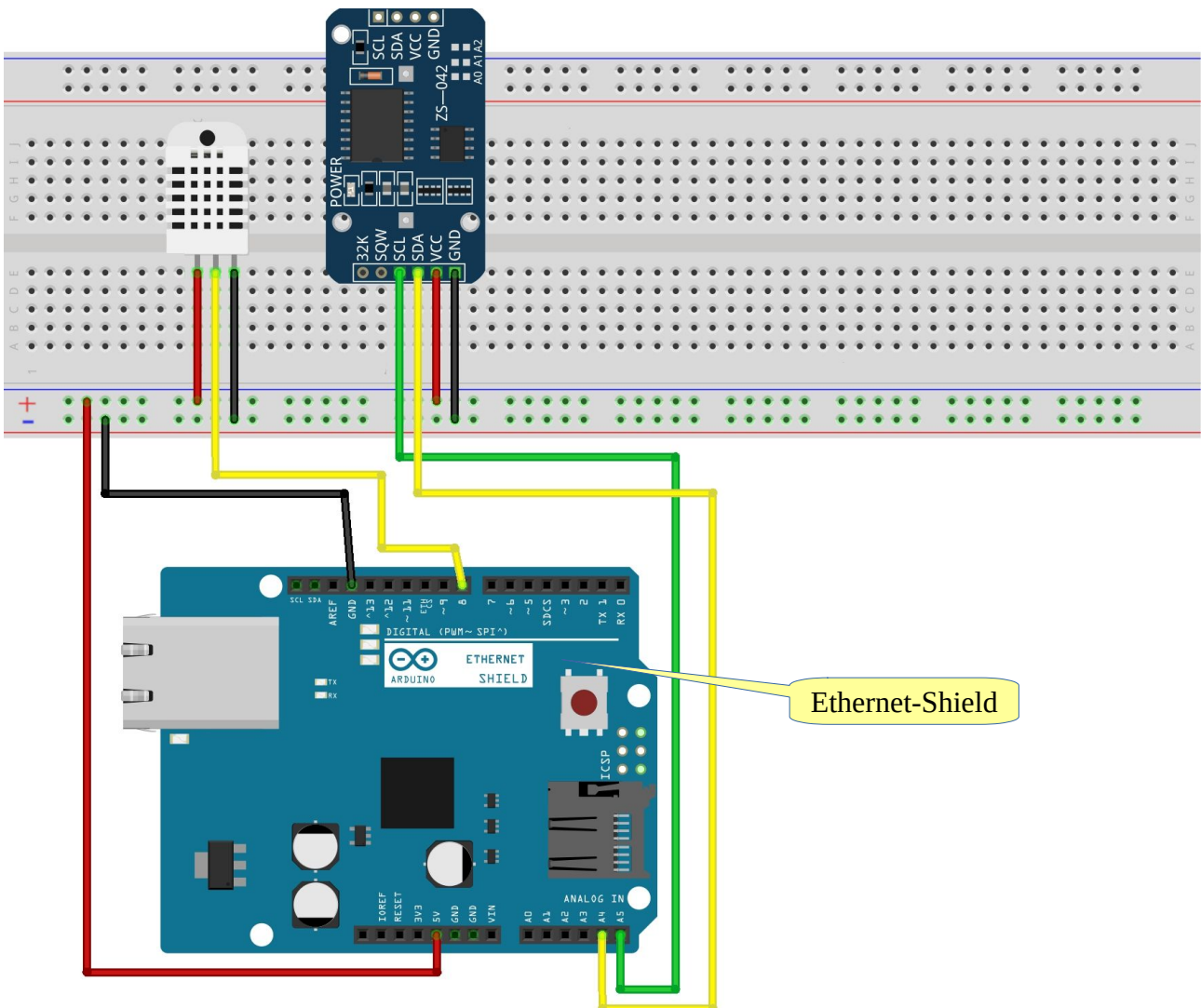
LAN-Anschluss

Für diese Aufgabe benötigst du ein sogenanntes „Shield“, eine Platine, die einfach auf den Arduino aufgesteckt wird. Auf ihr befindet sich ein LAN-Anschluss (RJ45).

Benötigte Bauteile:

- ➔ Temperatursensor DHT22
- ➔ Ethernet-Shield
- ➔ RTC-Modul DS 3231
- ➔ Leitungsdrähte

Baue die Schaltung auf:



Ethernet-Shield

Für das Programm brauchst du eine freie IP-Adresse und eine freie MAC-Adresse in deinem lokalen Netzwerk.

Im Regelfall befindet sich in einem lokalen Netzwerk ein DHCP-Server, der jedem Gerät im Netzwerk automatisch eine IP-Adresse zuteilt. Im Programm wird eine über DHCP vergebene Adresse verwendet.



