

# CO<sub>2</sub> Messstation mit Display und SD-Karte

## Benötigtes Material

- Arduino Uno
- USB-Kabel für den Arduino
- Co<sub>2</sub>-Sensor (MH-Z19B oder MH-Z19C)
- SD-Kartenmodul
- SD-Karte
- I<sup>2</sup>C Display
- 2x MM-Kabel
- 14x MW-Kabel
- Kleines Breadboard

## Benötigte Software

- Bibliothek SD von Arduino
- Bibliothek LiquidCrystal I<sup>2</sup>C von Marco Schwartz oder Frank de Brabander
- Bibliothek MH-Z19 von Jonathan Dempsey

## Beschreibung

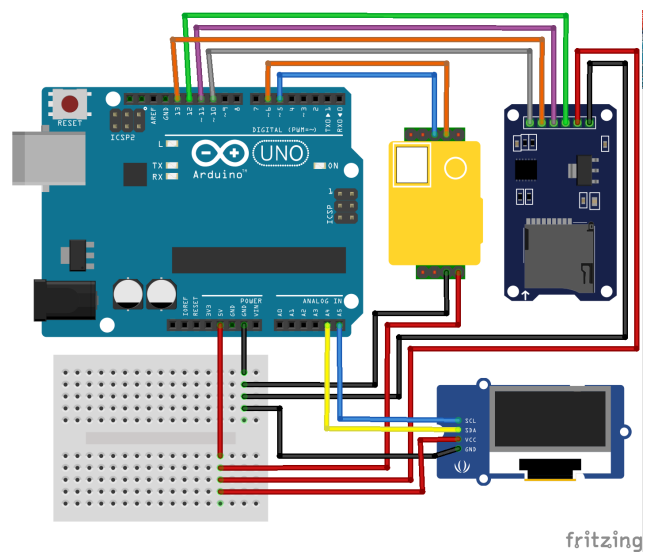
Dieses Arbeitsblatt beschreibt den Aufbau einer einfachen Messstation für den CO<sub>2</sub> Sensor. Die Daten werden direkt auf dem Display angezeigt und auf einer SD-Karte gespeichert.

## Verkabelung

MH-Z19B/MH-Z19C	Arduino	Anmerkungen
GND	GND	Über Breadboard
5V	5V	Über Breadboard
TX	5	
RX	6	

SD-Karten Modul	Arduino	Anmerkungen
GND	GND	Über Breadboard
VCC	5V	Über Breadboard
MOSI	11	
MISO	12	
SCK	13	
SS/CS	10	

I <sup>2</sup> C Display	Arduino	Anmerkungen
GND	GND	Über Breadboard
VCC	5V	Über Breadboard
SDA	A4	
SCL	A5	



