

CO₂ Ampel

Benötigtes Material

- Arduino Uno
- USB-Kabel für den Arduino
- Co2-Sensor (MH-Z19C)
- LED Grün
- LED Gelb
- LED Rot
- 4x MW-Kabel Lang für den Sensor
- 6x MM-Kabel für die LEDS

Benötigte Software

- Bibliothek MH-Z19 von Jonathan Dempsey

Beschreibung

Mit dem MH-Z19B lässt sich der CO₂ Gehalt in der Luft bestimmen. Hier bauen wir eine einfache CO₂-Ampel. Mit dieser lässt sich die Luftqualität in einen Raum schätzen.

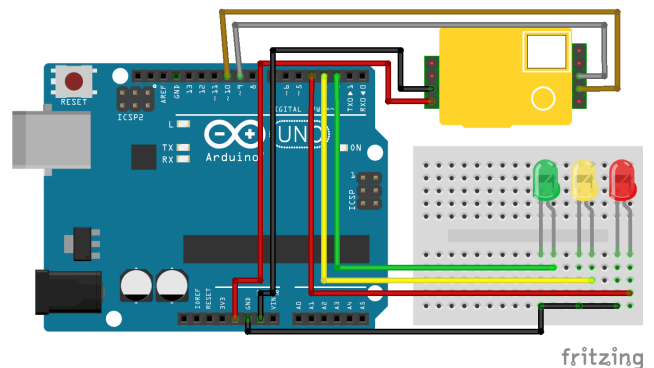
Hinweis!

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

Verkabelung

MH-Z19C	LEDs	Arduino
GND		GND
VIN		5V
TX		9
RX		10
	Grün	2
	Gelb	3
	Rot	4



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

int LED_GRUEN = 2; // Pin der grünen LED
int LED_GELB = 3; // Pin der gelben LED
int LED_ROT = 4; // Pin der roten LED

// 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() {
  // Setze LED Pins als Ausgabee
  pinMode(LED_GRUEN, OUTPUT);
  pinMode(LED_GELB, OUTPUT);
  pinMode(LED_ROT, OUTPUT);

  // 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);

  // 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);

  if (co2Gehalt == 0) {
    // Das sollte im normalen Betrieb nicht passieren.
    // Die Luft hat mindestens 400ppm.
    // Lasse alle LEDs leuchten um Fehler anzuzeigen
    digitalWrite(LED_GRUEN, HIGH);
    digitalWrite(LED_GELB, HIGH);
    digitalWrite(LED_ROT, HIGH);
  } else if (co2Gehalt < 700) {
    // Bis 700ppm lasse die grüne LED leuchten.
    digitalWrite(LED_GRUEN, HIGH);
    digitalWrite(LED_GELB, LOW);
    digitalWrite(LED_ROT, LOW);
  } else if (co2Gehalt < 1500) {
    // Zwischen 700ppm und 1500ppm lasse die gelbe LED leuchten
    digitalWrite(LED_GRUEN, LOW);
    digitalWrite(LED_GELB, HIGH);
    digitalWrite(LED_ROT, LOW);
  } else {
    // Ab 1500ppm rot augeben.
    digitalWrite(LED_GRUEN, LOW);
    digitalWrite(LED_GELB, LOW);
    digitalWrite(LED_ROT, HIGH);
  }

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

Hier in diesem Beispiel haben wir die Bereiche unter 700ppm CO₂, 700 bis 1500ppm und über 1500ppm gewählt.

Diese können je nach Bedarf geändert werden.

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 CO₂ 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 CO₂ 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!**

Antwort: _____

- ② **Was bedeuten die jeweiligen Farben der CO₂-Ampel?**

Antwort: _____

- ③ **Testet, was für einen Gehalt ihr beim Messen an einem offenen Fenster erhaltet und wie sich der Gehalt verändert, wenn ihr im Klassenraum in Richtung des Sensors atmet. Messt in Zeitabständen 0, 60, 120 Sekunden.**

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>)

