

Verkehrszählung mit Scratch oder Calliope mini



Abb. 1 Verkehrszählungen, Peter Rogoll, [CC-BY-SA 4.0](#)

Inhaltsverzeichnis

A ÜBERBLICK	3
B LERNAUFGABE	4
C BEZUG ZUM RAHMENLEHRPLAN	10
D ANHANG	13

A Überblick

Unterrichtsfach	Sachunterricht
Jahrgangsstufe/n	4
Niveaustufe/n	A, B, C
Zeitraumen	4-6 Unterrichtsstunden
Thema	Verkehrszählung

Themenfeld(er)	Kind → Womit kennen wir uns aus?
	Rad → „Was ist im Straßenverkehr zu beachten?“

Kontext	Planung und Durchführung einer Verkehrsbeobachtung/-zählung
Schlagwörter	Sachunterricht, Programmierung, Calliope mini, Scratch, Grundschule

Zusammenfassung	<p>Die Schülerinnen und Schüler planen eine Verkehrszählung in ihrer Schulumgebung. Statt einer Strichliste verwenden sie einen Calliope mini oder ein Tablet mit der App Scratch.</p> <p>Dafür planen sie zunächst gemeinsam die Verkehrszählung und entscheiden, was das Gerät für sie erledigen soll. Dann programmieren sie den Minicomputer/das Tablet.</p> <p>Nach einer ersten Erprobung im Klassenraum führen die Schülerinnen und Schüler eine Verkehrszählung innerhalb der Schulumgebung durch und werten sie anschließend gemeinsam aus.</p>
-----------------	--

