

Name: _____ Klasse: _____ Datum: _____

Die erste Station ist der Transport-Roboter, der Waren aus der Lade-Zone in die Anlage transportiert. Wir nennen ihn „Smarter Schieber“.

Siehst du die breite schwarze Linie auf der weißen Fahrbahn? Mit dem Multisensor an seiner Unterseite kann der Roboter schwarz und weiß unterscheiden. So erkennt er, ob er sich noch auf der Fahrbahn befindet oder davon abweicht.

Wenn du ihn richtig programmierst, fährt er sogar um die Kurven, ohne zu wissen, dass es welche sind.

Erklärung: Der Multisensor hat zwei Sensoren um die Linie zu erkennen. Dadurch entstehen vier verschiedene Möglichkeiten (Fahrtrichtung oben):

A	B	C	D

© Cornelsen Experimenta Illustrationen Arianna: Katjenka Krause; eXperiBot: Kinematics GmbH

Aufgabe:

Stelle eine Vermutung an, wie der Roboter seine zwei Motoren steuern muss (vorwärts oder rückwärts), um auf der Linie weiterfahren zu können:

- A) Ende der Linie: anhalten
- B) Motor links Drehrichtung: _____ Motor rechts Drehrichtung: _____
- C) Motor links Drehrichtung: _____ Motor rechts Drehrichtung: _____
- D) Motor links Drehrichtung: _____ Motor rechts Drehrichtung: _____

Programmieraufgabe

Schreibe ein Programm, das den Roboter aus der Lade-Zone bis zum Greifarm fahren lässt und dort anhält. Teste dein Programm und korrigiere mögliche Fehler.

Du kannst auch ein Warenstück in seine Gabel legen und schauen, ob er den Transport schafft.

Befehlsbox



Erweitere das Programm, sodass der Roboter seine Bewegung immer wiederholt und laufend Waren aus der Lade-Zone transportiert.

Du hast sicher bemerkt, dass der Roboter auch ohne Ladung seinen Weg fährt.

- ➡ Baue eine Objekt-Erkennung mit dem zweiten Multisensor in das Programm ein, damit der Roboter anhält, wenn er keine Ladung in der Gabel hat.

Name: _____ Klasse: _____ Datum: _____

```

Starte Programm
wiederhole unendlich
  mache
    + falls
      mache Double Motor setze Geschwindigkeit auf 0 % und Geschwindigkeit auf 10 % für 0.1 s
    sonst falls
      mache Double Motor setze Geschwindigkeit auf 10 % und Geschwindigkeit auf 0 % für 0.1 s
    sonst falls
      mache
        Double Motor fahre rückwärts für 1 s mit Geschwindigkeit mittel
        Double Motor setze Geschwindigkeit auf -10 % und Geschwindigkeit auf 10 %
        warte 1 s
        warte bis
      sonst
        Double Motor setze Geschwindigkeit auf 15 % und Geschwindigkeit auf 15 % für 0.1 s
  
```



Was ist denn hier los? In diesem Programm für den Schieber fehlen ja sämtliche Bedingungen! So kann der Multisensor gar nicht abgefragt werden! Kannst du die richtigen Bedingungen einsetzen, sodass der Schieber der Linie folgt?