Wenn die Temperaturmessung über einen längeren Zeitraum (mehrere Tage) eingesetzt werden soll, empfiehlt sich die Vergabe einer festen IP-Adresse, weil nach der sogenannten „leasetime“ die IP wieder neu vergeben wird.

Die MAC-Adresse ist die Hardware-Adresse jeder einzelnen Netzwerkschnittstelle (LAN oder WLAN), mit der jedes Gerät im Netzwerk eindeutig identifiziert werden kann. Sie besteht aus sechs Bytes in hexadezimaler Schreibweise, die durch „:“ oder „-“ getrennt werden.

Du kannst die im Programm verwendete „erfundene“ MAC-Adresse übernehmen: Die Gefahr, dass sich ein Gerät mit der gleichen MAC-Adresse im Netzwerk befindet, ist äußerst gering.

Benötigte Bibliotheken:

Sketch → Bibliothek einbinden → Bibliotheken verwalten



Binde die benötigten Bibliotheken ein:

```
# include <Ethernet.h>
# include <SimpleDHT.h>
# include <RTCLib.h>
```

```
// Name des RTC-Moduls
RTC_DS3231 rtc;

int SENSOR_DHT22 = 8;

// Namen des Sensors (dht22)
SimpleDHT22 dht22(SENSOR_DHT22);

// MAC-Adresse und IP definieren
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};

// festeIP = false -> IP-Adresse über DHCP vergeben
bool festeIP = false;

// feste IP
IPAddress ip(192, 168, 1, 200);

// Name des Servers vergeben
EthernetServer Server(80);

byte Zaehler = 0;
const byte AnzahlDaten = 10;

// Arrays für die gespeicherten Werte
String gespeicherteTemperatur[AnzahlDaten];
String gespeichertesDatum[AnzahlDaten];
```

Der setup-Teil. Beachte die Kommentare.

```
void setup()
{
  // RTC-Modul starten
  rtc.begin();

  /*
   wenn Datum und Zeit nicht korrekt -> Datum/Zeit setzen
   Jahr, Monat, Tag, Stunde, Minute, Sekunde
   keine führende 0 setzen
   Beispiel:
   rtc.adjust(DateTime(2018, 10, 25, 7, 2, 30));
  */
  // rtc.adjust(DateTime(2018, 12, 29, 21, 58, 30));

  Serial.begin(9600);

  // Ethernet starten feste IP
  if (festeIP) Ethernet.begin(mac, ip);

  // Ethernet starten DHCP
  else Ethernet.begin(mac);
```

```

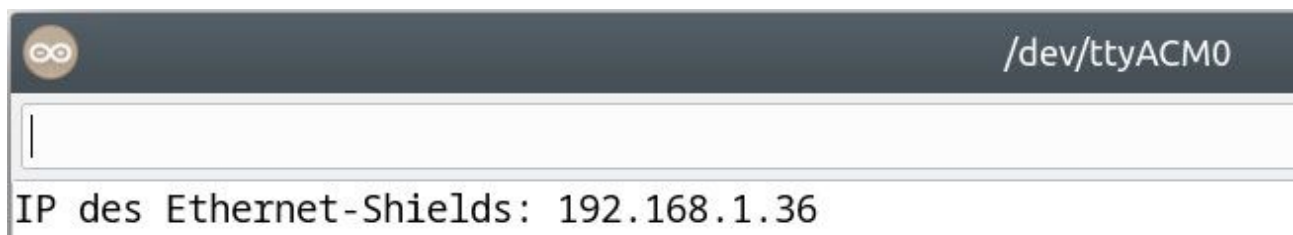
// Server starten
Server.begin();

Serial.begin(9600);

// zur Kontrolle IP-Adresse anzeigen
// localIP -> Adresse, die im Browser eingegeben wird
Serial.print(F("IP des Ethernet-Shields: "));
Serial.println(Ethernet.localIP());
}

```

Im Seriellen Monitor wird die IP des Ethernet-Shields angezeigt. Diese Adresse wird in einem Browser eingegeben.