B Lernaufgabe

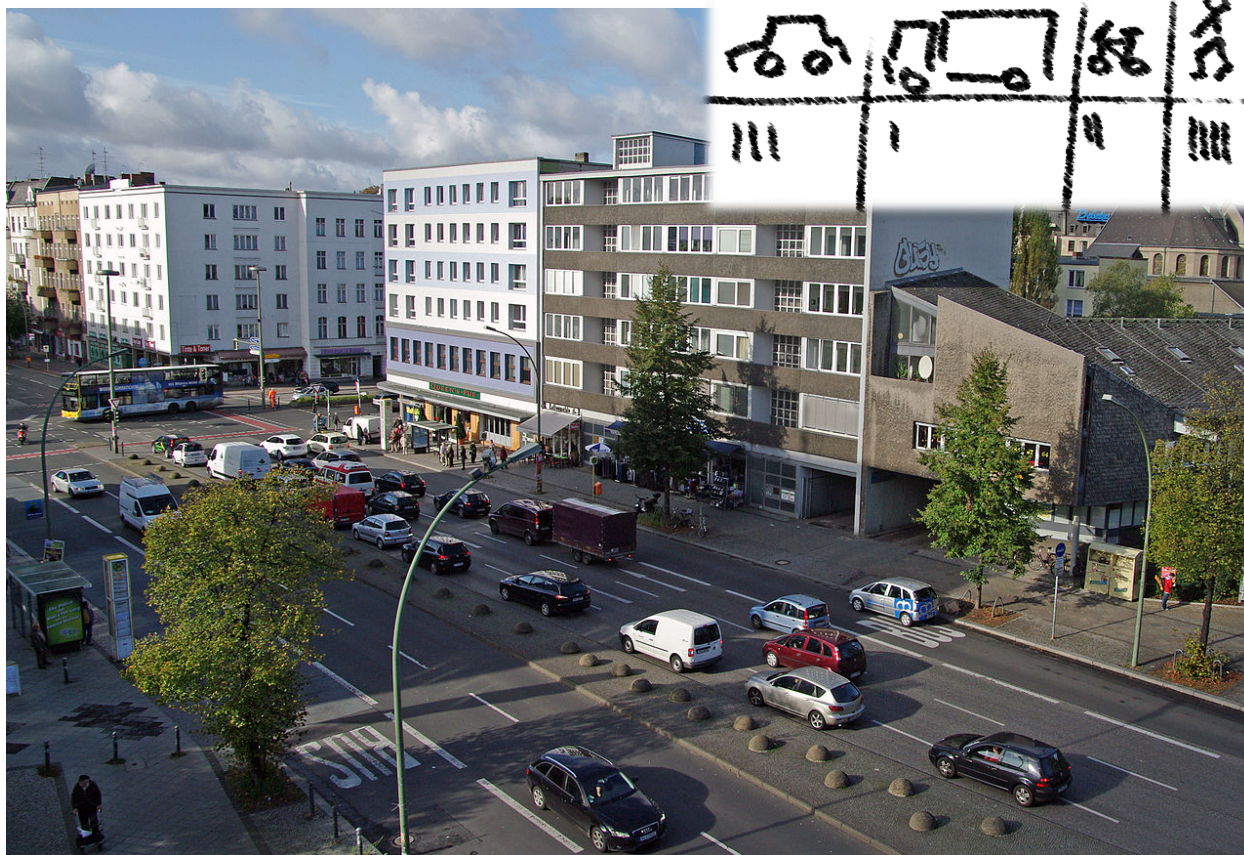


Abb. 2: Straßenverkehr, Crossing Hauptstraße/Dominicusstraße in Berlin-Schöneberg, Dirk Ingo Franke, CC-BY-SA 4.0

Man, hier kommt ja keiner rüber!

Es gibt Straßen, auf denen sehr viele Fahrzeuge unterwegs sind.

An einigen Stellen gibt es keine Ampel und keine Schülerlotsen. Dann ist es schwer, eine Straße sicher zu überqueren.

Bei einer Verkehrszählung zählt ihr Verkehrsteilnehmer (PKW, LKW, Fahrräder, usw.) an einer bestimmten Stelle.

So könnt ihr den Straßenverkehr an verschiedenen Orten vergleichen.

Mit den Ergebnissen eurer Verkehrszählungen könnt ihr vielleicht sogar Politikerinnen und Politiker davon überzeugen, dass an einer bestimmten Stelle eine Ampelanlage gebaut werden sollte.

Dafür könnt ihr eine Strichliste verwenden. Oder ihr lasst ein Gerät für euch zählen!



Abb. 3: Verkehrszählung, Peter Rogoll, CC-BY-SA 4.0



[CC-BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/)

iMINT Akademie Fachset Informatik
Lernaufgabe „Maschinelles Lernen (Verkehrsschilder)“, P. Rogoll,
Lizenz [CC-BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/), ebenda, unter Verwendung weiterer, bei den
jeweiligen Inhalten genannter Lizenzen

Senatsverwaltung
für Bildung, Jugend
und Familie


be  **Berlin**

Eine Verkehrszählung planen

Vor der Verkehrszählung müsst ihr euch auf einige Dinge einigen:

1. An welcher Straße möchtet ihr messen? 

2. Auf welcher Straßenseite (Richtung)? 

3. Welche Verkehrsteilnehmer sollen gezählt werden?
(PKW, LKW, Busse, Fahrräder, ...) 

4. Wie lange möchtet ihr messen (in Minuten/in Sekunden)? 


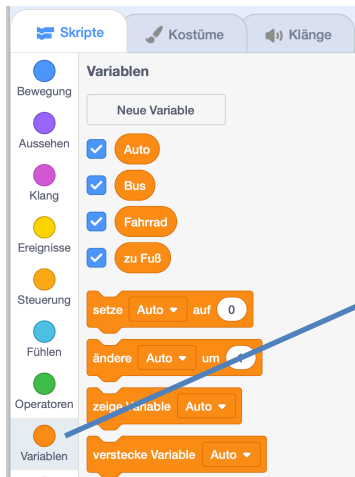
5. An welchem Tag und zu welcher Uhrzeit möchtet ihr messen? 

