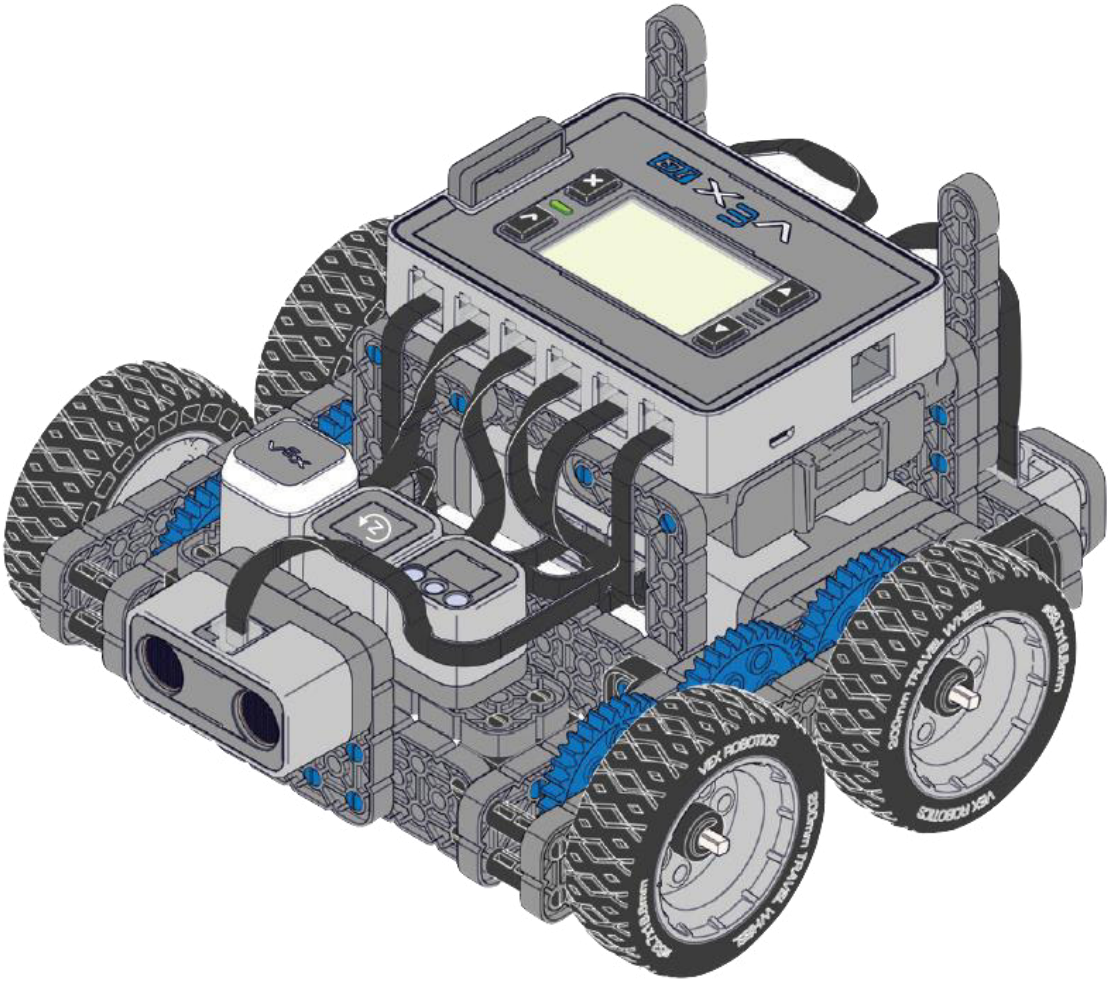


Kursprogramm
Autonomes Fahren

VEX IQ Kursprogramm Autonomes Fahren

Kursinhalte: Im VEX IQ Kurs Autonomes Fahren bauen die Schüler*innen Schritt für Schritt einen Fahrroboter auf. Dabei lernen sie von der Mechanik über die elektrischen und elektronischen Komponenten (Sensoren, Aktoren, Controller) bis hin zur Programmierung die wesentlichen Elemente kennen, die für ein Autonomes Fahrzeug benötigt werden.

