

Inhaltsverzeichnis

Ziel des Projekts.....	1
Die Hardware.....	1
Messprinzip von CO ₂ -Sensoren.....	1
Anschluss des MH-Z19C am UNO.....	2
Kalibrierung des Sensors.....	2
Händische Kalibrierung:.....	2
Automatische Kalibrierung:.....	2
Benötigte Bibliotheken.....	3
Benötigte Bauteile.....	4
Der Schaltplan.....	5
Das Programm.....	5
Bibliotheken und Variable.....	5
Der setup-Teil.....	6
Der loop-Teil.....	7
Quellen:.....	8

Ziel des Projekts

Eine Wetterstation soll neben Temperatur und Luftfeuchtigkeit auch den CO₂-Gehalt der Luft messen und anzeigen.

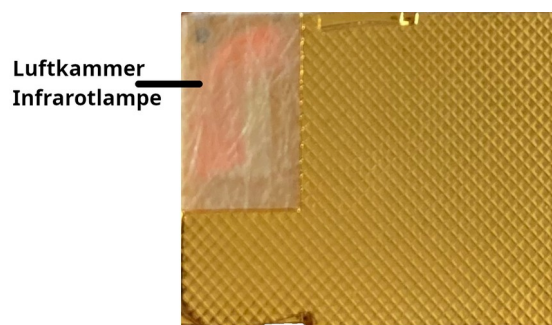
Das Programm misst mit dem Sensor DHT22 Temperatur und Luftfeuchtigkeit und mit dem MH-Z19C den CO₂-Gehalt der Luft. Das RTC-Modul DS3231 ermittelt Datum und Zeit und misst zusätzlich die Temperatur.



Die Hardware

Messprinzip von CO₂-Sensoren

Der Sensor MH-Z19C misst nach dem NDIR-Prinzip (Nichtdispersiver Infrarotsensor). Er besteht aus einer Infrarot-Lampe und einem Detektor. Dazwischen befindet sich die zu messende Luft. Je mehr CO₂ sie enthält, desto mehr Infrarot Strahlung wird absorbiert und entsprechend weniger kommt am Detektor an.



Anschluss des MH-Z19C am UNO



Kalibrierung des Sensors

Das Sensor MH-Z19C muss auf einen „Nullpunkt“ eingestellt werden. Es wird angenommen, dass draußen oder in einem gut gelüftetem Raum die CO₂-Konzentration 400 ppm (parts per million) beträgt. Dieser Wert wird als „Nullwert“ festgelegt. Meine Messungen haben gezeigt, dass dieser Wert in der Regel nur leicht überschritten wird.

Für die Kalibrierung gibt es drei Möglichkeiten:

Händische Kalibrierung:



Der Sensor muss mindestens 20 Minuten in einer gut gelüfteten Umgebung (am besten draußen) Messungen durchführen, dann musst du den HD Pin wird für mehr als 7 Sekunden mit GND verbinden.