Schaltplan

## Code

Bevor der Code auf dem Arduino ausgeführt werden kann, müssen die unter benötigte Software aufgezählten Libraries installiert werden.

```
// Bibliotheken für das Display
#include <LiquidCrystal_I2C.h> // LiquidCrystal_I2C Bibliothek einbinden.
#include <Wire.h> // Wire Bibliothek einbinden.

// Bibliotheken für die SD-Karte
#include <SD.h>
```

```
#include <SPI.h>

// Bibliothek für den CO2 Sensor
#include <MHZ19.h>
#include <SoftwareSerial.h>

// Pins vom CO2 Sensor
int MHZ_TX_PIN = 5; // TX Pin von MH-Z19B
int MHZ_RX_PIN = 6; // RX Pin von MH-Z19B

// Objekt für den Sensor
MHZ19 co2Sensor;
// Objekt für die Kommunikation mit dem Sensor.
// Verbinde Arduino RX mit MH-Z19B TX und
// Arduino TX mit MH-Z19B RX.
SoftwareSerial co2Serial(MHZ_TX_PIN, MHZ_RX_PIN);

// Objekt für das LCD Display
LiquidCrystal_I2C lcd(0x27, 16, 2);

// Anfang des Dateinamens für die Dateien auf der SD-Karte
// Datei hat dann z.B. den Namen "CO2_0.CSV"
// Der Name darf insgesamt nicht mehr als 8 Zeichen beinhalten.
String logPrefix = "CO2_";
// Name der Datei zum abspeichern der Datei
String logFileName;

void setup() {
  // Starte serielle Schnittstelle zur Kommunikation mit dem Arduino
  Serial.begin(9600);

  // Starte Kommunikationsschnittstelle für den Co2 Sensor
  co2Serial.begin(9600);
  // Starte den Sensor mit der Kommunikationsschnittstelle
  co2Sensor.begin(co2Serial);
  // Schalte automatische Kalibration ab
  co2Sensor.autoCalibration(false);

  // Starte das Display
  lcd.init();
  // Hintergrundbeleuchtung fürs Display einschalten
  lcd.backlight();

  // Warte solange bis eine SD-Karte eingesteckt wurde
  while (!SD.begin(SS_PIN)) {
    Serial.println(F("Konnte keine Verbindung zur SD-Karte aufbauen."));
    Serial.println(F("Versuche es in 2 Sekunden nochmal"));
    delay(2000);
  }
  // Finde Dateinamen der frei ist.
  int number = 0;
  while (SD.exists(logPrefix + number + ".CSV")) {
    number++;
  }

  logFileName = logPrefix + number + ".CSV"; // Setze den Namen zusammen
  Serial.print(F("Setup erfolgreich. Speichere Daten in: "));
  Serial.println(logFileName);
}
```

```
Serial.println(F("Warte 30 Sekunden damit der CO2 Sensor startet"));
delay(30000);
}

void loop() {
  // Warte 10 Sekunden
  delay(10000);

  // Hole den CO2 Gehalt vom Sensor in ppm
  int co2Content = co2Sensor.getCO2();

  // Gebe den CO2 Gehalt auf der seriellen Schnittstelle aus
  Serial.println(co2Content);

  // Gebe den CO2 Gehalt auf dem Display aus
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(F("CO2 Gehalt ppm:"));
  lcd.setCursor(0, 1);
  lcd.print(co2Content);

  //Öeffne die Datei auf der SD-Karte
  File logFile = SD.open(logFileName, FILE_WRITE);
  // Speichere den CO2 Gehalt mit der vergangenden Zeit auf der SD-Karte
  logFile.print(millis());
  logFile.print(F(", "));
  logFile.println(co2Content);

  // Schliesse die Datei auf der SD-Karte
  logFile.close();
}
```

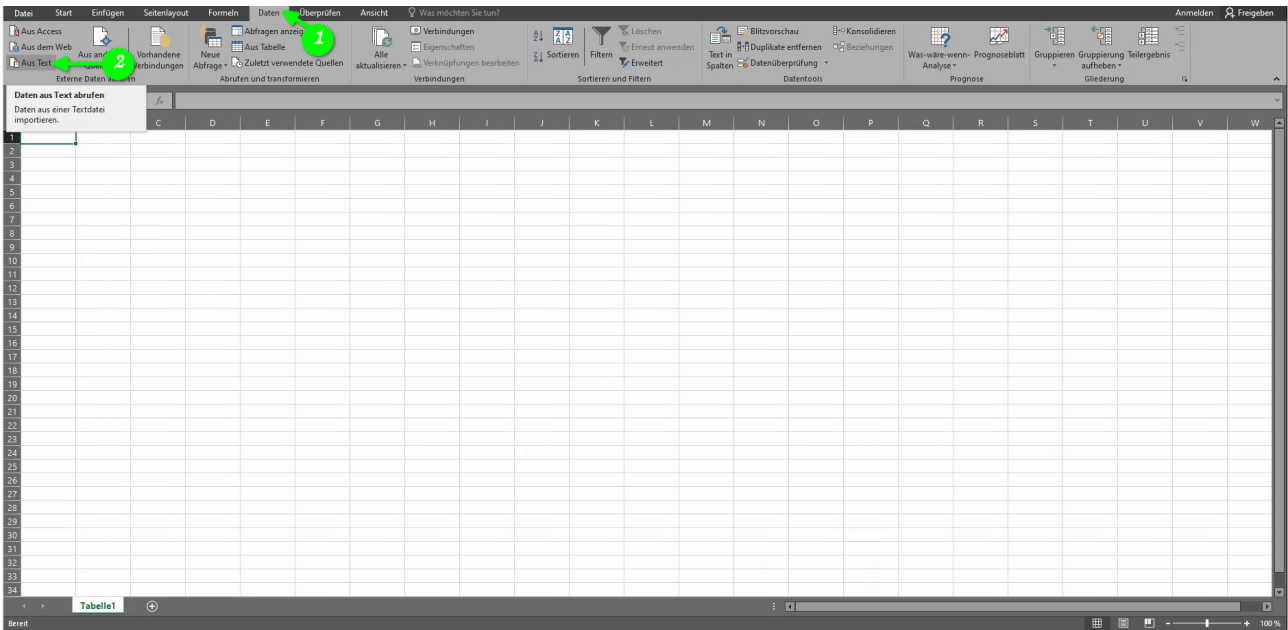
Die SD-Karte muss mit FAT32 formatiert sein, damit der Arduino die Messwerte darauf abspeichern kann.