Kursaufbau	
Übung 1	**BaseBot Aufbauen** Die Schüler*innen bauen den BaseBot als Basis des Fahrroboter auf und lernen daran die Bauteile und deren Funktion kennen. Der BaseBot wird abschließend mit der Fernbedienung in Betrieb genommen und als nicht-autonomes Fahrzeug betrieben.
Übung 2	**Geradeaus Fahren** Die Schüler*innen programmieren eine einfache Geradeaus Fahrt mit einer Zeit- bzw. einer Umdrehungsvorgabe bei unterschiedlichen Motorleistungen. Abschließend muss ein Programm für einen Pendelfahrt-Parcours erstellt werden.
Übung 3	**Kurven Fahren** Die Schüler*innen programmieren eine Drehbewegung mit einer Zeit- bzw. einer Umdrehungs-vorgabe. Abschließend muss ein Programm für einen Zickzack-Parcours erstellt werden.
Übung 4	**Verwendung des Kontaktschalters** Die Schüler*innen lernen die Funktionsweise eines Kontaktschalters und integrieren diesen in den Fahrroboter und die Programmierung. Abschließend wird ein Programm geschrieben, bei dem der Fahrroboter Gegenstände erkennt und selbstständig wendet.
Übung 5	**Verwendung der Touch-LED** Die Schüler*innen lernen die Funktionsweise der Touch-LED und integrieren diesen in den Fahrroboter und die Programmierung. Abschließend wird ein Programm geschrieben, bei dem die Touch-LED die Fahrtrichtung des Fahrroboters anzeigt.
Übung 6	**Verwendung des Abstandssensors** Die Schüler*innen lernen die Funktionsweise des Abstandssensors und integrieren diesen in den Fahrroboter und die Programmierung. Abschließend wird der Fahrroboter so programmiert, dass er mithilfe des Abstandssensors durch ein Würfelparcours findet.
Übung 7	**Verwendung des Farbsensors** Die Schüler*innen lernen die Funktionsweise des Farbsensors und integrieren diesen in den Fahrroboter und die Programmierung. Abschließend wird der Fahrroboter so programmiert, dass er mithilfe des Farbsensors in einer vorgegebenen Garage parkt.
Übung 8	**Verwendung des Gyro-Sensors** Die Schüler*innen lernen die Funktionsweise des Gyro-Sensors und integrieren diesen in den Fahrroboter und die Programmierung.
Projekt 1	**Fahren mit Assistenzsystemen und Teilautonomes Fahren** Die Schüler*innen recherchieren die Stufen des Autonomen Fahrens. Zum Fahren mit Assistenzsystemen und zum Teilautonomen Fahren sollen Fahrroboter aufgebaut und programmiert werden.
Projekt 2	**Autonomes Fahren** Die Schüler*innen analysieren drei vorprogrammierte, autonome Fahrprogramme und stellen die Funktionen in eigenen Programmen nach.
Projekt 3	**Fahrzeug mit Lenkgetriebe** Die Schüler*innen bauen einen Fahrroboter mit Lenkgetriebe wie im realen Straßenverkehr auf. Ein Park-Pilot als teilautonome Funktion soll programmiert werden.

BY NC SA

Übung 1

Basebot aufbauen

Übung 1 – Basebot aufbauen

Unterrichtsinhalte: In dieser Übung bauen die Schüler*innen anhand der mitgelieferten Bauanleitung den VEX IQ Basebot auf. Dieser Fahrroboter ist die Basis für alle weiteren Übungen dieses Kurses. Durch das Zuordnen von Begriffen zu Abbildungen auf einem vorbereiteten Arbeitsblatt lernen die Schüler*innen die einzelnen Komponenten des VEX IQ Bausatzes und deren Funktion und Verwendung kennen. Der aufgebaute Basebot wird mit der Fernsteuerung per Funk verbunden und kann abschließend von den Schülern durch den Raum bewegt werden.

Vermittelte Fähigkeiten	Altersgruppe	Dauer
Konstruktion, Aufbau, Funkgeräte verbinden	10-16	60 min

Lernziele

1. Die Schüler*innen können die einzelnen Komponenten des VEX IQ Bausatzes benennen und kennen deren Funktion zum Aufbau eines Roboters.
2. Die Schüler*innen können anhand einer grafischen Anleitung die erste Stufe des Fahrroboters selbstständig aufbauen.
3. Die Schüler*innen können eine Verbindung zwischen dem Robot Brain und der Fernsteuerung herstellen.

Dauer	Aktivitäten	Verbunden mit Lernziel
5 min	Einführung	1, 2, 3
10 min	Ausfüllen und Kontrolle der Arbeitsblätter zu Aufgabe 1.	1
ca. 30 min	Aufbau des VEX IQ Basebots in Kleingruppen. Die Dauer ist abhängig vom Alter und den Erfahrungen der Gruppe.	2
10 min	Verbinden des Robot Brain mit der Fernsteuerung und Test des Fahrroboters.	3
5 min	Reflexion der Übung anhand offener Fragen.	1, 2, 3
Zusatz-aufgabe	Bauen Sie einen Parcours im Klassenraum auf, auf dem die Schüler*innen ihre Geschicklichkeit im Umgang mit dem Fahrroboter zeigen können. Veranstalten Sie ein Wettrennen gegen die Uhr.	

