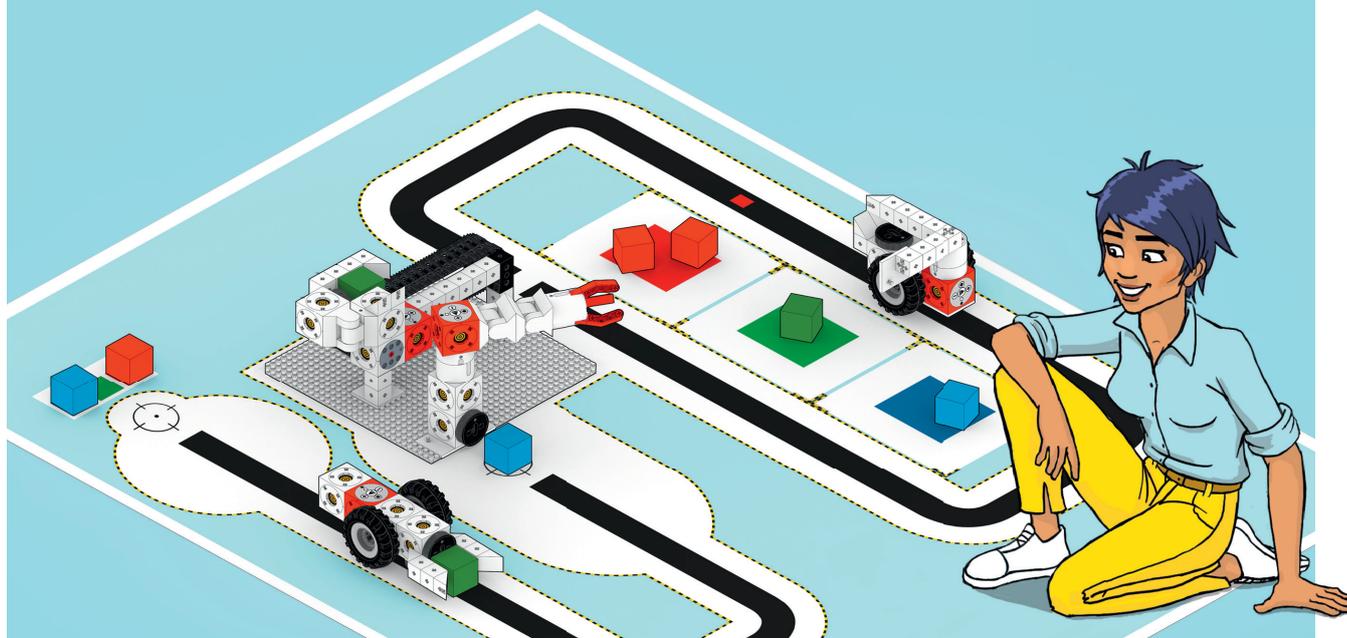


# eXperiBot

Unterrichtsmaterialien  
Smarte Fabrik



Handreichung

## Das gehört dazu

Zum eXperiBot Lernroboter gehören fertige Arbeitsblätter mit Projekten für den Unterricht natürlich dazu.

eXperiBot

Explorer-Sets

Fabrik-Set



+ weitere Unterrichtsmaterialien

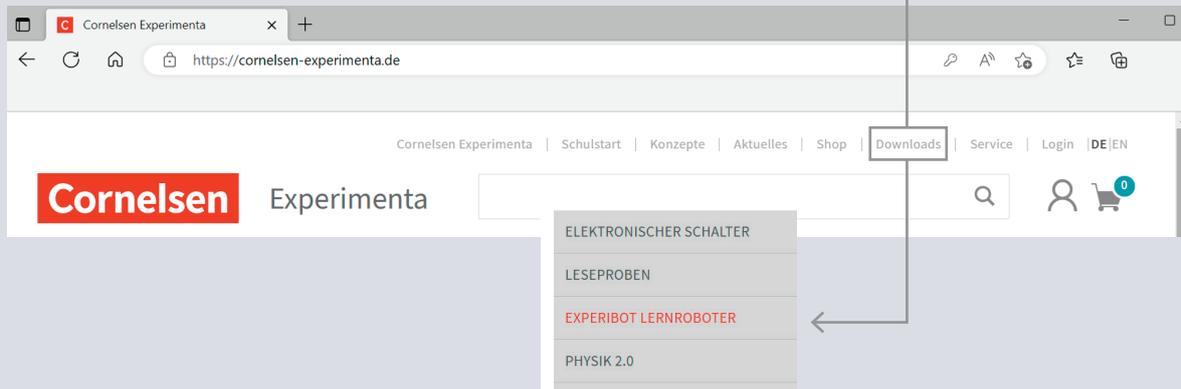


Alle Informationen zur den Produkten der eXperiBot-Serie finden Sie unter [cornelsen-experimenta.de](https://cornelsen-experimenta.de)



Arbeitsblätter  
online

Alle Arbeitsblätter finden Sie online unter [cornelsen-experimenta.de](https://cornelsen-experimenta.de) im Downloadbereich.



Produktentwicklung: Lars Pelz, Leonhard Oschütz, Uwe Biesel

Gesamtgestaltung: Katharina Meyer

Illustrationen Arianna: Katjenka Krause, Abbildungen eXperiBot: Leonhard Oschütz

[cornelsen-experimenta.de](https://cornelsen-experimenta.de)

Dieses Werk enthält Vorschläge und Anleitungen für Untersuchungen und Experimente. Vor jedem Experiment sind mögliche Gefahrenquellen zu besprechen. Beim Experimentieren sind die Richtlinien zur Sicherheit im Unterricht einzuhalten.

Die Webseiten Dritter, deren Internetadressen in diesem Lehrwerk angegeben sind, wurden vor Drucklegung sorgfältig geprüft. Cornelsen Experimenta übernimmt keine Gewähr für die Aktualität und den Inhalt dieser Seiten oder solcher, die mit ihnen verlinkt sind.

Das Werk und seine Teile sind urheberrechtlich geschützt. Jede Nutzung in anderen als den gesetzlich zugelassenen Fällen bedarf der vorherigen schriftlichen Einwilligung des Verlages. Hinweis zu §§ 60 a, 60 b UrhG: Weder das Werk noch seine Teile dürfen ohne eine solche Einwilligung an Schulen oder in Unterrichts- und Lehrmedien (§ 60 b Abs. 3 UrhG) vervielfältigt, insbesondere kopiert oder eingescannt, verbreitet oder in ein Netzwerk eingestellt oder sonst öffentlich zugänglich gemacht oder wiedergegeben werden. Dies gilt auch für Intranets von Schulen.

© 2024 Cornelsen Experimenta GmbH, Berlin

Mit der **eXperiBot Produktserie** können Kinder, Jugendliche und Erwachsene spielerisch coden lernen und experimentieren. Die **eXperiBot Sets** und **Unterrichtsmaterialien** sind passgenau für den Einsatz in Schulen, Workshops, Labs und anderen Bildungseinrichtungen konzipiert und auf die Rahmenlehrpläne abgestimmt. Hiermit erhalten Kinder und Jugendliche die Möglichkeit, sich mit den Themen *Programmierung*, *Robotik* und *Sensorik* vertraut zu machen.



**eXperiBot Blockly** ist eine grafische Programmierumgebung (Blockprogrammierung), die speziell für das Programmieren der eXperiBot Lernroboter entwickelt wurde. Mit der intuitiven Drag & Drop-Funktion können grundlegende Programmierkonzepte, wie Schleifen, Variablen, Bedingungen etc., vermittelt werden.

Mithilfe der verschiedenen Programmierblöcke können so schon Kinder ab 10 Jahren erste Erfahrungen mit dem Programmieren machen. Durch die Möglichkeit, sich den Code auch in der Programmiersprache *Python* anzeigen zu lassen, ist eXperiBot Blockly auch für die Nutzung in höheren Schulklassen bestens geeignet. Die Software läuft auf Windows- und macOS-Computern, sowie auf Android- und Apple-Tablets.

## Inhalt

<b>1</b>	<b>Allgemeine Informationen</b> .....	<b>4</b>
1.1	Lieferumfang .....	4
1.2	Software installieren .....	6
1.3	eXperiBot verbinden .....	6
1.4	eXperiBot programmieren (Beispiel) .....	6
1.5	Hinweise zum Umgang mit dem eXperiBot .....	6
1.6	Powerbrain: Buttons und LED-Status.....	7
<b>2</b>	<b>Didaktisches Konzept</b> .....	<b>8</b>
2.1	Die Begleitmaterialien .....	8
2.2	Arbeiten mit dem Set .....	8
2.3	Arbeitsblätter schnell und einfach gestalten .....	8
2.4	Differenzierung.....	9
<b>3</b>	<b>Aufbau der Fabrik</b> .....	<b>10</b>
<b>4</b>	<b>Arbeitsblätter (Kopiervorlagen) und Kommentare</b> .....	<b>12</b>
	Comic: Ariannas neuer Auftrag.....	13
4.1	Grundlagen.....	14
4.2	Smarter Schieber .....	16
4.3	Smarter Greifer .....	19
4.4	Smartes Fließband .....	22
4.5	Smarter Sortierer .....	24
4.6	Ergänzende Aufgaben - Debugging .....	28
<b>5</b>	<b>Hilfekarten (Kopiervorlage Vorder- und Rückseiten)</b> .....	<b>33</b>

# 1 Allgemeine Informationen

## 1.1 Lieferumfang

Die Sets **Smarte Fabrik 71630** (Ergänzungsset) und **71631** (Komplettset) enthalten

- **1 Spielplan 7163051**
- **Teile für den Aufbau der Fabrikroboter** (Übersicht der Teile im jeweiligen Set auf Seite 5)
- **1 Handreichung 716305** mit
  - Kopiervorlagen für Projekte
  - Kopiervorlagen für Hilfekarten
- **4 Hefte 7163052 mit Aufbauanleitungen** für die Fabrikroboter



Smarte Fabrik (Ergänzungsset) 71630



Smarte Fabrik (Komplettset) 71631

**Weitere Projekte zu den Fabrik-Stationen** zum kostenlosen Download unter [cornelsen-experimenta.de](http://cornelsen-experimenta.de).

Dort finden Sie

- alle eXperiBot-Projekte als pdf und editierbare Word-Dokumente.
- Druckvorlagen für die Hilfekarten und den Comic, um diesen z. B. auszudrucken oder Ihren Schülerinnen und Schülern als Einstiegs-Szenario auf das Whiteboard zu projizieren.
- Musterlösungen als Code-Dateien.
- die aktuelle Software.

Direkt zur eXperiBot-Mediathek:

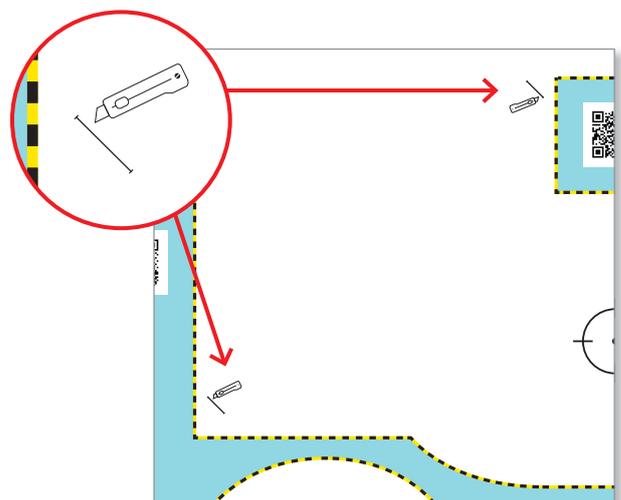


### Hinweise zum Spielplan

Bitte schneiden Sie vor der ersten Benutzung mit einem scharfen Cutter die Schnittmarken ein. In diese Schnitte können Sie die Noppen-Platte einschieben, um sie zu fixieren.

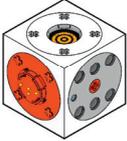
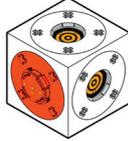
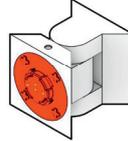
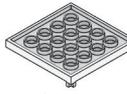
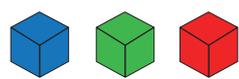
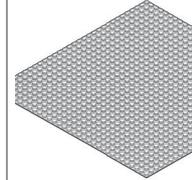
Den Spielplan bitte pfleglich behandeln, nicht knicken und nicht betreten. Nach der Nutzung aufrollen und in der mitgelieferten Verpackung lagern.