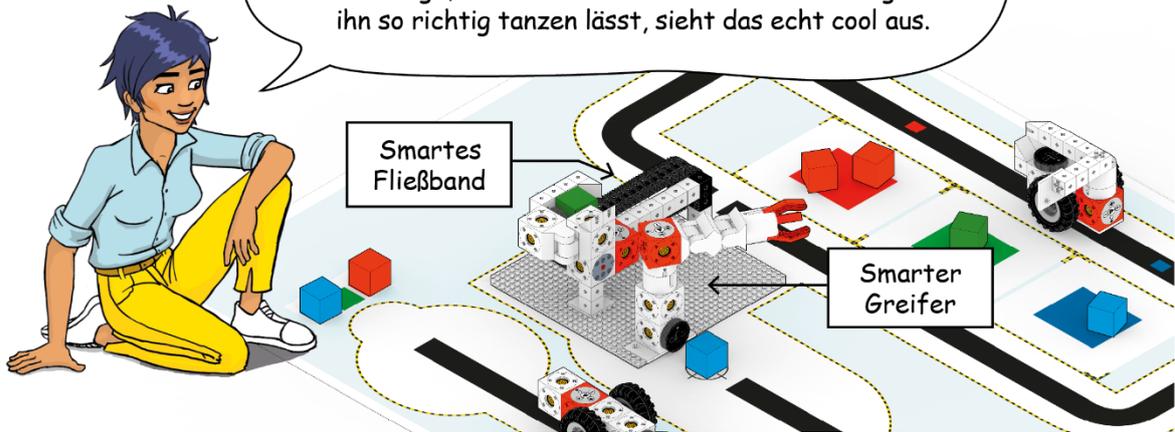
© Cornelsen Experimenta Illustrationen Arianna: Katjenka Krause

➡ Nutze dazu diese Blöcke:

Name: _____ Klasse: _____ Datum: _____

Jetzt wird es Zeit für's Roboterballett!
 Ich muss immer an eine Tanzaufführung denken, wenn sich viele Roboterarme gleichzeitig bewegen, um ihre Aufgaben zu erfüllen.
 Unser Roboterarm hat einen Greifer, mit dem er Waren greifen kann, die vom Transport-Roboter geliefert wurden.

Unser **Smarter Greifer** soll die Waren auf ein Fließband heben, damit sie in den Sortier-Bereich gelangen können.
 Keine Sorge, das ist nicht schwer. Wenn dein Programm ihn so richtig tanzen lässt, sieht das echt cool aus.



© Cornelsen Experimenta Illustrationen Arianna: Katjenka Krause; eXperiBot: Kinematics GmbH

1. Programmieraufgabe

Untersuche, welches Roboter-Modul welche Bewegungen ausführen kann. Schreibe dazu jeweils ein kleines Programm, das nur jeweils ein Modul steuert.

In der Befehlsbox ist ein Beispiel für den Twister:

So lernst du die einzelnen Module kennen. Die Winkelangaben für Twister und Pivot können Werte von -90° bis 90° annehmen.

- ➔ Gib dem Grabber mal etwas zum Festhalten, während er sich schließt.

Befehlsbox



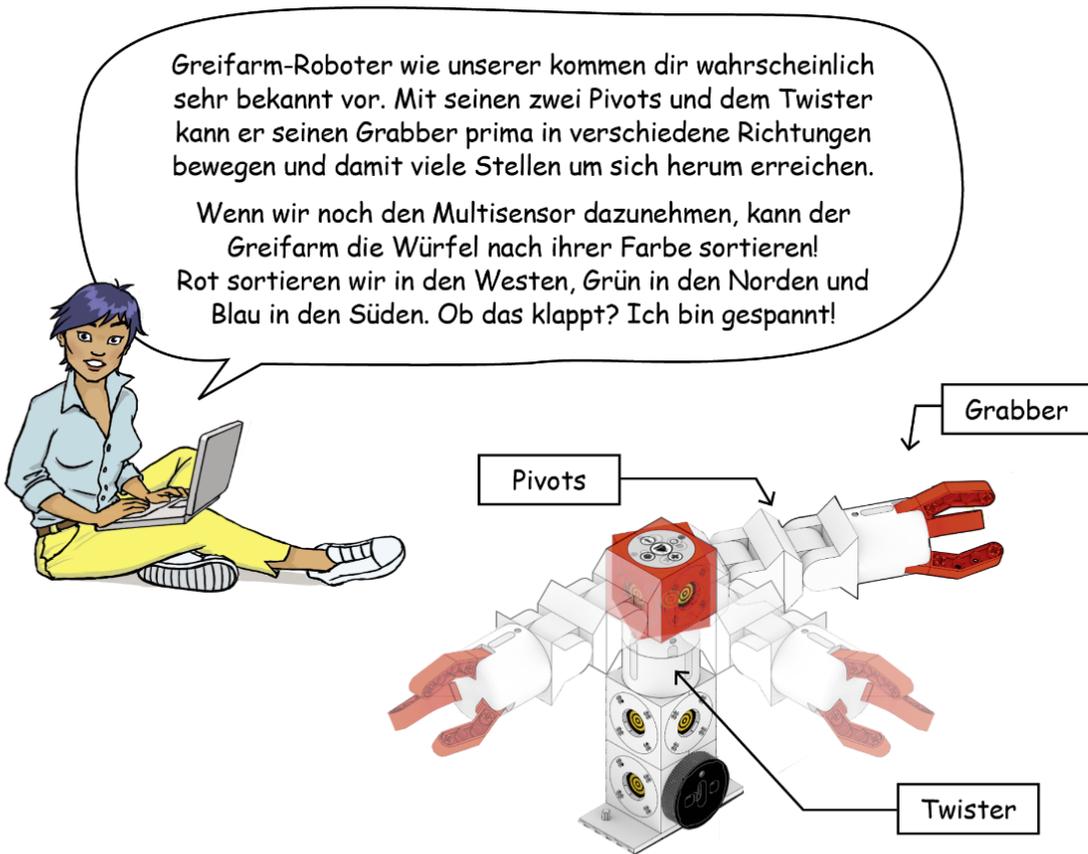
2. Programmieraufgabe:

- ➔ Plane den gesamten Bewegungsablauf. Lege fest, welche Winkel die Module nacheinander einnehmen sollen, um den Grabber zum Warenstück zu bewegen. Danach lass den Roboter zugreifen und das Warenstück zum Fließband transportieren.



Erweitere dein Programm, sodass die Bewegung erst dann beginnt, wenn ein Warenstück im markierten Bereich eintrifft. Nutze die Objekterkennung des Multisensors.

Name: _____ Klasse: _____ Datum: _____

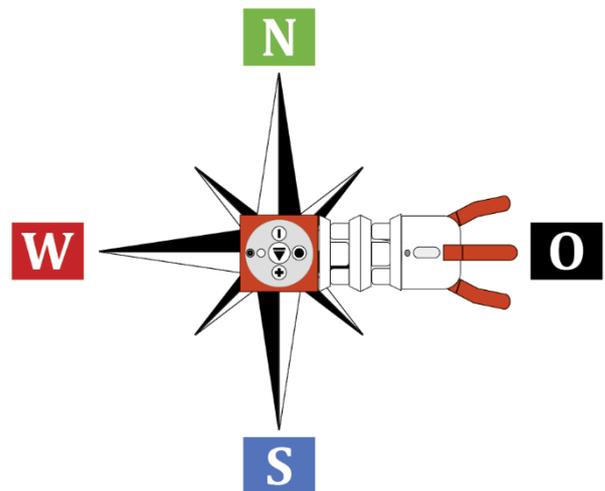


© Cornelsen Experimenta Illustrationen Arianna: Katjenka Krause; eXperiBot: Kinematics GmbH

Programmieraufgabe:

Schau dir die Windrose an: Im Osten des Greifarms ist die Ladezone, in der die Waren vom Schieber abgelegt werden können. Die Waren sollen im Osten gegriffen und im Norden, Süden oder Westen abgelegt werden, je nach ihrer Farbe.

1. *Programmiere die Bewegung für eine einzelne Farbe, z. B. rot.*
2. *Teste dein Programm und ändere es, bis es gut funktioniert.*
3. *Erweitere dein Programm um die Erkennung der zwei anderen Farben. Programmiere auch die Bewegungen.*
4. *Teste dein Programm und ändere es, bis es alle Anforderungen erfüllt.*



➡ **Tipp:** Das Twister-Modul kann sich zwar nur um 180° drehen, aber Du hast ja auch noch zwei Pivots, mit denen du mit etwas Programmiergeschick die dritte Richtung erreichen kannst.