Abb. 4: Symbole zum Thema Verkehrszählung, Peter Rogoll, CC-BY-SA 4.0

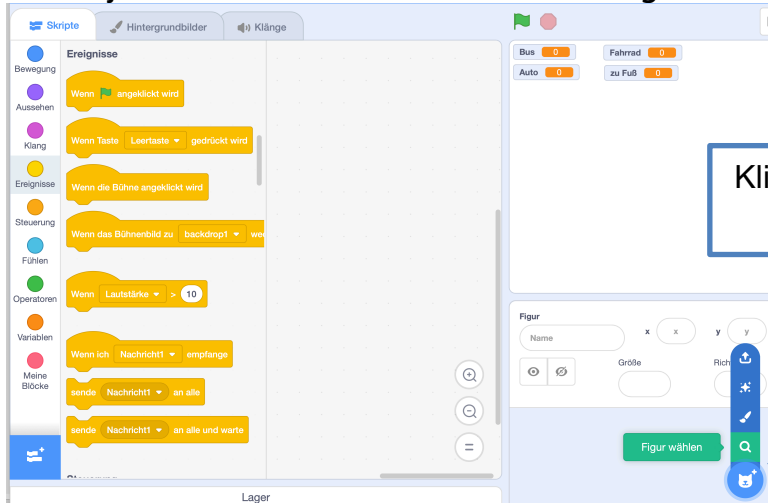
Programmiere das Programm in Scratch!

1. Zuerst musst du diese Merzkahlen/Variablen festlegen:



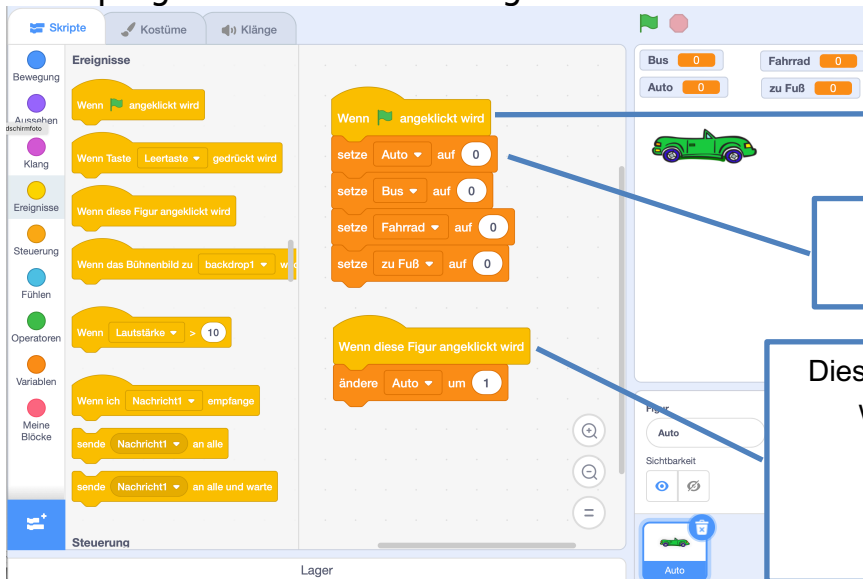
1. Klicke auf „Variablen“ und dann auf „Neue Variable“
 2. Gib jeder Variable einen Namen!
- So kann sich das Tablet merken, z.B. wie viele Autos du schon angeklickt hast.

2. Für jeden Verkehrsteilnehmer benötigst du eine Figur („Sprite“):



Klicke auf dieses Symbol und wähle ein Fahrzeug aus!

3. So programmierst du eine Figur:



Dieser Block wird gestartet, wenn du auf die grüne Fahne

Diese Blöcke findest du im orangen Bereich „Variablen“

Dieser Block wird immer wieder gestartet, wenn eine bestimmte Figur angeklickt wird.
Darunter ziehst du den Block für das Weiterzählen.

Abb. 5: Bildschirmausschnitte aus „Scratch“ (<https://scratch.mit.edu>), CC-BY-SA 4.0, erstellt von Peter Rogoll

So könnte der Verkehrszähler aussehen:

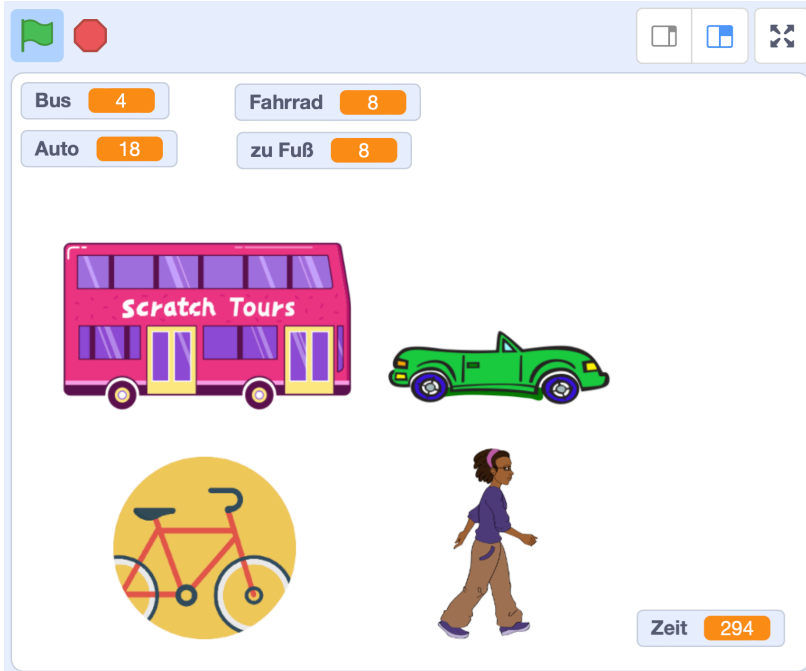
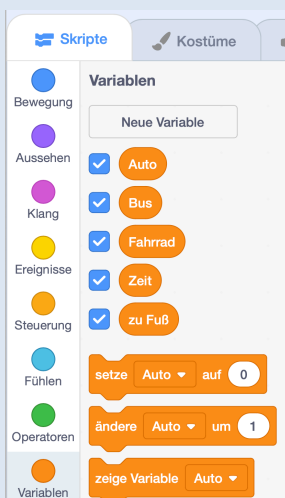


Abb. 6: Bildschirmausschnitt aus „Scratch“ (<https://scratch.mit.edu>), CC-BY-SA 4.0, erstellt von Peter Rogoll
 Abb. 7: Fahrrad-Symbol im Bildschirmausschnitt, Icon from the set Ballicons 2, pixelbuddha, CC-BY-SA 4.0
https://commons.wikimedia.org/wiki/File:Bicycle_ballonicon2.svg

So kannst du den Verkehrszähler noch verbessern:

1. Programmiere auch weitere Figuren für verschiedene Verkehrsmittel.
2. Programmiere einen Verkehrszähler mit Stoppuhr:
 Dann gibt das Programm nach einer bestimmten Zeit ein Signal.

Füge eine Variable für die Zeit hinzu!



So programmierst du die Zeitmessung:



Abb. 8: Bildschirmausschnitte aus „Scratch“ (<https://scratch.mit.edu>), CC-BY-SA 4.0, erstellt von Peter Rogoll

Programmiere das Programm in für den Calliope mini im NEPO-Editor!



ZÄHLEN:

Variable



Drücke auf das + neben Start, dann erscheint der Block „Variable“ Gib der Variablen den Namen „PKW“!

Wiederholungsblock



In diesen Wiederholungsblock kannst du alle anderen Blöcke hineinziehen. Die Blöcke im Wiederholungsblock werden unendlich oft wiederholt.

Entscheidungsblock



Drücke auf diesen Pfeil und wähle Pin0 aus.

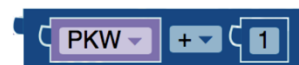
Achtung: Wenn du später einen Pin drücken möchtest, musst du gleichzeitig auch den Minus-Pin festhalten!

Block zum Verändern der PKW-Anzahl

Du benötigst diesen Mathematik-Block:



In diesen Block ziehst du die Variable und die 1 hinein:



Dann verbindest du den Block mit dem „Schreibe“-Block.



Mit „schreibe“ ist „verändere“ gemeint. Dieser Block verändert den Inhalt der Variable. Er addiert zu „PKW“ die Zahl 1 hinzu.

Warteblock



Ohne diese Wartezeit würde der Calliope mini sehr schnell weiter zählen, während du drückst.

