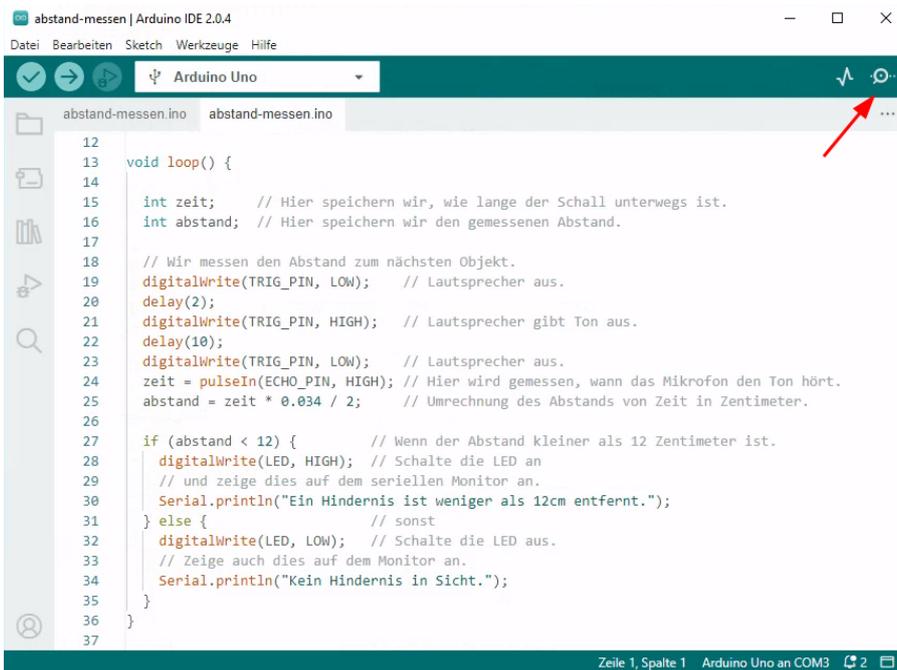


Serieller Monitor

Beschreibung

Der serielle Monitor kann dazu genutzt werden, um am PC zu verfolgen, welche Befehle der Arduino gerade ausführt. Das ist beispielsweise nützlich, wenn wir den exakten Wert ermitteln wollen, welchen ein angeschlossener Sensor misst. So spart man sich den Anschluss des LC-Displays, ist jedoch an den PC gebunden, weil man für den seriellen Monitor eine USB-Verbindung zum Arduino braucht.

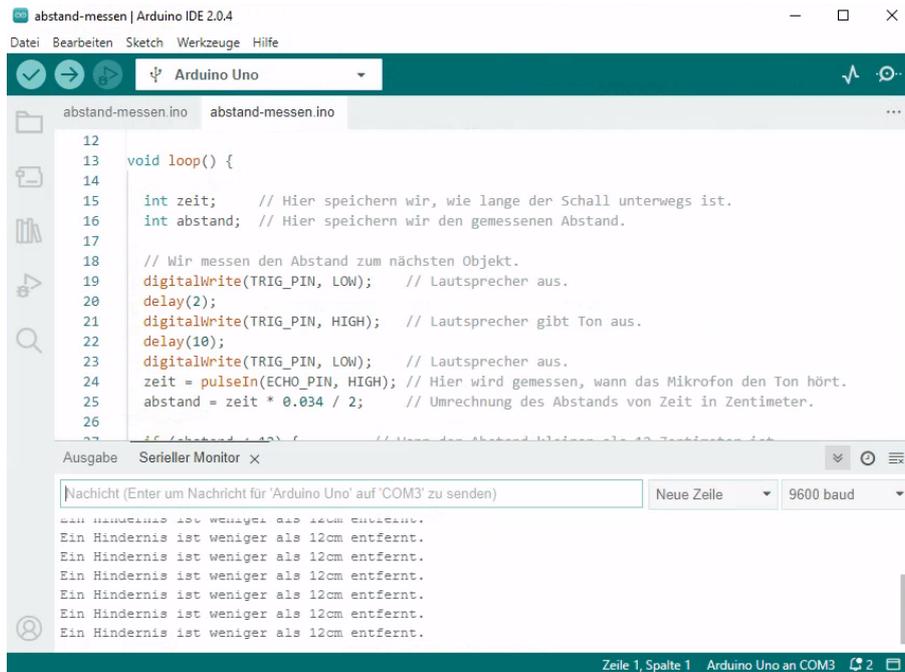
Aufrufen können wir ihn wie im Bild zu sehen oben rechts über das Lupensymbol.



```
12
13 void loop() {
14
15     int zeit; // Hier speichern wir, wie lange der Schall unterwegs ist.
16     int abstand; // Hier speichern wir den gemessenen Abstand.
17
18     // Wir messen den Abstand zum nächsten Objekt.
19     digitalWrite(TRIG_PIN, LOW); // Lautsprecher aus.
20     delay(2);
21     digitalWrite(TRIG_PIN, HIGH); // Lautsprecher gibt Ton aus.
22     delay(10);
23     digitalWrite(TRIG_PIN, LOW); // Lautsprecher aus.
24     zeit = pulseIn(ECHO_PIN, HIGH); // Hier wird gemessen, wann das Mikrophon den Ton hört.
25     abstand = zeit * 0.034 / 2; // Umrechnung des Abstands von Zeit in Zentimeter.
26
27     if (abstand < 12) { // Wenn der Abstand kleiner als 12 Zentimeter ist.
28         digitalWrite(LED, HIGH); // Schalte die LED an
29         // und zeige dies auf dem seriellen Monitor an.
30         Serial.println("Ein Hindernis ist weniger als 12cm entfernt.");
31     } else { // sonst
32         digitalWrite(LED, LOW); // Schalte die LED aus.
33         // Zeige auch dies auf dem Monitor an.
34         Serial.println("Kein Hindernis in Sicht.");
35     }
36 }
37
```

