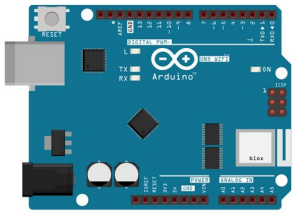
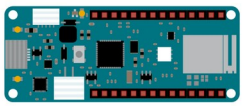
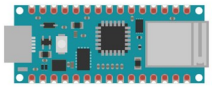
DHT22 – Wetterstation Anzeige im Internetbrowser mit WiFi

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

Für diese Anleitung benötigst du einen Arduino mit WiFi.


Arduino WiFi Rev 2

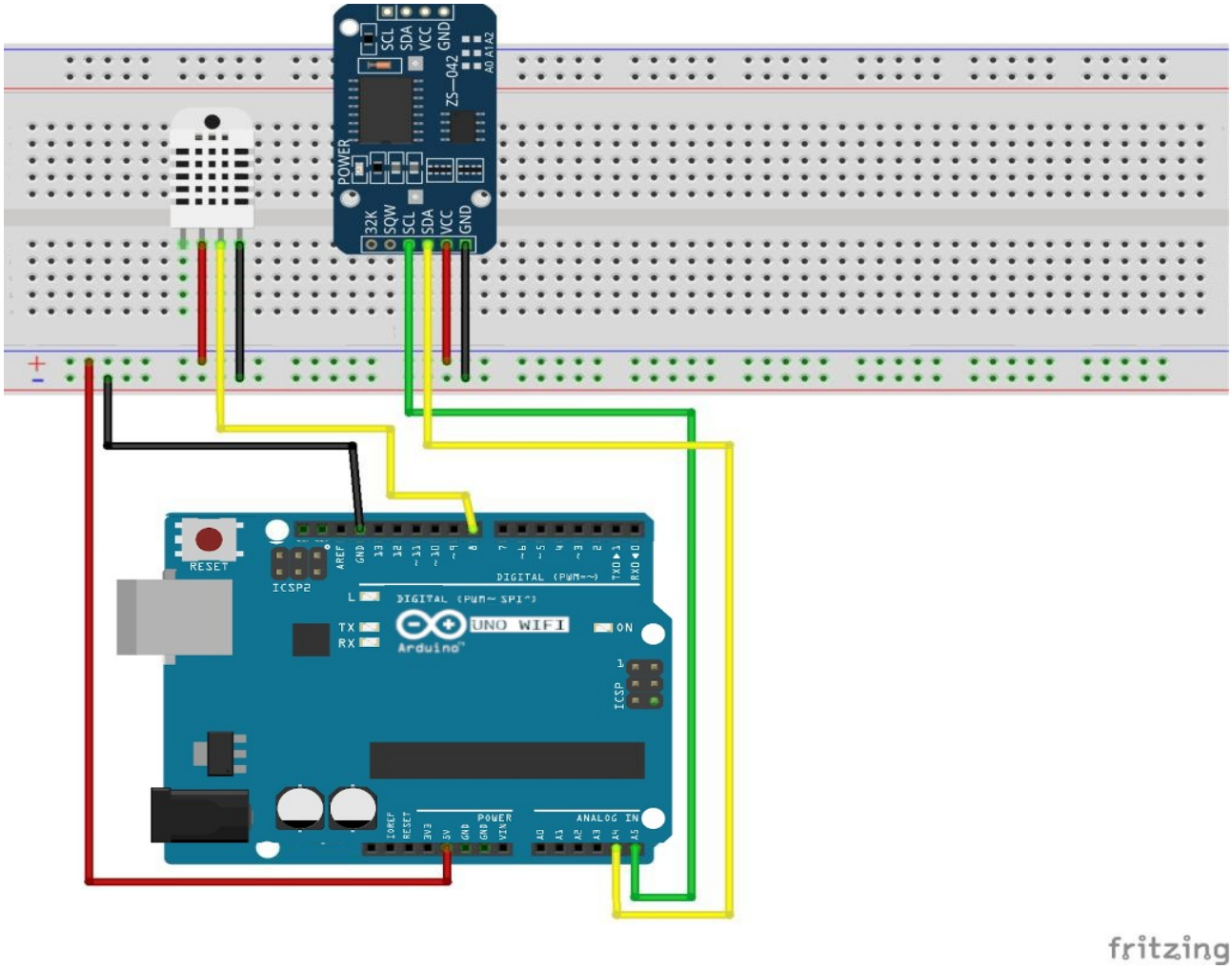