Materialien

PowerPoint Übung 1
Arbeitsblatt “Zuordnen”
VEX IQ Bausatz
Ggf. Materialien und frei Fläche für einen Parcours

BY NC SA
Erstellt von: Dreibein Lehrsysteme GmbH

Aufgabe 1: Ordne richtig zu!
Welche Komponenten sind abgebildet?

Name(n): _____

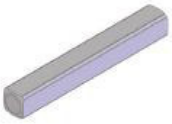
Lehrer/Klasse: _____

Datum: _____

Aufgabe: Ordne die aufgeführten Begriffe den Abbildungen zu.

Wortliste:

Winkelträger	Träger	Stift	Fernsteuerung
Winkelverbinder	Zahnrad	Gummiriemenhalter	Riemenscheibe
Platte	Funkmodul	Robot Brain	Gummiriemen
Gummimuffe	Welle	Wellenaufnahme	Smart Motor
Reifen	Stiftverbinder	Distanzstift	Felge

BY NC SA

Erstellt von: Dreibein Lehrsysteme GmbH

Flexibilität des VEX IQ Bausatzes

Die **VEX IQ** Bausätze bieten diverse Möglichkeiten, die vier Elemente der MINT-Fächer – Mathematik, Informatik, Naturwissenschaften und Technik – auf eine einfache und begeisternde Weise zu vermitteln. Dieser Kurs wird in aufeinander aufbauenden Übungen die Funktion und Verwendung der einzelnen Bestandteile des **VEX IQ** Bausatzes vermitteln.

Eine der größten Stärken der **VEX IQ** Serie ist die Flexibilität der Hardware. Lassen Sie der Kreativität Ihrer Schüler*innen freien Lauf und setzen Sie die Ideen mit **VEX IQ** um. Von mechanischen Modellen über motorgetriebenen Mechanismen und fernbedienbaren Maschinen bis hin zu vollständig autonomen Robotern ermöglichen die **VEX IQ** Bausätze ein breites Spektrum an Anwendungsmöglichkeiten.

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Dein erster Roboter

Das Ziel dieser Übung ist der Aufbau des VEX IQ BASEBOT durch die Schüler*innen anhand der im Bausatz mitgelieferten Aufbauanleitung.

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Arbeitsblatt

Lernziel

Die Schüler*innen können die einzelnen Komponenten des VEX IQ Bausatzes benennen und kennen deren Funktion zum Aufbau eines Roboters.

Aufgabe 1: Ordne richtig zu!

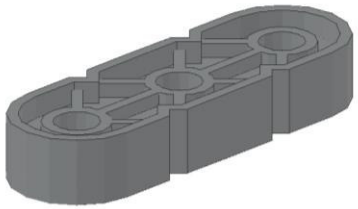
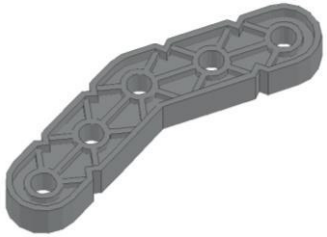
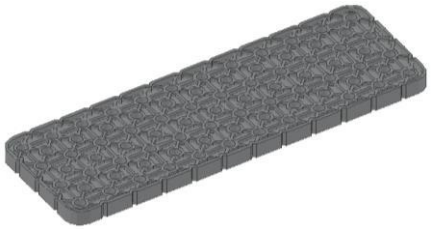
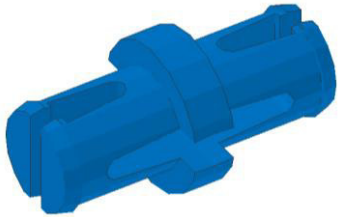
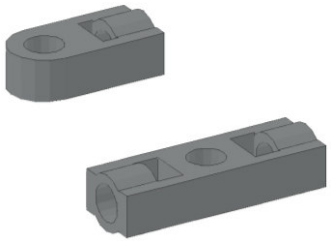
Ordne die auf dem Arbeitsblatt aufgeführten Begriffe den darunter stehenden Abbildungen zu.

