

Das Programm soll die Messdaten des Sensors DHT22 sammeln, auf dem Smartphone anzeigen und auf der SD-Karte speichern.

Taste 1: Daten sammeln und anzeigen

Taste 2: Daten auf der SD-Karte im CSV-Format speichern

Taste 3: Datei öffnen und im Seriellen Monitor anzeigen

Taste 4: Datei löschen

CSV-Dateien (Comma Separated Value) trennen Datensätze nach bestimmten Zeichen. Das Komma kann nicht verwendet werden, weil es in der Anzeige von Temperatur und Luftfeuchtigkeit benutzt wird. Deshalb dient das Semikolon als Trennzeichen.

Trennoptionen

Feste Breite Getrennt

Tabulator Komma Semikolon Leerzeichen Andere

Feldtrenner zusammenfassen Leerräume beschneiden Zeichenfolgen-Trennerzeichen: " ▾

Weitere Optionen

Felder in Hochkommata als Text formatieren Erweiterte Zahlenerkennung

Formeln auswerten

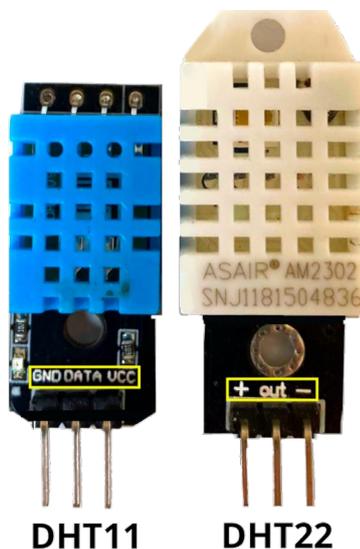
Feldbefehle

Spaltentyp: ▾

	Standard	Standard	Standard	Standard
1				
2	Datum	Zeit	Temperatur in C	Luftfeuchtigkeit in %
3	01.05.2023	08:27:55	18,60	41,00
4				
5	Datum	Zeit	Temperatur in C	Luftfeuchtigkeit in %
6	01.05.2023	08:27:58	18,60	41,00
7				
8	Datum	Zeit	Temperatur in C	Luftfeuchtigkeit in %
9	01.05.2023	08:28:02	18,70	42,00

