

CO₂ Messstation mit Display

Benötigtes Material

- Arduino Uno
- USB-Kabel für den Arduino
- Co₂-Sensor (MH-Z19B oder MH-Z19C)
- I₂C Display
- 2x MM-Kabel
- 8x MW-Kabel
- Kleines Breadboard

Benötigte Software

- Bibliothek LiquidCrystal I₂C von Marco Schwartz oder Frank de Brabander
- Bibliothek MH-Z19 von Jonathan Dempsey

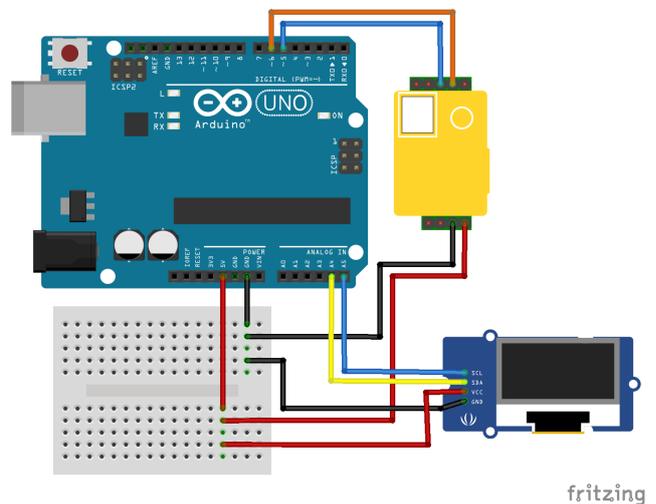
Beschreibung

Dieses Arbeitsblatt beschreibt den Aufbau einer einfachen Messstation für den CO₂ Sensor.

Verkabelung

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

I ₂ 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.

// 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);

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();

  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);
}
```

Aufgaben

① Beschreibe, was der Code tut!

Man setzt die Pins für den CO₂-Sensor und erstellt zwei Objekte. Dazu wird noch ein Objekt für das Display erstellt. Im Setup wird der Sensor und das Display initialisiert. In der Loop holen wir uns den CO₂-Gehalt vom Sensor und geben ihn auf der seriellen Schnittstelle und dem Display aus.

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

Häufige Fragen und Probleme

Mit dem CO₂ Sensor funktioniert etwas nicht.

In das FAQ des CO₂ Sensors 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>)