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Mechanische Bauteile

Aufgabe 1: Ordne richtig zu! – Antworten

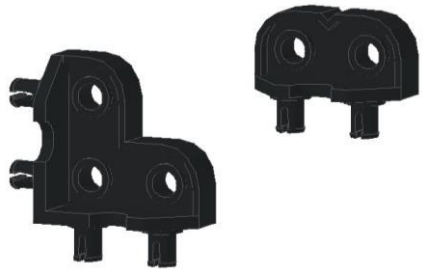
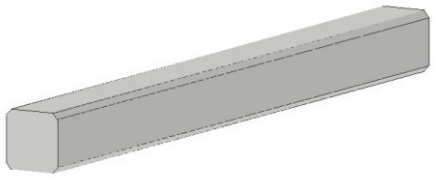
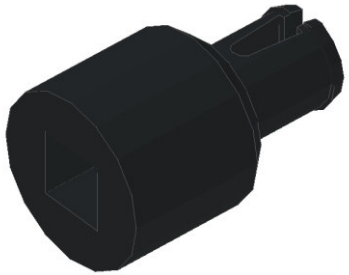
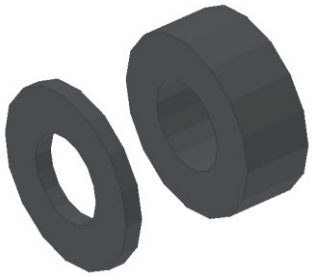
So werden die Komponenten bezeichnet und verwendet:

Träger verschiedene Größen	**Spezielle Träger** Winkel-, T-, rechtwinklige Träger	**Platte** verschiedene Größen
		
Mechanisches Grundbauteil	Mechanisches Grundbauteil	Mechanisches Grundbauteil
Stift verschiedene Längen	**Distanzstift** verschiedene Größen	**Stiftverbinder** verschiedene Typen
		
Zum Verbinden von Trägern, Platten, Winkelverbindern und anderen Bauteilen	Ermöglicht das Verbinden von Bauteilen mit einen festen Abstand	Verbindet Stifte und/oder Distanzstifte miteinander

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

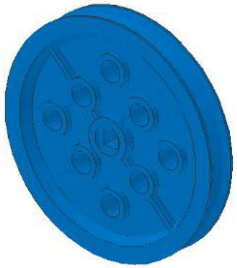
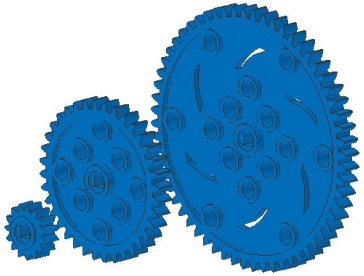
Mechanische Bauteile

Winkelverbinder verschiedene Typen	**Welle** Metall & Kunststoff, verschiedene Längen	**Wellenaufnahme**
		
Ermöglicht Eckverbindungen zwischen Trägern, Platten und anderen Bauteilen	Ermöglicht die Kraftübertragung zwischen drehbaren Teilen wie Motor, Zahnrad, Rädern, …	Stellt eine drehbare Verbindung zwischen Wellen und Trägern oder Platten her
Wellenaufnahme-Platten verschiedene Größen	**Gummimuffe**	**Distanzscheiben,- und hülsen**
		
Ermöglicht eine starre Verbindung zu Wellen	Fixiert Bauteile auf einer Welle oder die Achse selbst	Reduziert die Reibung bei drehenden Teilen und hält einen festen Abstand auf einer Welle

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Mechanische Bauteile

Riemenscheibe verschiedene Typen	**Gummiriemen** verschiedene Typen	**Gummiriemen-Halter**
		
Zur Aufnahme von Riemen, zum Aufbau von Flaschenzügen und zur Verwendung als kleines Rad	Zur Verwendung mit Riemen-scheiben, als Befestigungselement oder zum Speichern von Energie	Zum Fixieren von Gummiriemen und Bändern
Zahnräder verschiedene Größen	**Felgen, Reifen und Räder** Verschiedene Größen	**Smart Motor**
		
Zur Kraftübertragung zu anderen Zahnrädern und/oder Mechanismen	Zum Rollen und zur Kraftübertragung beim Fahren	Erzeugt eine Drehbewegung und misst den Drehwinkel

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Basebot Aufbau

Lernziel

Die Schüler*innen können anhand einer grafischen Anleitung die erste Stufe des Fahrroboters selbstständig aufbauen.

Aufgabe 2: Der Aufbau

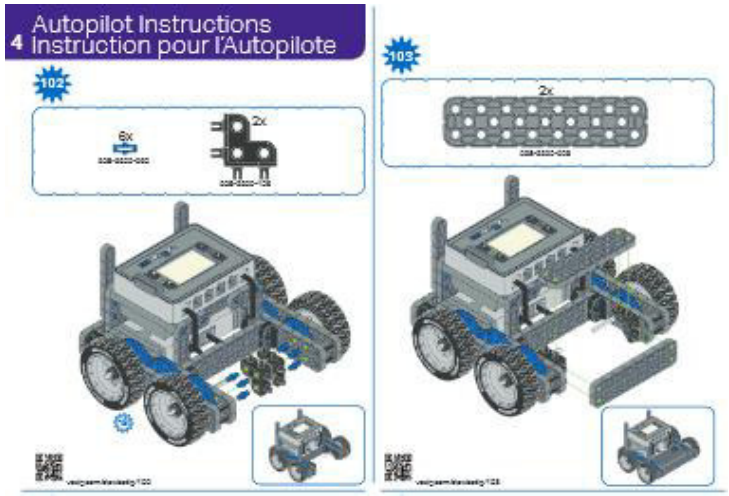
Baue und teste die erste Stufe des Fahrroboters anhand der Montageanleitung. Das Fahrzeug kann mit der Fernsteuerung von Dir bewegt werden. Welche Ideen hast du, das Fahrzeug zu verändern?