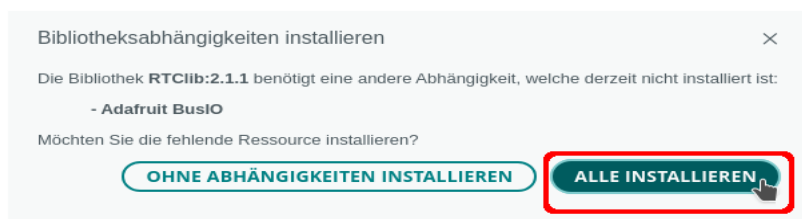
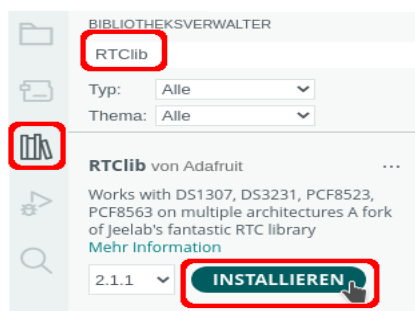
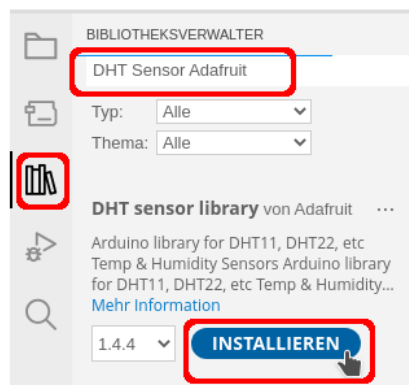
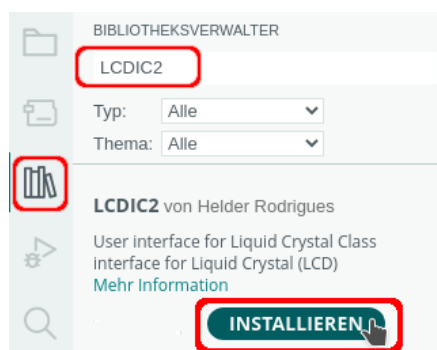
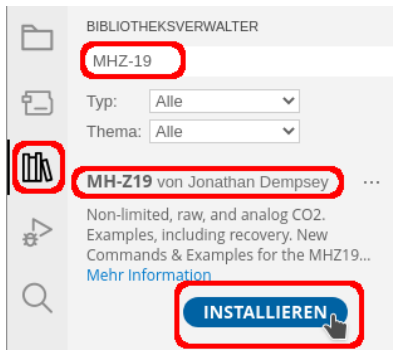
Automatische Kalibrierung:

Die automatische Kalibrierung kann nur dann funktionieren, wenn der Sensor über längere Zeit im Einsatz ist und sich zwischendurch immer wieder in einem gut durchlüftetem Raum befindet. Der „Nullpunkt“ von 400 ppm muss dann näherungsweise erreicht werden. Die automatische Kalibrierung funktioniert nur mit einer Bibliothek.

Sie wird mit

```
autoCalibration(false); ausgeschaltet  
autoCalibration(true); schaltet sie ein
```

