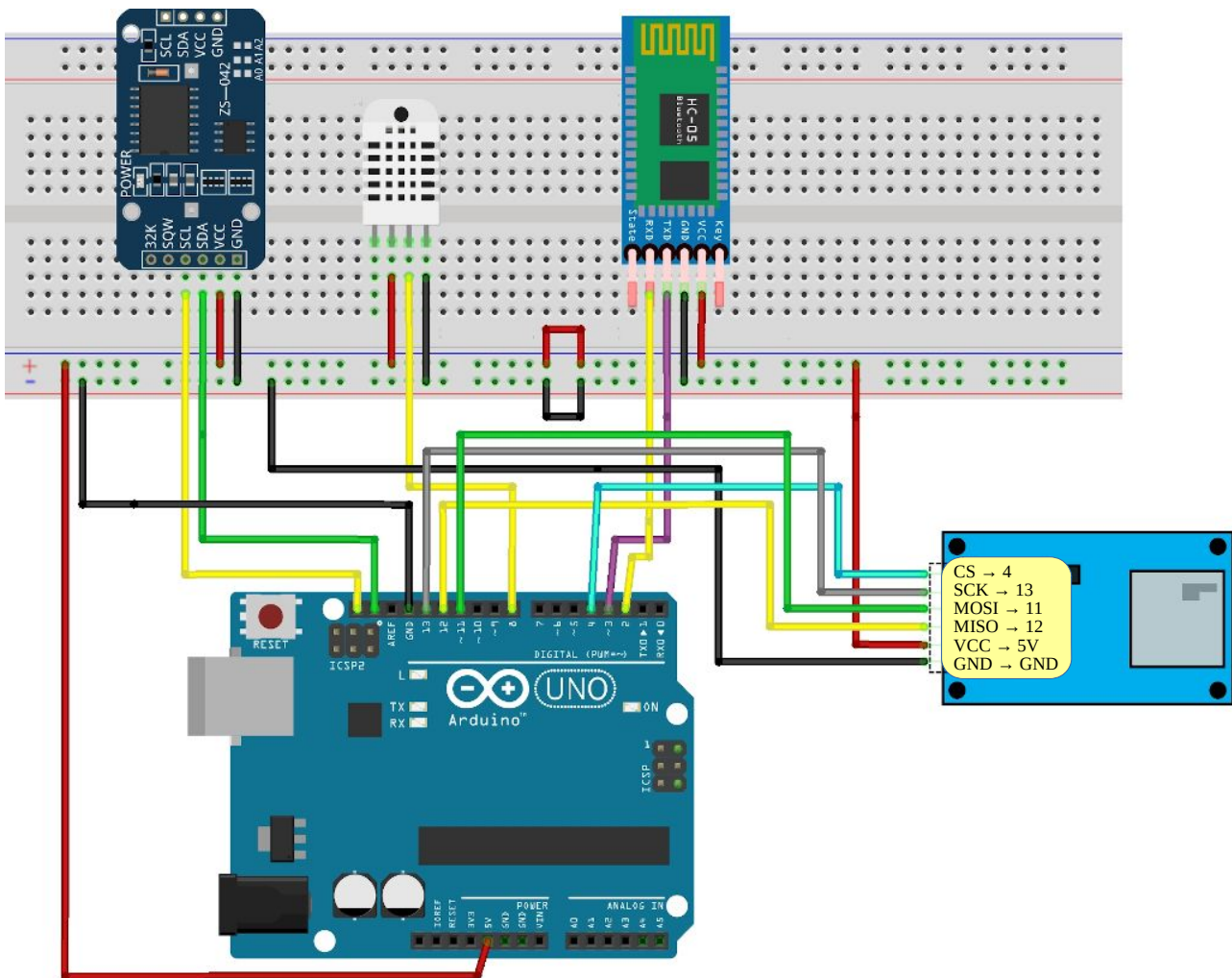


Wetterdaten mit Bluetooth übermitteln

Benötigte Bauteile:

- ➔ RTC-Modul DS3231
- ➔ Temperatur-/Feuchtigkeitssensor DHT22
- ➔ SD-Kartenleser
- ➔ Bluetooth-Module HC05/HM10
- ➔ Leitungsdrähte

Baue die Schaltung auf.



fritzing

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

Taste 1: Daten sammeln und anzeigen

Taste 2: Daten auf der SD-Karte speichern

Taste 3: Datei lesen und im Seriellen Monitor anzeigen

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 <SimpleDHT.h>
# include <RTCLib.h>
# include <SdFat.h>