Sollte der Spielplan verschmutzt sein, können Sie ihn mit einem feuchten Tuch abwischen.



# 1 Allgemeine Informationen

## Übersicht der Teile-Vorkommen in den Sets

							
Artikelnummer:	71444	71363	71448	71446	71447		
Artikel:	<b>Powerbrain</b>	<b>Motor</b>	<b>Cube</b>	<b>Pivot</b>	<b>Twister</b>		
Anz. in 71630 <i>Ergänzung:</i>	-	-	3	3	2		
Anz. in 71631 <i>Komplett:</i>	4	3	7	3	2		
	71445 Grabber-Modul, 1x 					female	
Artikelnummer:	71445, 71024		71619	71559	71041		
Artikel:	<b>Grabber</b>		<b>Multisensor</b>	<b>Wheel</b>	<b>Adapterplatte</b>		
Anz. in 71630 <i>Ergänzung:</i>	1		2	-	3		
Anz. in 71631 <i>Komplett:</i>	1		6	4	7		
							
Artikelnummer:	71406	71402	71400	71451	71525	71641	71638
Artikel:	<b>Double Cubie 2, Axis, schwarz</b>	<b>Cubie 1, weiß</b>	<b>Prisma 90 Grad, weiß</b>	<b>Achse mit Noppe</b>	<b>Abstandshalter</b>	<b>Noppenteller</b>	<b>Slider</b>
Anz. in 71630 <i>Ergänzung:</i>	1	31	14	-	-	-	-
Anz. in 71631 <i>Komplett:</i>	1	31	14	4	4	4	2
							
Artikelnummer:	71683	71686	71682	71685	71684		
Artikel:	<b>Achse lang</b>	<b>Achse kurz</b>	<b>Spacer, schwarz</b>	<b>Zahnrad</b>	<b>Kettenglied</b>		
Anz. in 71630 <i>Ergänzung:</i>	1	1	4	2	41		
Anz. in 71631 <i>Komplett:</i>	1	1	4	2	41		
	71076 Prisma 90 Grad, gelb, 1x 						
Artikelnummer:	71076/ 71031/ 71032		71179/ 71720/ 71721	71687	71614	71618	
Artikel:	<b>Separator</b>		<b>Schaumstoffwürfel (Pakete) blau, grün, rot</b>	<b>Platte mit 32 x 32 Noppen</b>	<b>Netzteil</b>	<b>Bluetooth-Dongle</b>	
Anz. in 71630 <i>Ergänzung:</i>	2		je 3	1	-	-	
Anz. in 71631 <i>Komplett:</i>	4		je 3	1	4	4	

# 1 Allgemeine Informationen

## 1.2 Software installieren

- eXperiBot Blockly App aus den App-Stores herunterladen bzw. folgenden Link öffnen: [blockly.app.experibot.com](https://blockly.app.experibot.com)
- Nur für Windows-PC mit Windows 10 in der Version 21H1 oder älter:  
Bluetooth-Dongle (USB-Stick) einstecken
- App starten

## 1.3 eXperiBot verbinden

- „Play“ drücken, um den eXperiBot einzuschalten
- eXperiBot Blockly app starten
- Workspace erstellen und öffnen
- Projekt erstellen und öffnen
- In der Projektansicht, oben auf  klicken
- Powerbrain auswählen

eXperiBot ist jetzt startklar.

## 1.4 eXperiBot programmieren (Beispiel)

- Links in der Toolbar auf „Powerbrain“ klicken
- Gewünschten Block in die Arbeitsfläche rechts ziehen
- Alle Blöcke, die der eXperiBot ausführen soll, müssen mit dem "starte Programm"-Block verbunden sein.
- Zum Starten des Programms unten auf "Programm ausführen" klicken
- Der eXperiBot beginnt sofort mit der Ausführung des Programms.

Eine ausführliche Anleitung mit Tipps für die verschiedenen Roboter-Module, sowie die Programmierumgebung finden Sie unter <https://docs.experibot.com/>

## 1.5 Hinweise zum Umgang mit dem eXperiBot

Da auf dem Spielplan bis zu vier Gruppen gleichzeitig aktiv sind, sollten Sie die Powerbrains (rote Module) mit einer Markierung versehen, um sie beim Verbinden mit den PC/Mac/Tablets besser identifizieren zu können.

- Dazu starten Sie die App und lassen sich nacheinander – einzeln – die verfügbaren Powerbrains anzeigen, indem Sie auf den „Verbinden“-Button klicken.
- Jedes Powerbrain hat eine dreistellige Kennung, die aus einer Buchstaben/Zahlenkombination besteht.
- Sobald ein Powerbrain detektiert wird, erscheint die Kennung in der Liste.
- Diese Kennung können Sie nun auf einem kleinen Aufkleber auf dem jeweiligen Powerbrain notieren oder zur Not direkt auf das Powerbrain schreiben.



Bei der Programmierung des eXperiBot gibt es zwei Möglichkeiten, den Roboter ein Programm ausführen zu lassen:

- Sie verbinden den eXperiBot wie oben beschrieben mit der App und starten das Programm aus der App heraus. Dies hat den Vorteil, dass die Schülerinnen und Schüler sehr schnell Änderungen vornehmen können und das Programm ständig aktualisiert wird.

Manchmal kann es von Vorteil sein, das Programm ohne Computeranbindung auszuführen, z.B. wenn mehrere Robotermodelle mit einem Gerät programmiert werden oder die Computer zu weit vom Einsatzort der Roboter entfernt stehen. Dann gehen Sie wie folgt vor:

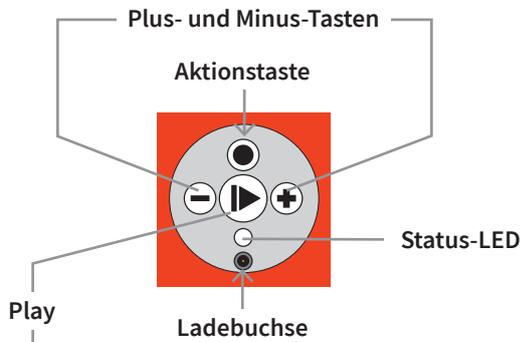
- Sie führen alle unter 1. beschriebenen Punkte aus und starten das Programm einmal aus der App heraus. Das Programm ist jetzt automatisch im Powerbrain hinterlegt.
- Trennen Sie den eXperiBot von der App. Sie können das Powerbrain auch ausschalten. Das Programm bleibt im Speicher.
- Falls Sie das Powerbrain ausgeschaltet haben, schalten Sie es wieder ein. Drücken Sie nun die Play-Taste. Der eXperiBot führt das zuletzt aufgespielte Programm aus.

# 1 Allgemeine Informationen

## 1.6 Powerbrain: Buttons und LED-Status

**Einschalten:** Play-Taste gedrückt halten

**Neustart:** Plus- und Minus-Taste für sieben Sekunden gleichzeitig gedrückt halten



**Drücken:** Programm starten, das zuvor mit der eXperiBot Blockly App auf das Powerbrain geladen wurde

**Langes Drücken:** Powerbrain ein- und ausschalten

LED		Powerbrain ...
leuchtet	—	
blinkt	....	
gelb		... ist vollständig geladen
		... lädt
grün		... ist eingeschaltet und einsatzbereit
blau		... ist mit der App verbunden
hellblau		... ist mit der App verbunden und führt ein Programm aus
weiß		... ist nicht mit der App verbunden und führt das zuletzt auf das Powerbrain geladene Programm aus
violett		... benötigt Firmware-Update
		... Firmware wird upgedated

## 2 Didaktisches Konzept

### 2.1 Die Begleitmaterialien

Das Set *eXperiBot Unterrichtsmaterialien 71630/71631* enthält schriftliche Begleitmaterialien für vier ausgearbeitete Einheiten zum Thema „Fabrik“. Zu jeder Einheit finden Sie weitere Arbeitsblätter mit differenzierenden Angeboten unter [cornelsen-experimenta.de](http://cornelsen-experimenta.de).

In dieser Handreichung finden Sie zu jeder Einheit neben einem Kommentar mit didaktisch-methodischen Hinweisen, Lösungen und Tipps für Sie auch ein Arbeitsblatt für Ihre Schülerinnen und Schüler als Kopiervorlage. Komplettiert wird das Anleitungspaket durch Hilfekarten und die Debugging-Arbeitsblätter zu den Projekten, die Sie bei der Differenzierung umfassend unterstützen.

### 2.2 Arbeiten mit dem Set

Das Set ist dafür ausgelegt, dass sich die Lernenden in Teams mit verschiedenen Stationen beschäftigen. Zu den hier zu erlangenden Erfahrungen gehören die Absprachen zwischen den Team-Mitgliedern (Aufgabenverteilung) und zwischen den Teams (Schnittstellen). Daher stehen weniger allgemein gültige Algorithmen im Vordergrund, sondern stabil laufende individuelle Lösungen, die zu einem gemeinsamen Erfolgserlebnis führen.

Die Bearbeitung der Lösungen sollte zyklisch ablaufen:

1. Planung der Roboterbewegung als Pseudocode oder Programmablaufplan
2. Implementierung eines Teils der Bewegung
3. Mehrmaliges Testen des Ablaufs
4. „Finetuning“, d.h. Verändern der Bewegungsparameter (Winkel, Geschwindigkeit, Dauer)
5. Zurück zu Schritt 2, falls Fehler auftreten oder weitere Bewegungen hinzugefügt werden sollen

Ziel der Arbeit ist ein stabiler, fehlertoleranter Ablauf. Die im Heft gezeigten Programmbeispiele sollen diesem Anspruch genügen, wenn mit den vorgegebenen Robotermodellen gearbeitet wird. Es kann effizientere und einfachere Lösungen geben, die gern von den Lernenden ausprobiert werden können. Lassen Sie es daher zu, wenn die Lernenden anfangen, auch die Robotermodelle anzupassen, um Schwierigkeiten zu beheben.

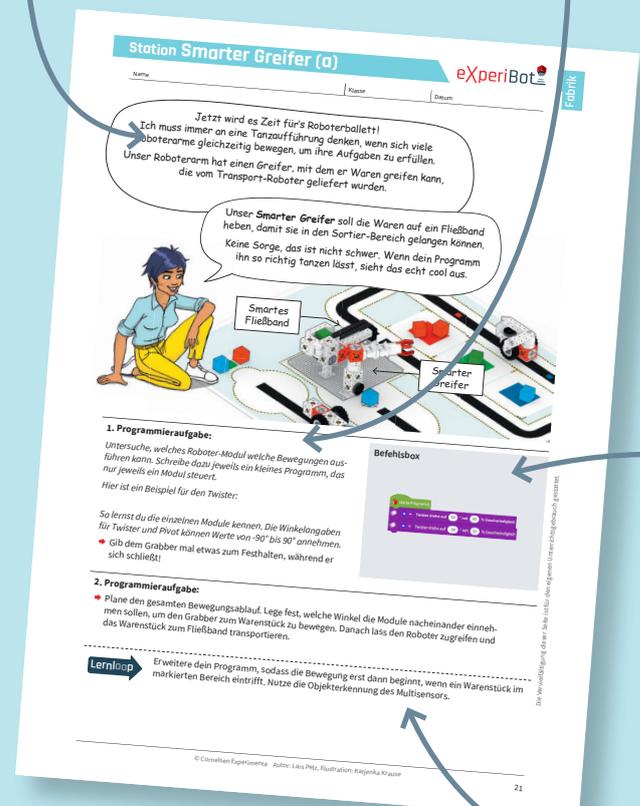
### 2.3 Arbeitsblätter schnell und einfach gestalten

Anstatt der Kopiervorlagen im Heft empfehlen wir Ihnen die **kostenlosen, frei editierbaren Arbeitsblätter auf unserer Website** zu nutzen, um die für Ihre Lerngruppe nötigen Anpassungen vorzunehmen.

Ein Arbeitsblatt umfasst vier Bereiche:

1. **Kontextualisierte Einführung:** In kleinen Geschichten wird das Thema aufgegriffen und motiviert.

2. **Szenario sowie die Aufgabenstellung zur Programmierung:** Das Szenario sowie die Anforderungen an das Programm werden definiert.