Abb. 9: Logo „Block-Puzzle“, Peter Rogoll, CC-BY-SA 4.0
 Abb. 10: Bildschirmausschnitte aus „Scratch“ (<https://scratch.mit.edu>), Peter Rogoll, CC-BY-SA 4.0

ANZEIGEN:

<p>Noch ein Entscheidungsblock</p>		<p>Wenn Taste A gedrückt wird, soll die Zählung beendet werden.</p>
<p>Anzeige-Block</p>		<p>Dieser Block sorgt dafür, dass der Calliope mini „PKW“ anzeigt.</p>
<p>Anzeige-Block und Variablenblock</p>		<p>Dieser Block zeigt den Inhalt der Variable PKW an, wenn die Zählung beendet ist.</p>

Beispiellösung:

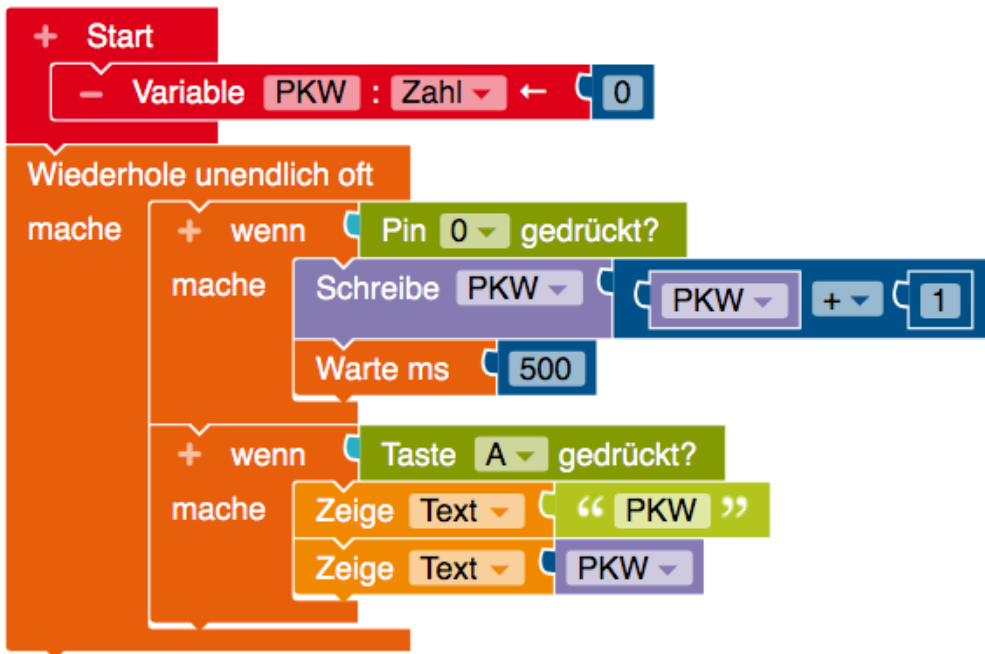


Abb. 11: Bildschirmausschnitte aus NEPO lab.open-roberta.org, CC-BY-SA 4.0, erstellt von Peter Rogoll

So kannst du den Verkehrszähler noch verbessern:

1. **Programmiere den Calliope mini so, dass er verschiedene Verkehrsmittel zählen kann.** Für jedes Verkehrsmittel, das du zählen möchtest, musst du einen anderen Pin programmieren, z.B.

Pin0 für PKW,

Pin1 für Busse,

...

Du benötigst also **mehrere Variablen** und **mehrere „Wenn“-Blöcke**.

2. Du kannst den Calliope mini so programmieren, dass er verschiedene Töne abspielt, wenn du verschiedene Pins oder Tasten drückst. So weißt du genau, ob du richtig gedrückt hast.

Dazu benötigst du diesen Block, bei dem du die Dauer und die Höhe des Tones verändern kannst:



Abb. 12: Bildschirmausschnitt aus NEPO lab.open-roberta.org, CC-BY-SA 4.0, erstellt von Peter Rogoll

C Bezug zum Rahmenlehrplan

Diese Lernaufgabe ist folgenden sachunterrichtlichen Perspektiven zuzuordnen:

Naturwissenschaftliche und technischen Perspektive

„Die naturwissenschaftliche Perspektive des Sachunterrichts eröffnet eine neue Sichtweise auf ... Phänomene: Alltäglich Erscheinendes wird hinterfragt, Komplexes reduziert.“

Der Rahmenlehrplan weist darauf hin, dass Kinder technische Geräte oft nur als „Blackboxes“ nutzen und fordert, dass sie Technik nicht nur nutzen, sondern auch mit ihr experimentieren sollen. Daneben sollen technische Gegenstände nacherfunden und konstruiert werden. Bei der Mediennutzung soll auch das Programmieren berücksichtigt werden (Teil C, Sachunterricht, S. 24-26).