Benötigte Bibliotheken



In den Beispielen zur Bibliothek MHZ-19 befindet sich das Programm Calibration, ich habe es ein wenig angepasst.

```
#include "Arduino.h"
#include "MHZ19.h"
#include "SoftwareSerial.h"

// RX/TX Pins zuordnen
// TX MH-Z19C auf 2, RX-Pin MH-Z19C auf 3
# define RX 2
# define TX 3

// Name des Moduls MH-Z19
MHZ19 MHZC02;

// SoftwareSerial -> Name zuordnen
SoftwareSerial MHZSerial(RX, TX);

unsigned long Wartezeit = 0;
```



```
void setup()
{
  Serial.begin(9600);
  delay(500);

  MHZSerial.begin(9600);
  MHZCO2.begin(MHZSerial);
  // automatische Kalibrierung bei Dauerbetrieb einschalten
  MHZCO2.autoCalibration(true);

  Serial.println("20 Minuten warten um die Messwerte zu stabilisieren...");
  Wartezeit = 12e5;
  delay(Wartezeit);
  Serial.println("Kalibriere..");

  // Sensor kalibrieren
  MHZCO2.calibrate();
}

void loop()
{
  if (millis() - Wartezeit >= 2000)
  {
    int CO2;
    CO2 = MHZCO2.getCO2();

    Serial.print("CO2 (ppm): ");
    Serial.println(CO2);

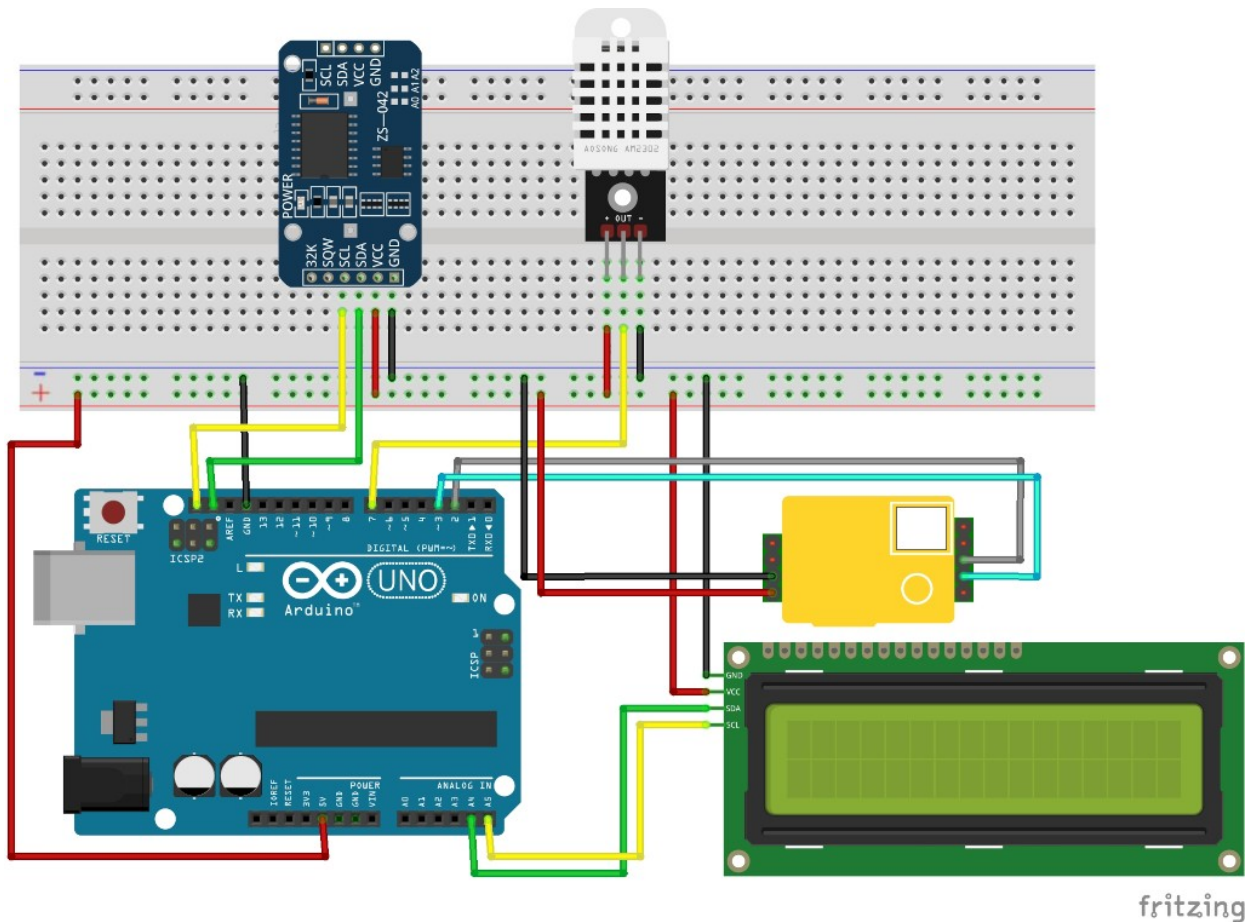
    Wartezeit = millis();
  }
}
```

Nach der Kalibrierung darfst du das Programm nicht erneut hochladen!

Benötigte Bauteile

- ➔ CO₂-Sensor MH-Z19C
- ➔ RTC-Modul DS3231
- ➔ Temperatur-/Feuchtigkeitssensor DHT22 oder DHT11
- ➔ 4-zeiliges LCD mit I²C-Schnittstelle
- ➔ Leitungsdrähte

Der Schaltplan



Das Programm

Bibliotheken und Variable

```
#include "Arduino.h"
#include "MHZ19.h"
#include "SoftwareSerial.h"
#include "RTCLib.h"
#include "DHT.h"
#include "LCDIC2.h"

// 4-zeiliges LCD
LCDIC2 lcd(0x27, 20, 4);

// Pin des DHT-Sensors
int SENSOR_DHT = 7;

// Sensortyp festlegen
// DHT22 oder DHT11
# define SensorTyp DHT22
```



```
// Sensor DHT einen Namen zuweisen
DHT dht(SENSOR_DHT, SensorTyp);

// Name des RTC-Moduls (rtc)
RTC_DS3231 rtc;

// RX/TX Pins zuordnen
# define RX 2
# define TX 3

// Name des Moduls MH-Z19
MHZ19 MHZCO2;

// SoftwareSerial -> Name zuordnen
SoftwareSerial MHZSerial(RX, TX);

unsigned long VerstricheneZeit = 0;
```

