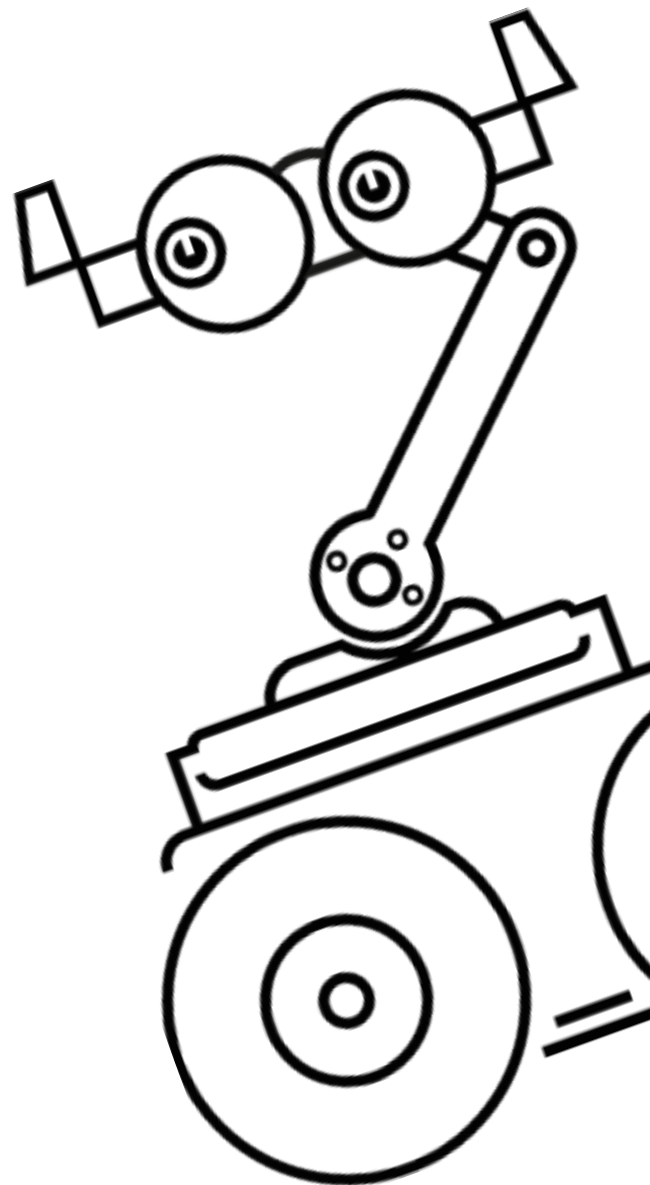


# Programmieren mit Open Roberta

Unterrichtsbeispiele mit dem Lego WeDo 2.0 Set



**ROBERTA**  
INITIATIVE

Roberta ist ein eingetragenes Warenzeichen der Fraunhofer-Gesellschaft e.V.  
Roberta ist seit 2010 Mitglied der Fraunhofer Academy

# Inhalt

Dieses Dokument stellt eine Unterrichtseinheit aus der Reihe [Roberta-Lernmaterialien für den Lego Education WeDo](#) dar.

Unterrichtseinheit

## Automatismen und Reflexe

Die Roberta-Lernmaterialien für den Lego Education WeDo umfassen folgende Kapitel:

**Kapitel 1:** Einführung in Open Roberta® und das Lego® WeDo 2.0 Set

**Kapitel 2:** Der Übergang von Lego WeDo zu NEPO®

Weitere Unterrichtseinheiten mit Open Roberta und Lego WeDo 2.0

- Objekte orten
- Einfacher Regelkreis

# Automatismen und Reflexe mit Lego WeDo

## Kurz

Im vorliegenden Unterrichtsentwurf wird der Bewegungssensor von Lego WeDo 2.0 so genutzt, dass bei Unterschreitung eines Mindestabstands eine Ampel auf Rot gestellt und/oder der Motor aktiviert und eine Schranke gesenkt wird.

