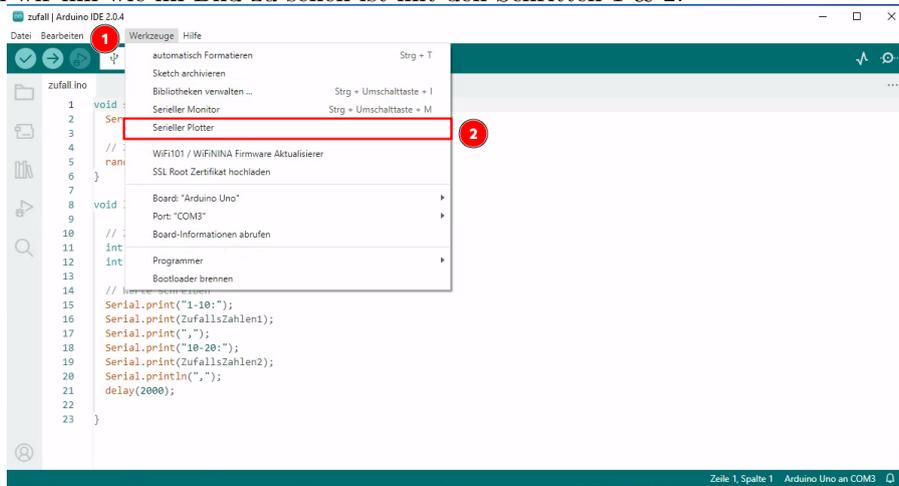


# Serieller Plotter

## Beschreibung

Genauso wie der serielle Monitor kann auch der serielle Plotter dazu genutzt werden, um am PC zu verfolgen, welche Messwerte der Arduino gerade verarbeitet. Die Messwerte, welcher ein angeschlossener Sensor misst, werden graphisch dargestellt. So hat man einen direkt sichtbaren Verlauf der Messung, jedoch ist der Arduino nicht portabel, weil für die Ausgabe durch den seriellen Monitor oder Plotter eine USB-Verbindung vom Arduino zum PC benötigt wird.

Aufrufen können wir ihn wie im Bild zu sehen ist mit den Schritten 1 & 2.



Schritte 1 & 2

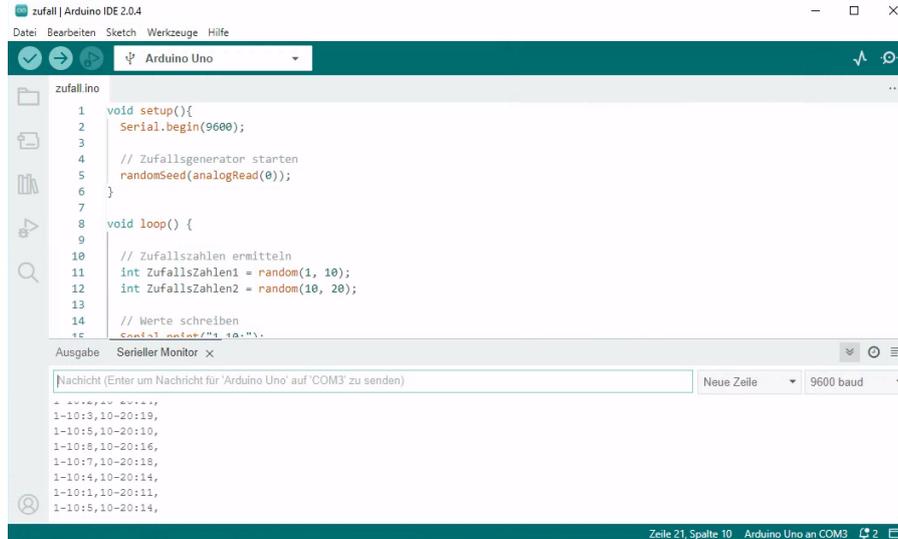
Um vom seriellen Monitor Gebrauch zu machen, müssen wir folgende Zeilen in unseren setup Codeblock einfügen: `Serial.begin(9600);`. Mithilfe von `Serial.print("1-10:");` und `Serial.print(ZufallsZahlen1);` können wir die Graphenbezeichnung und die Messwerte bzw. in diesem Beispiel die generierten Zufallszahlen in einem bestimmten Intervall, welche in Variablen gehalten werden, auf dem seriellen Plotter graphisch abbilden lassen.

So könnte ein Graph auf dem seriellen Plotter aussehen. Als Beispiel wurden Zufallszahlen in zwei verschiedenen Intervallen, nämlich die Zahlen 1-10 und 10-20, generiert und abgebildet. Der Plotter erstellt selbstständig eine Legende, da wir ihm die Beschriftung mit dem Befehl `Serial.print("1-10:");` übergeben haben. In blau zeichnet der Plotter alle Zufallszahlen im Intervall 1-10 und in rot alle Zahlen im Bereich 10-20 ein.



Plotterbeispiel-Zufallszahlen

Durch den Befehl `Serial.print(",");` werden alle Zahlen in die gleiche Zeile geschrieben. Für einen Zeilenumbruch nach dem ersten Intervall wird der Befehl `Serial.println(",");` benötigt. Um einen Vergleich zum seriellen Monitor zu bekommen, so würden die Messwerte hier aussehen. Es wird zuerst das Intervall genannt und danach die generierte Zufallszahl, wobei beide Intervalle durch ein Komma getrennt sind.



Als Vergleich zum Plotter die Darstellung auf dem seriellen Monitor

## Code

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

  // Zufallsgenerator starten
  randomSeed(analogRead(0));
}

void loop() {
  // Zufallszahlen ermitteln
  int ZufallsZahlen1 = random(1, 10);
  int ZufallsZahlen2 = random(10, 20);

  // Werte schreiben
  Serial.print("1-10:");
  Serial.print(ZufallsZahlen1);
  Serial.print(",");
  Serial.print("10-20:");
  Serial.print(ZufallsZahlen2);
  Serial.println(",");
  delay(2000);
}
```

## Aufgaben

### ① Was kann ein serieller Plotter?

Ähnlich zum seriellen Monitor können die Messwerte, diesmal graphisch, abgebildet werden. Jedoch lassen sie sich oft nicht exakt ablesen, weil man die Achsenverteilung beachten muss oder der Text mit abgebildet wird.

### ② Schnappt euch einen Sensor, schreibt euch einen simplen Code und bildet die Messwerte anhand des seriellen Plotters graphisch ab.

## Häufige Fragen und Probleme

### **Kann ich mit dem seriellen Plotter auch exakte Werte wiedergeben?**

Wenn die Achsen relativ klein sind kann man sie darstellen, um jedoch wirklich exakte Messwerte zu erhalten oder verarbeitete Befehle nachzuverfolgen ist der serielle Monitor besser geeignet.

### **Der Graph schwankt zu sehr hin und her oder braucht zu lange bis er die Messwerte ordentlich anzeigt, was kann ich tun?**

- Verkabelung überprüft?
- Verkürze das delay, damit der Graph schneller gezeichnet wird.
- Falls ihr wie im Beispiel mit einem Pulssensor messt, achtet darauf den Finger nicht zu fest auf den Sensor zu drücken und beim Messen ihn nicht zu bewegen.

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>)