**Editierbare Arbeitsblätter online:**  
[cornelsen-experimenta.de](http://cornelsen-experimenta.de)

## 2 Didaktisches Konzept

### 2.4 Differenzierung

Mit den **Hilfekarten** lassen sich einzelne Befehle oder Programmiertechniken selbstständig entdecken und erlernen. Die Karten enthalten Befehle, die speziell für den eXperiBot gelten, und einige Programmstrukturen.



**Hilfekarten**

Eine Kopiervorlage finden Sie auf den Seiten 33 und 34; im Downloadbereich unter [cornelsen-experimenta.de](http://cornelsen-experimenta.de) finden Sie darüber hinaus eine Druckvorlage.

Auf der Vorderseite einer Karte ist jeweils der Befehl, auf der Rückseite dessen Bedeutung aufgeführt. Legen Sie die Karten mit der Befehlsseite nach oben aus, erhalten Sie eine „Hilfestation“, an der die Schülerinnen und Schüler die Bedeutung verschiedener Blöcke nachschauen können, ohne dass sie ihnen ständig am Platz zur Verfügung stehen.

Wenn Sie die Karten stapeln, immer die oberste aufdecken und jeweils nach der Rückseite fragen, erhalten Sie ein Übungsspiel.

Das **Nachprogrammieren** von funktionierenden Programmen ist der einfachste Weg in die Programmierung. Darauf aufbauend ist das **Auffinden von kleinen Fehlern in fertigen Programmen** (Debugging) ein weiterer Schritt hin zur selbstständigen Programmierung. Aus diesem Grund gibt es für einige Projekte ein bzw. mehrere



**Debugging**

Beispielprogramme, von denen eins eine funktionierende Lösung ist. Die anderen Programme sind der Lösung ähnlich, jedoch an einer Stelle fehlerhaft.

Im Zuge der Differenzierung können Sie:

- Das richtige Programm zum Nachprogrammieren bereitstellen.
- Das richtige sowie ein oder zwei fehlerhafte Programme zur Verfügung stellen.

Weiterhin können Sie die Programme für Zusatzaufgaben verwenden und das richtige Programm oder die Fehler finden lassen.

**Hinweis:** Wie alle unsere Produkte werden der eXperiBot und die dazugehörige App kontinuierlich verbessert und überarbeitet. Die hier abgedruckten Coding-Beispiele und Screenshots werden mit jeder neuen Auflage angepasst und aktualisiert. Sollte es also einmal vorkommen, dass etwas nicht wie in der Print-Anleitung beschrieben funktioniert, schauen Sie bitte immer zuerst auf unserer Webseite, bevor Sie sich an unseren Kundendienst wenden.

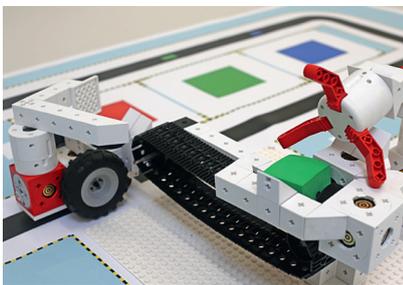
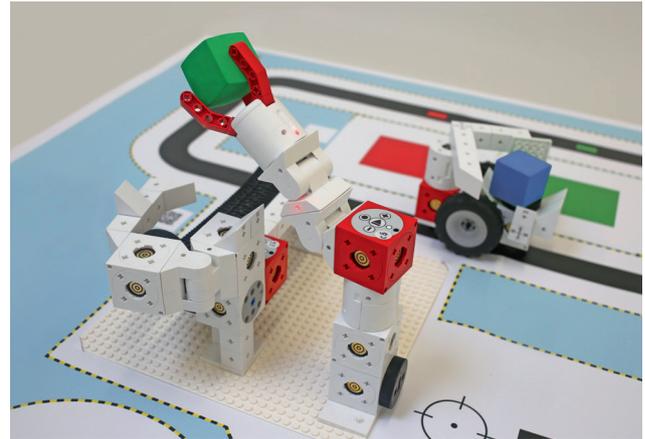
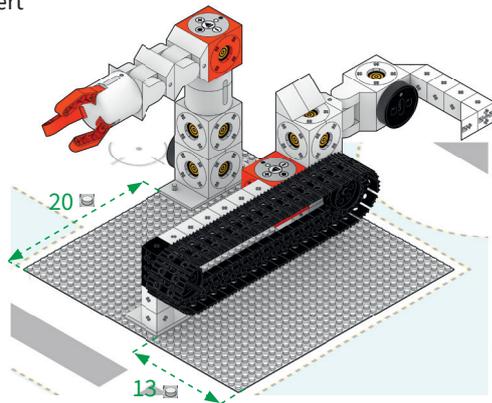
Unter [cornelsen-experimenta.de](http://cornelsen-experimenta.de) finden Sie immer die aktuellsten Versionen der Arbeitsblätter und der Software.

- Befehlsbox zur Differenzierung:** In der Befehlsbox sind wichtige Befehle aufgeführt, die zur Lösung des Problems notwendig sind. Möchten Sie den Schwierigkeitsgrad erhöhen, bieten sich die folgenden Möglichkeiten:
  - Bereits bekannte Befehle aus der Box löschen und nur neue Befehle listen.
  - Alle voreingetragenen Befehle aus der Box löschen und die SchülerInnen und Schüler auffordern, neu erlernte Befehle in die Box einzutragen.

- Der Lernloop mit vertiefenden Aufgaben** schließt sich der erfolgreich gelösten Programmieraufgabe an. Er liefert zusätzliche Problemstellungen, mit denen der in einem Szenario behandelte Algorithmus verändert und erweitert wird, um ihn zu vertiefen.

### 3 Aufbau der Fabrik

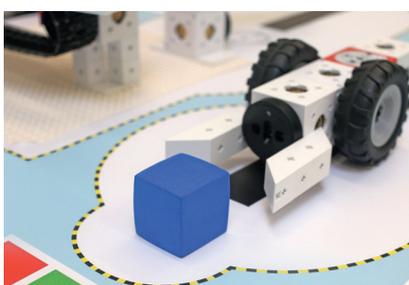
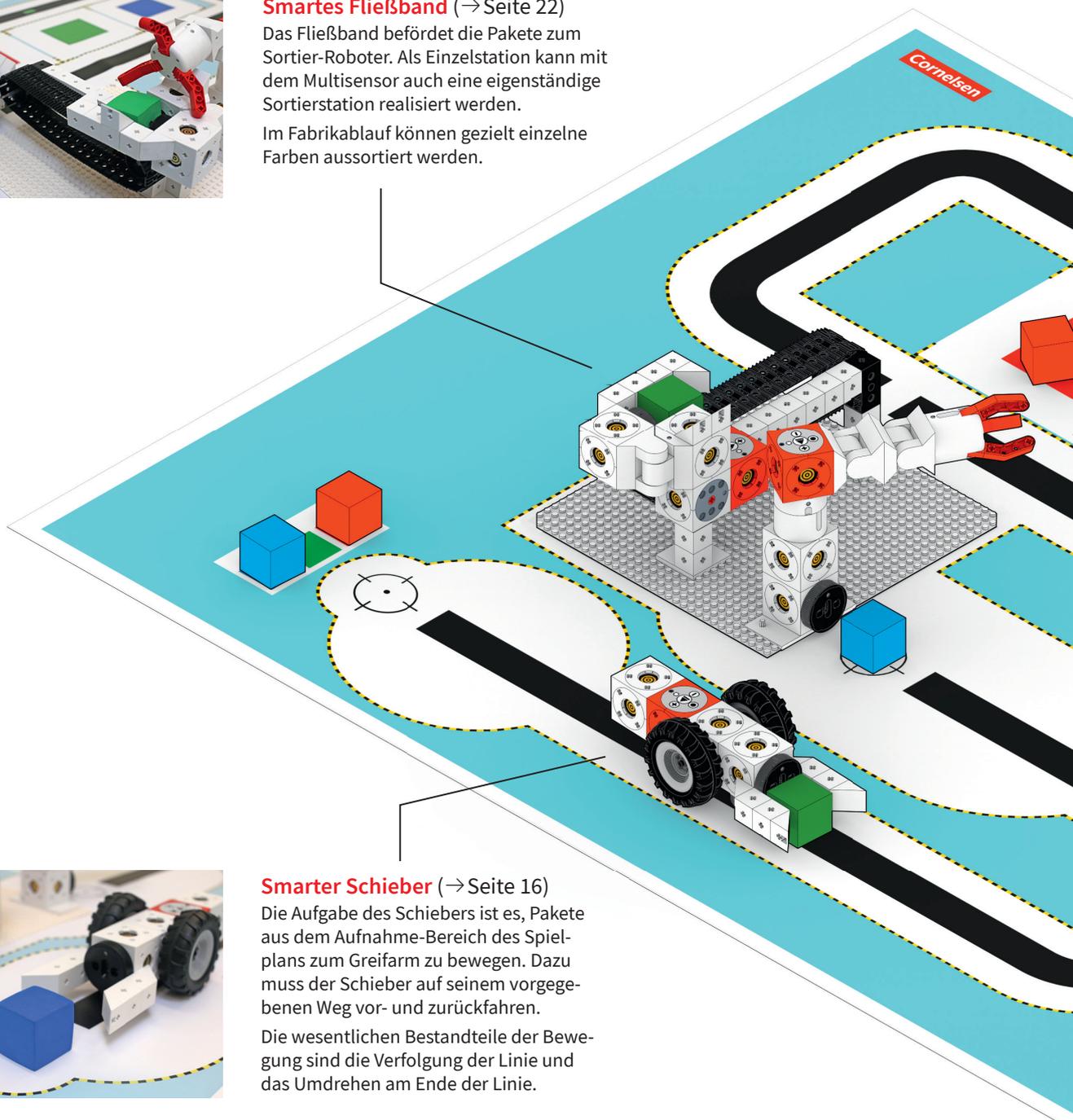
So werden Greifer und Fließband auf der Noppenplatte platziert



#### Smartes Fließband (→ Seite 22)

Das Fließband befördert die Pakete zum Sortier-Roboter. Als Einzelstation kann mit dem Multisensor auch eine eigenständige Sortierstation realisiert werden.

Im Fabrikablauf können gezielt einzelne Farben aussortiert werden.



#### Smarter Schieber (→ Seite 16)

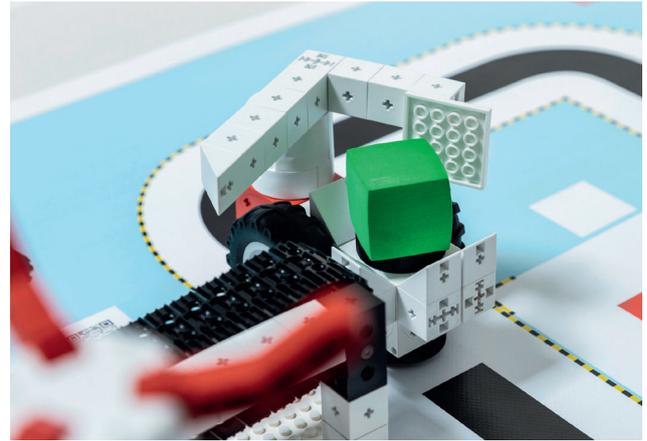
Die Aufgabe des Schiebers ist es, Pakete aus dem Aufnahme-Bereich des Spielplans zum Greifarm zu bewegen. Dazu muss der Schieber auf seinem vorgegebenen Weg vor- und zurückfahren.

Die wesentlichen Bestandteile der Bewegung sind die Verfolgung der Linie und das Umdrehen am Ende der Linie.

### 3 Aufbau der Fabrik

Unten sehen Sie die komplette Fabrik mit den vier Stationen. Jeder Roboter wird von einer Gruppe gebaut und programmiert.

Damit die Fabrik reibungslos funktioniert, kommt es auf präzise Absprachen und gute Programmierung an den Schnittstellen an. Das funktioniert nur, wenn alle ihre individuellen Lösungen aufeinander abstimmen und zu einem kollaborativen Gesamtkonzept verbinden – nicht ganz einfach, aber sehr herausfordernd und mit viel Freude für alle Coding-Fans!