Der loop-Teil. Beachte die Kommentare.

```

void loop()
{
  DateTime aktuell = rtc.now();
  float Temperatur;
  float Luftfeuchtigkeit;
  String Nummer;

  // auf Clienten warten ...
  EthernetClient Client = Server.available();

  // neue Anfrage
  if (Client)
  {
    // solange der Client verbunden ist ...
    while (Client.connected())
    {
      if (Client.available())
      {
        char Zeichen = Client.read();

        // \n = Seite vom Clienten vollständig geladen
        if (Zeichen == '\n')
        {
          // HTTP-Anforderung senden
          Client.println(F("HTTP/1.1 200 OK"));
        }
      }
    }
  }
}

```

```

Client.println(F("Content-Type: text/html"));

// Leerzeile zwingend erforderlich
Client.println();

/*
  HTML-Seite aufbauen
  die folgenden Anweisungen müssen
  mit print oder println gesendet werden
  println "verschönert" den Quelltext
  (jede Anweisung in einer eigenen Zeile)
  " innerhalb einer HTML-Anweisung muss mit \" maskiert werden
*/
Client.println(F("<!doctype html>"));
Client.println(F("<html>"));
Client.println(F("<body>"));

// alle 60 Sekunden aktualisieren mit meta-Tag
Client.println(F("<meta http-equiv=\"refresh\" content=\"60\">"));

Client.println(F("<h1> Temperatur und Luftfeuchtigkeit messen"));
Client.println(F("</h1>"));
Client.println(F("<hr />"));
Client.print(F("<h2>Letzte Messung: "));
/*
  Wochentag anzeigen
  0 = Sonntag
  1 = Montag
  ...
  6 = Samstag
*/
switch (aktuell.dayOfTheWeek())
{
  case 0:
    Client.print(F("Sonntag"));
    break;
  case 1:
    Client.print(F("Montag"));
    break;
  case 2:
    Client.print(F("Dienstag"));
    break;
  case 3:
    Client.print(F("Mittwoch"));
    break;
  case 4:
    Client.print(F("Donnerstag"));
    break;
  case 5:
    Client.print(F("Freitag"));
    break;
}

```

```
        case 6:
            Client.print(F("Samstag"));
            break;
    }
    Client.print(", ");

    // Datum und Zeit schreiben
    char Datum[] = "DD.MM.YYYY ";
    char Zeit[] = "hh:mm:ss Uhr";

    Client.print(aktuell.toString(Datum));
    Client.print(F(" "));
    Client.print(aktuell.toString(Zeit));
    Client.print(F("</h2><hr> "));

    // Button formatieren
    Client.print(F("<input style=\"font-size:16pt; font-weight:bold;\""));
    Client.print(F("background-color:#55A96B;\""));
    Client.print(F("display:block; cursor:pointer;\"type=\"button\""));
    Client.println(F("onClick=\"history.go(0)\""));
    Client.println(F("value=\"aktualisieren\""));
    Client.println(F("</form>"));
    Client.println(F("<hr />"));
    gespeichertesDatum[Zaehler] = aktuell.toString(Zeit);

    dht22.read2(&Temperatur, &Luftfeuchtigkeit, NULL);

    // in Strings umwandeln, . durch , ersetzen
    String AnzeigeTemperatur = String(Temperatur);
    AnzeigeTemperatur.replace(".", ",");

    // Temperatur speichern
    gespeicherteTemperatur[Zaehler] = AnzeigeTemperatur;

    // Luftfeuchtigkeit lesen
    String AnzeigeLuftfeuchtigkeit = String(Luftfeuchtigkeit);
    AnzeigeLuftfeuchtigkeit.replace(".", ",");

    Client.print(F("<b>Temperatur: <br>"));
    Client.println(AnzeigeTemperatur + " &deg;C</b>");
    Client.println(F("<br>"));
    Client.println(F("<hr>"));
    Client.println(F("<b>Gespeicherte Werte:<br>"));
    for (int i = 0; i <= Zaehler; i++)
    {
        Client.println("<b>" + String(i + 1) + ": ");
        Client.println(gespeichertesDatum[i]);
        Client.println(" &#9654; " + gespeicherteTemperatur[i]);
        Client.println(F(" &deg;C"));
        Client.println(F("</b><br>"));
    }
}
```

```

Client.println(F("<hr>"));
Client.print(F("<br><b>Luftfeuchtigkeit: <br>"));
Client.println(AnzeigeLuftfeuchtigkeit + " %</b><hr>");
Client.println(F("<form>"));

Client.print(F("<b>Eigene IP: "));
Client.print(Client.remoteIP());
Client.print(F("</b>"));
Client.print(F("<br><b>IP des Ethernet-Shields: "));
Client.print(Ethernet.localIP());
Client.print(F("</b>"));
Client.println(F("</body>"));
Client.print(F("</html>"));

// Zeit, um die Antwort zu übertragen
delay(1);

// Verbindung beenden
Client.stop();

// Zaehler erhöhen und bei > AnzahlDaten zurücksetzen
if (Zaehler < AnzahlDaten) Zaehler ++;
if (Zaehler >= AnzahlDaten) Zaehler = 0;
    }
  }
}
}
}
}
}

```



Wenn die Seite mit Aufruf der IP nicht angezeigt wird, versuche ein http:// davor zu setzen.