

Die mit dem Temperatursensor DHT11/DHT22 gemessene Temperatur soll mit aktuellem Datum und aktueller Zeit auf eine SD-Karte gespeichert werden.

So soll es aussehen:

```

Initialisiere SD-Karte
Initialisierung abgeschlossen
10.07.2022 Uhrzeit: 11:46:08
Temperatur: 22,40
Luftfeuchtigkeit in %: 60,80
Schreibe Messdaten in die Datei Messung.csv ...
-----
Abgeschlossen.

10.07.2022 Uhrzeit: 11:46:38
Temperatur: 22,40
Luftfeuchtigkeit in %: 60,50
Schreibe Messdaten in die Datei Messung.csv ...
-----
Abgeschlossen.

10.07.2022 Uhrzeit: 11:47:08
Temperatur: 22,50
Luftfeuchtigkeit in %: 60,60
Schreibe Messdaten in die Datei Messung.csv ...
-----
Abgeschlossen.

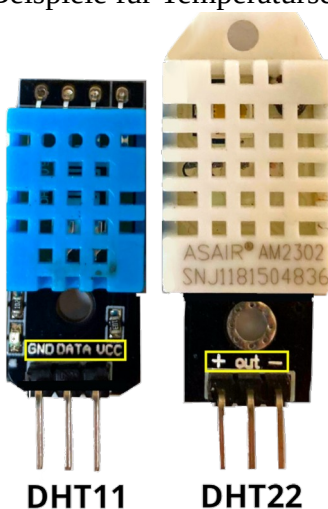
10.07.2022 Uhrzeit: 11:47:38
Temperatur: 23,10
Luftfeuchtigkeit in %: 66,00
Schreibe Messdaten in die Datei Messung.csv ...
-----
Abgeschlossen.
    
```

	A	B	C	D
1	Datum	Zeit	Temperatur in °C	Luftfeuchtigkeit in %
2	10.07.2022	Uhrzeit: 11:45:52	22,4	59,9
3				
4	Datum	Zeit	Temperatur in °C	Luftfeuchtigkeit in %
5	10.07.2022	Uhrzeit: 11:45:57	22,4	61
6				
7	Datum	Zeit	Temperatur in °C	Luftfeuchtigkeit in %
8	10.07.2022	Uhrzeit: 11:46:08	22,4	60,8
9				
10	Datum	Zeit	Temperatur in °C	Luftfeuchtigkeit in %
11	10.07.2022	Uhrzeit: 11:46:38	22,4	60,5
12				
13	Datum	Zeit	Temperatur in °C	Luftfeuchtigkeit in %
14	10.07.2022	Uhrzeit: 11:47:08	22,5	60,6
15				
16	Datum	Zeit	Temperatur in °C	Luftfeuchtigkeit in %
17	10.07.2022	Uhrzeit: 11:47:38	23,1	66
18				
19	Datum	Zeit	Temperatur in °C	Luftfeuchtigkeit in %
20	10.07.2022	Uhrzeit: 11:48:08	23,2	60,9
21				

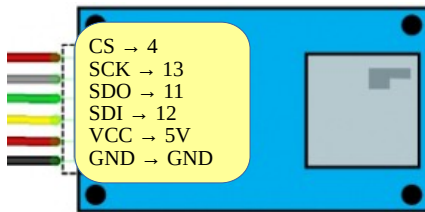
Benötigte Bauteile:

- ➔ RTC-Modul DS3231
- ➔ Temperatur-/Feuchtigkeitssensor DHT11/DHT22
- ➔ SD-Kartenleser (Datenlogger-Shield oder Ethernet-Shield)
- ➔ Leitungsdrähte

