

# CO<sub>2</sub>-Messung mit dem CCS811-Sensor

Mit dem CCS811 Luftqualitätssensor Breakout Board von Adafruit können wir eine Raumluftüberwachung durchführen. Der Sensor von AMS kann eine Vielzahl von flüchtigen organischen Verbindungen (VOCs) erfassen und wurde für die Innenraumluftüberwachung entwickelt. Über I2C mit einem Mikrocontroller oder Arduino verbunden, lassen sich mithilfe der Adafruit-Bibliothek die Summe der Konzentrationen der Einzelverbindungen, auch Total Volatile Organic Compound (TVOC) genannt, sowie das äquivalente CO<sub>2</sub>-Level (eCO<sub>2</sub>) auslesen.

Die eCO<sub>2</sub>-Konzentration wird im Bereich von 400 bis 8192 parts per million (ppm) und die TVOC-Konzentration im Bereich von 0 bis 1187 parts per billion (ppb). Laut Datenblatt werden Alkohole, Aldehyde, Ketone, organische Säuren, Amine, Aliphate und aromatische Kohlenwasserstoffe erfasst. Adafruit hat darüber hinaus ein 10 kΩ NTC-Thermistor verbaut, damit der Sensor die Temperatur ermitteln kann.

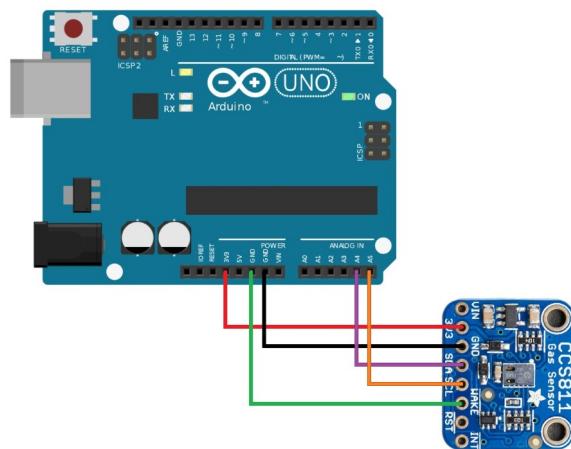
Bitte beachten, dass VOC-/Gassensoren üblicherweise kalibriert werden müssen, bevor präzise Messungen möglich sind. Das heißt, dass der Sensor gut zur groben Einschätzung und zur Beobachtung der Entwicklung der Raumluftqualität eingesetzt werden kann, präzise Ergebnisse jedoch vorab eine Kalibrierung mit einer bekannten Quelle erfordern.

Anschlüsse beim Arduino UNO:	SDA an A4
	SCL an A5
	GND und WAK an GND
	VCC an 5 V (NIC)

Hier ist ein erster Sketch:

```
#include <Adafruit_CCS811.h>
Adafruit_CCS811 ccs;

void setup() {
    Serial.begin(9600);
    Serial.println("CCS811 test");
    if (!ccs.begin()) {
```



```

Serial.println("Failed to start sensor! Please check your wiring.");
while(1); }
Serial.print("high");
while(!ccs.available()); //Kalibrieren des Temperatursensors
float temp = ccs.calculateTemperature();
ccs.setTempOffset(temp - 25.0); }

void loop() {
if(ccs.available()){
float temp = ccs.calculateTemperature();
if(!ccs.readData()){
Serial.print("CO2: ");
Serial.print(ccs.getCO2());
Serial.print("ppm, TVOC: ");
Serial.print(ccs.getTVOC());
Serial.print("ppb Temp:");
Serial.println(temp);
} else {
Serial.println("ERROR!");
while(1);}
}
delay(3000);
}

```

Die Daten werden am seriellen Monitor (am PC) ausgegeben.

Alternativ kann auch ein LCD-Display verwendet werden, um bei Batteriebetrieb die Daten einsehen zu können.