Name: _____ Klasse: _____ Datum: _____

Programm A

```

Starte Programm
wiederhole unendlich
  mache
    Pivot drehe nach 0 °
    Pivot drehe nach 90 °
    Twister drehe nach 90 °
    warte bis Multisensor erkennt Objekt
    Powerbrain spiele Note C6 für 1/4 Schlag
    warte 2 s
    Grabber öffnen
    Pivot drehe nach -30 °
    Pivot drehe nach -60 °
    Grabber schließen
    Pivot drehe nach 0 °
    Pivot drehe nach 90 °
    Twister drehe nach -90 °
    Pivot drehe nach -90 °
    Grabber öffnen
    
```

Programm B

```

Starte Programm
wiederhole unendlich
  mache
    Pivot drehe nach 0 °
    Pivot drehe nach 0 °
    Twister drehe nach 0 °
    warte bis Multisensor erkennt Objekt
    Powerbrain spiele Note C6 für 1/4 Schlag
    warte 6 s
    Twister drehe nach -90 °
    Pivot drehe nach -90 °
    Pivot drehe nach -60 °
    Grabber schließen
    Pivot drehe nach 0 °
    Grabber öffnen
    Pivot drehe nach 90 °
    Grabber öffnen
    Pivot drehe nach -30 °
    
```

Programm C

```

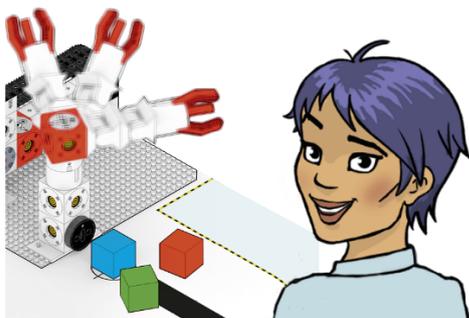
Starte Programm
wiederhole unendlich
  mache
    Pivot drehe nach 0 °
    Grabber schließen
    Pivot drehe nach 0 °
    Powerbrain spiele Note C6 für 1/4 Schlag
    Twister drehe nach -90 °
    Pivot drehe nach -90 °
    Grabber öffnen
    warte 6 s
    Grabber öffnen
    Pivot drehe nach -30 °
    Pivot drehe nach -60 °
    Pivot drehe nach 90 °
    warte bis Multisensor erkennt Objekt
    Pivot drehe nach 90 °
    Twister drehe nach 90 °
    
```

Jetzt schau dir das an!
 Was macht denn der Greifarm da?
 Das sieht ja aus wie Breakdance!

Und die Waren lässt er einfach stehen!
 Da muss jemand komplett den Bewegungs-
 ablauf durcheinandergebracht haben.

Ich sehe schon: Hier wartet eine große
 Aufgabe auf uns.

Hilfst du mir, das Programm für den
richtigen Ablauf zu finden?

