

# CO<sub>2</sub>-Ampel

## Benötigtes Material

- Arduino Uno
- USB-Kabel für den Arduino
- Co2-Sensor (MH-Z19C)
- microSD-Lesegerät (Arduino)
- microSD-Karte
- Servomotor (SG90)
- Piezo Pieper
- LED Ring (Adafruit NeoPixel 24)
- 9V Batterie mit Anschluss für den Arduino
- Halter für 3x AA Batterien
- 3x AA Batterien
- 20x MM-Kabel
- Anzeigenadel (3D-Druck)

## Benötigte Software

- Bibliothek SD von Arduino
- Bibliothek Adafruit NeoPixel von Adafruit
- Bibliothek MH-Z19 Adafruit von Jonathan Dempsey

### Schülerprojekt

Dieses Projekt wurde von Schülerinnen und Schülern der Kieler Gelehrtenschule erarbeitet.

## Beschreibung

In diesem Projekt bauen wir eine CO<sub>2</sub>-Ampel, welche in der Lage ist, auf den aktuellen CO<sub>2</sub>-Gehalt in der Luft aufmerksam zu machen. Dazu nutzen wir einen CO<sub>2</sub>-Sensor, der Werte in der Einheit ppm messen kann. Um die Werte anzuzeigen, kommt ein LED-Ring und ein Servomotor zum Einsatz. Damit das überschreiten eines kritischen Wertes nicht verpasst wird, lassen wir die Ampel noch piepen. Zusätzlich speichern wir die Werte auf einer SD-Karte ab, um später auf eine Tabelle mit den gemessenen Daten zugreifen zu können. So eignet sich dieser Aufbau auch für Langzeitexperimente.

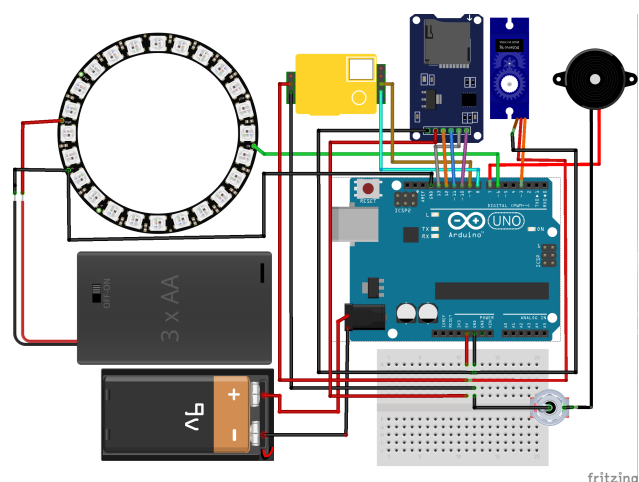
Der Servo funktioniert als Tachonadel für den CO<sub>2</sub>-Gehalt in einem Bereich von 400 bis 5000ppm. Dafür kann eine Nadel auf den Servo geklebt werden und eine Scheibe mit der Anzahl dahinter gelegt werden.

Damit die CO<sub>2</sub>-Ampel auch ohne einen Rechner betrieben werden kann, muss der Arduino mit einer 9V Batterie verbunden werden.

## Verkabelung

Arduino	CO <sub>2</sub> -Sensor	Servomotor	SD-Kartenleser
GND	GND	Braun	GND
5V	VIN	Rot	VCC
3		Orange	
8	RX		
9	TX		
10			SS oder CS
11			MOSI
12			MISO
13			SCK

Arduino	Piezo Pieper	LED-Ring	Batteriefach
GND	GND	GND	Schwarz
5V		VCC	Rot
6		Data IN	
7	VCC		