## Thema

Technische Automatismen und biologische Reflexe

## Klassenstufe

3 bis 6

## Fächer

Mathematik, Sachunterricht, Biologie, Naturphänomene und Technik

## Zeitaufwand

ca. 90 Minuten

## Material (für je 1 Kind)

- Ein Tablet mit Internetanschluss (<https://lab.open-roberta.org>) oder mit vorinstallierter App »Open Roberta Mobile« für Android und iOS
- Ein Lego Education WeDo 2.0 Set

## Voraussetzungen

- Grundlegende Kenntnis der basalen Programmierblöcke von NEPO® (Open Roberta Lab)

## Kompetenzen

- Die Schüler\*innen beschreiben die grundsätzliche Funktionsweise eines Automatismus und/oder eines Reflexes in eigenen Worten.
- Die Schüler\*innen setzen die grundlegenden Blöcke der Programmiersprache NEPO® zielgerichtet ein und nutzen Kontrollblöcke (Entscheidungen, Schleifen, Warten).
- Die Schüler\*innen arbeiten im Team und erweitern ihre Kollaborations- und Kommunikationskompetenz.
- Die Schüler\*innen entwickeln ihre arithmetischen und informatischen Kompetenzen.

Über diese Kompetenzen hinaus werden mit dem vorliegenden Projekt weitere Kompetenzen gefördert, die den jeweiligen Medienkompetenzrichtlinien der einzelnen Bundesländer zugeordnet werden können. Da diese jedoch sehr vielfältig sind (sowohl was deren Bezeichnung als auch deren Struktur angeht), werden sie an dieser Stelle nicht explizit erwähnt, sondern sind situativ von der Lehrperson auszuwählen.

# 1. Grundlagen

Informatische Systeme werden häufig dazu genutzt, dass bei Eintritt eines bestimmten Ereignisses eine definierte Aktion ausgelöst («getriggert») wird. Ein gängiges Beispiel sind Türen, die sich automatisch öffnen, sobald sich eine Person nähert: Die Person geht auf die Tür zu und wird von einem Sensor registriert; die Rechneinheit greift auf den Sensorwert zu und setzt die Türsteuerung in Bewegung.

In unserer Umwelt findet sich eine Vielzahl an weiteren Beispielen, die alle nach diesem grundlegenden Prinzip funktionieren. In manchen Fällen ist der Trigger ein Knopf, der gedrückt wird, z. B. um einen Aufzug zu rufen, eine Fußgängerampel auf Grün zu schalten oder einen Kaffee zu bekommen. In anderen Fällen – wie der oben angesprochenen Tür – sind Sensoren beteiligt. Hierzu gehören beispielsweise der Rauchmelder in der Küche oder die Diebstahlsicherung im Kaufhaus.

Solche Automatismen »Ereignis → Aktion« finden sich in der Natur z. B. als Reflexe. Bekannt sind beim Menschen insbesondere der Lidschluss- und der Kniesehnenreflex als angeborene Reflexe. Es gibt aber auch konditionierte Reflexe. Einer der bekanntesten konditionierten Reflexe ist vermutlich das, was als »Pawlow'scher Hund« in die Psychologiegeschichte einging.<sup>1</sup>

Wenn Sie eines der angesprochenen Themen im Unterricht behandeln, können Sie Lego WeDo 2.0 anhand der Beispiele in dieser Unterrichtseinheit leicht zur Festigung, Vertiefung und Erweiterung der fachlichen Inhalte und Kompetenzen nutzen. Bezüge zum Bildungsplan finden Sie ab Seite 11. Der Übergang vom Fachinhalt zum Arbeiten mit WeDo 2.0 gestaltet sich dabei denkbar einfach – in der Regel genügt eine Formulierung im Sinne von »... und das wollen wir nun einmal selbst programmieren!«, gefolgt von der jeweiligen Zielsetzung als Aufgabenstellung.