Themenfeld	Themen	Inhalte	Unterrichtsanregungen des RLPs
3.2 Kind	Womit kennen wir uns aus?	- Straßenverkehr ... - Computer und Internet	Recherchen durchführen: Wie funktionieren Computer? (Handy, Roboter u. a.)
3.4 Rad	Was ist im Straßenverkehr zu beachten? (Mit dem Fahrrad unterwegs)	- auf der Straße (der richtige Weg, Verkehrszeichen, Fairness) - Vorfahrtsregeln	

Die Schülerinnen und Schüler können

2.1 Erkennen	• Fragen zu einem Phänomen/Thema stellen	A
	• sich an der Planung und Ausführung von Arbeits- und Lernschritten beteiligen	A
	• einen Sachverhalt auf Grundlage einer Fragestellung untersuchen ... Beobachtungen durchführen ...	B
	• ein Vorhaben ... nach Vorgaben planen und ... durchführen	C
	• ein Phänomen ... im Hinblick auf eine Fragestellung untersuchen	C
2.4 Handeln	• Dinge oder Informationen nach Kriterien vergleichen.	B/C
	• Eine Aufgabe auswählen und ausführen	A
	• Lern- und Arbeitsergebnisse selbst kontrollieren	B
	• Materialien, Instrumente, Geräte, Apparate und Medien aufgabenbezogen nutzen ...	B

Bezüge zum Basiscurriculum Sprachbildung¹

Standards des BC Sprachbildung	Die Schülerinnen und Schüler können...
1.3.2 Rezeption/ Leseverstehen	
Texte verstehen und nutzen	<ul style="list-style-type: none"> • Aus Texten gezielt Informationen ermitteln • Informationen aus Texten zweckgerichtet nutzen
Lesetechniken und Lesestrategien anwenden	Lesetechniken ... entsprechend der Leseabsicht anwenden

Bezüge zum Basiscurriculum Medienbildung²

Standards des BC Medienbildung	Die Schülerinnen und Schüler können ...
Produzieren	<ul style="list-style-type: none"> • Medientechnik einschließlich Hard- und Software nach Vorgaben einsetzen. • Medientechnik einschließlich Hard- und Software unter Verwendung von Anleitungstexten oder Tutorials handhaben.

Bezüge zu übergreifenden Themen³

Gleichstellung und Gleichberechtigung der Geschlechter

Inklusive Aspekte der Lernaufgabe

	Standards der iMINT-Akademie
Zugänge	Computergestütztes Arbeiten
Sprache	Möglichst einfache Satzstrukturen
Aufgabenstellungen	Verschiedene Differenzierungsansätze
Methoden	Rückmeldung durch Computereinsatz
Experimente	Partnerarbeit
IT	Online-Programmierplattform für Kinder

¹ vgl. Rahmenlehrplan Jahrgangsstufen 1-10, Teil B, S. 6-10, Berlin, Potsdam 2015

² vgl. Rahmenlehrplan Jahrgangsstufen 1-10, Teil B, S. 15-22, Berlin, Potsdam 2015

³ vgl. Rahmenlehrplan Jahrgangsstufen 1-10, Teil B, S. 24 ff., Berlin, Potsdam 2015

D Anhang

Didaktische Hinweise

Der Straßenverkehr, z.B. im Schulumfeld, hat eine hohe Alltagsrelevanz. Neben Fragen der Verkehrssicherheit geht es hier auch um Fragen der zukunftsfähigen und nachhaltigen Mobilität.

Die Beobachtung des einzelner Verkehrsteilnehmer*innen kann die Kinder dabei unterstützen, sich besser in diese hineinzusetzen. Es fördert die Fähigkeit, sich anbahnende Gefahren besser zu antizipieren.

