

Wetterstation

Benötigtes Material

- Arduino Uno
- USB-Kabel für den Arduino
- Echtzeituhr (RTC DS3231 AT24C32 I2C)
- Digitaler Temperatur Sensor (DHT22/AM2302)
- Luftdruck Sensor/ Barometer (GY-68 BMP180 Bosch)
- SD Kartenmodul für Arduino
- 16 GB Micro SD Karte
- 17x MW-Kabel
- 2x MM-Kabel

Benötigte Software

- Bibliothek Adafruit BMP085 Library von Adafruit
- Bibliothek DHT sensor library von Adafruit
- Bibliothek SD von Arduino

Beschreibung

 Schulprojekt

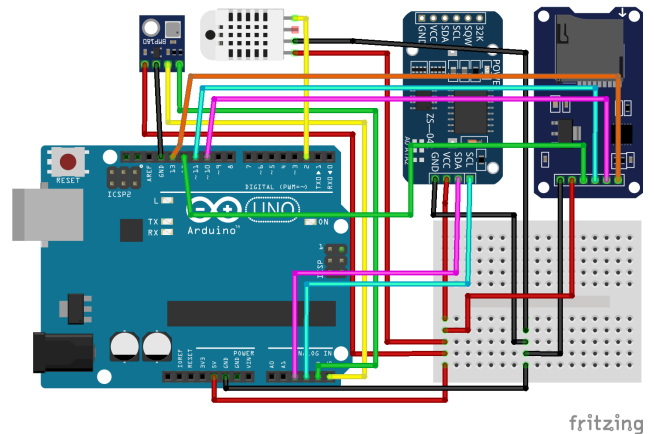
Dieses Projekt wurde durch Schülerinnen und Schüler im Rahmen eines Schulprojektes an der Kieler Gelehrtenschule erarbeitet.

Der folgende Aufbau umfasst eine Wetterstation mit einem Barometer, Thermometer und einer Uhr. Die erfassten Daten werden auf einer SD-Karte für eine spätere Auswertung abgespeichert.

Verkabelung

Arduino	Barometer	Temperatur	SD-Kartenleser
GND	GND	-	GND
5V	VIN	+	VCC
A5	SCL		
A4	SDA		
2		Out	
10			SS oder CS
11			MOSI
12			MISO
13			SCK
A2			
A3			

Arduino	Echtzeituhr
GND	GND
5V	VCC
A2	SDA
A3	SCL



Schaltplan

Code

```
#include <Adafruit_BMP085.h>
#include <SD.h>
#include <SPI.h>
#include <Wire.h>
```

```
#include "DHT.h"
#define DHTPIN 2 // Pin Nummer, in dem der Tempaeratur Sensor angesteckt ist
#define DHTTYPE DHT22 // Art des Sensors wird definiert

Adafruit_BMP085 barometer; // Instanzvariable des Barometers (085 ist
                          // Vorgänger)
DHT dht(DHTPIN, DHTTYPE);

String logPrefix = "DATEN_"; // Prefix für die Logdateien
String logFileName; // Variable welche Dateinamen der Exceltabelle enthält

// Abstand, in dem Werte auf die SD-Karte geschrieben werden (1 Minute)
unsigned long intervall = 1000UL * 60 * 1;
unsigned long lastTime = 0; // Letzter Zeitpunkt, an dem Daten gespeichert
                          // wurden

void setup() {
  Serial.begin(9600); // Starte die Serielle Schnittstelle
  Wire.begin();
  // RTC.begin();

  // Warte, solange bis eine SD-Karte eingesteckt wurde
  while (!SD.begin(SS_PIN)) {
    Serial.println("Konnte keine Verbindung zur SD-Karte aufbauen.");
    Serial.println("Versuche es in 2 Sekunden nochmal");
    delay(2000);
  }
  // Warte, bis eine Verbindung zum Barometer hergestellt werden kann
  while (!barometer.begin()) {
    Serial.println(
      "Konnte kein valides Barometer finden, überprüfe Verkabelung!");
    delay(2000);
  }
  int number = 0;
  while (SD.exists(logPrefix + number + ".CSV")) { // Finde ungenutzten
                                                    // Dateinamen
    number++;
  }
  logFileName = logPrefix + number + ".CSV"; // Setze den Namen zusammen
  delay(1000);
  File logFile = SD.open(logFileName, FILE_WRITE);
  delay(1000);

  if (!logFile) { // Beschreibe die Kopfzeile (sofern moeglich)
    Serial.println("Konnte die Datei nicht öffnen");
  } else {
    logFile.println(
      "Ms seit Programmstart;Temperatur in °C;Luftfeuchtigkeit in "
      "%;Luftdruck in Pa");
    logFile.close();
  }

  dht.begin();
  Serial.println("Setup erfolgreich. Speichere Daten in: " + logFileName);
}

void loop() {
  delay(2000); // Warte zwischen den Messungen ein paar Sekunden
```

```
// Ermittle die vergangene Zeit seit Programmbeginn in Millisekunden
unsigned long currentTime = millis();

// Lese Luftfeuchtigkeit, Temperatur und Luftdruck und speichere in Variablen
// ab
float luftfeuchtigkeit = dht.readHumidity();
float temperatur       = dht.readTemperature();
long luftdruck        = barometer.readPressure();

// Gebe die Daten aus auf der seriellen Schnittstelle aus
Serial.println("Uhrzeit: " + String(currentTime));
Serial.println("Temperatur: " + String(temperatur) + "°C");
Serial.println("Luftfeuchtigkeit: " + String(luftfeuchtigkeit) + "%");
Serial.println("Luftdruck: " + String(luftdruck) + "Pa\n");

// Teste, ob die Zeit schon einen Intervall weiter ist.
if (currentTime - lastTime >= intervall) {
    lastTime      = currentTime; // Aktualisiere letzten Datenzugriff
    File logFile = SD.open(logFileName, FILE_WRITE); // Oeffne Datei

    if (!logFile) {
        Serial.println("Konnte die Datei nicht öffnen");
    } else {
        // Schreibe die Daten auf die SD-Karte
        logFile.print(String(currentTime) + ";");
        logFile.print(temperatur);
        logFile.print(";");
        logFile.print(luftfeuchtigkeit);
        logFile.print(";");
        logFile.println(luftdruck); // println fügt zusätzlich einen
                                    // Zeilenumbruch ein
        logFile.close();           // Schliesse die Datei
    }
}
}
```

Aufgaben

- ① Aktuell werden die Daten der Sensoren alle 5 Minuten gespeichert. Kannst du das Programm so anpassen, dass es jede Minute die Daten abspeichert?

Antwort: _____

- ② Kannst du die Daten visualisieren?

Antwort: _____

- ③ Baue ein Gehäuse für die Wetterstation, um sie auch wetterfest zu machen! (3D Druck)

Antwort: _____

Das Material und dessen Inhalte sind - sofern nicht anders angegeben - lizenziert unter der Creative Commons Lizenz CC BY-NC-SA 4.0 (für den vollständigen Lizenztext siehe <https://creativecommons.org/licenses/by-sa/4.0/legalcode>)