// Bezeichnung der SD-Karte
SdFat SD;

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

// Bezeichnung der Textdatei
File Temperaturmessung;

// Name und Pin des Sensors
SimpleDHT22 dht22(SENSOR_DHT22);
int SENSOR_DHT22 = 8;

// RTC
RTC_DS3231 rtc;

// TX → 2, RX → 3
SoftwareSerial BTSerial(2, 3);
```

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

int SensorLesen;

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

Der setup-Teil. Beachte die Kommentare.

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

    pinMode(SENSOR_DHT22, INPUT);
    Serial.begin(9600);

    // rtc.adjust(DateTime(2018, 10, 25, 17, 28, 30));
    rtc.begin();
    /*
    wenn Datum und Zeit nicht korrekt -> Datum/Zeit setzen
    Jahr, Monat, Tag, Stunde, Minute, Sekunde
    rtc.adjust(DateTime(2018, 10, 25, 17, 28, 30));
    */
    if (!SD.begin(CSPIN))
    {
        Serial.println(F("Initialisierung fehlgeschlagen!"));
    }
    else Serial.println(F("Initialisierung abgeschlossen"));

    // wenn die Datei gelöscht werden soll
    if (SD.exists("Messung.csv")) SD.remove("Messung.csv");
    else Serial.println(F("Datei 'Messung.csv' nicht gefunden!"));
}
```



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();
      float Temperatur;
      float Luftfeuchtigkeit;

      // Daten lesen
      dht22.read2(&Temperatur, &Luftfeuchtigkeit, NULL);
      // in Strings umwandeln, . durch , ersetzen
      AnzeigeTemperatur = String(Temperatur);
      AnzeigeTemperatur.replace(".", ",");

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

      BTSerial.println(F("Temperatur: "));
      BTSerial.println(AnzeigeTemperatur);

      BTSerial.println("Luftfeuchtigkeit:");
      BTSerial.println(AnzeigeLuftfeuchtigkeit);
      BTSerial.println();
      BTSerial.println(F("-----"));

      aktuell = rtc.now();
      char Zeit[] = "hh:mm";
      AktuelleZeit = aktuell.toString(Zeit);
      BTSerial.print(AktuelleZeit);
    }
    if (btDaten == '2')
    {
      float Temperatur;
      float Luftfeuchtigkeit;

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

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

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

      /*
         Datei zum Schreiben (FILE_WRITE) öffnen
         wenn sie noch nicht existiert, wird sie erstellt
         wenn Schreiben nicht möglich -> Fehlermeldung
      */
    }
  }
}
```

```
if (Temperaturmessung.open("Messung.csv", FILE_WRITE))
{
  // wenn die Datei geöffnet werden konnte ...
  if (Temperaturmessung)
  {
    BTSerial.println(F("Schreibe Messdaten"));

    schreibeUeberschrift();

    DateTime aktuell = rtc.now();
    char Datum[] = "DD.MM.YY";
    AktuellesDatum = aktuell.toString(Datum);

    char Zeit[] = "hh:mm";
    AktuelleZeit = aktuell.toString(Zeit);
    Temperaturmessung.print(AktuellesDatum + Trennzeichen);
    Temperaturmessung.print(AktuelleZeit + Trennzeichen);

    Temperaturmessung.print(AnzeigeTemperatur + Trennzeichen);
    Temperaturmessung.print(AnzeigeLuftfeuchtigkeit);
    Temperaturmessung.println();

    // Schreibfehler abfragen
    if (!Temperaturmessung.sync() || Temperaturmessung.getWriteError())
    {
      BTSerial.println(F("Schreibfehler!"));
    }
    else
    {
      // Datei schließen
      Temperaturmessung.close();
      BTSerial.println(F("Abgeschlossen."));
    }
  }
}
else BTSerial.println(F("Datei kann nicht geöffnet werden!"));
}

if (btDaten == '3')
{
  Temperaturmessung = SD.open("Messung.csv");
  if (Temperaturmessung)
  {
    // solange sich Zeilen in der Datei befinden ...
    while (Temperaturmessung.available())
    {
      // ... werden sie gelesen und im Seriellen Monitor ausgegeben
      Serial.write(Temperaturmessung.read());
    }
    // Datei schließen
    Temperaturmessung.close();
  }
}
}
```

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();
}
```