Schaltplan

## Code

Die oben unter **Benötigte Software** angegebenen Bibliotheken müssen installiert werden, damit der Code erfolgreich ausgeführt werden kann. Für Details siehe Hintergrundwissen **Bibliotheken einbinden**.

```
#include <Adafruit_NeoPixel.h>
#include <MHZ19.h>
#include <SD.h>
#include <SPI.h>
#include <Servo.h> //Wir binden die Bibliothek für den Servomotor ein
#include <SoftwareSerial.h>

String logPrefix = "DATEN_"; // Prefix für die Logdateien
String logFileName; // Name der Datei zum abspeichern der Daten

int MHZ_TX_PIN = 9; // TX Pin von MH-Z19B
int MHZ_RX_PIN = 8; // RX Pin von MH-Z19B

int PIEPER_PIN = 7; // Pin des Pieper
int SERVO_PIN = 3; // Pin für den Servo

int LED_PIN = 6; // Pin des NeoPixels
int LED_COUNT = 24; // Anzahl der LEDs des Neopixels
Adafruit_NeoPixel strip(LED_COUNT, LED_PIN, NEO_GRB + NEO_KHZ800);

// Objekt für den Sensor
MHZ19 co2Sensor;
// Objekt für die Kommunikation mit dem Sensor.
// Verbinde Arduino RX mit MH-Z19C TX und
// Arduino TX mit MH-Z19C RX.
SoftwareSerial co2Serial(MHZ_TX_PIN, MHZ_RX_PIN);

// Servomotor für die Nadel
Servo servomotor;

void setup() {
  pinMode(PIEPER_PIN, OUTPUT);

  servomotor.attach(SERVO_PIN); // Bestimme den Digitalen Pin 3 als Anschluss
  // 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);

  // Warte solange bis eine SD-Karte eingesteckt wurde
  while (!SD.begin(SS_PIN)) {
    Serial.println("Konnte keine Verbindung zur SD-Karte aufbauen.");
    Serial.println("Versuche es in 2 Sekunden nochmal");
    delay(2000);
  }
  // Finde Dateinamen der frei ist.
  int number = 0;
  while (SD.exists(logPrefix + number + ".CSV")) {
    number++;
  }
  logFileName = logPrefix + number + ".CSV"; // Setze den Namen zusammen
```

```
Serial.println("Setup erfolgreich. Speichere Daten in: " + logFileName);

strip.begin();           // Initialisiere den NeoPixel
strip.show();            // Schalte alle LEDs aus
strip.setBrightness(50); // Setzte Helligkeit auf ca. 1/5

// Gebe dem Sensor Zeit zu starten (30 Sekunden).
delay(30000);
}

void loop() {
  // Hole den CO2 Gehalt in ppm vom Sensor
  int co2Gehalt = co2Sensor.getCO2();

  // Gebe diesen Wert zur Kontrolle über die Seriellchnittstelle aus
  Serial.println(co2Gehalt);

  // Oeffne Datei auf der SD-Karte
  File logFile = SD.open(logFileName, FILE_WRITE);
  if (!logFile) {
    Serial.println("Konnte die Datei nicht öffnen");
  }
  // Speichere den Wert auf der SD-Karte
  logFile.println(co2Gehalt);
  // Schliesse die Datei
  logFile.close();

  // Schalte alle auf LEDs aus.
  for (int led = 0; led < 24; led = led + 1) {
    strip.setPixelColor(led, 0, 0, 0, 255);
  }

  if (co2Gehalt < 800) {
    // Anzahl der extra grünen LEDs die angeschaltet werden.
    // Zahl wischen 0 und 8
    int anzahlLedsGruen = map(co2Gehalt, 400, 800, 0, 8);

    // Led 0 bis 7
    for (int led = 0; led < 8; led = led + 1) {
      strip.setPixelColor(led, 0, 255, 0);
    }

    // Led 8 bis 15
    for (int ledNR = 0; ledNR < anzahlLedsGruen; ledNR = ledNR + 1) {
      strip.setPixelColor(8 + ledNR, 0, 255, 0);
    }

    // Led 23 bis 16
    for (int ledNR = 0; ledNR < anzahlLedsGruen; ledNR = ledNR + 1) {
      strip.setPixelColor(23 - ledNR, 0, 255, 0);
    }
  }

  if (co2Gehalt > 800 and co2Gehalt < 1200) {
    // Anzahl der extra gelben LEDs die angeschaltet werden.
    // Zahl wischen 0 und 8
    int anzahlLedsGelb = map(co2Gehalt, 800, 1200, 0, 8);

    // Led 8 bis 15
```

```
for (int led = 8; led < 16; led = led + 1) {
  strip.setPixelColor(led, 255, 255, 0);
}

// Led 16 bis 23
for (int ledNR = 0; ledNR < anzahlLedsGelb; ledNR = ledNR + 1) {
  strip.setPixelColor(16 + ledNR, 255, 255, 0);
}

// Led 7 bis 0
for (int ledNR = 0; ledNR < anzahlLedsGelb; ledNR = ledNR + 1) {
  strip.setPixelColor(7 - ledNR, 255, 255, 0);
}
}

if (co2Gehalt > 1200) {
  // Anzahl der extra grünen LEDs die angeschaltet werden.
  // Zahl wischen 0 und 8
  int anzahlLedsRot = map(co2Gehalt, 1200, 5000, 0, 8);

  // Led 16 bis 23
  for (int led = 16; led < 24; led = led + 1) {
    strip.setPixelColor(led, 255, 0, 0);
  }

  // Led 15 bis 8
  for (int ledNR = 0; ledNR < anzahlLedsRot; ledNR = ledNR + 1) {
    strip.setPixelColor(15 - ledNR, 255, 0, 0);
  }

  // Led 0 bis 7
  for (int ledNR = 0; ledNR < anzahlLedsRot; ledNR = ledNR + 1) {
    strip.setPixelColor(0 + ledNR, 255, 0, 0);
  }
}
// Sende die neuen Farben an den NeoPixel
strip.show();

// Schalte den Pieper ein sobald der CO2Gehalt über 1200ppm ist
if (co2Gehalt > 1200) {
  digitalWrite(PIEPER_PIN, LOW);
} else {
  digitalWrite(PIEPER_PIN, HIGH);
}

// Warte solange bis der Sensorwert sich ändert
int alterCo2Gehalt = co2Gehalt;
while (alterCo2Gehalt == co2Gehalt) {
  delay(10000); // Warte 10 Sekunden
  co2Gehalt = co2Sensor.getCO2();
}

// Verdrehe den Servomotor nur falls sich der Wert um mehr als 50ppm geändert
// hat.
if (abs(alterCo2Gehalt - co2Gehalt) > 50) {
  int winkel = map(co2Gehalt, 0, 5000, 0, 180);
  servomotor.write(winkel);
}
}
```