Bearbeite die Seiten 4…9 (Schritte 1…19)

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Basebot Aufbau

Wenn das geschafft ist, vervollständige den Aufbau noch mit den Schritten 102 und 103 auf Seite 32 der Montageanleitung.



FRAGE: Mit diesen Schritten wird das Chassis des Fahrzeugs ergänzt. Warum ist das sinnvoll?

ANTWORT: Die zusätzlichen Träger machen das Chassis steifer und somit den Fahrroboter robuster. Außerdem dient der Träger als Befestigung für Sensoren.

Dieser Aufbau ist die Basis für alle weiteren Übungen dieses Kurses.

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Funkverbindung herstellen

Lernziel

Die Schüler*innen können eine Verbindung zwischen dem Robot Brain und der Fernsteuerung herstellen.

Aufgabe 3: Robot Brain und Fernsteuerung verbinden

Benötigte Komponenten: Robot Brain mit Batterie und Funkmodul, Fernsteuerung mit Batterie und Funkmodul, blaues Verbindungskabel zur Fernsteuerung

FRAGE: Warum muss das Robot Brain und die Fernsteuerung zunächst per Kabel miteinander verbunden werden?

ANTWORT: Um eine Funkverbindung zwischen Robot Brain und Fernsteuerung zu ermöglichen und um bei Verwendung von mehreren Robotern eine eindeutige Zuordnung zu erhalten.

Das Verbinden des Robot Brain mit der Fernsteuerung ist in Kapitel 2 der Bedienungsanleitung (Seiten 9 und 10) beschrieben.

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Verständnisfragen

Lasst uns gemeinsam folgende Fragen beantworten:

1. Was sind die wichtigsten Komponenten des VEX IQ Bausatzes?
2. Wie paart man das Robot Brain und die Fernsteuerung?
3. Warum paart man das Robot Brain und die Fernsteuerung?
4. Wozu dienen die LEDs am Robot Brain und der Fernsteuerung?

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Zusammenfassung

Lernziel

Die Schüler*innen können:

- die einzelnen Komponenten des VEX IQ Bausatzes benennen und kennen deren Funktion zum Aufbau eines Roboters
- anhand einer grafischen Anleitung die erste Stufe des Fahrroboters selbstständig aufbauen
- eine Verbindung zwischen dem Robot Brain und der Fernsteuerung herstellen

Heute hast du:

- *die verschiedenen Komponenten des VEX IQ Bausatzes kennengelernt*
- *die Funktion und Verwendung der Bauteile verstanden*
- *gelernt, wie das Robot Brain und die Fernsteuerung gepaart werden*

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Übung 2

Geradeaus Fahren

Übung 2 – Geradeaus Fahren

Unterrichtsinhalte: In dieser Übung programmieren die Schüler*innen eine einfache Geradeausfahrt. Das Fahren mit Umdrehungsvorgabe wird mit dem Fahren mit Zeitvorgabe verglichen und der Einfluss unterschiedlicher Leistungsvorgaben getestet.

Vermittelte Fähigkeiten	Altersgruppe	Dauer
Programmieren, Parameter variieren, Testen, Code kompilieren und übertragen	10-16	60 min

Lernziele

1. Die Schüler*innen können den Aufbau und die Funktionsweise eines Drehwinkelgebers erklären und diesen für die Steuerung des Fahrroboters verwenden.
2. Die Schüler*innen können ein eigenes Programm in ROBOTC Graphical erstellen und anschließend auf das Robot Brain übertragen.
3. Die Schüler*innen können den *„forward“* Befehl zur Programmierung des Roboters einsetzen und dabei zwischen der Zeit- und Umdrehungsvorgabe unterscheiden.