---

<sup>1</sup> Siehe z. B. [https://de.wikipedia.org/wiki/Pawlowscher\\_Hund](https://de.wikipedia.org/wiki/Pawlowscher_Hund)

## 2. Aufbau

Für den Bau des Lego-Modells werden benötigt:

- ein WeDo-Smarthub
- ein Motor
- ein Bewegungssensor
- weitere Bauteile aus dem WeDo-Kasten

Der exakte Zusammenbau des Modells kann den Schüler\*innen überlassen werden.

Im einfachsten Fall genügt es, den Sensor auf den Hub zu stecken, ohne dass weitere Bauteile erforderlich sind. Wichtig ist dabei jedoch, dass der Sensor »freie Sicht« hat. Die folgende Abbildung soll lediglich als ein Beispiel dienen, wie das Objekt aussehen kann.

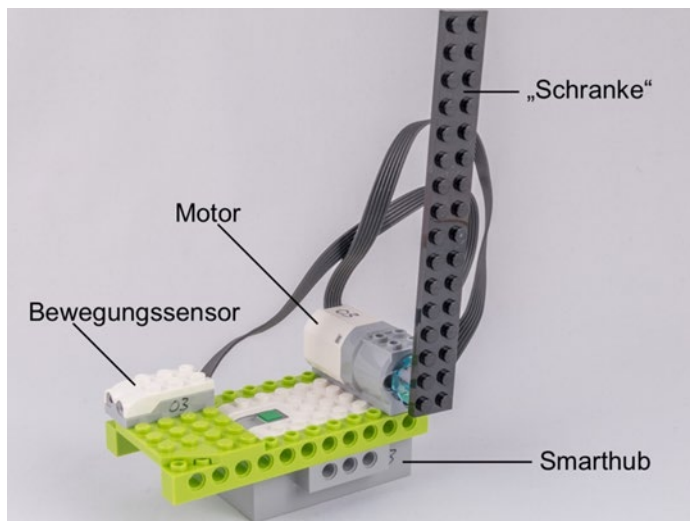


Abbildung 1: Das Lego-Modell für die vorliegende Unterrichtseinheit

**Tipp:** Wenn Sie den Zusammenbau den Schüler\*innen überlassen, geben Sie ihnen eine Zeit vor (z. B. 10 Minuten). Erfahrungsgemäß zieht sich der Zusammenbau ansonsten sehr in die Länge.

Im Open Roberta Lab bzw. in der Open Roberta Mobile App können Sie in »Roboterkonfiguration« einstellen, welches Bauteil mit welchem Anschluss verwendet werden soll. Hier ist es wichtig, dass der Infrarotsensor dem Anschluss zugeordnet ist, an welchem er am Roboter tatsächlich angeschlossen ist (Reihenfolge von links an rechts. Der linke Anschluss ist der Erste). Welches Bauteil welchen Anschluss erhält, ist nicht relevant.