Indem Kinder das Verkehrsgeschehen zahlenmäßig erfassen, können sie ihre individuelle Wahrnehmung mit genaueren Messergebnissen vergleichen bzw. überprüfen. Insofern ist es günstig, wenn Kinder vor den Messungen auch Schätzungen z.B. zum Anteil einzelner Verkehrsmittel am gesamten Verkehrsaufkommen notieren.

Bei einer anschließenden Auswertung lässt sich so die Einsicht anbahnen, dass Zählungen/Messungen eine Grundlage für Entscheidungen sind. Es wird auch ersichtlich, dass ein sinnvoller Vergleich zweier Verkehrssituationen nur anhand genauer Zahlen möglich ist.

Genauere Beobachtungen und Zahlen ermöglichen es auch, an Politiker*innen mit dem Ziel heranzutreten, Verkehrssituationen im Schulumfeld zu verbessern.

Das Arbeitsblatt (S. 5) kann bei der Vorbereitung einer Verkehrszählung hilfreich sein. Es ist auch möglich, dass die Kinder in einem Unterrichtsgespräch (zunächst ohne das Arbeitsblatt) wesentliche Fragen/Aufgaben selbst formulieren.

In dieser Lernaufgabe werden zwei Möglichkeiten vorgeschlagen, wie ein Verkehrszähler programmiert werden kann:

1. Mit der Programmiersprache „**Scratch**“ (**einfacher**):

Die Programmierung kann direkt am Tablet mit der Scratch-App stattfinden. Dasselbe Gerät kommt auch bei der Zählung zum Einsatz.

2. Mit dem „**NEPO-Editor**“ und einem **Calliope mini** (**etwas aufwändiger**):

Die Programmierung findet an einem PC oder Laptop statt. Anschließend wird das Programm auf den Mikrocontroller Calliope mini übertragen. Mit diesem wird dann die Zählung durchgeführt.

Zu Scratch:

Scratch ist eine der verbreitetsten grafischen Programmiersprachen. Die Oberfläche setzt wenig Einarbeitung voraus. Dennoch sind die Möglichkeiten sehr vielseitig. Die Programmierung findet auf dieser Seite der Universität Massachusetts Institute of Technology (MIT) statt: <https://scratch.mit.edu>

Für den Einstieg eignen sich u.a. folgende Seiten:

<https://tueftelakademie.de/programmieren-lernen/scratch/>

<https://unterrichten.zum.de/wiki/Scratch>

Scratch bietet den Kindern eine attraktive Programm-Oberfläche, auf der sie nach eigenen Vorstellungen programmieren können.

Die Programmierung ist bei dieser Aufgabe bewusst einfach gehalten, so dass schnelle Erfolge möglich sind. Auch aus diesem Grund findet bei dieser Aufgabe vor allem nachvollziehendes Programmieren statt.

Zur weiteren Vereinfachung oder zeitlichen Straffung ist es möglich, eine Grundversion des Programms zu erstellen den Kindern diese Datei zur Verfügung zu stellen.

Zur Differenzierung sind viele einfache und komplexere Erweiterungen der Programmierung möglich:

- Ein Bühnenbild, z.B. mit Verkehrscharakter kann selbst gezeichnet oder ausgewählt werden.
- Die einzelnen Figuren/Fahrzeuge können so programmiert werden, dass sich beim Anklicken/Antippen vergrößern und dann wieder verkleinern.
- Es können auch verschiedene Audio-Dateien beim Antippen abgespielt werden.

Zum NEPO-Editor und Calliope mini:

Auch der NEPO-Editor vom Fraunhofer-Institut stellt eine grafische Programmiersprache zur Verfügung: <https://lab.open-roberta.org>

Diese Unterrichtseinheit setzt voraus, dass Schülerinnen und Schüler Vorerfahrungen mit dem Calliope mini und seiner Programmierung gesammelt haben. Voraussetzung ist auch, dass Kinder Programme vom PC/Laptop auf den Calliope mini übertragen können.

