

# CO<sub>2</sub> Sensor

## Benötigtes Material

- Arduino Uno
- USB-Kabel für den Arduino
- Co2-Sensor (MH-Z19C)
- 4x MW-Kabel Lang für den Sensor

## Benötigte Software

- Bibliothek MH-Z19 von Jonathan Dempsey

## Beschreibung

Mit dem MH-Z19B lässt sich der CO<sub>2</sub> Gehalt in der Luft bestimmen.

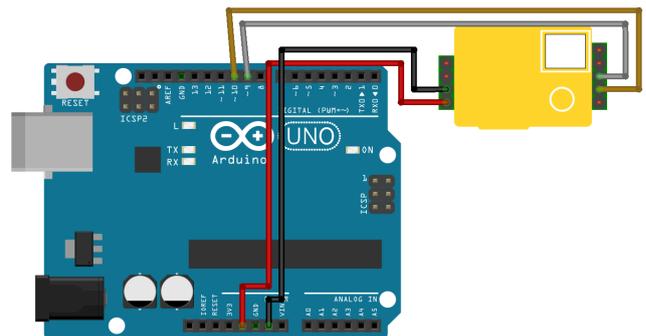
### Hinweis!

Der Sensor liefert zwar sinnvolle Werte, aber sollte nicht für sicherheitsrelevante Anwendungen verwendet werden.

Derzeit sind gefälschte Sensoren im Umlauf, die nicht richtig funktionieren. Woran man diese erkennt, findet man hier: [https://revspace.nl/MH-Z19B#Fake\\_MH-Z19B\\_.28black\\_PCB.29](https://revspace.nl/MH-Z19B#Fake_MH-Z19B_.28black_PCB.29)

## Verkabelung

| MH-Z19C | LEDs | Arduino |
|---------|------|---------|
| GND     |      | GND     |
| VIN     |      | 5V      |
| TX      |      | 9       |
| RX      |      | 10      |



fritzing

Schaltplan

## Code

```
#include <MHZ19.h>
#include <SoftwareSerial.h>

int MHZ_TX_PIN = 9; // TX Pin von MH-Z19B
int MHZ_RX_PIN = 10; // 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);

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 Kalibrierung ab
co2Sensor.autoCalibration(false);

// Gebe dem Sensor Zeit zu starten (30 Sekunden).
delay(30000);
}

void loop() {
  // Hole den CO_2 Gehalt in ppm vom Sensor
  int co2Gehalt = co2Sensor.getCO2();

  // Gebe diesen Wert zur Kontrolle über die Seriellchnittstelle aus
  Serial.println(co2Gehalt);

  // Gebe dem Sensor Zeit neue Werte zu messen
  delay(1000);
}
```

## Werte als Graph darstellen

Mit dem Seriellen Plotter den man in der Arduino IDE unter Werkzeuge findet, lassen sich die Werte auch einfach als Graph darstellen.

## Sensor kalibrieren

Wenn man mehrere Sensoren verwendet sieht man, dass die nicht immer die gleichen Werte anzeigen. Dies liegt meistens daran, dass die Sensoren nicht perfekt kalibriert sind. Es gibt zwei verschiedene Möglichkeiten den Sensor zu kalibrieren.

### In Software

Nur den Sensor mit dem Arduino verbinden. Danach den Code aus der **kalibrierung.zip** auf den Arduino hochladen. Den Arduino danach mit dem Sensor eine ca. 400ppm CO<sub>2</sub> Umgebung legen (z.B. draußen) und 20 Minuten warten. Nach dieser Zeit sollte im seriellen Monitor die Meldung **Sensor kann nun entfernt werden**. zu sehen sein. Danach kann der Arduino vom Strom getrennt werden und entweder ein anderer Sensor zum kalibrieren angeschlossen werden oder der Arduino wieder mit dem alten Code bespielt werden, falls keine Kalibrierungen mehr durchgeführt werden sollen.

### In Hardware

Wenn kein neuer Code auf den Arduino hochgeladen werden soll, kann der Sensor auch nur mit Hardware kalibriert werden.

1. Den Sensor normal anschließen und ein extra MW-Kabel an HD vom MZ-19B anschließen.
2. Den Sensor in eine Umgebung mit ca. 400ppm CO<sub>2</sub> legen (z.B. nach draußen).
3. Den Sensor mindestens 20min Werte messen lassen. Diese Werte sollten ab einem Zeitpunkt fast konstant sein.
4. Das Kabel was mit HD verbunden ist für 7 Sekunden in einen GND des Arduinos stecken und danach wieder entfernen.
5. Nun ist der Sensor kalibriert und die Werte des Sensors sollten bei ca. 400ppm sein.

## Aufgaben

- ① Beschreibe, was der Code tut!



Der Arduino holt sich die Daten vom CO<sub>2</sub>-Sensor und speichert sie in der Variable co2Gehalt. Anschließend wird der Gehalt über die serielle Schnittstelle ausgegeben.

## Häufige Fragen und Probleme

### Sensor reagiert nicht sofort auf Änderungen.

Der Sensor misst nur alle paar Sekunden und rechnet mit Durchschnittswerten. Daher kann es sein, dass wenn man in den Sensor z.B. pustet sich der Wert erst einige Zeit später ändert.

### Der Sensor gibt nach dem Anschalten falsche Werte.

Das ist normal, nachdem der Sensor startet braucht er ein bisschen Zeit, bis er richtige Ergebnisse liefert. Daher warten wir auch im Code am Anfang 30 Sekunden. Wenn (noch) keine Verbindung zum Sensor besteht, wird eine 0 als Wert zurück gegeben. Sonst startet er bei 5000 und geht dann zum richtigen runter. Nach 3-4 Minuten sollte er die richtigen Werte anzeigen.

### **!Error: Timed out waiting for response oder nur 0 als Ausgabe**

Wenn der Sensor richtig verkabelt ist, sollte dir dieser in regelmäßigen Abständen leicht rot blinken. Wenn dies nicht der Fall ist, nochmal die Verkabelung prüfen und den Arduino vom Strom trennen und nochmal verbinden.

### Wo finde ich mehr Informationen zum MH-Z19B?

- Datenblatt: <https://www.winsen-sensor.com/d/files/MH-Z19B.pdf>
- Hilfreiche Informationen: <https://revspace.nl/MH-Z19B>

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>)