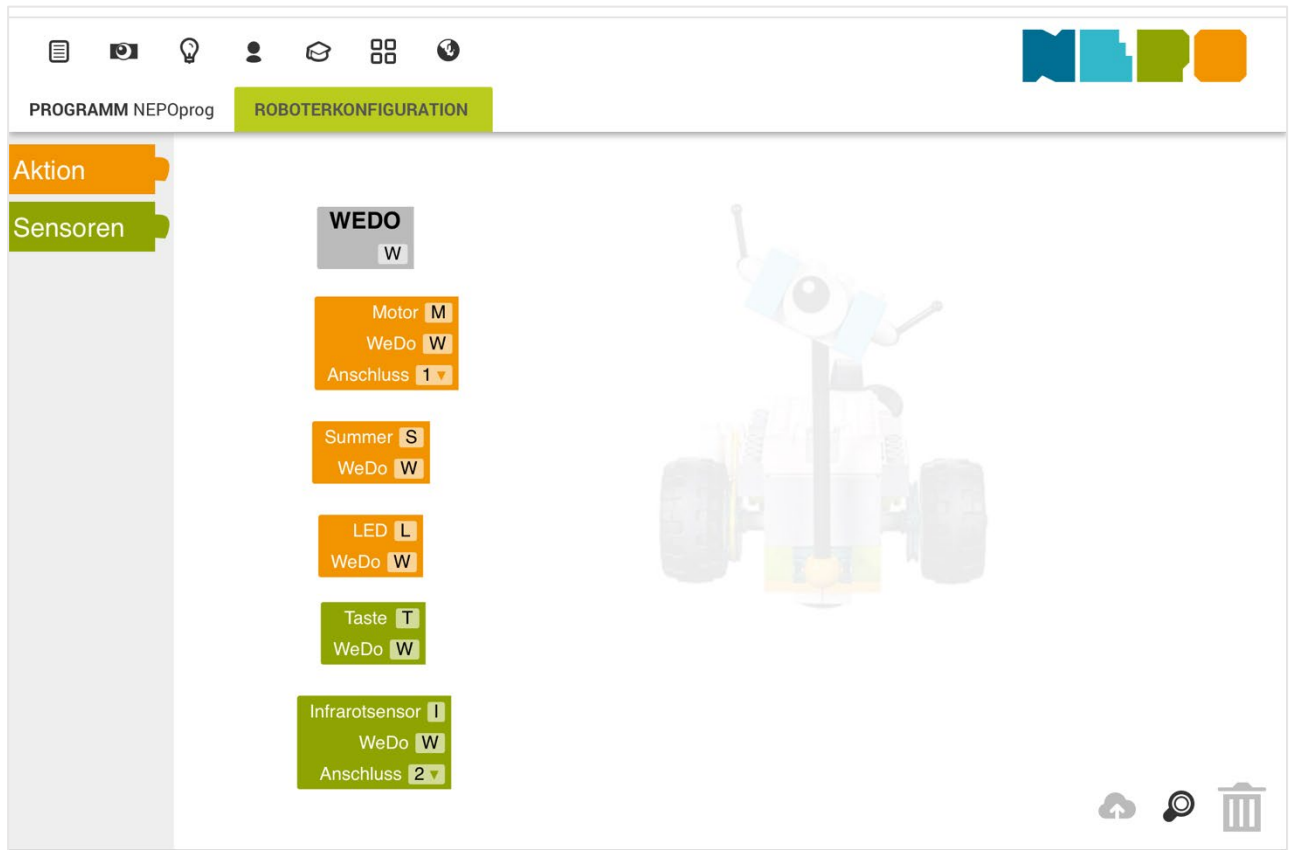


Abbildung 2: Die Roboterkonfiguration

### 3. Programmierung

Die Programmierung in NEPO kann auf verschiedenen Schwierigkeitsstufen durchgeführt werden.

#### Einfach: Die Ampel

**Zielsetzung:** Wenn sich ein Objekt nahe am Bewegungssensor befindet, soll die LED am Smarthub rot leuchten. Wenn sich kein Objekt nahe am Bewegungssensor befindet, soll sie grün leuchten.

Für die einfache Variante kann auf den Motor und die dort befestigte »Schranke« verzichtet werden. Alle benötigten Befehle befinden sich in »NEPO-Blöcke Anfänger«. In seiner einfachsten Form sieht das Programm wie folgt aus:

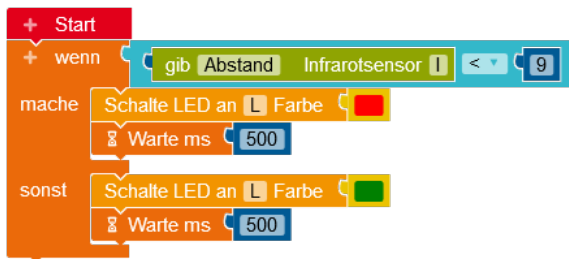


Abbildung 3: Die Ampel-Programmierung (einfach)

Der Nachteil dieses Programms besteht darin, dass pro Durchgang nur eine Prüfung durchgeführt wird. Wenn man eine zweite Prüfung durchführen möchte, muss das Programm ein zweites Mal gestartet werden. Dies lässt sich durch das Hinzufügen einer Schleife umgehen. Besonders praktisch ist hierbei die Schleife »Wiederhole ist« (in »NEPO-Blöcke Experte«) in Verbindung mit dem Sensor »Taste T gedrückt?«. Damit läuft die Schleife so lange, bis die grüne Taste am Smarthub gedrückt wird.