Beispiele für Temperatursensoren

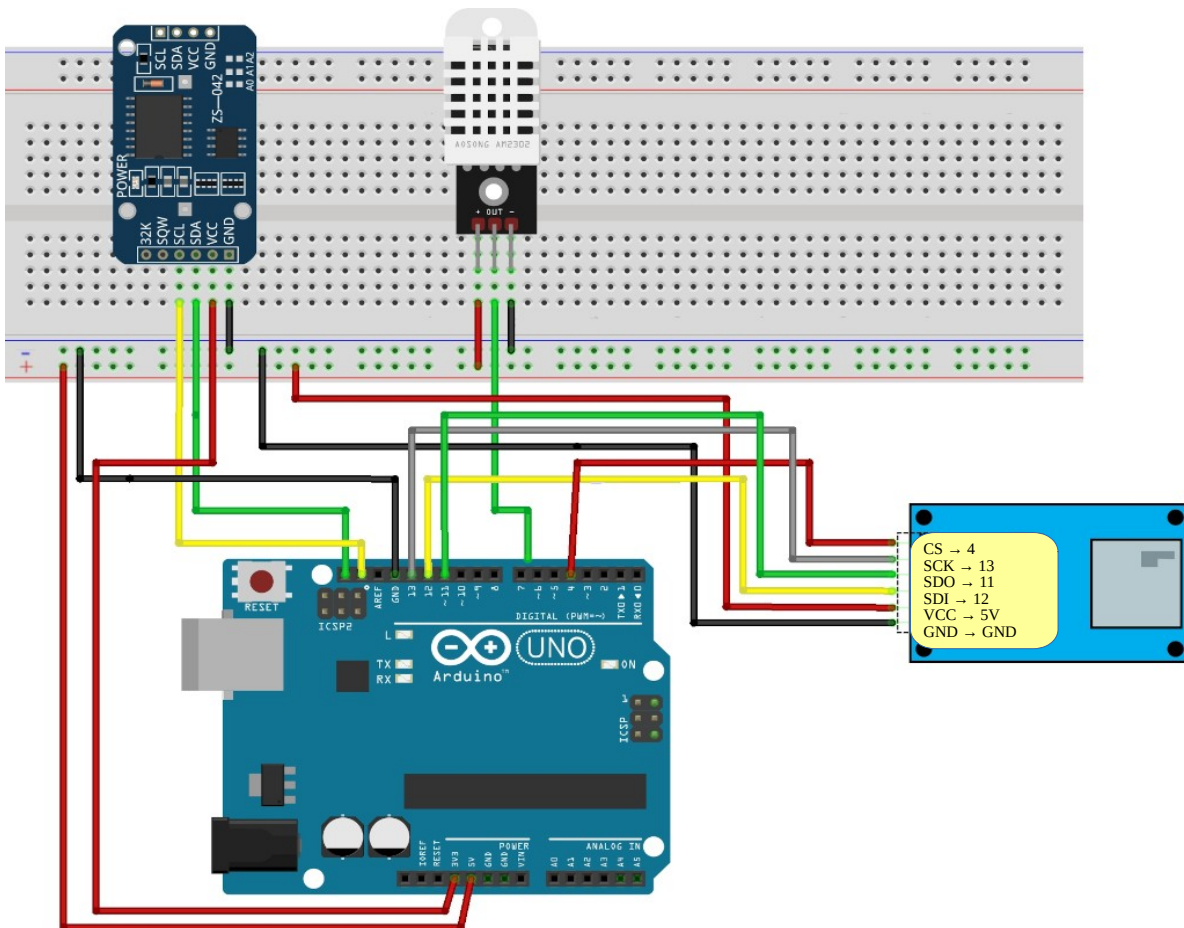


Die Pinbelegung kann sich von der hier gezeigten unterscheiden. Achte auf die Beschriftung auf dem Modul!



Achte auf die Pin-Belegung des SD-Kartenlesers!
Die SD-Karte muss mit FAT32 formatiert sein!
SDO = Serial Data Out
SDI = Serial Data In

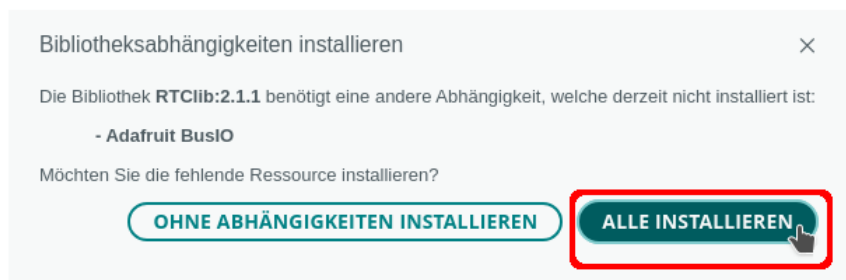
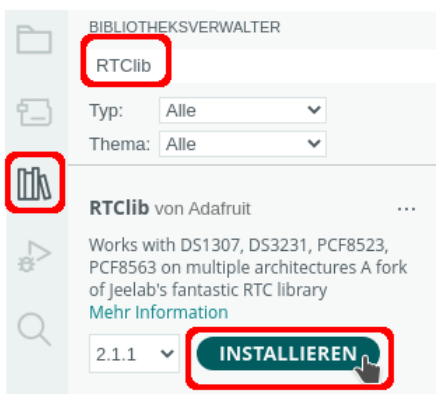
Baue die Schaltung auf.

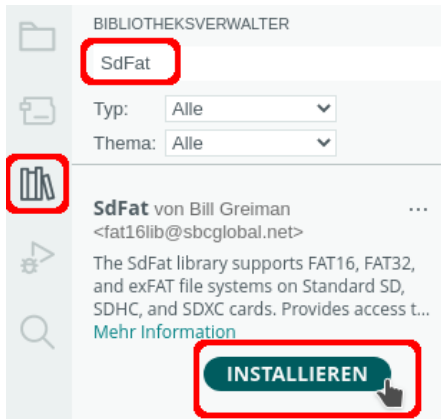
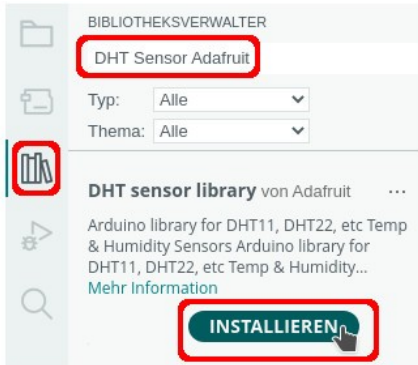


fritzing

Benötigte Bibliotheken:

Sketch → **Bibliothek einbinden** → **Bibliotheken verwalten**





Im Kopf des Programms werden die benötigten Bibliotheken eingebunden und die Variablen definiert. Beachte die Kommentare:

```
#include "SdFat.h"
#include "DHT.h"

// Bezeichnung der SD-Karte
SdFat SD;

// Pin des DHT-Sensors
int SENSOR_DHT = 8;

// Sensortyp festlegen
// DHT22 oder DHT11
#define SensorTyp DHT11

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

// Bibliothek für das RTC-Modul
# include <RTClib.h>
RTC_DS3231 rtc;

// Bezeichnung der Textdatei
File Temperaturmessung;
```

```
// Datenpin für das SD-Kartenmodul
// int DatenPin = 4;

// Datenpin für das Datenlogger-Shield
int DatenPin = 10;
```

Der setup-Teil initialisiert die SD-Karte, startet das RTC-Modul und legt den pinMode für den Sensor fest.

```
void setup()
{
  Serial.begin(9600);

  // auf serielle Verbindung warten
  while (!Serial) {;}
  Serial.println(F("Initialisiere SD-Karte"));

  if (!SD.begin(DatenPin))
  {
    Serial.println(F("Initialisierung fehlgeschlagen!"));
  }
  else Serial.println(F("Initialisierung abgeschlossen"));

  // rtc starten
  rtc.begin();
  /*
   wenn Datum und Zeit nicht korrekt → Datum/Zeit setzen
   Jahr, Monat, Tag, Stunde, Minute, Sekunde
   rtc.adjust(DateTime(2023, 5, 15, 10, 30, 30));
  */
  // Sensor DHT starten
  dht.begin();
}
```

Der loop-Teil: Beachte die Kommentare.

```
void loop()
{
  DateTime aktuell = rtc.now();

  // Daten lesen
  float Temperatur = dht.readTemperature();
  float Luftfeuchtigkeit = dht.readHumidity();

  // Funktion aufrufen
  ZeitAusgeben(aktuell);

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

```
String AnzeigeLuftfeuchtigkeit = String(Luftfeuchtigkeit);
AnzeigeLuftfeuchtigkeit.replace(".", ",");
Serial.print(F("Temperatur:\t\t"));
Serial.println(AnzeigeTemperatur);
Serial.print(F("Luftfeuchtigkeit in %:\t"));
Serial.println(AnzeigeLuftfeuchtigkeit);

/*
  in Datei schreiben
  O_CREAT -> Datei erstellen, wenn sie nicht existiert
  O_WRITE -> in die Datei schreiben
  O_AT_END -> Startposition zum Schreiben an das Ende der Datei setzen
*/
if (!Temperaturmessung.open("Messung.csv", O_CREAT | O_WRITE | O_AT_END))
{
  Serial.print(F("Datei kann nicht ge\u00f6ffnet werden!"));
}
// wenn die Datei ge\u00f6ffnet werden konnte ...
if (Temperaturmessung)
{
  Serial.println(F("Schreibe Messdaten in die Datei Messung.csv ... "));
  Serial.println(F("-----"));

  // \u00dcberschrift in Datei schreiben
  // Funktion aufrufen
  schreibeUeberschrift();
  Temperaturmessung.print(AktuellesDatum + ";");
  Temperaturmessung.print(AktuelleZeit + ";");
  Temperaturmessung.print(AnzeigeTemperatur + ";");
  Temperaturmessung.print(AnzeigeLuftfeuchtigkeit);
  Temperaturmessung.println();

  // Schreibfehler abfragen
  if (!Temperaturmessung.sync() || Temperaturmessung.getWriteError())
  {
    Serial.print(F("Schreibfehler!"));
  }
  else
  {
    // Datei schlie\u00dfen
    Temperaturmessung.close();
    Serial.println(F("Abgeschlossen."));
    Serial.println();
  }
}

// Zeit bis zur n\u00e4chsten Messung (eine Minute)
delay(60000);
}
```

Im loop-Teil werden zwei Methoden aufgerufen:

```
void ZeitAusgeben(DateTime aktuell)
{
  char Datum[] = "DD.MM.YYYY";
  AktuellesDatum = aktuell.toString(Datum);
  char Zeit[] = "Uhrzeit: hh:mm:ss";
  AktuelleZeit = aktuell.toString(Zeit);
  Serial.println(AktuellesDatum + " " + AktuelleZeit);
}

void schreibeUeberschrift()
{
  Temperaturmessung.println();
  Temperaturmessung.print(F("Datum"));

  // ; als Trennzeichen für die CSV-Datei
  Temperaturmessung.print(";");
  Temperaturmessung.print(F("Zeit"));
  Temperaturmessung.print(";");
  Temperaturmessung.print(F("Temperatur in °C"));
  Temperaturmessung.print(";");
  Temperaturmessung.print(F("Luftfeuchtigkeit in %"));
  Temperaturmessung.println();
}
```

Hartmut Waller (hartmut-waller.info/arduino-blog) Letzte Änderung: 19.02.25