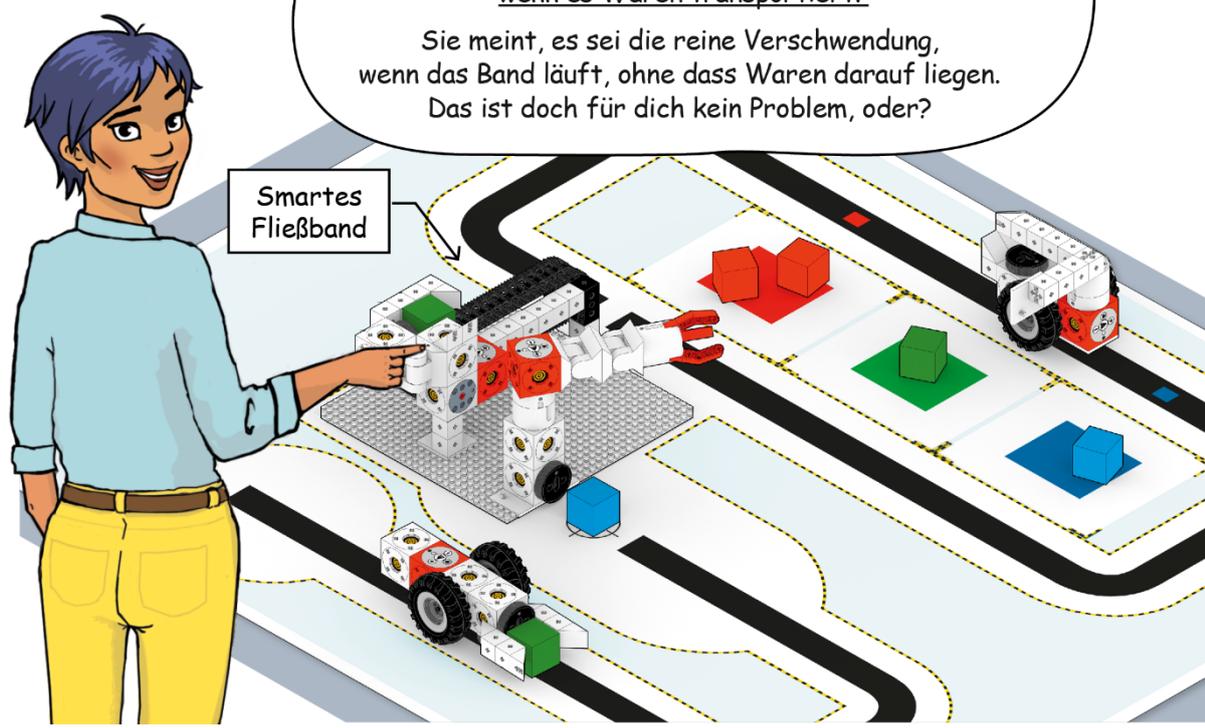


© Cornelsen Experimenta Illustrationen Arianna: Katjenka Krause; eXperiBot: Kinematics GmbH

Name: _____ Klasse: _____ Datum: _____

An dieser Stelle soll das **Smarte Fließband** entstehen. Außer Waren zur nächsten Station zu befördern, kann es auch erkennen, wie schnell sie transportiert werden dürfen oder welche Station ihr Ziel ist.

Von unserer Energiebeauftragten haben wir noch eine **zusätzliche Anforderung** erhalten:
Das Fließband soll sich nur dann bewegen, wenn es Waren transportiert.
 Sie meint, es sei die reine Verschwendung, wenn das Band läuft, ohne dass Waren darauf liegen. Das ist doch für dich kein Problem, oder?

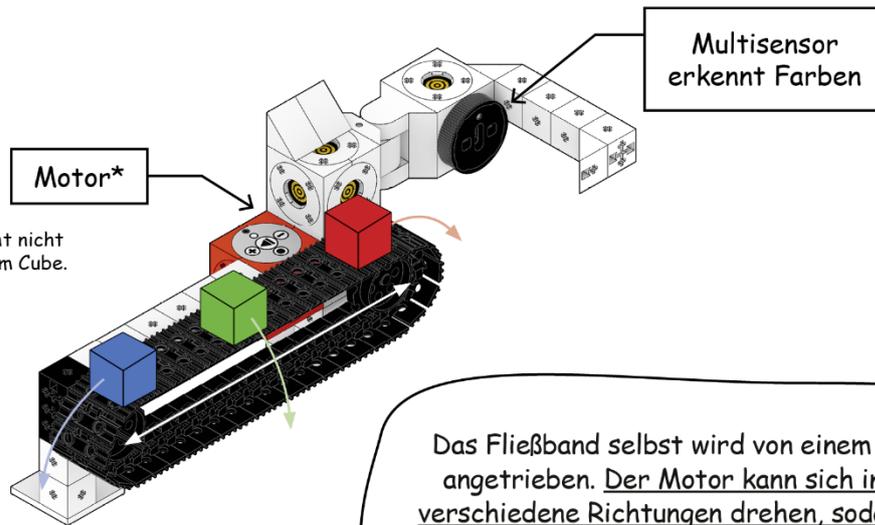


© Cornelsen Experimenta Illustrationen Arianna: Katjenka Krause; eXperiBot: Kinematics GmbH

<p>Programmieraufgabe</p> <p>Das Fließband soll sich einschalten, wenn der Multisensor ein Objekt erkennt.</p> <p>Erstelle das Programm und probiere es aus.</p>	<p>Befehlsbox</p>
---	--------------------------

Lernloop → Pakete oder Objekte, die nicht rot, grün oder blau sind, sollen aussortiert werden. Erweitere das Programm und nutze den Pivot, um die Pakete zu sortieren.