## Daten in Excel als Graph darstellen

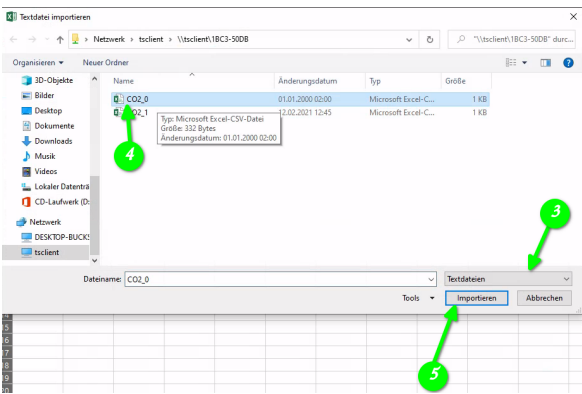
Damit die Daten von der SD-Karte verwendet werden können, als erstes den Arduino vom Strom trennen und danach die SD-Karte entfernen. Danach befinden sich Dateien in dem Format CO<sub>2</sub>\_NUMMER.CSV wobei NUMMER eine Zahl ist die hoch gezählt wird.

Hier eine Schritt für Schritt Anleitung um die Daten in Excel zu importieren.

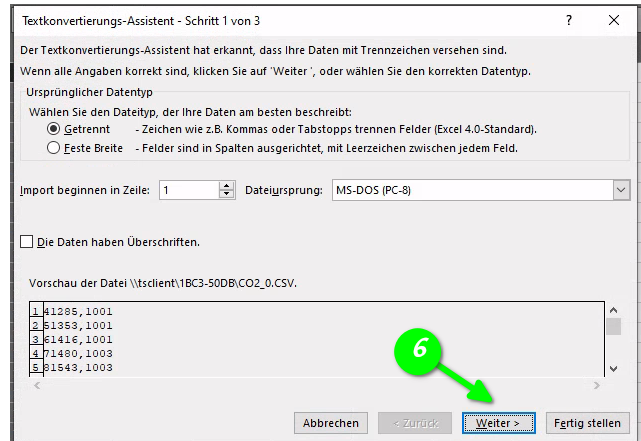
1. Eine leere Excel Datei aufmachen und auf den Reiter „Daten“ wechseln
2. Auf „Aus Text“
3. Überprüfen ob „Textdateien“ als Dateiformat gewählt ist
4. Die CO<sub>2</sub>\_NUMMER.CSV auswählen
5. Auf „Importieren“ klicken
6. Auf „Weiter“ klicken
7. Den Haken bei „Komma“ setzen
8. Auf „Weiter“ klicken
9. Auf „Fertig stellen“ klicken
10. Auf „Ok“ klicken
11. Nun befinden sich in der ersten Spalte die Zeiten in Millisekunden und in der zweiten die Messwerte