Dauer	Aktivitäten	Verbunden mit Lernziel
5 min	Einführung	1, 2, 3
5 min	Vermittlung Aufbau Smart Motor und Funktionsweise Drehwinkelgeber	1, 3
10 min	Einführung in die Programmieroberfläche und den „forward“ Befehl	2, 3
20 min	Bearbeitung des Arbeitsblattes zur Verwendung des „forward-rotations“ und „forward-seconds“ Befehls mit Durchführung von Testläufen.	2, 3
15 min	Aufbau eines Pendelfahrt-Parcours und Umsetzung der Programmieraufgabe durch die Schüler*innen.	2, 3
5 min	Reflexion der Übung anhand offener Fragen.	1, 2, 3

Materialien

PowerPoint Übung 2
Arbeitsblatt “Geradeaus fahren”
VEX IQ Bausatz
VEX IQ Wettbewerbsfeld und Würfelset
Computer mit RobotC Programmierumgebung

BY NC SA
Erstellt von: Dreibein Lehrsysteme GmbH

Name(n): _____

Lehrer/Klasse: _____

Datum: _____

Aufgabe 1: „forward“ Befehl mit Umdrehungen

1. Erstelle folgendes Programm auf dem Computer und übertrage es auf das Robot Brain:

```
1 forward ( 3 , rotations , 50 );
2
```

2. Lass den Roboter das Programm dreimal durchlaufen und miss jeweils, wie weit der Roboter fährt. Trage das Ergebnis in die Tabelle unten ein und berechne den Durchschnitt.
3. Wiederhole Schritte 1 und 2 jeweils mit 75% und 100% Leistung.

Leistung	Fahrt 1	Fahrt 2	Fahrt 3	Durchschnitt
50%				
75%				
100%				

Aufgabe 2: „forward“ Befehl mit Zeit

1. Erstelle folgendes Programm auf dem Computer und übertrage es auf das Robot Brain:

```
1 forward ( 3 , seconds , 50 );
2
```

2. Lass den Roboter das Programm dreimal durchlaufen und miss jeweils, wie weit der Roboter fährt. Trage das Ergebnis in die Tabelle unten ein und berechne den Durchschnitt.
3. Wiederhole Schritte 1 und 2 jeweils mit 5 s und 7 s Fahrzeit.

Zeit	Fahrt 1	Fahrt 2	Fahrt 3	Durchschnitt
3 s				
5 s				
7 s				

Was sind deine Schlussfolgerungen?
Was geschieht, wenn in Aufgabe 2 die Leistung erhöht wird?

BY NC SA

Erstellt von: Dreibein Lehrsysteme GmbH

Deine erste Programmierung

In dieser Übung wirst du mit den Befehlen “Time” und “Rotation” dem Basebot das Geradeaus Fahren beibringen.

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

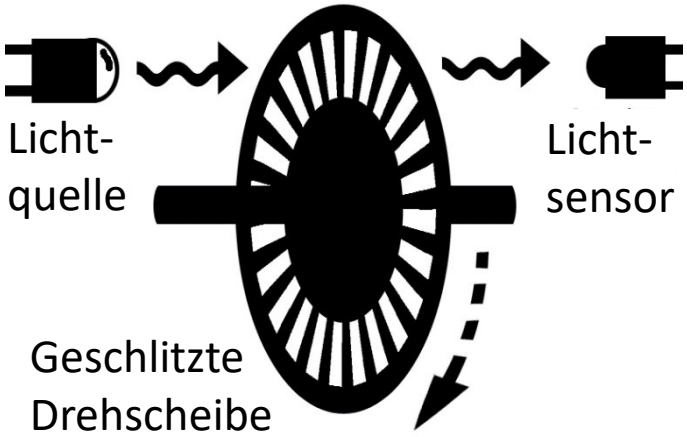
Motor mit Drehwertgeber

Lernziel

Die Schüler*innen können den Aufbau und die Funktionsweise eines Drehwinkelgebers erklären und diesen für die Steuerung des Fahrroboters verwenden.

Der VEX IQ Smart Motor

Der Motor wandelt die elektrische Energie aus der Batterie in eine Drehbewegung.
Das besondere am VEX IQ Smart Motor ist der integrierte optische Drehwinkelgeber. Der Drehwinkelgeber besteht aus einer Scheibe mit winzigen Schlitzen, einer Lampe und einem Lichtsensor. Der Smart Motor zählt mithilfe des Lichtsensors, wie oft bei der Bewegung des Motors Licht durch die Scheibe kommt und gibt diesen Wert an das Robot Brain weiter.

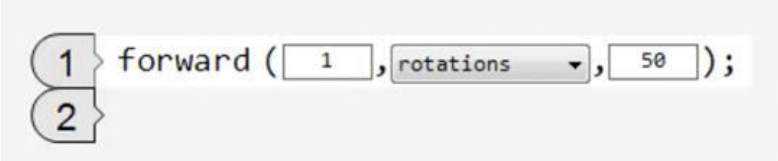


Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Programmieren mit ROBOTC

Lernziel

Die Schüler*innen können ein eigenes Programm in ROBOTC Graphical erstellen und anschließend auf das Robot Brain übertragen.