#### Smarter Greifer (→ Seite 19)

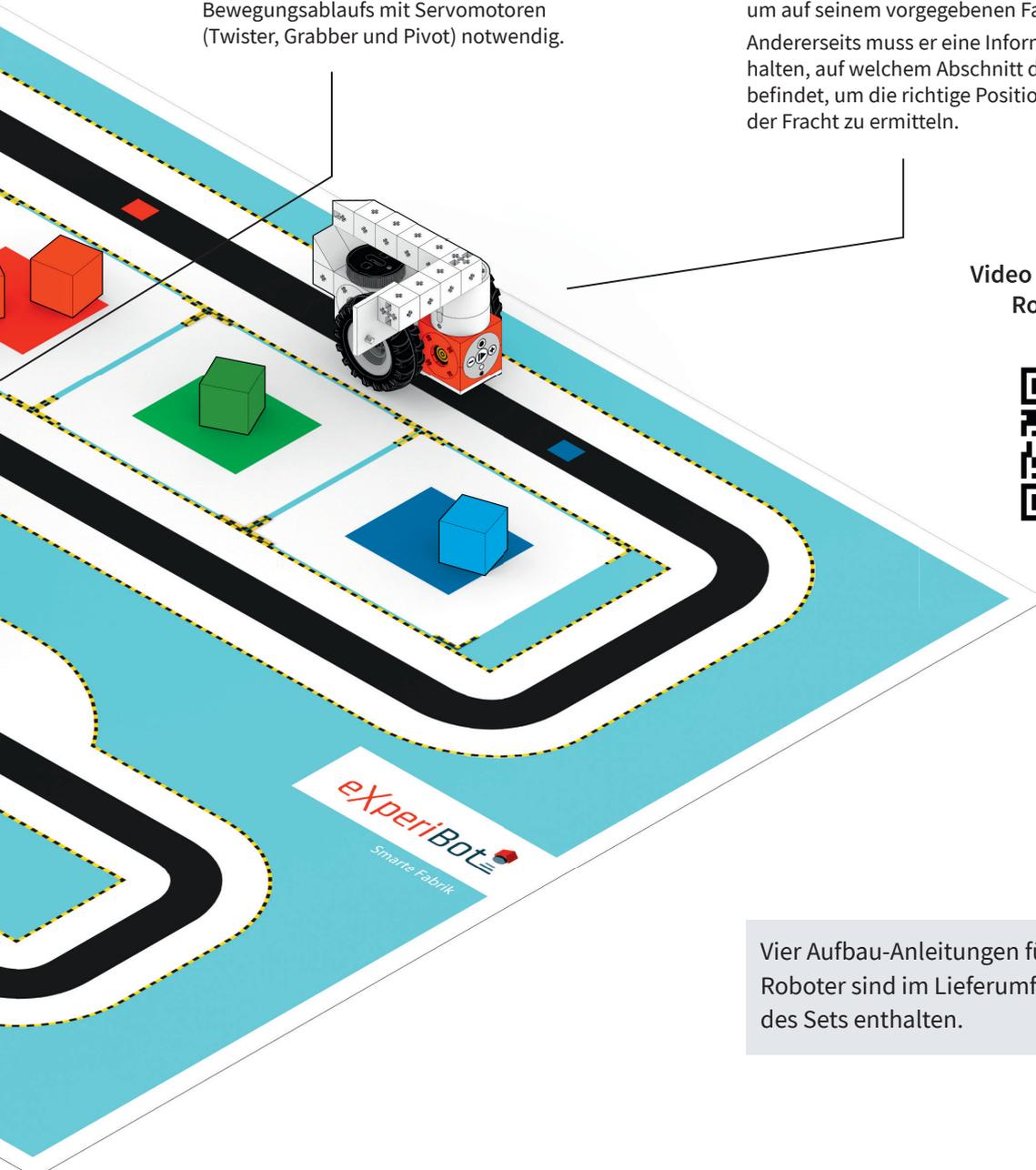
Die Aufgabe des Greifarms ist es, vom Schieber transportierte Pakete auf das Förderband zu heben.

Hier ist die Programmierung eines Bewegungsablaufs mit Servomotoren (Twister, Grabber und Pivot) notwendig.

#### Smarter Sortierer (→ Seite 24)

Der Sortierer erhält Pakete vom Fließband, die er entsprechend ihrer Farbe in die richtigen Ausgabebereiche transportieren muss.

Dafür muss er einerseits die Linie richtig erkennen um auf seinem vorgegebenen Fahrweg zu bleiben. Andererseits muss er eine Information darüber erhalten, auf welchem Abschnitt des Fahrwegs er sich befindet, um die richtige Position zum „Auswerfen“ der Fracht zu ermitteln.



Video einer kompletten  
Roboter-Runde:



Vier Aufbau-Anleitungen für die  
Roboter sind im Lieferumfang  
des Sets enthalten.



## 4 Arbeitsblätter und Kommentare

Auf den folgenden Seiten finden Sie die Arbeitsblätter für Ihre Schülerinnen und Schüler mit den vorangestellten Hinweisen zu den Aufgaben.

Das neue Konzept von Cornelsen Experimenta zum lösungsorientierten Programmieren (auch „Coding“ oder kurz „Coden“ genannt) ist im methodischen Rahmen von Storytelling angelegt.

Hierbei werden anspruchsvolle Programmieraufgaben mithilfe der Programmiererin Arianna in bedeutungsvolle Zusammenhänge gestellt. Damit können Sie Ihre Schülerinnen und Schüler begeistern und ihnen helfen, die Grundlagen des Coding und das Entwickeln von Algorithmen zu be„greifen“ und nachhaltig zu verstehen.

Ein wichtiges Element jeder Aufgabe ist das Wechselspiel zwischen dem Erstellen oder Erweitern eines Programms (Code) und dem anschließenden Test des Programms mit dem eXperiBot. Die sofortige Übertragung des Virtuellen ins Reale macht den Entwicklungsvorgang als krea-

tive Handlung für die Schülerinnen und Schüler erfahrbar und wirkt sehr motivierend.

Zur Einführung in das Thema können Sie den Comic „Ariannas neuer Auftrag“ von Seite 11 aus dieser Handreichung verwenden. Sie können ihn wahlweise kopieren oder die Druckvorlage nutzen, die Sie auf unserer Website im Downloadbereich finden.

Aufbauend auf dieser einführenden Geschichte enthält jedes Arbeitsblatt eine weitere kurze Kontextualisierung. Mit dieser wird eine bedeutsame Programmieraufgabe entwickelt, die dann von den Schülerinnen und Schülern – am besten in Partnerarbeit – gelöst und mit dem eXperiBot auf dem jeweiligen Spielplan ausprobiert wird.

Die einzelnen Aufgaben bauen stufenweise aufeinander auf, können aber je nach Vorwissen und Leistungsstand auch unabhängig voneinander bearbeitet werden.

### Projekte

Bevor Sie mit Ihren Schülerinnen und Schülern in das Fabrik-Set einsteigen, empfiehlt es sich, **als Grundlage die Arbeitsblätter des eXperiBot Lernroboter (Basis)-Sets durchzuarbeiten** – quasi als Projekt 0, um die Grundlagen des Coding zu verstehen und die Module kennenzulernen.

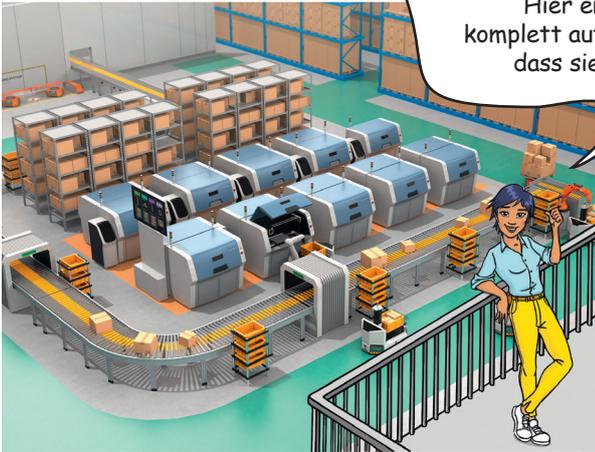
Insgesamt gibt es in dieser Anleitung **vier vorgefertigte Arbeitsblätter** zu folgenden Themen:

- Projekt **Smarter Schieber** (Seite 16–18)
- Projekt **Smarter Greifer** (Seite 19–21)
- Projekt **Smartes Fließband** (Seite 22–23)
- Projekt **Smarter Sortierer** (Seite 24–27)

**Ergänzende Debugging-Aufgaben** zu den Projekten finden Sie auf den Seiten 28–31.

**Weitere Arbeitsblätter** zu den einzelnen Stationen und editierbare Word-Dokumente von allen Projekten finden Sie unter [cornelsen-experimenta.de](http://cornelsen-experimenta.de).

# Ariannas neuer Auftrag



Beeindruckend, oder? Das ist eine „Smarte Fabrik“. Hier erledigen Roboter die ganze Arbeit, komplett automatisiert. Sie sortieren die Waren so, dass sie korrekt versendet werden können.

Ach so... **Hi erst einmal**, ich bin Arianna und unterstütze als Programmiererin das Team hier vor Ort. Wir sind gerade dabei, eine neue Sortier- und Lagerhalle zu planen.

Wo welche Anlagen stehen und welche Wege die Roboter dann zurücklegen sollen, wissen wir schon.



Auch wie die Roboter aussehen müssen, damit sie die Container bewegen können, ist schon klar.

Hey, der hier ist ja ganz schön bepackt ...



Die Vervielfältigung dieser Seite ist für den eigenen Unterrichtsgebrauch gestattet.



Und heute ist es soweit. **Wir programmieren die Roboter!** Das hier ist übrigens eXperiBot, mein Test-Roboter.

Ups, ´tschuldigung. Ich stehe hier auf der Fahrlinie.



Hep... noch einer. Die sind ja schnell. Wir suchen uns mal besser irgendwo ein ruhigeres Plätzchen. Kommt mal mit, es wird ...

biieeeeep



... **spannend!** Wir bauen uns gerade ein Modell der Anlage. In der echten Anlage darf nämlich kein Fehler passieren. Das kann sehr teuer werden, wenn die Roboter z.B. zusammenstoßen oder vom Weg abkommen. Und im Modell erkennen wir auch sehr genau, wo wir die Programmierung korrigieren müssen. **Also los!**



# 4.1 Grundlagen

In Lektion 1 lernen die Schülerinnen und Schüler die Funktionen eines einfachen Robotermodells kennen und erforschen anschließend die Bestandteile. Als Vorbereitung für die späteren Lektionen programmieren die Schülerinnen und Schüler ihr erstes Programm.

## 4.1.1 Lernziele Grundlagen

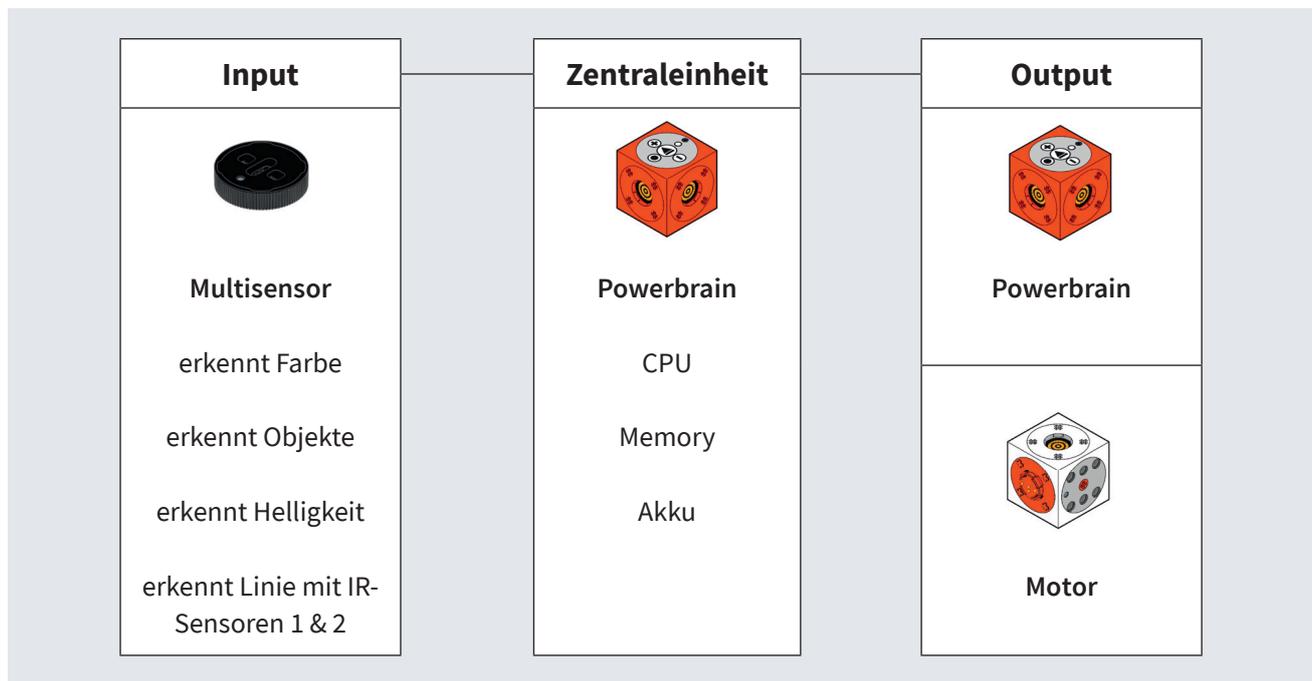
Die Schülerinnen und Schüler

- verstehen, dass der eXperiBot als Minicomputer Eingaben aufnimmt und nach der Verarbeitung der Eingabe eine Ausgabe produziert.
- lernen die Vielfalt der verschiedenen Arten von Informationen kennen, die der eXperiBot als Eingabe verarbeiten kann.