Arduino MKR WiFi 1010


Arduino Nano 33 IoT

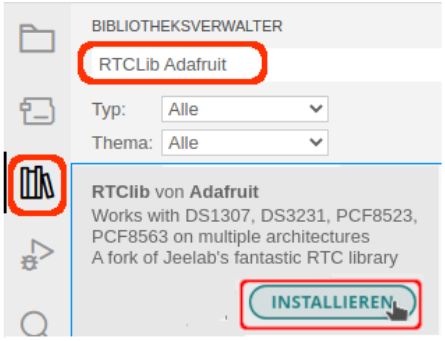
So sieht es aus:

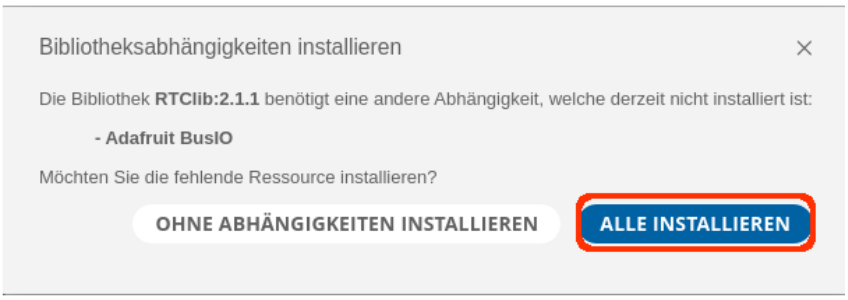
Benötigte Bauteile:

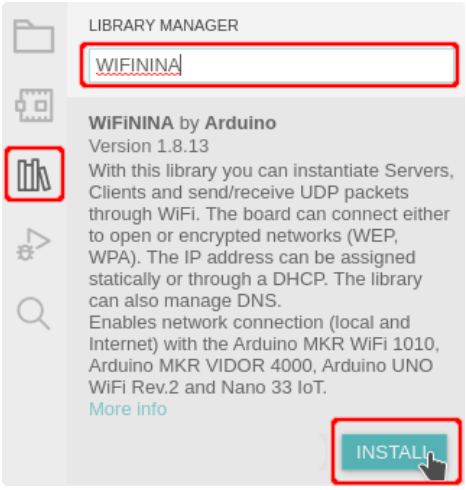
- Temperatursensor DHT11 oder DHT22
- RTC-Modul DS 3231
- Leitungsdrähte



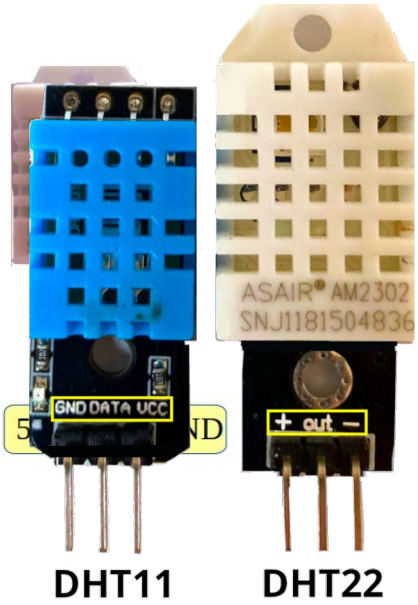
Benötigte Bibliotheken installieren:





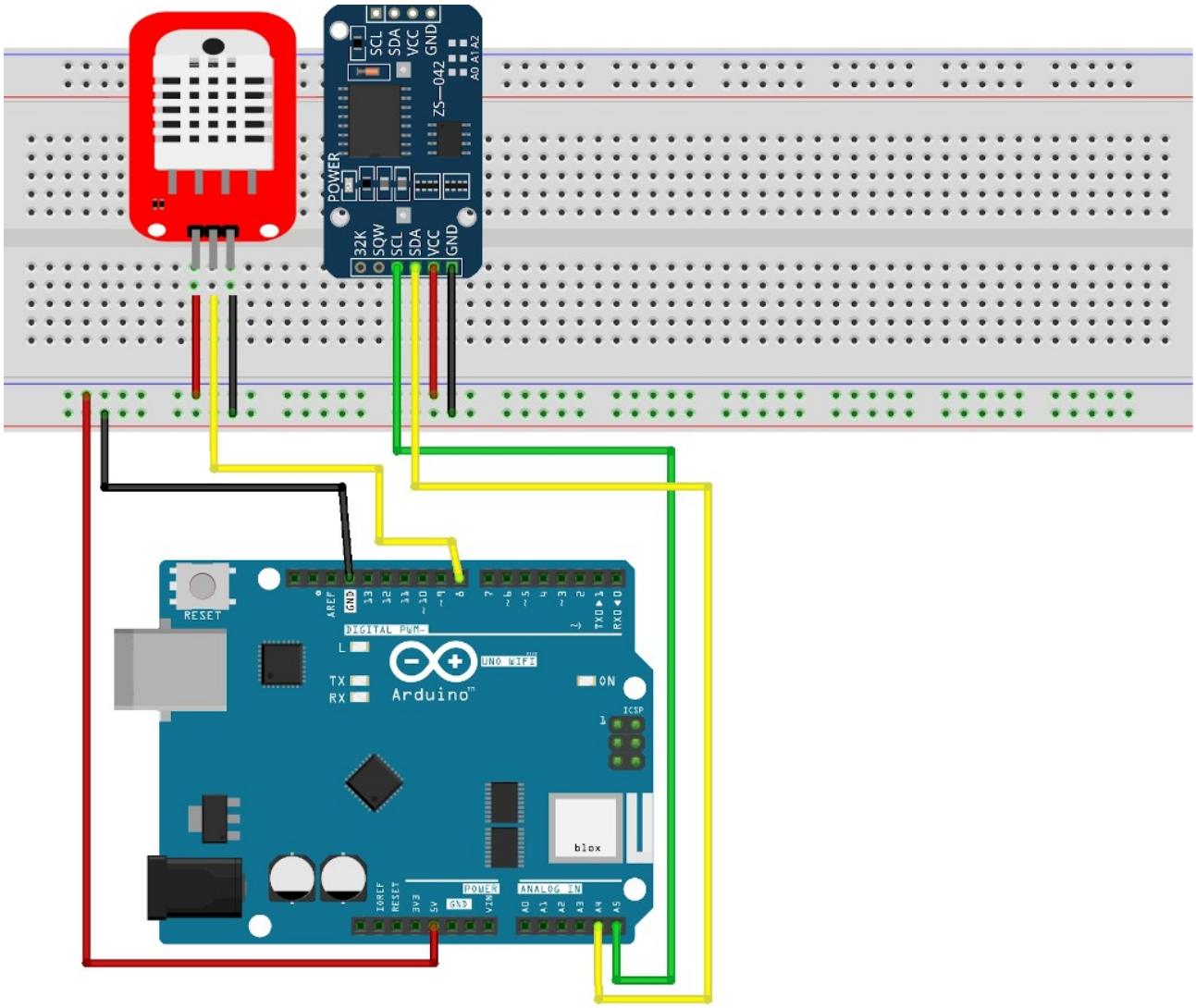


Beispiele für DHT11/DHT22 Sensoren



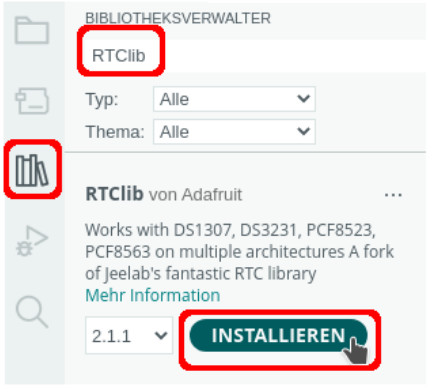
Die Pinbelegung kann sich von der hier gezeigten unterscheiden. Achte auf die Beschriftung auf dem Modul!

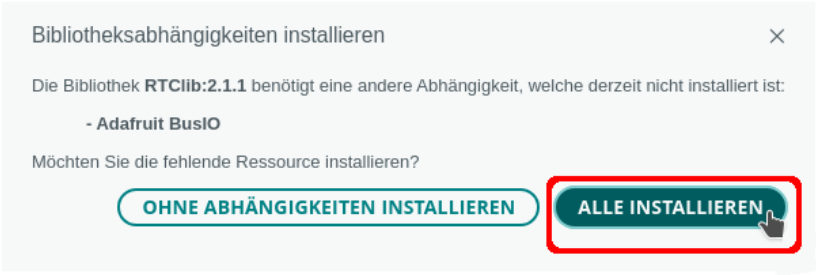
Baue die Schaltung auf:

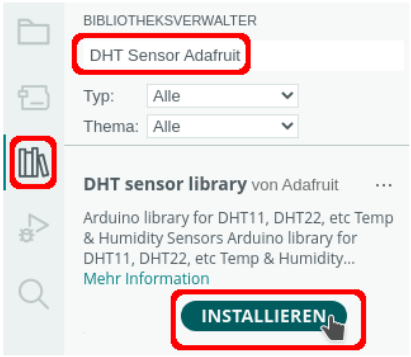


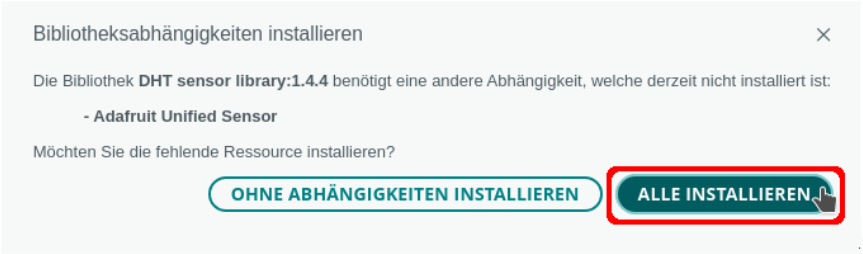
Für das Programm brauchst du eine freie IP-Adresse und eine freie MAC-Adresse in deinem lokalen Netzwerk.
Im Regelfall befindet sich in einem lokalen Netzwerk ein DHCP-Server, der jedem Gerät im Netzwerk automatisch eine IP-Adresse zuteilt. Im Programm wird eine über DHCP vergebene Adresse verwendet.

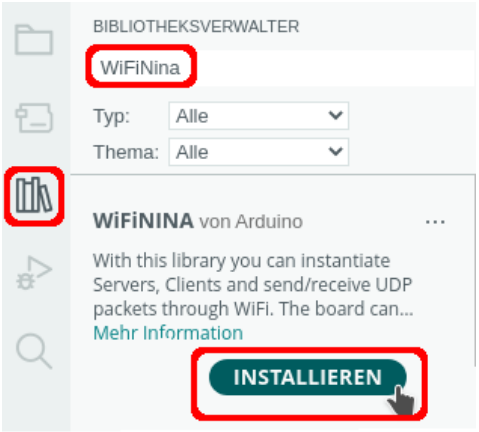
Benötigte Bibliotheken:











Binde die benötigte Bibliotheken ein und definiere die Variablen.

```
# include <DHT.h>
# include <RTClib.h>
# include <WiFiNINA.h>

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

```
// DHT11
// # define DHTTYPE DHT11

// DHT22
# define DHTTYPE DHT22
# define DHTPIN 8
DHT dht(DHTPIN, DHTTYPE);

// Router: Name des Routers
// Passwort: WLAN-Passwort
char Router[] = "FRITZ!Box 7590 LB";
char Passwort[] = "xxxxxxx";

// Netzwerkstatus
int status = WL_IDLE_STATUS;
WiFiServer server(80);
WiFiClient client = server.available();
```

Der setup-Teil:

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

  dht.begin();

  Serial.begin(9600);

  // 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);
    delay(2000);
  }

  server.begin();
  Serial.print("Router: ");
  Serial.println(WiFi.SSID());

  IPAddress ip = WiFi.localIP();
  Serial.print("IP Adresse Arduino: ");
  Serial.println(ip);

  DateTime aktuell = rtc.now();

  // Datum und Zeit holen
  char Datum[] = "DD.MM.YYYY ";
```

```
  char Zeit[] = "hh:mm:ss Uhr";

  Serial.print(aktuell.toString(Datum));
  Serial.println(aktuell.toString(Zeit));
}
```

Im Seriellen Monitor werden die IP des Arduinos und das aktuelle Datum und die aktuelle Zeit angezeigt. Diese Adresse wird in einem Browser eingegeben.

Da die IP-Adresse über DHCP bergeben wird, kann sie nach einem erneuten Hochladen anders sein.

Der loop-Teil. Beachte die Kommentare.

```
void loop()
{
  float Temperatur;
  float Luftfeuchtigkeit;
  String Nummer;
  String AnzeigeTemperatur;
  String AnzeigeLuftfeuchtigkeit;

  client = server.available();
  if (client) {
    String SchaltungLesen = "";
    while (client.connected())
    {
      if (client.available())
      {
        char Zeichen = client.read();
        if (Zeichen == '\n')
        {
          if (SchaltungLesen.length() == 0)
          {
            // HTTP-Anforderung senden
            client.println("HTTP/1.1 200 OK");
            client.println("Content-Type: text/html");

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

            // Leerzeile zwingend erforderlich
            client.println();
```

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

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

client.println("<h1> Temperatur und Luftfeuchtigkeit messen</h1>");
client.println("<hr />");

// Daten lesen
AnzeigeTemperatur = String(dht.readTemperature());
AnzeigeTemperatur.replace(".", ",");

AnzeigeLuftfeuchtigkeit = String(dht.readHumidity());
AnzeigeLuftfeuchtigkeit.replace(".", ",");
client.print("<h2>Letzte Messung: ");
DateTime aktuell = rtc.now();
char Datum[] = "DD.MM.YYYY ";
char Zeit[] = "hh:mm:ss Uhr";

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(", ");
client.print(aktuell.toString(Datum));
client.println(aktuell.toString(Zeit));
client.println("</h2>");
```

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

            client.print("<b>Temperatur:</b><blockquote>");
            client.println(AnzeigeTemperatur + " &deg;C</blockquote>");
            client.println("<br>");

            client.print("<b>Luftfeuchtigkeit:</b><blockquote>");
            client.println(AnzeigeLuftfeuchtigkeit + " %</blockquote><hr>");
            client.println("<form>");

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

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

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

            // Seite vollständig geladen -> loop verlassen
            break;
          }
          else
          {
            SchaltungLesen = "";
          }
        }

        // bei einem anderen Zeichen als return (\r)
        // -> Zeichen dem String SchaltungLesen hinzufügen
        else if (Zeichen != '\r')
        {
          SchaltungLesen += Zeichen;
        }
      }
    }
    client.stop();
  }
}
```