Durch Anklicken der Abbildung wird ein englischsprachiges Video im Browser geöffnet

Das erste Programm – nur ein Befehl:
Eine Umdrehung vorwärts fahren mit 50% Geschwindigkeit

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Programmieren mit ROBOTC

Übertrage das Programm vom Computer auf das Robot Brain.

Detaillierte Infos findest Du in der Anleitung zur Programmierung mit ROBOTC.

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Fahren nach Zeit und Umdrehung

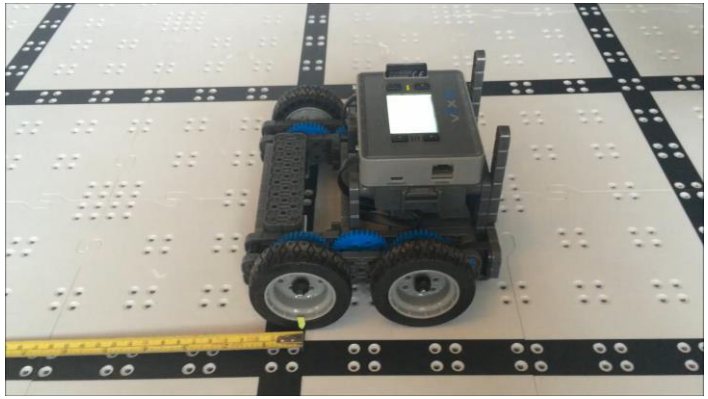
Lernziel

Die Schüler*innen können den „*forward*“ Befehl zur Programmierung des Roboters einsetzen und dabei zwischen der Zeit- und Umdrehungsvorgabe unterscheiden.

Aufgabe 1: ‘forward’ Befehl mit Umdrehungen

```
1 forward ( 3 , rotations , 50 );
2
```

Programmiere eine Vorwärtsfahrt (“forward”) für 3 Umdrehungen (“rotations”) mit einer Leistung von 50%, 75% und 100%.



Aufgabe 2: ‘forward’ Befehl mit Zeit

```
1 forward ( 3 , seconds , 50 );
2
```

Programmiere eine Vorwärtsfahrt (“forward”) für 3, 5 und 7 Sekunden (“seconds”) mit einer Leistung von 50%.

Was sind deine Schlussfolgerungen?

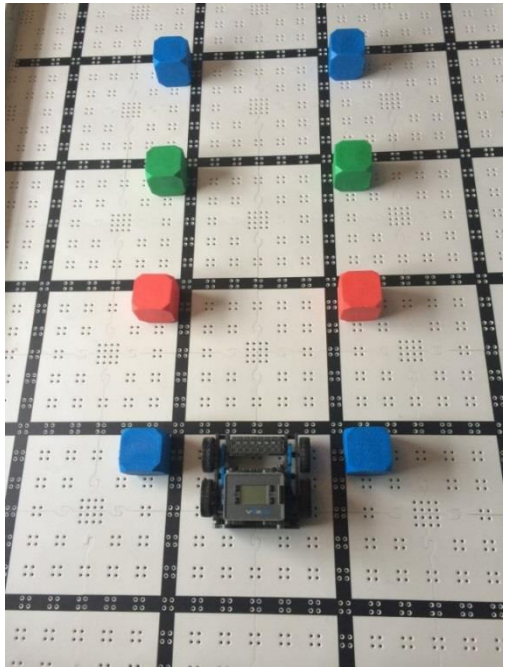
Was geschieht, wenn in Aufgabe 2 die Leistung erhöht wird?

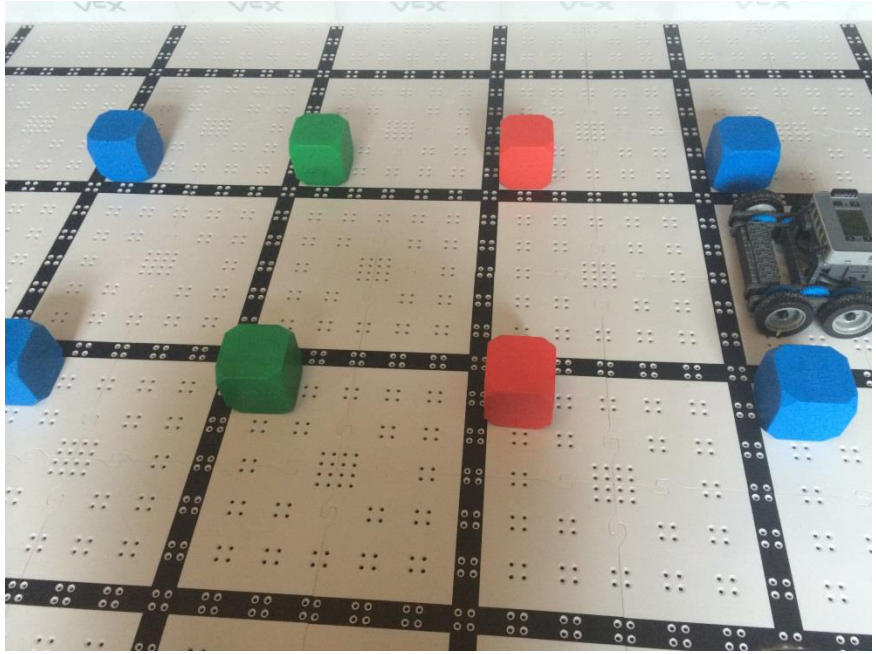
Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Pendelfahrt