Dieses kostenlose Arbeitsheft kann hierbei hilfreich sein:

<https://calliope.cc/schulen/arbeitsheft>

Das Material ist als „Block-Puzzle“ konzipiert, so dass Schülerinnen und Schüler mit etwas Unterstützung selbst eine Programmstruktur finden können. Ausgangspunkt kann aber auch das Beispielprogramm sein, das anschließend von den Kindern erweitert wird.

Auch für den Calliope mini könnte eine Stoppuhr-Funktion programmiert werden. Die Programmierung ist hier etwas aufwändiger bzw. weniger übersichtlich als bei Scratch. Insofern bietet sich eher die Nutzung einer herkömmlichen Stoppuhr an.



[CC-BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/)

iMINT Akademie Fachset Informatik
Lernaufgabe „Maschinelles Lernen (Verkehrsschilder)“, P. Rogoll,
Lizenz [CC-BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/), ebenda, unter Verwendung weiterer, bei den
jeweiligen Inhalten genannter Lizenzen

Senatsverwaltung
für Bildung, Jugend
und Familie

be  Berlin

Material für den Einsatz dieser Lernaufgabe

Anzahl	Name des Materials
1	<i>Bei der Arbeit mit „Scratch“:</i> Tablet für jeweils 2 Kinder
1	<i>Bei der Arbeit mit Calliope mini:</i> Calliope mini für 2 Kinder
1	PC oder Laptop für 2 Kinder

Literaturquellen

Philipp Spitta: Praxisbuch Mobilitätsbildung - Unterrichtsideen zu Mobilität, Verkehr und Bildung für nachhaltige Entwicklung für die Klassen 1-6, Baltmannsweiler 2020

Landesinstitut für Schule und Medien Berlin-Brandenburg (LISUM): Handreichung für das übergreifende Thema MOBILITÄTSBILDUNG UND VERKEHRSERZIEHUNG, Ludwigsfelde 2018

Bildnachweis

Bildtitel	Seite	Bildquelle
Abb. 1: Verkehrszählungen	1	Peter Rogoll, CC-BY-SA 4.0
Abb. 2: Straßenverkehr	4	Collage aus Abb1 (s.o.) und Crossing Hauptstraße/Dominicusstraße in Berlin-Schöneberg, Dirk Ingo Franke, CC-BY-SA 4.0 https://commons.wikimedia.org/wiki/File:Berlin_Schoeneberg_kreuzung_haupt-dominicusstrasse_12.10.2011_09-51-07.JPG
Abb. 3: Verkehrszählung	4	Peter Rogoll, CC-BY-SA 4.0
Abb. 4: Symbole zum Thema Verkehrszählung	5	Peter Rogoll, CC-BY-SA 4.0
Abb. 5: Bildschirmausschnitte aus „Scratch“	6	https://scratch.mit.edu , CC-BY-SA 4.0, erstellt von Peter Rogoll
Abb. 6: Bildschirmausschnitt aus „Scratch“	7	https://scratch.mit.edu , CC-BY-SA 4.0, erstellt von Peter Rogoll
Abb. 7: Fahrrad-Symbol im Bildschirmausschnitt	7	Icon from the set Ballicons 2, pixelbuddha, CC-BY-SA 4.0 https://commons.wikimedia.org/wiki/File:Bicycle_ballonicon2.svg
Abb. 8: Bildschirmausschnitte aus „Scratch“	7	https://scratch.mit.edu , CC-BY-SA 4.0, erstellt von Peter Rogoll
Abb. 9: Logo „Block-Puzzle“		Peter Rogoll, CC-BY-SA 4.0
Abb. 10: Bildschirmauss Bildschirmausschnitte aus NEPO	8	lab.open-roberta.org, CC-BY-SA 4.0, erstellt von Peter Rogoll
Abb. 11: Bildschirmausschnitte aus NEPO	8	lab.open-roberta.org, CC-BY-SA 4.0, erstellt von Peter Rogoll
Abb. 12: Bildschirmausschnitte aus NEPO	9	lab.open-roberta.org, CC-BY-SA 4.0, erstellt von Peter Rogoll
Abb. 13: Bildschirmausschnitte aus NEPO	10	lab.open-roberta.org, CC-BY-SA 4.0, erstellt von Peter Rogoll