## 4.1.2 Diskussion

Besprechen Sie mit den Schülerinnen und Schülern die Bestandteile eines eXperiBot Roboters. Gehen Sie dabei auf die folgenden Komponenten ein und zeichnen Sie dabei das folgende Schema an die Tafel.

- **Memory:** Hier merkt sich der eXperiBot Roboter Programme und Daten.
- **Akku:** Ist eine wiederaufladbare Batterie im Powerbrain und speichert elektrische Energie.
- **Input:** So nimmt ein eXperiBot Roboter Informationen von seiner Umwelt auf. Beim Menschen kommt der Input über seine Sinnesorgane. Bei dem eXperiBot Roboter gibt es Sensoren, die Objekte, Farben, Helligkeiten und Linien erkennen. Fast alle Module des Roboters enthalten eigene Sensoren.
- **Output:** Hierüber zeigt der eXperiBot Roboter Informationen an oder bewegt sich. Menschen kommunizieren Informationen u.a. mit dem Mund durch Sprache. Sie bewegen sich mithilfe ihrer Gliedmaßen. Der eXperiBot Roboter bewegt sich mithilfe seiner Bewegungsmodule. Mit dem Lautsprecher kann er Signale ausgeben.



- **Zentraleinheit:** Das Powerbrain bildet die zentrale Einheit jedes eXperiBot Roboters.
- **CPU:** Die CPU ist der wichtigste Teil des Computers und steht für „Central Processing Unit“. Hier werden Daten verarbeitet und umgewandelt.

# 4.1 Grundlagen

## 4.1.3 Lernziele Module erforschen

Die Schülerinnen und Schüler

- lernen die eXperiBot Module haptisch kennen.
- lernen, welche Funktionen die Bestandteile haben.

## 4.1.4 Diskussion

Lassen Sie die Schülerinnen und Schüler die eXperiBot-Bestandteile nun selbst erforschen und diskutieren Sie mit ihnen die folgenden Fragestellungen:

- Was bedeutet Powerbrain übersetzt?
- Wozu benötigt ein eXperiBot Roboter ein Powerbrain?
- Was bedeuten die Farben an den zwei unterschiedlichen Achsen des Motor-Moduls?
- Wie können zwei Bausteine miteinander verbunden werden?
- Wie bewegt sich das Pivot-Modul und wie unterscheidet es sich vom Motor-Modul?
- Wie können Räder an dem eXperiBot Roboter befestigt werden?
- Warum muss das Powerbrain aufgeladen werden?

## 4.1.5 Einstieg

Als Einstieg in die Welt des Coding empfiehlt es sich, die Themen und Arbeitsblätter aus dem eXperiBot Lernroboter (Basis)-Set durchzuarbeiten. Damit wird der Einstieg erleichtert und viele Fertigkeiten, die im Fabrik-Set von Nutzen sind, stehen dann schon zur Verfügung.

Folgender Ablauf hat sich bewährt:

1. eXperiBot Einstiegs-Projekt: Partner-Roboter
2. eXperiBot\_Zusammenbau\_Software-Installation
3. eXperiBot Einstiegs-Projekt: Powerbrain macht Musik
4. eXperiBot Projekt: Hindernissen ausweichen

Danach können Sie die einzelnen Roboter-Modelle aus dem Fabrik-Set nachbauen und später anpassen und modifizieren – bestimmt haben Ihre Schülerinnen und Schüler viele eigene sehr gute Ideen, was man noch alles verbessern und optimieren kann.

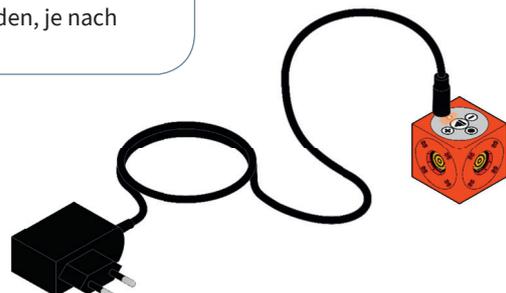
Wir empfehlen die Arbeitsblätter an die Schülerinnen und Schüler auszuteilen. Natürlich können Sie die Arbeitsblätter auch vortragen oder an ein Smartboard projizieren.

## 4.1.6 Beurteilen

Besprechen Sie mit den Schülerinnen und Schülern die Lösungen. Lassen Sie einige Schülerinnen und Schüler Ihre Lösungen präsentieren und gehen Sie auf mögliche Fehler und Probleme ein.



Alle Roboter benötigen elektrische Energie um zu funktionieren. Ein eXperiBot Roboter kann mit dem beigelegten 9V Netzteil geladen werden. Eine Ladung dauert etwa 45 bis 60 Minuten und hält anschließend etwa zwei bis acht Stunden, je nach Anwendung.



## 4.2 Smarter Schieber



Die Aufgabe des Schiebers ist es, Pakete aus dem Aufnahme-Bereich des Spielplans zum Greifarm zu bewegen. Dazu muss der Schieber auf seinem vorgegebenen Weg vor- und zurück fahren. Die wesentlichen Bestandteile der Bewegung sind die Verfolgung der Linie und das Umdrehen am Ende der Linie.

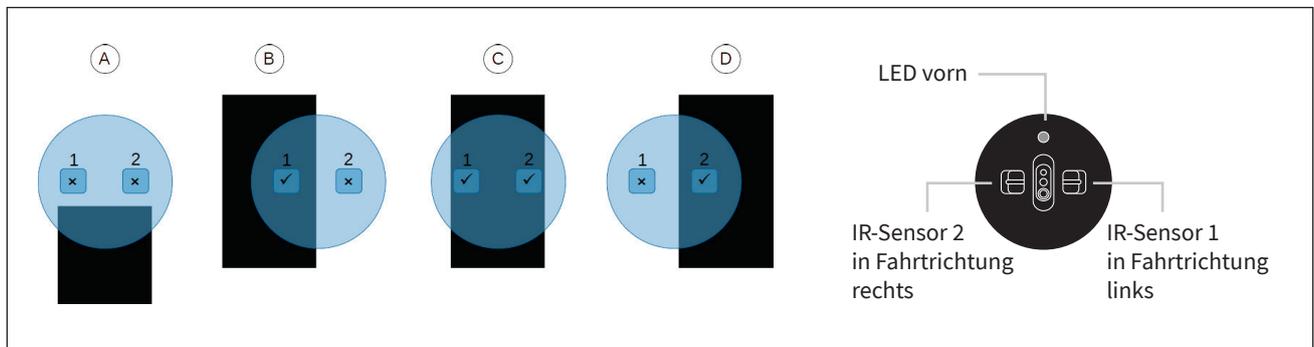
### 4.2.1 Linie verfolgen

Um dem Weg zu folgen, der auf dem Spielplan für den Schiebe-Roboter vorgezeichnet ist, muss die schwarze Linie durch die Infrarot-Sensoren des unterseitig angebrachten Multisensors erfasst werden. Dabei können vier verschiedene Fälle auftreten, die genutzt werden können, um Entscheidungen zur Roboter-Bewegung zu treffen:

führt. Die konkreten Werte für Geschwindigkeit und Zeit können dabei individuellen Wünschen angepasst werden. Höhere Geschwindigkeiten verkürzen die Arbeitszeit, erhöhen aber das Risiko, dass der Roboter die Linie gänzlich verliert. Gleiches gilt für die Drehung: Ist sie zu langsam oder zu schnell, schafft der Roboter es nicht, um die Kurve zu fahren.

### 4.2.2 Umdrehen

Am Ende der Linie muss der Schieber eine Wende vollführen, um auf der Linie zum Aufnahme-Bereich zurückzukehren. Genauso soll er wieder im Aufnahme-Bereich wenden, um kontinuierlich Pakete zum Greifarm zu transportieren.



- A. **Linienende:** Beide Sensoren melden, dass keine Linie erkannt wird. In diesem Fall ist der Roboter an das Ende einer Linie gelangt. Der Schieber muss sich auf der Stelle um etwa 180° drehen, um die Linie wieder zu finden und auf ihr zurückzufahren.
- B. **Übertretung rechts:** Der Roboter ist zu weit nach rechts gefahren. Er muss sich nach links drehen, um wieder zur Linie zurückzukehren.
- C. **Linie gefunden:** Der Roboter fährt genau auf der Linie. Er kann seine Fahrt geradeaus fortsetzen.
- D. **Übertretung links:** Der Roboter ist zu weit nach links gefahren. Er muss sich nach rechts drehen, um wieder zur Linie zurückzukehren.

Die Bewegung ist hier im Beispiel (Abb. 4.2.b) wie folgt gelöst:

- a. Eine Drehung auf der Stelle, um die Linie vollständig zu verlassen.
- b. Viele kurze Drehbewegungen auf der Stelle, bis beide Liniensensoren die Linie erkennen.

Die Nutzung der vordefinierten Dreh-Befehle hat sich für diese Aufgabe in Tests nicht bewährt, da der Roboter zu viel Drehmoment aufnimmt und sich über die Linie hinaus weiterdreht. Sie können trotzdem Versuche durchführen oder Ihre Lernenden dazu animieren, eine Lösung mit Dreh-Befehlen zu finden.

### 4.2.3 Vollständiger Ablauf

Im Folgenden finden Sie eine **einfache Möglichkeit, den Schieber kontinuierlich Pakete transportieren zu lassen** (Abb. 4.2.c). Dies kann das Ziel des Arbeitsauftrags für das Team sein, das am Schieber arbeitet.

Die Abbildung rechts (Abb. 4.2.a) zeigt eine mögliche Lösung für den Linienfolger. Die Gesamt-Bewegung wird durch viele kurze Drehungen des Motors (jeweils eine Zehntelsekunde) in die erforderliche Richtung ausgeführt. So kann der Roboter schnell auf das Überfahren des Linienrandes reagieren.

Der Bewegungsblock (Bedeutung: „rückwärts fahren“ für eine Sekunde) zwischen den Funktionsaufrufen sorgt dafür, dass der Schieber das transportierte Paket am Zielort liegen lässt. Danach vollführt er die Wende und kehrt in den Aufnahme-Bereich zurück.

## 4.2 Smarter Schieber



Sie können die Aufgabenstellung erweitern, indem Sie den zweiten, nach vorn gerichteten Multisensor in die Programmierung mit einbeziehen:

- Der Schieber soll nur losfahren, wenn er ein Objekt in seiner Transportvorrichtung erkennt.
- Der Schieber soll rote Pakete langsamer transportieren, weil sie „zerbrechliche“ Produkte enthalten.

- Für die Bearbeitung durch einzelne Schülerinnen und Schüler oder in Gruppen an separaten Tischen kann die Aufgabenstellung des Schiebers um eine Farberkennung erweitert werden.