Damit ändert sich die Zielsetzung folgendermaßen: Die LED am Smarthub soll solange grün leuchten, bis ein Objekt dem Bewegungssensor genähert wird. Dann soll sie rot leuchten. Wird das Objekt wieder entfernt, soll die LED erneut grün leuchten.

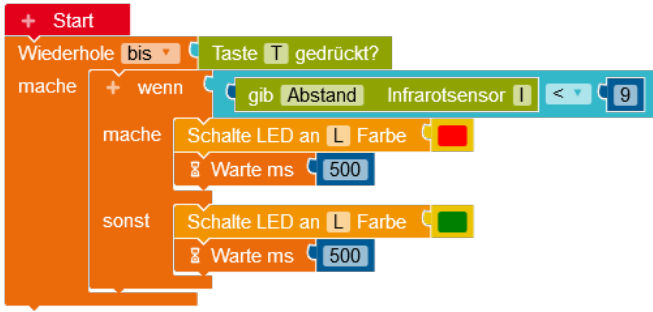


Abbildung 4: Die Ampel-Programmierung mit Schleife

## Mittel: Die Schranke ohne »Wenn«

**Zielsetzung:** Wenn ein Objekt dem Bewegungssensor genähert wird, soll sich die Schranke schließen, 5 Sekunden unten bleiben und sich dann wieder öffnen.

Für den mittleren Schwierigkeitsgrad werden Motor und »Schranke« benötigt (siehe Abbildung 1). Mithilfe von »Warte bis« wird das Schließen der Schranke ausgelöst, sobald der Abstand eines Objekts zum Bewegungssensor einen bestimmten Wert unterschreitet. Nach dem Schließen der Schranke wird fünf Sekunden gewartet, bis sie wieder öffnet.

Beachten Sie:

- Je nachdem, in welcher Orientierung der Motor im Lego-Modell eingebaut ist, muss ggf. das Vorzeichen getauscht werden. In Abbildung 5 steht das negative Vorzeichen bei »Tempo %« für das Schließen und das positive Vorzeichen für das Öffnen der Schranke.
- Bei den Werten für »Zeit ms« muss ein wenig experimentiert werden. Es ist durchaus möglich, dass für das Schließen und Öffnen der Schranke verschiedene Werte eingetragen werden müssen.

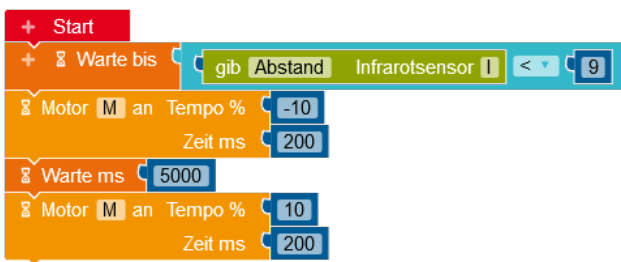


Abbildung 5: Die Schranken-Programmierung mit »Warte bis«



## Schwierig: Die Schranke mit »Wenn«

**Zielsetzung:** Die Schranke soll solange offen sein, bis ein Objekt dem Bewegungssensor genähert wird. Dann soll sie sich schließen. Wird das Objekt wieder entfernt, soll sich die Schranke erneut öffnen.