## Änderungen:

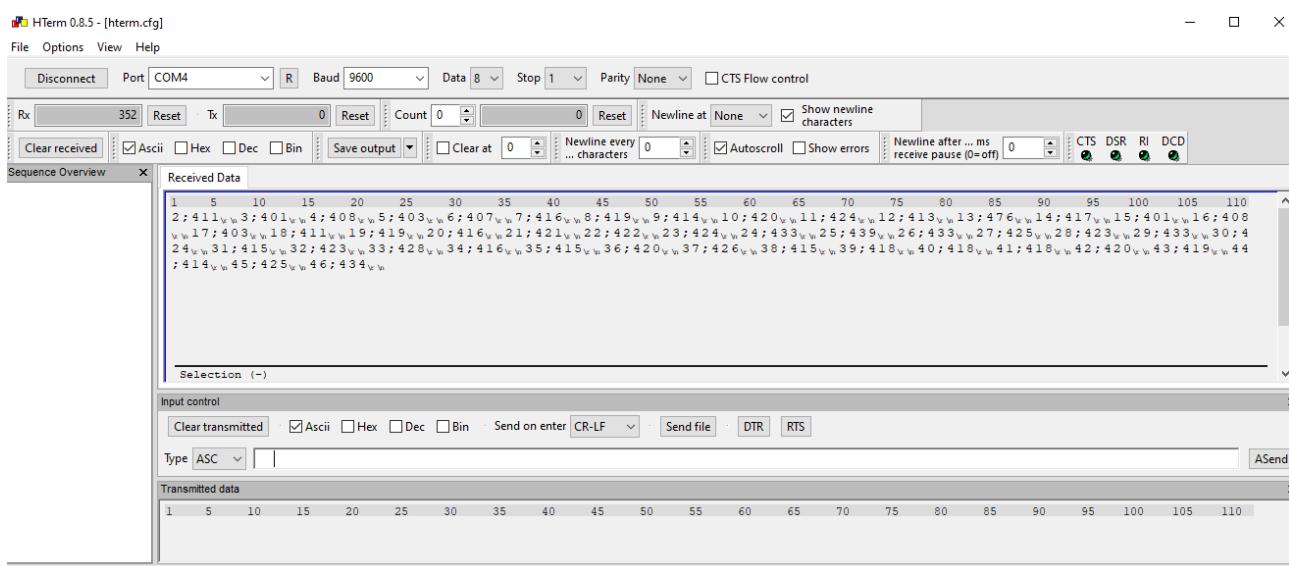
1. 5 Messungen durchführen und den Mittelwert ausgeben
2. Ausgabe an das LCD-Display (und dann mit 9V betreiben)

Um ein Daten-Logging zu betreiben, d.h. Daten zu speichern und visuell mittels eines Diagramms aufzubereiten, empfiehlt sich ein Terminal-Programm, z.B. hterm. Damit ist es möglich, statt das von Arduino bereitgestellte Fenster des seriellen Monitors zu verwenden, die Daten nach hterm zu senden. Diese Daten können nun abgespeichert (z.B. als csv-Datei) und mit Hilfe eines Kalkulationsprogramms aufbereitet werden.

## Beispielsketch:

```
#include <Adafruit_CCS811.h>
Adafruit_CCS811 ccs;
int i=0;
void setup() {
Serial.begin(9600);
if (!ccs.begin()) {
  Serial.println("Failed to start sensor! Please check your wiring.");
  while(1);
}
}
void loop() {
i=i+1;
int summe = 0;
for (int k = 0; k < 5; k++) {
  if(ccs.available()){
    float temp = ccs.getCO2();
    if(!ccs.readData()){
      summe = summe + temp;
    }
    else {
      Serial.println("ERROR!");
      while(1);
    }
  }
  delay(3000);
}
Serial.print(i);
Serial.print(";");
summe = summe/5;
Serial.println(summe);
}
```

## Beispielansicht in hterm:



Mit dem Button send output speichert man die Daten zur weiteren Verarbeitung in einem Kalkulationsprogramm. Einstellungen dazu rechts daneben: raw-Format, Rest am besten leer)

Anmerkung: Prinzipiell ist auch eine Speicherung auf SD-Karte möglich, doch wird der IC2-Bus verwendet wie auch für den Gassensor.