**Zusätzliche Debugging-Aufgabe** (→ Arbeitsblatt Seite 29)

### 4.2.a Lösung für den Linienfolger

```

Funktion: linie_folgen
wiederhole bis
  Multisensor erkennt hellen Untergrund links und Multisensor erkennt hellen Untergrund rechts (A)
mache
  falls
    Multisensor erkennt dunklen Untergrund links und Multisensor erkennt hellen Untergrund rechts (B)
  mache
    Motor setze Geschwindigkeit der Achse(n) auf 25 % für 0.1 s
  sonst falls
    Multisensor erkennt hellen Untergrund links und Multisensor erkennt dunklen Untergrund rechts (D)
  mache
    Motor setze Geschwindigkeit der Achse(n) auf 25 % für 0.1 s
  sonst
    Motor setze Geschwindigkeit der Achse(n) auf 25 % für 0.1 s (C)
  
```

### 4.2.b Lösung für das Drehen auf der Stelle

```

Funktion: umdrehen
Motor setze Geschwindigkeit der roten Achse auf -20 % und der blauen Achse auf 20 % für 0.5 s (A)
wiederhole bis
  Multisensor erkennt dunklen Untergrund links und Multisensor erkennt dunklen Untergrund rechts
mache
  Motor setze Geschwindigkeit der roten Achse auf -20 % und der blauen Achse auf 20 % für 0.1 s (B)
  
```

### 4.2.c Hauptprogramm für den Schieber

```

starte Programm
wiederhole unendlich
  mache
    linie_folgen
    Motor fahre rückwärts für 1 s mit mittlerer Geschwindigkeit
    umdrehen
    linie_folgen
    umdrehen
  
```

Musterlösungen als Code-Dateien für die App finden Sie im Download-bereich unter [cornelsen-experimenta.de](http://cornelsen-experimenta.de)

Name \_\_\_\_\_

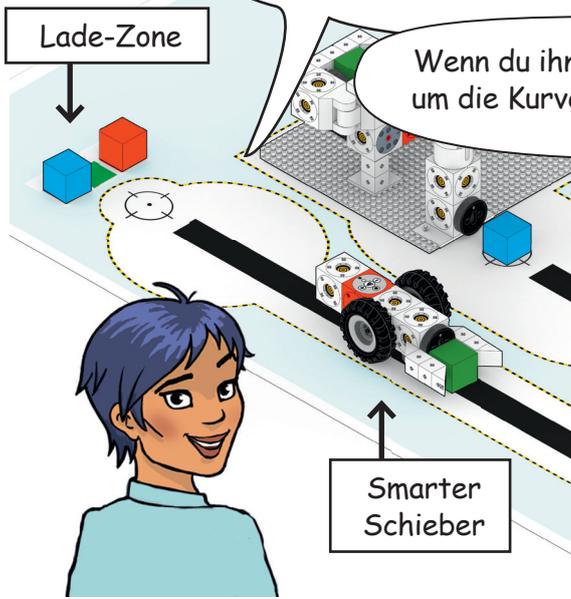
Klasse \_\_\_\_\_

Datum \_\_\_\_\_

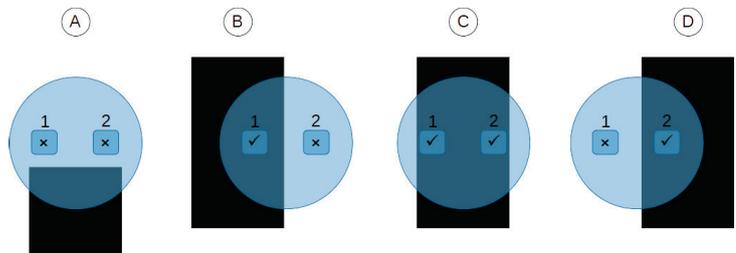
Die erste Station ist der Transport-Roboter, der Waren aus der Lade-Zone in die Anlage transportiert. Wir nennen ihn „**Smarter Schieber**“.

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):



**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**

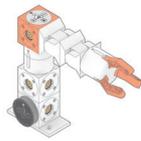


Erweitere das Programm, sodass der Roboter seine Bewegung immer wiederholt und laufend Waren aus der Lade-Zone transportiert.

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.

## 4.3 Smarter Greifer



Die Aufgabe des Greifarms ist es, vom Schieber transportierte Pakete auf das Förderband zu heben. Hier ist die Programmierung eines Bewegungsablaufs mit Servomotoren (Twister, Grabber und Pivot) notwendig.

Im folgenden Beispiel (Abb. 4.3.a) startet der Bewegungsablauf zwei Sekunden nachdem ein zu greifendes Objekt erkannt wurde. Die Wartezeit gibt dem Schieber die Möglichkeit, sich aus dem Ablagebereich zu entfernen, damit die Roboter nicht zusammenstoßen. Die Klaue des Greifarms (Grabber) erkennt selbstständig, wie weit sie sich schließen muss, um einen Gegenstand festzuhalten. Daher kann im entsprechenden Block nichts eingestellt werden. Die Drehwinkel können Ihren Wünschen entsprechend verändert werden, sofern die Bewegung angepasst werden soll.

Bitte beachten Sie außerdem, dass der Greifarm viel Energie benötigt (vier Motoren, Hubarbeit). Halten Sie daher den **Akku gut aufgeladen**. Verzichten Sie auf die gleichzeitige Ansteuerung mehrerer Elemente, da eventuell nicht genug Energie zur Verfügung steht. Sie erkennen eine Überlastung daran, dass die LEDs an den Elementen flackern, kurz erlöschen oder alle „rot“ anzeigen.

**Erweiterte Aufgaben** (→ Arbeitsblätter  im **Download** unter [cornelsen-experimenta.de](http://cornelsen-experimenta.de)): Für die Bearbeitung durch einzelne Schülerinnen und Schüler oder in Gruppen an separaten Tischen kann die Aufgabenstellung des Greifarms um eine **Farberkennung** erweitert werden. Damit ist es möglich, die Pakete anhand ihrer Farbe an verschiedenen Stellen abzulegen. Im Beispiel (Abb. 4.3.b) wird die erkannte Farbe in eine Zahl umgewandelt und in einer Variablen gespeichert. Abhängig vom Wert der Variablen werden unterschiedliche Bewegungen ausgeführt, die jeweils das Paket an eine andere Stelle transportieren.

Es ist möglich, die Aufgabe ohne die Einführung bzw. Nutzung einer Variablen zu lösen (Abb. 4.3.c). Dabei muss die Abfrage des Farbsensors sofort erfolgen, wenn der Multisensor das Vorhandensein eines Objekts erkannt hat. Sobald das Paket vom Greifarm angehoben wird ist die Farbinformation nicht mehr verfügbar. Der Unterschied ist gering: Die Funktion „aufheben“ muss nach der Abfrage der Farbinformation aufgerufen werden.

**Zusätzliche Debugging-Aufgabe** (→ Arbeitsblatt  Seite 31)

4.3.a Beispiel für die Bewegungssteuerung des Greifarms

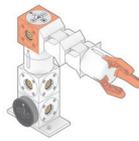
Bitte beachten Sie, dass die Farbkodierung der Roboter Elemente in Ihrem Modell von denen im Beispiel abweichen kann. Am blau markierten „Pivot“ ist die Klaue des Greifarms befestigt.

Mit dem **eXperiBot Manager** können Sie die Identifikationsfarben der Module an Ihr Programm anpassen – d.h. Sie müssen nicht mehr den Code ändern, wenn die Vergabe der Modulfarben durch das Powerbrain sich geändert hat, sondern können einfach die ursprüngliche Kombination wieder herstellen.



Klicken sie, während ein eXperiBot mit der App verbunden ist, auf den eXperiBot-Button oben rechts, um den eXperiBot Manager zu öffnen.

# 4.3 Smarter Greifer



## 4.3.b Programm für farbabhängigen Transport

## 4.3.c Programm für farbabhängigen Transport ohne Variable

Musterlösungen  
als Code-Dateien  
für die App finden  
Sie im Download-  
bereich unter  
[cornelsen-  
experimenta.de](http://cornelsen-experimenta.de)

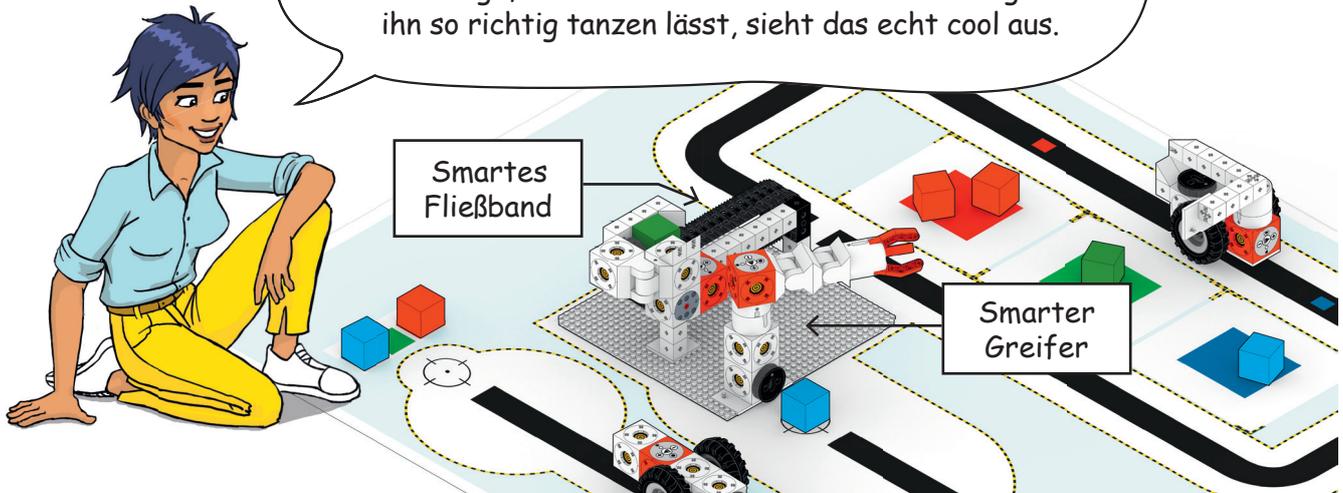
Name \_\_\_\_\_

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.



## 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  $-90^\circ$  bis  $90^\circ$  annehmen.

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

### Befehlsbox



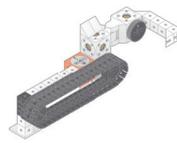
## 2. Programmieraufgabe:

- 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.

# 4.4 Smartes Fließband



## 4.4.a Einfaches Programm für Fließband

Musterlösungen als Code-Dateien für die App finden Sie im Downloadbereich unter [cornelsen-experimenta.de](http://cornelsen-experimenta.de)

Die grundsätzliche Aufgabe des Fließbandes im Verbund der Fabrik-Geräte ist der Transport eines Pakets zum Sortier-Roboter. Das lässt sich mit einem einfachen Programm umsetzen (siehe oben Abb. 4.4.a)

Insofern kann die Bearbeitung der Fließband-Aufgabe ggf. schwächeren Teams bei der gemeinsamen Bearbeitung zugeordnet werden. Auch Anfängerinnen und Anfänger finden sich hier gut zurecht und haben mehr Zeit, verschiedene Versuche durchzuführen. Der von Arianna in ihrer Einleitung zur Aufgabe erwähnte „Energiespar-Ansatz“ ist ebenfalls integriert.

Im **Lernloop** werden die Schülerinnen und Schüler angeleitet, das Fließband mit einer Sortierfunktion auszustatten. Links (Abb. 4.4.b) findet sich eine beispielhafte Lösung für ein Programm, mit dem alle Pakete aussortiert werden können, die nicht als rot, grün oder blau gefärbt erkannt werden. Damit der Multisensor die Farben besser erkennt, wird das Paket zunächst durch eine kontinuierliche rückläufige Bewegung des Bandes an den Sensor herangeführt. Das läuft der Energiespar-Vorgabe zuwider, führt aber zu korrekter Farberkennung. Eventuell können durch Anpassung des Robotermodells beide Ziele erreicht werden.

**Erweiterte Aufgaben** (→ Arbeitsblätter im Download unter [cornelsen-experimenta.de](http://cornelsen-experimenta.de)):

Für die Bearbeitung durch einzelne Schülerinnen und Schüler oder in Gruppen an separaten Tischen kann die Aufgabenstellung des smarten Fließbands um eine **Farberkennung** erweitert werden (Abb. 4.4.c). Damit ist es möglich, die Pakete anhand ihrer Farbe in verschiedene Richtungen zu transportieren.