Die Schranke analog zur Ampel-Programmierung mit einem »Wenn« zu erstellen, ist etwas für fortgeschrittene Programmiererinnen und Programmierer. Der Hintergrund ist folgender: Beim Ampel-Beispiel wird das Licht auf Rot gestellt, sobald der Abstand zum Sensor unterschritten wird. Bleibt der Abstand unterschritten, leuchtet die rote LED einfach weiter. Überträgt man dieses Vorgehen auf die Schranke, so senkt sie sich, sobald der Abstand zum Bewegungssensor zum ersten Mal unterschritten wird. Da der Sensor jedoch regelmäßig den Abstand misst, wird der Motor weitere Male angesteuert – aus der Schranke wird dann eine Windmühle! Abhilfe schafft eine Kontrollvariable:

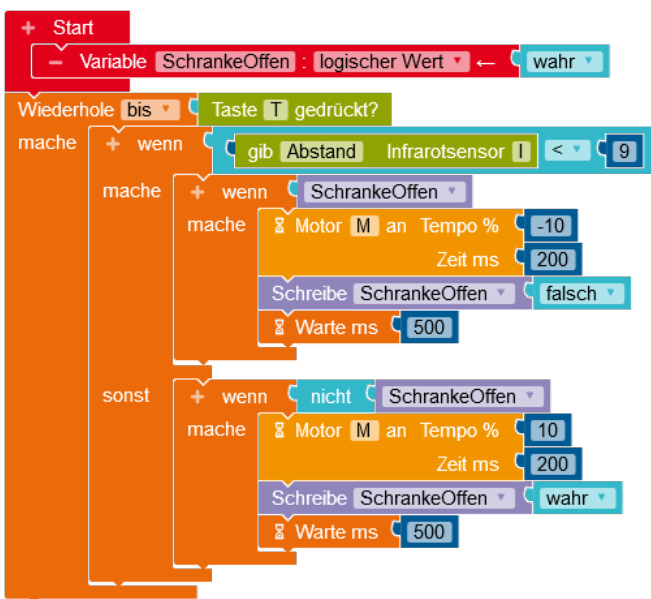


Abbildung 6: Die Schranken-Programmierung mit »Wenn« und einer Kontrollvariablen

### Erläuterung der relevanten Befehle:

Durch Antippen des +-Symbols bei »Start« wird eine Variable namens »SchrankeOffen« definiert (Name frei wählbar). In diesem Beispiel ist die Variable vom Typ »logischer Wert« und kann damit die Werte »wahr« und »falsch« annehmen. Der Ausgangswert ist »wahr« (= die Schranke ist offen). Alternativ kann ein anderer Variablentyp gewählt werden; dann müssen die folgenden Punkte entsprechend abgeändert werden.

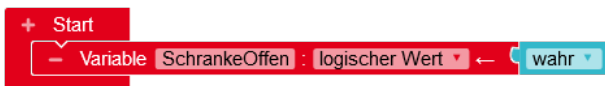


Abbildung 7: Die Variable »SchrankeOffen«

Innerhalb des »mache«-Abschnitts im äußeren Wenn-Block kommt die Kontrollvariable zum Einsatz:  
Wenn die Schranke offen ist, mache ...

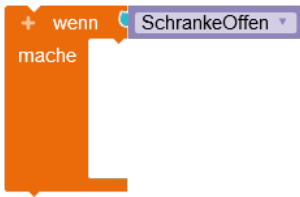


Abbildung 8: Bedingung, wenn Schranke geöffnet

Nach dem Schließen der Schranke über den »Motor«-Befehl wird die Variable auf »falsch« gesetzt. Dadurch wird vermieden, dass der Motor ein weiteres Mal in Schließrichtung aktiviert wird.



Abbildung 9: Änderung des Wertes der Variable

Für das Öffnen der Schranke gilt das Gesagte mit »umgekehrtem Vorzeichen«: Geöffnet werden darf die Schranke nur dann, wenn sie nicht bereits offen ist.