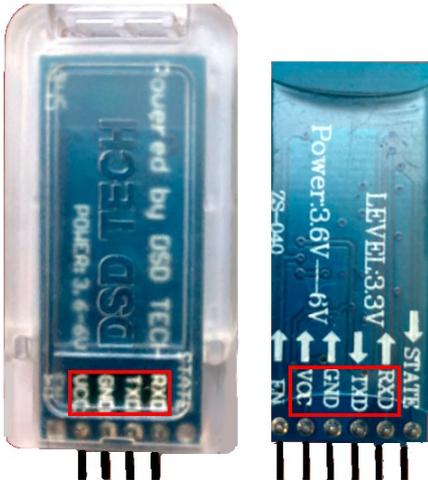
Hilfe Abbrechen

Beispiele für DHT11/DHT22 Sensoren



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

Beispiele für Bluetooth-Module HM-10

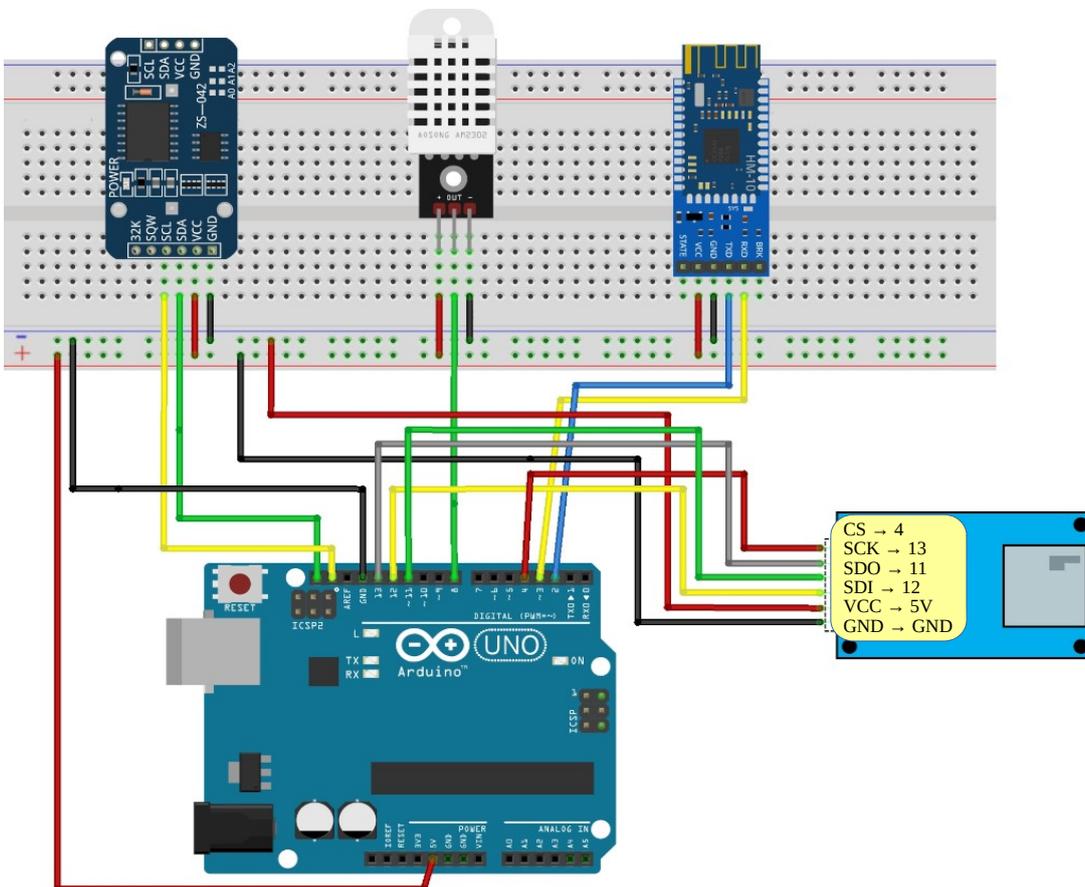


Das Bluetooth-Modul HM-10 arbeitet mit BLE (Bluetooth Low Energy) und kann mit Android-Betriebssystemen und iOS umgehen.

Benötigte Bauteile:

- ➔ RTC-Modul DS3231
- ➔ Temperatur-/Feuchtigkeitssensor DHT11/DHT22
- ➔ SD-Kartenleser
- ➔ Bluetooth-Modul HM10
- ➔ Leitungsdrähte

Baue die Schaltung auf.



So sieht es aus:

DSD-Tech Bluetooth

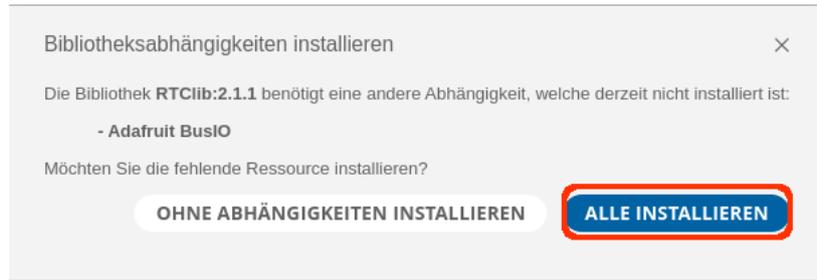
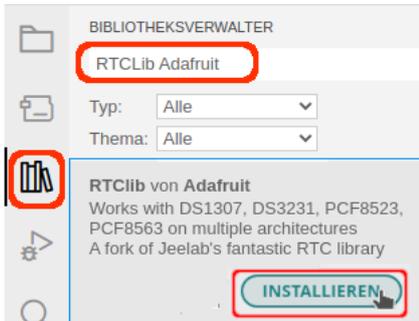
<p>Bluetooth-Modul scannen</p>	<p>Daten senden</p>

Arduino Bluetooth Controller

<p>Bluetooth-Modul scannen</p>	<p>Terminal auswählen</p>	<p>Einstellungen</p>
<p>Tastatur wählen</p>	<p>Daten senden</p>	

Benötigte Bibliotheken:

Sketch → Bibliothek einbinden → Bibliotheken verwalten



Binde die benötigten Bibliotheken ein und definiere die Variablen:

```
# include <SoftwareSerial.h>
# include <DHT.h>
# include <RTCLib.h>
# include <SdFat.h>

// Bezeichnung der SD-Karte
SdFat SD;

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

// Bezeichnung der Textdatei
File Temperaturmessung;

RTC_DS3231 rtc;

SoftwareSerial BTSerial(2, 3);
```

```
// über Bluetooth vom Smartphone gesendeter Wert
char btDaten;

// 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);

const String Trennzeichen = ";";
String AnzeigeTemperatur;
String AnzeigeLuftfeuchtigkeit;
```