## 4.4.b Programm zum Aussortieren nicht erkannter Farben

## 4.4.c Erweitertes Programm für Fließband

Name \_\_\_\_\_

Klasse \_\_\_\_\_

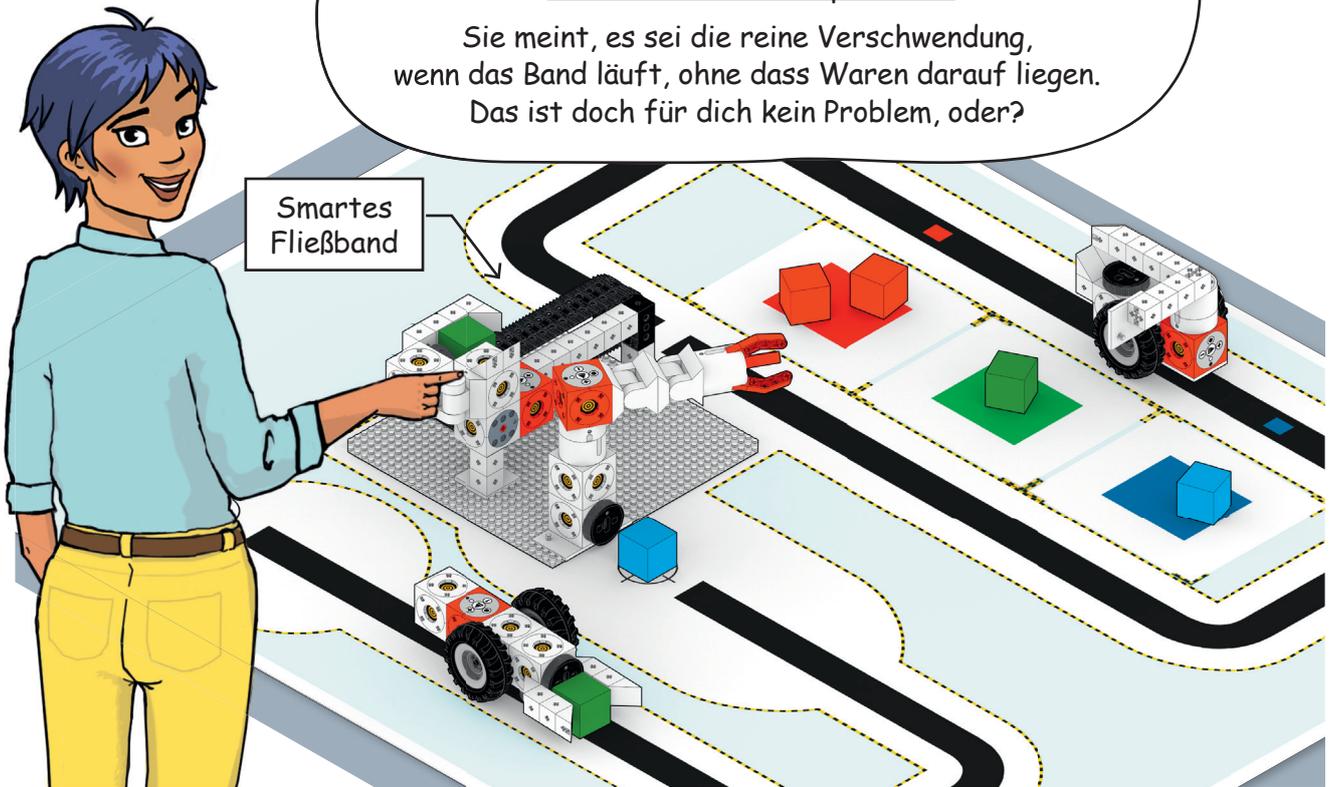
Datum \_\_\_\_\_

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?



### Programmieraufgabe:

Das Fließband soll sich einschalten, wenn der Multisensor ein Objekt erkennt.

Erstelle das Programm und probiere es aus.

### Befehlsbox

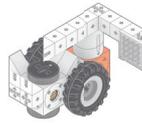
Multisensor erkennt Objekt mit normaler Empfindlichkeit

Motor setze Geschwindigkeit der Achse(n) auf 50 % für 5 s

### 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.

# 4.5 Smarter Sortierer



Der Sortierer erhält Pakete vom Fließband, die er entsprechend ihrer Farbe in die richtigen Ausgabebereiche transportieren muss. Dafür muss er einerseits die Linie richtig erkennen um auf seinem vorgegebenen Fahrweg zu bleiben. Andererseits muss er eine Information darüber erhalten, auf welchem Abschnitt des Fahrwegs er sich befindet, um die richtige Position zum „Auswerfen“ der Fracht zu ermitteln.

(Abb. 4.5.b): Um auf der Linie zu bleiben, kann die Funktion „linie\_folgen“ aus dem Programm des Smarten Schiebers angepasst werden (siehe Abb. 4.2.a auf Seite 17). Die Fälle B, C und D des Multisensor-Zustands treten weiterhin auf. In den Fällen B und C muss sich der Roboter durch Drehung zur Linie zurückbewegen. In Fall D kann er geradeaus weiterfahren. Fall A wird zum Fehlerzustand.

Zum Erkennen der Farbe des aufgelegten Pakets dient die folgende Funktion (Abb. 4.5.a). Sie ordnet der jeweils erkannten Farbe einen Zahlenwert zu und liefert ihn als Rückgabewert zurück. Damit können später Vergleiche besser durchgeführt werden.

Wird eine der vordefinierten Farben erkannt, so liefert die Funktion den entsprechenden Wert. Wird keine Farbe erkannt, so liefert sie den Wert 0. Damit kann der Roboter entscheiden, wann er auf einen Farbwert reagieren muss.

4.5.a Funktion zur Farbwertermittlung (oben)

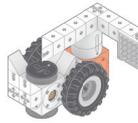
```
Funktion: farbe_oben
falls Multisensor erkennt Farbe Rot mit normaler Empfindlichkeit gib zurück 1
falls Multisensor erkennt Farbe Grün mit normaler Empfindlichkeit gib zurück 2
falls Multisensor erkennt Farbe Blau mit normaler Empfindlichkeit gib zurück 3
gib zurück 0
```

Musterlösungen als Code-Dateien für die App finden Sie im Downloadbereich unter [cornelsen-experimenta.de](http://cornelsen-experimenta.de)

4.5.b Linienfolger für den Sortierroboter

```
Funktion: folge_bis_farbe mit farbe
wiederhole bis farbe_unten = farbe
mache
falls Multisensor erkennt dunklen Untergrund links und Multisensor erkennt hellen Untergrund rechts
mache Motor setze Geschwindigkeit der roten Achse auf 0 % und der blauen Achse auf 25 %
sonst falls Multisensor erkennt hellen Untergrund links und Multisensor erkennt dunklen Untergrund rechts
mache Motor setze Geschwindigkeit der roten Achse auf 25 % und der blauen Achse auf 0 %
sonst Motor setze Geschwindigkeit der roten Achse auf 25 % und der blauen Achse auf 25 %
Motor stoppe Achse(n)
```

# 4.5 Smarter Sortierer



Die Position auf dem Rundkurs erkennt der Roboter durch den Farbsensor im nach unten gerichteten Multisensor. Unten abgebildet (Abb. 4.5.c) findet sich die **Funktion zur Farbwertzuordnung**. Dabei wird ein vierter Farbwert eingeführt, der für „weiß“ steht. Damit kann der Roboter die Position des Fließbandes ermitteln, die durch einen weißen Streifen markiert ist.

Das **Hauptprogramm** (Abb. 4.5.d) des Sortier-Roboters besteht aus den folgenden Abschnitten:

1. Warten, bis ein Paket auf die Transportfläche des Roboters fällt.
2. Warten, bis die Farbe des Pakets sicher erkannt wurde. Das ist nötig, damit der Roboter erst mit seiner Runde beginnt, wenn eine der drei vordefinierten Farben erkannt wurde. Ließe man die Prüfung weg, würde eventuell der Farbwert 0 ermittelt und an die Funktion „folge\_bis\_farbe“ übergeben. Der Roboter würde das Paket sofort auswerfen und eine „Leer-Runde“ fahren.

3. Fahren, bis der auf den Fahrweg gerichtete Multi-sensor dieselbe Farbe erkennt wie der auf das Paket gerichtete: folge\_bis\_farbe (farbe\_oben)
4. Stoppen und das Paket auswerfen: Twister dreht sich.
5. Weiterfahren, bis der Fahrweg-Multi-sensor „weiß“ erkennt: folge\_bis\_farbe (4), 4 = weiß
6. Stoppen. Der Roboter befindet sich wieder bei seiner Ausgangsposition am Fließband und kann neue Pakete laden.

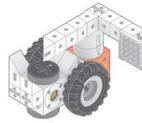
### Hinweise zur Farberkennung

Die Farberkennungsleistung des Multisensors hängt stark von der Umgebungsbeleuchtung und Oberflächenbeschaffenheit der Unterlage ab. Der Block „Multisensor erkennt Farbe (rot) mit angepasster Empfindlichkeit (100%)“ kann verwendet werden, um Fehler bei der Farberkennung zu reduzieren (Abb. 4.5.e).

4.5.c Funktion zur Farbermittlung (unten)

4.5.d Hauptprogramm des Smarten Sortierers

# 4.5 Smarter Sortierer



Wird als Empfindlichkeit ein Wert größer 100% ausgewählt, so wird die gewählte Farbe seltener erkannt. Dies verhindert z. B. die Erkennung eines blauen Bereichs auf der schwarzen Fahrlinie des Sortierers.

Bei der Einstellung sollten die Empfindlichkeit zunächst in 5-er Schritten in die gewünschte Richtung geändert werden, bis der erhoffte Effekt eintritt. Bei weiteren Tests kann eine Fein-Einstellung in 1-er Schritten die Erkennungsleistung weiter verbessern.

Wird als Empfindlichkeit ein Wert kleiner 100% ausgewählt, so wird die gewählte Farbe häufiger bzw. früher erkannt. Dies kann jedoch zu ungewollter Erkennung führen, obwohl eine Farbe nicht vorhanden ist.

### Debugging

Ein Beispiel für ein Programm zum Testen der Farbeinstellung findet sich in den Debugging-Übungen. (→ Arbeitsblatt Seite 30); Lösung siehe [Abb. 4.5.f](#)

4.5.e Funktion zur Farbwertermittlung mit dem unteren Sensor inkl. Beispiel für eine Fein-Einstellung

```

Funktion: farbe_unten
falls (Multisensor erkennt Farbe Rot mit angepasster Empfindlichkeit von 70 %) gib zurück 1
falls (Multisensor erkennt Farbe Grün mit angepasster Empfindlichkeit von 120 %) gib zurück 2
falls (Multisensor erkennt Farbe Blau mit angepasster Empfindlichkeit von 130 %) gib zurück 3
falls (Multisensor gib Farbinintensität von Rot > 1500 und Multisensor gib Farbinintensität von Grün < 1500) gib zurück 0

```

4.5.f Lösung für den Farbsensor-Debugger

```

starte Programm
wiederhole unendlich
mache setze piepser auf 0
falls (Multisensor erkennt Farbe Rot mit normaler Empfindlichkeit)
mache setze piepser auf 1
falls (Multisensor erkennt Farbe Grün mit normaler Empfindlichkeit)
mache setze piepser auf 2
falls (Multisensor erkennt Farbe Blau mit normaler Empfindlichkeit)
mache setze piepser auf 3
wiederhole (piepser) -mal:
mache spiele Note C6 für 0.2 s
warte 0.2 s
warte 1 s

