

```

#include <SPI.h>
#include <SD.h>
#include <Wire.h>
#include "RTClib.h"
#include <DHT.h>
#include <DHT_U.h>
#include <Adafruit_Sensor.h>
#include "DHT.h" //DHT Bibliothek laden

#define DHTPIN 2 //Der Sensor wird an PIN 2 angeschlossen.
#define DHTTYPE DHT22 // Es handelt sich um den DHT22 Sensor.

// Globale Variablen
#define LEN 9
#define SECONDS 30000
unsigned char incomingByte = 0;
unsigned char buf[LEN];
// Variablen, um Sensorwerte auszulesen.
float PM2_5Val = 0;
float PM10Val = 0;
float Temperatur;
float Luftfeuchtigkeit;
// Variablen, um Zeichenkette für SD-Card zu erzeugen
String temp;
String humid;
String value1;
String value2;
// Variable fuer LED, welche als Kontrolle dient
int LED = 8;
// Pin 10 ist fuer Data Logging Shield (by default: Pin 4)
const int chipSelect = 10;
// Messung der vergangenen Zeit - Typ muss "long" sein
unsigned long time_1 = 0;

String timestring;

RTC_DS1307 rtc;
//Der Sensor wird ab jetzt mit "dht" angesprochen
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  dht.begin(); //DHT22 Sensor starten
  pinMode(LED,OUTPUT); // Die LED dient zur aeusseren Kontrolle, dass Daten
uebertragen werden
  Serial.begin(9600); // Baudrate
  delay(3000);

  Serial.println("Initializing SD card...");
  if (!SD.begin(chipSelect)) {
    Serial.println("SD Card error");
    return;
  }
  Serial.println("card initialized");

  if (! rtc.begin()) {

```

```

    Serial.println("No RTC found");
} else {
    Serial.println("RTC clock found");
}
if (! rtc.isrunning()) {
    Serial.println("RTC is not configured");
}
// rtc.adjust(DateTime(F(__DATE__), F(__TIME__))); //Uncomment this line to set
Date and Time
}

```

```

void loop() {
    if( millis() >= time_1 + SECONDS){
        do{
            get_logvalue(); // Wert wird eingelesen
            get_time(); // Zeitdaten werden genommen
            if((PM10Val > 0) && (PM2_5Val >0)){
                break;
            }
        }while(true);

        float Luftfeuchtigkeit = dht.readHumidity(); //die Luftfeuchtigkeit auslesen
        float Temperatur = dht.readTemperature(); //die Temperatur auslesen
        // Werte werden in String umgewandelt
        value1 = PM2_5Val;
        value2 = PM10Val;
        temp = Temperatur;
        humid = Luftfeuchtigkeit;
        // Werte am Monitor ausgeben
        Serial.println(value1);
        Serial.println(value2);
        Serial.println(Temperatur);
        Serial.println(Luftfeuchtigkeit);

        write_data(); //Schreiben auf SD-Card
        time_1 += SECONDS;
    }
}

```

```

void get_logvalue() {
    unsigned char checksum;
    if (Serial.available() > 0) {
        // Einlesen des ersten Bytes: Dieses muss Wert (0xAA = 170) haben.
        incomingByte = Serial.read();
        // Ueberpruefen, ob Bedingung zutrifft. Dann wird weiter eingelesen.
        if (incomingByte == 0xAA) {
            // Die naechsten Bytes werden eingelesen.
            Serial.readBytes(buf, LEN);
            // Laut Datenblatt ist das zweite Element der Bytefolge (also buf[0])
            (0xC0 = 192) und das zehnte Byte der gesamten Bytefolge (buf[8]) (0xAB = 171)
            if ((buf[0] == 0xC0) && (buf[8] == 0xAB)) {
                // Stimmt die Bedingung ueberein, muss laut Datenblatt die Summe
                von dritten bis achten Byte gleich dem neunten Byte sein - dies wird hier
            }
        }
    }
}

```

ueberprueft.

```
    for (int i=1; i<=6; i++) {
        checksum = checksum + buf[i];
    }
    // Überprüfung der Kontrollzahl
    if (checksum == buf[7]) {
        // Laut Datenblatt is PM2_5 = (highbyte*256 + lowbyte)/10 -
        die Multiplikation entspricht einem Bytesift << 8. (2^8 = 256) Analog funktioniert
        dies mit dem PM10 Wert
        PM2_5Val=((buf[2]<<8) + buf[1])/10;
        PM10Val =((buf[4]<<8) + buf[3])/10;
        // Einschalten der LED, wenn die Daten ok sind, sodass von
        aussen ersichtlich ist, ob Daten gespeichert werden koennen.
        digitalWrite(LED,HIGH);
        //      Serial.print("PM2.5: ");
        //      Serial.print(PM2_5Val);
        //      Serial.println(" ug/m3");
        //      Serial.print("PM10 : ");
        //      Serial.print(PM10Val);
        //      Serial.println(" ug/m3");
        //      Serial.println();
    } // Wenn Kontrollzahl nicht stimmt, wird diese else{} Funktion
    durchgefuehrt
    else {
        Serial.println("checksum Error");
        PM2_5Val = -9; //checksum error = -9
        PM10Val = -9; // checksum error = -9
        digitalWrite(LED,LOW);
    }
    } // wenn begrenzende Bytes nicht passen, dann wird dieses Else{}
    durchgefuehrt
    else {
        Serial.println("frame error"); // (buf[0] == 0xC0) && (buf[8]
    == 0xAB) not okay
        PM2_5Val = -8; // frame error = -8
        PM2_5Val = -8; // frame error = -8
        digitalWrite(LED,LOW);
    }
    }
}
```

```
void get_time(){ // Einlesen der Zeit
    DateTime now = rtc.now();
    timestring = now.day();
    timestring += "-";
    timestring += now.month();
    timestring += "-";
    timestring += now.year();
    timestring += " ";
    timestring += now.hour();
    timestring += ":";
    timestring += now.minute();
    timestring += ":";
    timestring += now.second();
    // Serial.println(timestring);
}
```

```
void write_data() { // Auf SD-Card schreiben im CSV Format
```

```
// Format: PM2_5Wert, PM10Wert, Temperatur, Luftfeuchtigkeit, Monat - Tag - Jahr
Stunden : Minuten : Sekunden
String dataString = value1 + "," + value2 + "," + temp + "," + humid + "," +
timestring;
File dataFile = SD.open("datalog.txt", FILE_WRITE);
if (dataFile) {
  dataFile.println(dataString);
  dataFile.close();
  Serial.println(dataString);
}
else {
  Serial.println("error writing datalog.txt");
}
}
```