Der setup-Teil. Beachte die Kommentare.

```
void setup()
{
  BTSerial.begin(9600);
  Serial.begin(9600);
  delay(500);

  rtc.begin();
  /*
   wenn Datum und Zeit nicht korrekt -> Datum/Zeit setzen
   Jahr, Monat, Tag, Stunde, Minute, Sekunde
   rtc.adjust(DateTime(2023, 3, 30, 10, 30, 30));
  */

  // SD-Kartenmodul starten
  if (!SD.begin(DatenPin)) Serial.println(F("Initialisierung fehlgeschlagen!"));
  else Serial.println(F("Initialisierung abgeschlossen"));

  // DHT-Sensor starten
  dht.begin();
}
```



Der loop-Teil. Beachte die Kommentare.



```

void loop()
{
  if (BTSerial.available())
  {
    // vom Smartphone gesendeten Wert lesen
    btDaten = BTSerial.read();

    if (btDaten == '1')
    {
      DateTime aktuell = rtc.now();
      char Datum[] = "DD.MM.YYYY ";
      char Zeit[] = "hh:mm:ss";

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

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

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

      // Daten in der App ausgeben
      BTSerial.println(aktuell.toString(Zeit));
      BTSerial.println(F("Temperatur: "));
      BTSerial.println(AnzeigeTemperatur + " °C");
      BTSerial.println("Luftfeuchtigkeit:");
      BTSerial.println(AnzeigeLuftfeuchtigkeit + "%");
      BTSerial.println(F("-----"));

      // Daten im Seriellen Monitor ausgeben
      Serial.print(aktuell.toString(Datum));
      Serial.println(aktuell.toString(Zeit));
      Serial.print(F("Temperatur: "));
      Serial.println(AnzeigeTemperatur + " C");
      Serial.print("Luftfeuchtigkeit:");
      Serial.println(AnzeigeLuftfeuchtigkeit);
      Serial.println(F("-----"));
    }

    // Daten in Datei Messung.csv schreiben
    if (btDaten == '2')
    {
      // Daten lesen
      float Temperatur = dht.readTemperature();
      float Luftfeuchtigkeit = dht.readHumidity();
    }
  }
}
    
```

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

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

/*
  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
*/
Temperaturmessung.open("Messung.csv", O_CREAT | O_WRITE | O_AT_END);

// wenn die Datei geöffnet werden konnte ...
if (Temperaturmessung)
{
  BTSerial.println(F("Schreibe Messdaten"));

  // Überschrift schreiben
  schreibeUeberschrift();

  DateTime aktuell = rtc.now();
  char Datum[] = "DD.MM.YYYY ";
  char Zeit[] = "hh:mm:ss";
  Temperaturmessung.print(aktuell.toString(Datum) + Trennzeichen);
  Temperaturmessung.print(aktuell.toString(Zeit) + Trennzeichen);
  Temperaturmessung.print(AnzeigeTemperatur + Trennzeichen);
  Temperaturmessung.print(AnzeigeLuftfeuchtigkeit);
  Temperaturmessung.println();

  // Datei schließen
  Temperaturmessung.close();
  BTSerial.println(F("Abgeschlossen."));
}
}

// Daten lesen
if (btDaten == '3')
{
  Temperaturmessung = SD.open("Messung.csv", O_RDONLY);
  if (Temperaturmessung)
  {
    // solange sich Zeilen in der Datei befinden ...
    while (Temperaturmessung.available())
    {
      // ... bis return lesen ('\n')
      String GeleseneZeile = Temperaturmessung.readStringUntil('\n');

      // für die bessere Lesbarkeit ; durch Leerzeichen ersetzen
      GeleseneZeile.replace(";", " ");

      // in der App und im Seriellen Monitor ausgeben
      Serial.println(GeleseneZeile);
      BTSerial.print(GeleseneZeile);
    }
  }
}
```

```
// Datei schließen
Temperaturmessung.close();
}
}

// Datei entfernen
if (btDaten == '4')
{
  if (SD.exists("Messung.csv"))
  {
    SD.remove("Messung.csv");
    Serial.println(F("Datei 'Messung.csv' entfernt!"));
    BTSerial.println(F("Datei 'Messung.csv' entfernt!"));
  }
}
}
}
```

Die Methode `schreibeUeberschrift`:

```
void schreibeUeberschrift()
{
  Temperaturmessung.println();
  Temperaturmessung.print(F("Datum"));
  Temperaturmessung.print(Trennzeichen);
  Temperaturmessung.print(F("Zeit"));
  Temperaturmessung.print(Trennzeichen);
  Temperaturmessung.print(F("Temperatur in °C"));
  Temperaturmessung.print(Trennzeichen);
  Temperaturmessung.print(F("Luftfeuchtigkeit in %"));
  Temperaturmessung.println();
}
```

Hartmut Waller (<https://hartmut-waller.info/arduino-blog>) letzte Änderung: 16.05.24