Der setup-Teil

```
void setup()
{
  Serial.begin(9600);
  delay(500);

  // LCD starten
  lcd.begin();

  // Cursor "verstecken"
  lcd.setCursor(false);

  // Serielle Kommunikation MH-Z19C starten
  MHZSerial.begin(9600);
  MHZCO2.begin(MHZSerial);

  // automatische Kalibrierung ausschalten, true schaltet sie ein
  MHZCO2.autoCalibration(false);

  // RTC-Modul starten
  rtc.begin();

  // Datum/Zeit einmalig setzen, beim nächsten Starten auskommentieren
  // rtc.adjust(DateTime(2023, 4, 23, 8, 50, 30));

  // Sensor DHT starten
  dht.begin();
}
```



Der loop-Teil

```
void loop()
{
    // 5 Sekunden warten
    if (millis() - VerstricheneZeit >= 5000)
    {
        // rtc.now() -> aktuelle Zeit holen
        DateTime aktuell = rtc.now();

        /*
            Datumsformat festlegen
            DD -> Tag mit führender 0
            MM -> Monat mit führender 0
            YYYY -> vollständige Angabe des Jahres
        */
        char Datum[] = "DD.MM.YYYY ";

        // Datum in Zeichenkette (String) umwandeln und anzeigen
        Serial.print(aktuell.toString(Datum));

        // Format der Zeitangabe festlegen
        char Zeit[] = "hh:mm:ss Uhr";

        // Zeitangabe in Zeichenkette (String) umwandeln und anzeigen
        Serial.println(aktuell.toString(Zeit));
        // Daten lesen
        // CO2-Wert MH-Z19C
        int CO2 = MHZCO2.getCO2();

        // Temperatur MH-Z19
        int TemperaturMHZ = MHZCO2.getTemperature();

        // Temperatur RTC (DS3231) ermitteln
        String TemperaturRTC = String(rtc.getTemperature());
        String AnzeigeTemperaturRTC = String(TemperaturRTC);
        AnzeigeTemperaturRTC.replace(".", ",");

        // Temperatur DHT
        float TemperaturDHT = dht.readTemperature();
        String AnzeigeTemperaturDHT = String(TemperaturDHT);
        AnzeigeTemperaturDHT.replace(".", ",");

        // Luftfeuchtigkeit DHT
        String AnzeigeLuftfeuchtigkeit = String(dht.readHumidity());
        AnzeigeLuftfeuchtigkeit.replace(".", ",");
        Serial.print("Messwert CO2: ");
        Serial.println(String(CO2) + " ppm");

        Serial.print("Temperatur MH-Z19: ");
        Serial.println(String(TemperaturMHZ) + "°C");

        Serial.print("Temperatur RTC: ");
        Serial.println(AnzeigeTemperaturRTC + "°C");
        Serial.print("Temperatur DHT: ");
```



```
Serial.println(AnzeigeTemperaturDHT + "°C");
Serial.println("-----");

lcd.setCursor(0, 0);
lcd.print(aktuell.toString(Zeit));
lcd.setCursor(0, 1);
lcd.print("CO2 " + String(CO2) + " ppm");
lcd.setCursor(0, 2);
lcd.print("Temperatur " + AnzeigeTemperaturDHT + "\337C");
lcd.setCursor(0, 3);
lcd.print("Feuchtigkeit " + AnzeigeLuftfeuchtigkeit + "%");

VerstricheneZeit = millis();
}
}
```

Quellen:

[Datenblatt MH-Z19C](#) bei Winsen Eletronics

Beitrag bei [Wikipedia](#) zum NDIR-Prinzip

Fotos: Hartmut Waller

Hartmut Waller letzte Änderung: 21.11.25