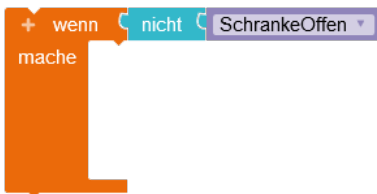


Abbildung 10: Bedingung, wenn Schranke geschlossen

Wurde die Schranke geöffnet, muss die Variable wieder auf »wahr« gesetzt werden.



Abbildung 11: Änderung des Wertes der Variable

## 4. Bezug zum Bildungsplan

Die vorliegende Unterrichtseinheit bietet sich unmittelbar an, wenn technische Automatismen oder biologische Reflexe behandelt werden. Weitere Anknüpfungspunkte und Kompetenzen werden im Folgenden genannt.<sup>2</sup>

### Klasse 3/4

#### Mathematik

Beim Arbeiten im Team und beim Erarbeiten der Programme entwickeln die Schüler\*innen die prozessbezogenen Kompetenzen »Kommunizieren« und »Argumentieren« sowie »Problemlösung« und »Modellieren«.

Durch die Eingabe verschiedener Werte bei der Abstandsmessung mit dem Sensor festigen die Schüler\*innen ihr Vorstellungsvermögen zu Raum und Größen.

#### Sachunterricht

Durch den Bezug auf zugrundeliegende Phänomene wie automatische Türen, Druckknöpfe bei Fußgängerampeln oder Rauchmeldern entwickeln die Schüler\*innen die prozessbezogenen Kompetenzen »Welt erleben und wahrnehmen« und »Welt erkunden und verstehen«. Durch das Programmieren wird zudem »In der Welt handeln – Welt gestalten« angesprochen.

Weiterhin kann das Thema Reflexe (Lidschluss-, Kniesehenreflex) als Experiment im Bereich »Körper und Gesundheit« behandelt und als Basis für die Programmierung genutzt werden.

---

<sup>2</sup> Als Grundlage dienen die Bildungspläne des Landes Baden-Württemberg, siehe <http://www.bildungsplaene-bw.de>

## Klasse 5/6

### Biologie, Naturphänomene und Technik

Beim Arbeiten im Team und beim Erarbeiten der Programme entwickeln die Schüler\*innen die prozessbezogenen Kompetenzen »Erkenntnisgewinnung«, »Kommunikation«, »Bewertung« und »Herstellung«.

Inhaltlich kreieren die Schüler\*innen eigene Wege, um die ihnen gestellte (Programmier-) Aufgabe zu lösen. Anhand eines konkreten Beispiels wie der Ampel oder der Schranke lernen sie, dass Menschen technische Objekte anfertigen, um eine bestimmte Aufgabe zu erfüllen.

### Mathematik

Beim Arbeiten im Team und beim Erarbeiten der Programme entwickeln die Schüler\*innen die prozessbezogenen Kompetenzen »Argumentieren«, »Problemlösung«, »Modellieren«, »mit symbolischen, formalen und technischen Elementen (der Mathematik) umgehen« sowie »Kommunizieren«.

Inhaltlich lassen sich die folgenden Leitideen wiederfinden:

- Zahl – Variable – Operation
- Messen
- Raum und Form
- Funktionaler Zusammenhang

# Kontakt

Die Roberta-Initiative im Web

[roberta-home.de](https://roberta-home.de)

[lab.open-roberta.org](https://lab.open-roberta.org)

FAQ rund um die Roberta-Initiative

[roberta-home.de/faq](https://roberta-home.de/faq)

Informationen zum Datenschutz

[roberta-home.de/datenschutz](https://roberta-home.de/datenschutz)



## Info

Dieses Material wurde zusammen mit Prof. Dr. Julia Knopf und Prof. Dr. Silke Ladel entwickelt.

Dieses Material entstand mit Unterstützung der Google Zukunftswerkstatt.

Lizenz: CC-BY-SA 4.0

Version: 1.2

Stand: November 2019

## Warenzeichen

Roberta, Open Roberta und NEPO sind eingetragene Warenzeichen der Fraunhofer-Gesellschaft e.V.

Lego ist ein eingetragenes Markenzeichen der Lego Group.