Die Ampel mit dem NeoPixel leuchtet wie folgt:

- 400 bis 800ppm: LEDs leuchten Grün.
- 800 bis 1200ppm: LEDs leuchten Gelb.
- ab 1200ppm: LEDs leuchten Rot.

Es leuchten zuerst immer acht LEDs in der Farbe und falls der CO<sub>2</sub>-Gehalt weiter steigt, leuchten weitere LEDs auf bis alle erleuchtet sind. Anschließend ändert sich die Farbe.

Ab 1200ppm wird der Pieper eingeschaltet.

## Experiment zur Fotosynthese

### Material

- Die CO<sub>2</sub>-Ampel
- Eine Pflanze
- Wasser
- Stromversorgung für den Arduino (z.B. Powerbank, eine 9V Batterie hält nur wenige Stunden)
- Großes verschließbares durchsichtiges Gefäß

### Durchführung

1. Besorge alle Materialien
2. Gieße die Pflanze
3. Stelle die Ampel mit der Pflanze ins Gefäß und verschließe es
4. Beobachte wie sich der CO<sub>2</sub>-Gehalt in der Luft des Gefäßes verändert

### Aufgabe

Werte den CO<sub>2</sub>-Gehalt mithilfe der Anzeige und den Daten auf der SD-Karte aus. Was lässt sich aus den Daten erkennen? Begründe deine Vermutung.

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>)