Name: _____ Klasse: _____ Datum: _____



* in dieser Ansicht nicht sichtbar unter dem Cube.

Hier siehst du ein super cleveres Fließband, mit dem Waren nach ihren Farben sortiert werden sollen. Noch klappt das nicht so ganz, aber gemeinsam schaffen wir das bestimmt.



Das Fließband selbst wird von einem Motor angetrieben. Der Motor kann sich in zwei verschiedene Richtungen drehen, sodass sich die Waren auf dem Fließband vor- oder zurückbewegen können.

Der Multisensor an einem Ende des Fließbands kann die Farben der Waren erkennen. Dann wird der Motor so gesteuert, dass die blauen Waren an einem Ende und die roten Waren am anderen Ende abgelegt werden.
Doch wie schaffen wir es, dass grünen Waren in der Mitte vom Band abgeladen werden? Knifflig!

© Cornelsen Experimenta Illustrationen Arianna: Katjenka Krause; eXperiBot: Kinematics GmbH

Programmieraufgabe

1. Schreibe ein Programm, das ein Paket auf dem Band erkennt und die Paketfarbe erfassen kann.
2. Programmiere die Farberkennung mithilfe des Multisensors.
3. Lege fest, welche Bewegungen das Fließband und der Pivot für welche Farbe ausführen sollen.
4. Teste dein Programm und ändere Winkel, Wartezeiten oder Abfolge der Befehle.