Um vom seriellen Monitor Gebrauch zu machen, müssen wir folgende Zeilen in unseren setup Codeblock einfügen: `Serial.begin(9600);`. Mithilfe von `Serial.print("Dieser Text wird nun auf dem seriellen Monitor ausgegeben");` oder `Serial.print(Co2Messwert);` können wir Text oder Variablen auf dem seriellen Monitor anzeigen lassen.

So könnte ein Messung auf dem seriellen Monitor aussehen. Als Beispiel wurde der Abstand zwischen einem Hindernis und dem Arduino mit einem Ultraschallsensor gemessen (siehe **AB Ultraschallsensor - Eine Einführung**).



Abstandsmessung mit einem Ultraschallsensor auf dem seriellen Monitor

Code

```

int TRIG_PIN = 9; // Trig_Pin angeschlossen an Pin 9.
int ECHO_PIN = 10; // Echo_Pin angeschlossen an Pin 10.
int LED = 13; // Initiiere LED unter Pin 13.

void setup() {
  // Starte serielle Schnittstelle zur Kommunikation mit dem Arduino.
  Serial.begin(9600);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(LED, OUTPUT);
}

void loop() {

  int zeit; // Hier speichern wir, wie lange der Schall unterwegs ist.
  int abstand; // Hier speichern wir den gemessenen Abstand.

  // Wir messen den Abstand zum nächsten Objekt.
  digitalWrite(TRIG_PIN, LOW); // Lautsprecher aus.
  delay(2);
  digitalWrite(TRIG_PIN, HIGH); // Lautsprecher gibt Ton aus.
  delay(10);
  digitalWrite(TRIG_PIN, LOW); // Lautsprecher aus.
  zeit = pulseIn(ECHO_PIN, HIGH); // Hier wird gemessen, wann das Mikrofon den Ton hört.
  abstand = zeit * 0.034 / 2; // Umrechnung des Abstands von Zeit in Zentimeter.

  if (abstand < 12) { // Wenn der Abstand kleiner als 12 Zentimeter ist.
    digitalWrite(LED, HIGH); // Schalte die LED an
    // und zeige dies auf dem seriellen Monitor an.
    Serial.println("Ein Hindernis ist weniger als 12cm entfernt.");
  } else { // sonst
    digitalWrite(LED, LOW); // Schalte die LED aus.
    // Zeige auch dies auf dem Monitor an.
  }
}

```

```
    Serial.println("Kein Hindernis in Sicht.");  
  }  
}
```

Aufgaben

① Was kann ein serieller Monitor?

Mit dem seriellen Monitor können vom Arduino ausgeführte Befehle verfolgt und Messwerte direkt ausgegeben und abgelesen werden ohne ein externes Display anschließen zu müssen.

② Schnappt euch einen Sensor, schreibt euch einen simplen Code und überprüft die Messwerte anhand des seriellen Monitors. Vergleicht anschließend die Ausgaben des seriellen Monitors mit denen auf einem LCD-Display.

Häufige Fragen und Probleme

Kann ich mit dem seriellen Monitor auch Graphen abbilden?

Nein, dafür ist der serielle Plotter eher geeignet.

Meine Messwerte werden brauchen zu lange bis sie angezeigt werden, was kann ich tun?

Verkürze die delays und/oder überprüfe deine Verkabelung. Wenn gar nichts klappt resete den Arduino.

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>)

