CO_2 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

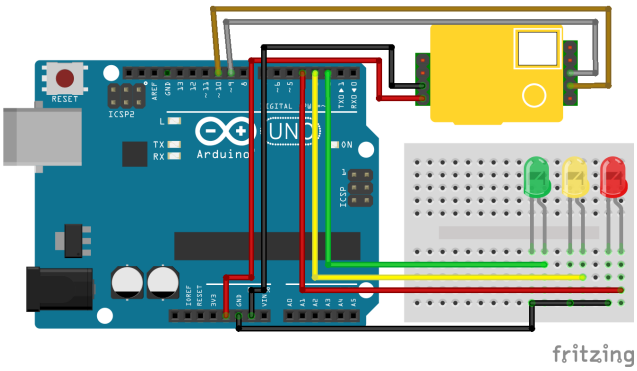
Mit dem MH-Z19B lässt sich der CO_2 Gehalt in der Luft bestimmen. Hier bauen wir eine einfache CO_2-Ampel. Mit dieser lässt sich die Luftqualität im einen Raum schätzen.

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.
// Verbine 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 Ausagabe
  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 Serielleschnittstelle 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 CO_2, 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 kalibieren

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_2 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 kalibieren 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_2 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_2-Sensor und speichert sie in der Variable co2Gehalt. Anschließend wird der Gehalt in der Luft anhand einer Co2-Ampel dargestellt.

(2) Was bedeuten die jeweiligen Farben der CO_2-Ampel?

Die rote LED zeigt sehr hohe Werte im Bereicht ab 1500ppm an, heißt es ist wenig Sauerstoff im Raum. Die gelbe LED gibt einen mittleren Bereich zwischen 700ppm und 1500ppm an und die grüne LED leuchtet bei Werten bis 700ppm an, was einem Sauerstoff gesättigten Raum gleicht.

(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.

Am Fenster sollte die Ampel grün anzeigen. Innen im Klassenzimmer sollte nach dem Atmen die Ampel sich allmählich in Richtung rot bewegen.

Häufige Fragen und Probleme

Sensor reagiert nicht sofort auf Änderungen.

Der Sensor misst nur alle paar Sekunden und rechnet mit Durschnittswerten. 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