Befehlsbox

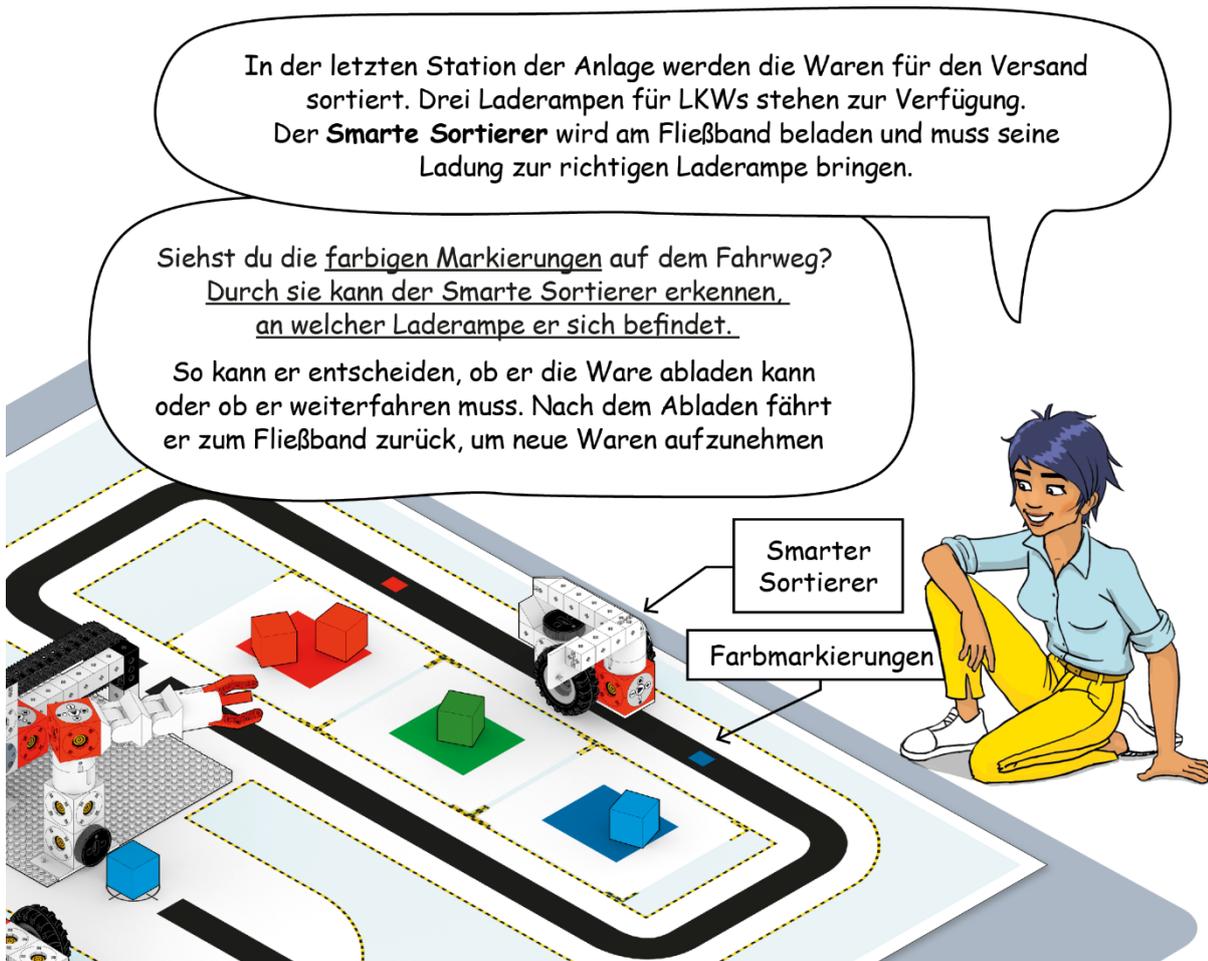
Multisensor erkennt Objekt

Multisensor erkennt Farbe Rot



Konstruiere ein neues Fließband, welches eine Paket-Art mithilfe eines weiteren Pivots in der Mitte des Fließbandes auswirft.

Name: _____ Klasse: _____ Datum: _____



© Cornelsen Experimenta Illustrationen Arianna: Katjenka Krause; eXperiBot: Kinematics GmbH

Programmieraufgabe

1. Programmiere den Sortier-Roboter so, dass er eine Runde um den Sortier-Bereich herumfährt und am Fließband wieder stoppt.
2. Erweitere das Programm, sodass die Farbe des Pakets erkannt wird und der Roboter am richtigen Ausgabe-Bereich stoppt.
3. Lass den Roboter das Paket in den richtigen Ausgabe-Bereich auswerfen (Twister bewegen).
4. Vervollständige das Programm, indem du den Roboter nach dem Auswerfen weiter zum Fließband fahren lässt.

Befehlsbox

```

um farbe_oben
falls Multisensor erkennt Farbe Rot gib zurück 1
falls Multisensor erkennt Farbe Grün gib zurück 2
falls Multisensor erkennt Farbe Blau gib zurück 3
gib zurück 0
    
```



Das bisherige Programm funktionierte zwar, dafür konnte der Roboter aber nur sehr langsam fahren. Das kostet Zeit!
Schaffst du es, dass er den Rundkurs in weniger als einer Minute schafft?

Name:

Klasse:

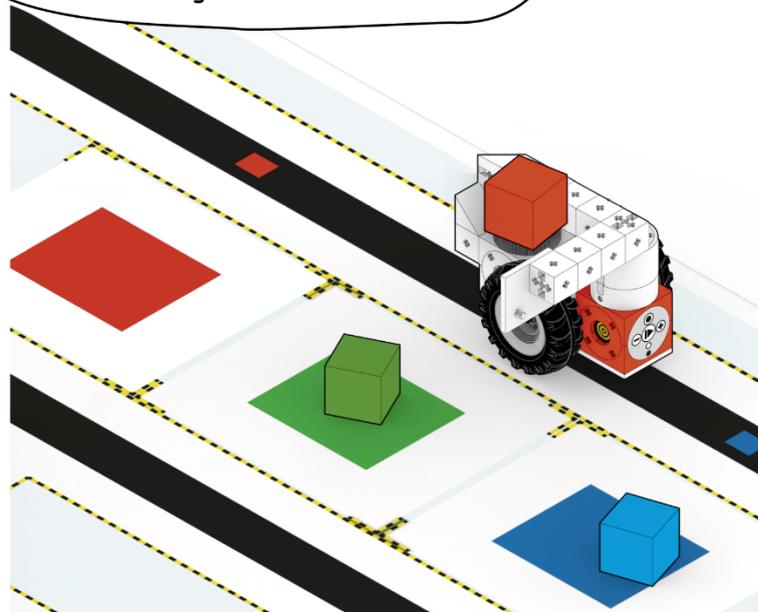
Datum:

Verflix, schon wieder ist der Sortierer über die rote Markierung gefahren, ohne das rote Paket auszuwerfen. Vorhin hat das doch geklappt!

Ach, das liegt bestimmt daran, dass jetzt die Sonne in unser Büro scheint und den Farb-Sensor stört. Oder stimmt etwa die Farbgestaltung des Roboter-Weges nicht?

Wir brauchen unbedingt ein Programm, mit dem wir testen können, ob der Multi-sensor die Farben richtig erkennt.

Kriegst du das hin??



© Cornelsen Experimenta Illustrationen Arianna: Katjenka Krause; eXperiBot: Kinematics GmbH

- ➔ **Hinweis:** Schreibe ein Programm, das eine erkannte Farbe durch eine Anzahl Pieptöne anzeigt, z. B.: 1 x Piep = Rot ... 2 x Piep = Grün ... 3 x Piep = Blau.

Name: _____ Klasse: _____ Datum: _____

Hier hat einer unserer Entwickler eine Funktion für den Sortier-Roboter programmiert.

Sie soll „wahr“ zurückgeben, wenn die Farbe einer Ware (Sensor oben / grün) gleich der Farbe auf dem Fahrweg ist (Sensor rot / unten).

Irgendwie klappt das nicht so ganz... Die Funktion gibt immer „wahr“ zurück! Sie sieht für mich aber ganz okay aus. Kannst du den Fehler finden?



© Cornelsen Experimenta Illustrationen Arianna: Katjenka Krause

- ➔ Finde und korrigiere die Fehler. Übernimm dazu erst das Programm in den eXperiBot-Editor, probiere es aus und ändere es.

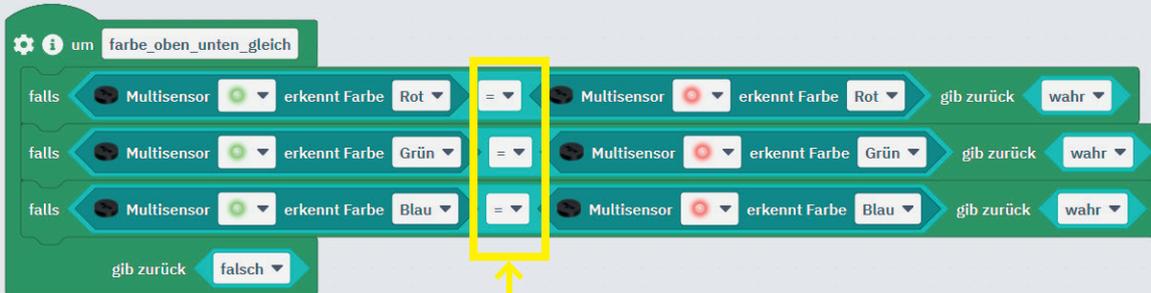
```
um farbe_oben_unten_gleich
falls (Multisensor oben erkennt Farbe Rot) = (Multisensor unten erkennt Farbe Rot) gib zurück wahr
falls (Multisensor oben erkennt Farbe Grün) = (Multisensor unten erkennt Farbe Grün) gib zurück wahr
falls (Multisensor oben erkennt Farbe Blau) = (Multisensor unten erkennt Farbe Blau) gib zurück wahr
gib zurück falsch
```

Name

Klasse

Datum

Fehlerhaftes Programm



Statt „=“ muss „und“ verwendet werden.

Korrektes Programm