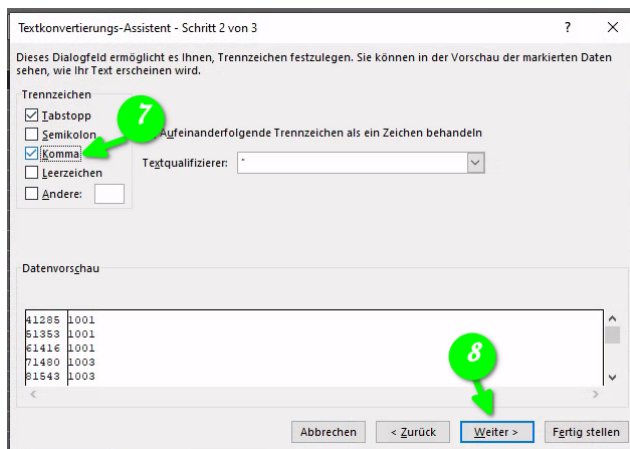
Schritt 1 und 2



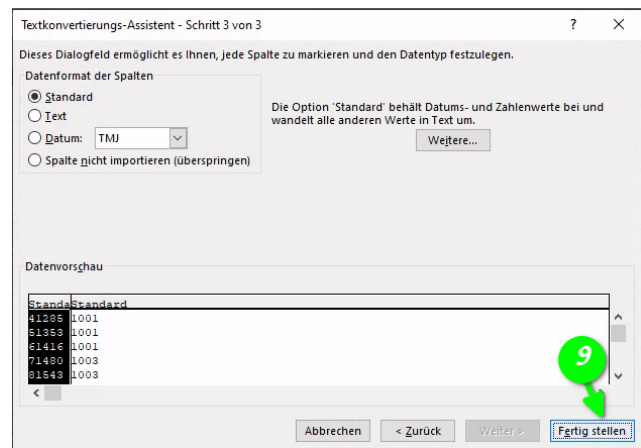
Schritt 3 bis 5



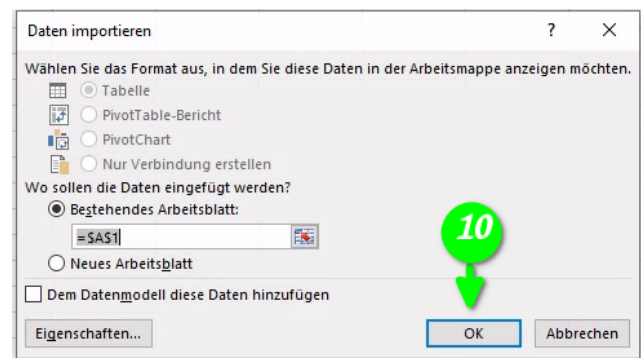
Schritt 6



Schritt 7 und 8



Schritt 9



Schritt 10

## Aufgaben

### ① Beschreibe, was der Code tut!

Man setzt die Pins für den CO<sub>2</sub>-Sensor und erstellt zwei Objekte. Dazu wird noch ein Objekt für das Display erstellt, der Dateiname und Name des Dateipfades gesetzt. Initialisieren den Sensor, Display und die SD-Karten Eigenschaften im Setup. In der Loop holen wir uns den CO<sub>2</sub>-Gehalt vom Sensor und geben ihn auf der seriellen Schnittstelle und dem Display aus. Anschließend werden die Werte auf der SD-Karte gespeichert.

### ② Trage die Werte wie im Video gezeigt in eine Tabelle ein und erstelle einen Graphen.

## Häufige Fragen und Probleme

**Mit der SD-Karte, CO<sub>2</sub> Sensor funktioniert etwas nicht.**

In das FAQ des jeweiligen Materials gucken.

**Was bedeutet F("text")?**

F("text") bedeutet das der Text direkt vom Speicher geladen wird. Damit wird dem Arduino mehr Speicherplatz zum Arbeiten gegeben. Diese Funktion nutzen wir, da die benutzten Bibliotheken schon sehr viel Speicherplatz benötigen

**Beim Hochladen wird folgende Meldung angezeigt: „Wenig Arbeitsspeicher verfügbar, es können Stabilitätsprobleme auftreten.“**

Dies liegt daran, dass die eingebundenen Bibliotheken sehr viel Speicherplatz benötigen. Solange der Code normal funktioniert kann diese Meldung ignoriert werden.

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>)