```

Name \_\_\_\_\_

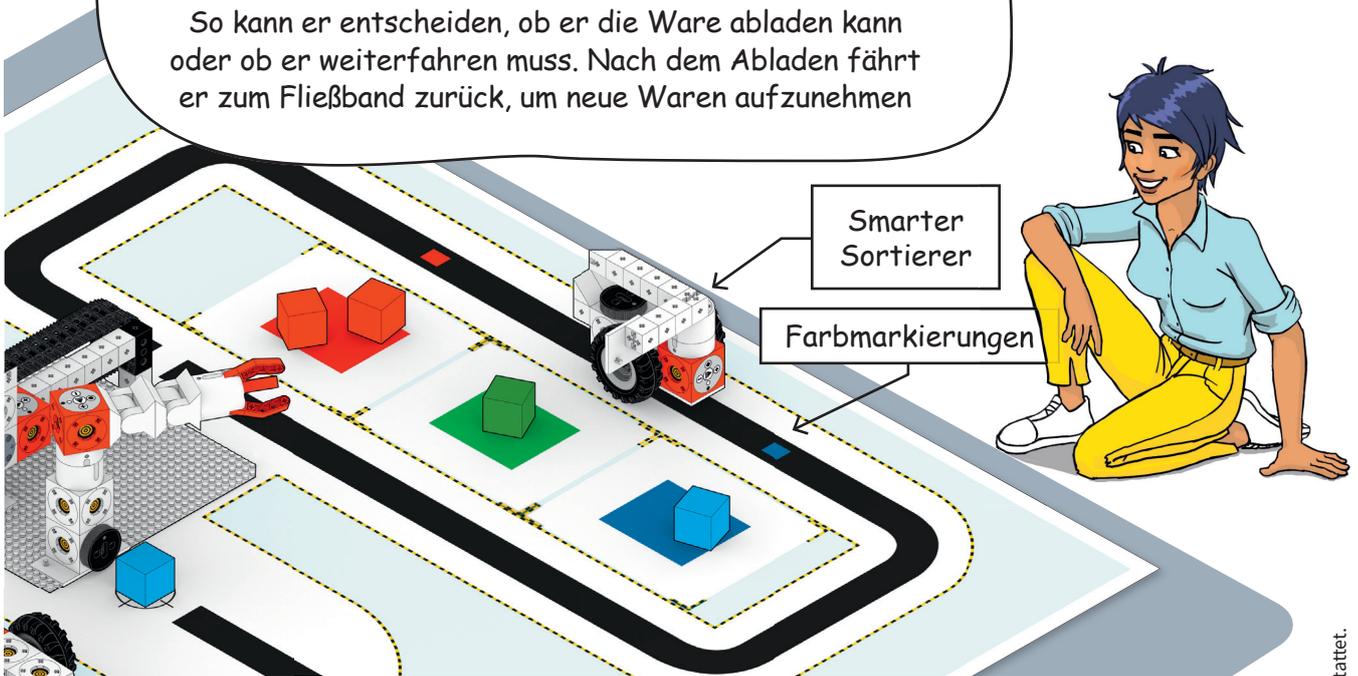
Klasse \_\_\_\_\_

Datum \_\_\_\_\_

In der letzten Station der Anlage werden die Waren für den Versand sortiert. Drei Laderampen für LKWs stehen zur Verfügung. Der **Smarte Sortierer** wird am Fließband beladen und muss seine Ladung zur richtigen Laderampe bringen.

Siehst du die farbigen Markierungen auf dem Fahrweg? Durch sie kann der Smarte Sortierer erkennen, an welcher Laderampe er sich befindet.

So kann er entscheiden, ob er die Ware abladen kann oder ob er weiterfahren muss. Nach dem Abladen fährt er zum Fließband zurück, um neue Waren aufzunehmen



## 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

The screenshot shows a programming block titled 'Funktion: farbe\_oben'. It contains three 'falls' (if) conditions, each with a 'Multisensor erkennt Farbe' block. The first condition is for 'Rot' (red), the second for 'Grün' (green), and the third for 'Blau' (blue). Each condition is followed by 'mit normaler Empfindlichkeit' and 'gib zurück' with a number (1, 2, and 3 respectively). Below these conditions is a 'gib zurück 0' block.

## Lernloop

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?



**Weitere Arbeitsblätter** zu den einzelnen Stationen und editierbare Word-Dokumente von allen Projekten finden Sie im Downloadbereich unter [cornelsen-experimenta.de](https://cornelsen-experimenta.de).

Name \_\_\_\_\_

Klasse \_\_\_\_\_

Datum \_\_\_\_\_

```

starte Programm
wiederhole unendlich
  mache
    falls
      mache Motor setze Geschwindigkeit der Achse(n) auf 25 % für 0.1 s
    sonst falls
      mache Motor setze Geschwindigkeit der Achse(n) auf 25 % für 0.1 s
    sonst falls
      mache Motor fahre rückwärts für 1 s mit mittlerer Geschwindigkeit
      Motor setze Geschwindigkeit der roten Achse auf -20 % und der blauen Achse auf 20 %
      warte 1 s
      warte bis
    sonst
      mache Motor setze Geschwindigkeit der Achse(n) auf 25 % 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?

► Nutze dazu diese Blöcke:

[Multisensor erkennt hellen Untergrund links]
   
 [Multisensor erkennt hellen Untergrund rechts]
   
 [Multisensor erkennt dunklen Untergrund links]
   
 [Multisensor erkennt dunklen Untergrund rechts]
   
 [und]

Die Vervielfältigung dieser Seite ist für den eigenen Unterrichtsgebrauch gestattet.

Name \_\_\_\_\_

Klasse \_\_\_\_\_

Datum \_\_\_\_\_

Verflixt, 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??



- **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 \_\_\_\_\_

Programm A

Programm B

```

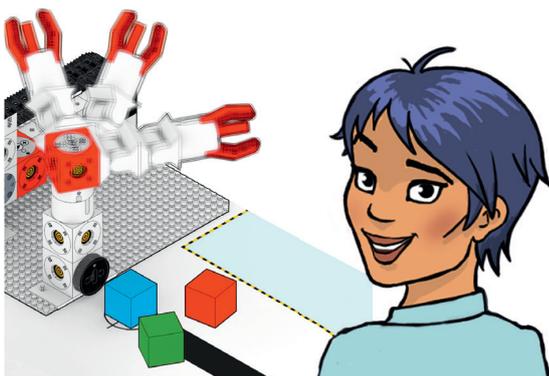
starte Programm
wiederhole unendlich
mache
  Pivot drehe auf 0 ° mit 50 % Geschwindigkeit
  Pivot drehe auf 90 ° mit 50 % Geschwindigkeit
  Twister drehe auf 90 ° mit 50 % Geschwindigkeit
  warte bis Multisensor erkennt Objekt mit normaler Empfindlichkeit
  spiele Note C6 für 1 s
  warte 6 s
  Grabber öffnen
  Pivot drehe auf -30 ° mit 50 % Geschwindigkeit
  Pivot drehe auf -60 ° mit 50 % Geschwindigkeit
  Grabber schließen
  Pivot drehe auf 0 ° mit 50 % Geschwindigkeit
  Pivot drehe auf 90 ° mit 50 % Geschwindigkeit
  Twister drehe auf -90 ° mit 50 % Geschwindigkeit
  Pivot drehe auf -90 ° mit 50 % Geschwindigkeit
  Grabber öffnen
  
```

```

starte Programm
wiederhole unendlich
mache
  Pivot drehe auf 0 ° mit 50 % Geschwindigkeit
  Pivot drehe auf 0 ° mit 50 % Geschwindigkeit
  Twister drehe auf 0 ° mit 50 % Geschwindigkeit
  warte bis Multisensor erkennt Objekt mit normaler Empfindlichkeit
  spiele Note C6 für 1 s
  warte 6 s
  Twister drehe auf -90 ° mit 50 % Geschwindigkeit
  Pivot drehe auf -90 ° mit 50 % Geschwindigkeit
  Pivot drehe auf -60 ° mit 50 % Geschwindigkeit
  Grabber schließen
  Pivot drehe auf 0 ° mit 50 % Geschwindigkeit
  Grabber öffnen
  Pivot drehe auf 90 ° mit 50 % Geschwindigkeit
  Grabber öffnen
  Pivot drehe auf -30 ° mit 50 % Geschwindigkeit
  
```

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?



Programm C

```

starte Programm
wiederhole unendlich
mache
  Pivot drehe auf 0 ° mit 50 % Geschwindigkeit
  Grabber schließen
  Pivot drehe auf 0 ° mit 50 % Geschwindigkeit
  spiele Note C6 für 1 s
  Twister drehe auf -90 ° mit 50 % Geschwindigkeit
  Pivot drehe auf -90 ° mit 50 % Geschwindigkeit
  Grabber öffnen
  warte 6 s
  Grabber öffnen
  Pivot drehe auf -30 ° mit 50 % Geschwindigkeit
  Pivot drehe auf -60 ° mit 50 % Geschwindigkeit
  Pivot drehe auf 90 ° mit 50 % Geschwindigkeit
  warte bis Multisensor erkennt Objekt mit normaler Empfindlichkeit
  Pivot drehe auf 90 ° mit 50 % Geschwindigkeit
  Twister drehe auf 90 ° mit 50 % Geschwindigkeit
  
```

Die Vervielfältigung dieser Seite ist für den eigenen Unterrichtsgebrauch gestattet.

Lösung für die Debugging-Aufgabe auf Seite 29

Lösung für den Farbsensor-Debugger

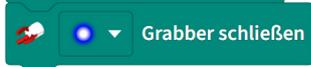
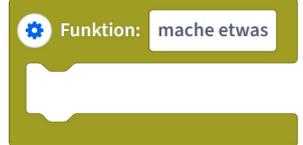
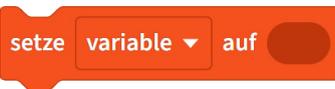
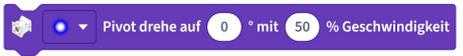
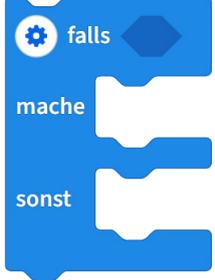
Korrektes Greifarm-Programm: A

Die Vervielfältigung dieser Seite ist für den eigenen Unterrichtsgebrauch gestattet.

# 5 Hilfekarten

## (Kopiervorlage Vorderseiten)

Sie können die beiden Seiten auf ein Blatt vorder- und rückseitig kopieren und ausschneiden.  
Die Karten als pdf zum Ausdrucken finden Sie hier: [cornelsen-experimenta.de/experibot](http://cornelsen-experimenta.de/experibot)

# 5 Hilfekarten

## (Kopiervorlage Rückseiten)

<p><b>Funktions-Block</b></p> <p>Fasst häufig benötigte Befehlsblöcke zu einem „neuen“ einzelnen Block (Unterprogramm) zusammen.</p>	<p><b>Vorprüfende Schleife</b></p> <p>Die Schleife läuft, <b>solange</b> die Bedingung erfüllt ist oder <b>bis</b> die Bedingung erfüllt ist.</p>	<p><b>Bewegungs-Block</b></p> <p>Öffnet oder schließt die Arme eines Grabber-Moduls.</p>
<p><b>Bewegungs-Block</b></p> <p>Lässt das Pivot-Modul zu einem bestimmten Winkel (-90° bis 90°) drehen. Dein Programm wird erst weiter ausgeführt, wenn der angegebene Winkel erreicht ist.</p>	<p><b>Kontroll-Block</b></p> <p>Lässt dein Programm für die eingegebene Zeit pausieren.</p>	<p><b>Variable</b></p> <p>Variablen sind veränderliche Werte, die im laufenden Programm angepasst werden. Sie können mit beliebigen Namen angelegt werden.</p>
<p><b>Verzweigung</b></p> <p>Entscheidet je nach Wert der Bedingung, ob die Befehle in der Klammer „dann“ oder in der Klammer „sonst“ ausgeführt werden.</p>	<p><b>Sensor-Block</b></p> <p>Gibt „wahr“ zurück, wenn im Sichtfeld des Farbsensors die angegebene Farbe erkannt wurde.</p>	<p><b>Bewegungs-Block</b></p> <p>Lässt das Twister-Modul zu einem bestimmten Winkel (-90° bis 90°) drehen. Dein Programm wird erst weiter ausgeführt, wenn der angegebene Winkel erreicht ist.</p>



Handreichung (71630 5)

eXperiBot Unterrichtsmaterialien  
*Smarte Fabrik*

Cornelsen Experimenta GmbH  
Holzhauser Straße 76  
13509 Berlin

Für Bestellungen und Anfragen:  
Service **Tel.:** 0800 435 90 20  
Tel.: +49 (0)30 435 902-0  
Service **Fax:** 0800 435 90 22  
Fax: +49 (0)30 435 902-22

E-Mail:  
[info@cornelsen-experimenta.de](mailto:info@cornelsen-experimenta.de)

[cornelsen-experimenta.de](http://cornelsen-experimenta.de)

Ref. 03.22