Aufgabe 3: Baue einen Pendelfahrt-Parcours auf einer flachen Oberfläche auf. Schreibe ein Programm mit dem „forward-rotations“ und/oder dem „forward-seconds“ Befehl, sodass der Fahrroboter nach dem Erreichen eines Würfels immer zum Start zurückkehrt, bevor er zum nächsten Würfel fährt.





Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Pendelfahrt - Video

Link zur Online-Version auf Youtube: https://youtu.be/VPorlnht3jg

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Verständnisfragen

Lasst uns gemeinsam folgende Fragen beantworten:

1. Was war der wesentliche Unterschied zwischen dem „forward-rotations“ und „forward-seconds“ Befehl?

2. Wurde bei beiden Befehlen die selbe Strecke zurückgelegt?

3. Hat die Änderung der Leistung die zurückgelegte Strecke beeinflusst?

4. Wie könnten Zeit und zurückgelegte Strecke vom Ladezustand der Batterie beeinflusst werden?

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Zusammenfassung

Lernziel

Die Schüler*innen können:

- den Aufbau und die Funktionsweise eines Drehwinkelgebers erklären und diesen für die Steuerung des Fahrroboters verwenden
- ein eigenes Programm in ROBOTC Graphical erstellen und anschließend auf das Robot Brain übertragen
- den „*forward*“ Befehl zur Programmierung des Roboters einsetzen und dabei zwischen der Zeit- und Umdrehungsvorgabe unterscheiden

Heute hast du:

- *gelernt, wie du mit Hilfe eines Drehwertgebers einen Fahrroboter steuern kannst*
- *den Fahrroboter mit dem „forward“ Befehl programmiert*
- *untersucht, wie sich der Zeit- und Umdrehungsbefehl auf den Roboter auswirken*
- *den Pendelfahrt-Parcours mit dem Fahrroboter absolviert*

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Übung 3

Kurven Fahren

Übung 3 – Kurven Fahren

Unterrichtsinhalte: In dieser Übung programmieren die Schüler*innen eine einfache Drehbewegung des Fahrroboters. Dazu werden die Befehle *„turnRight“* und *„turnLeft“* verwendet. Das Drehen mit Umdrehungsvorgabe (Motorumdrehungen) wird mit dem Drehen mit Zeitvorgabe verglichen und der Einfluss unterschiedlicher Leistungsvorgaben diskutiert.

Vermittelte Fähigkeiten	Altersgruppe	Dauer
Programmieren, Parameter variieren, Testen, Code kompilieren und übertragen	10-16	60 min

Lernziele

1. Die Schüler*innen können mithilfe einfacher Befehle in ROBOTC Graphical die Fahrrichtung des Roboters ändern.
2. Die Schüler*innen können bei den Befehlen *„turnRight“* und *„turnLeft“* zwischen der Zeit- und der Umdrehungsvorgabe unterscheiden.
3. Die Schüler*innen können die Befehle *„turnRight“* und *„turnLeft“* bei der Programmierung selbstständig mit den passenden Vorgabewerten einsetzen.

Dauer	Aktivitäten	Verbunden mit Lernziel
5 min	Einleitung	1, 2, 3
5 min	Einführung der Befehle *“turnRight”* und *„turnLeft“*	1, 2, 3
20 min	Bearbeitung des Arbeitsblattes zur Verwendung der Umdrehungs- und Zeitvorgabe mit Durchführung von Testläufen.	2
25 min	Aufbau eines Labyrinth-Parcours und Umsetzung der Programmieraufgabe durch die Schüler*innen.	3
5 min	Reflexion der Übung anhand offener Fragen.	1, 2, 3

Materialien

PowerPoint Übung 3
Arbeitsblatt “Kurven fahren”
VEX IQ Bausatz
VEX IQ Wettbewerbsfeld und Würfelset
Laptop mit RobotC Programmierumgebung

BY NC SA
Erstellt von: Dreibein Lehrsysteme GmbH

Name(n): _____

Lehrer/Klasse: _____

Datum: _____

Aufgabe 1: „turnRight“ Befehl mit Umdrehungen

1. Erstelle folgendes Programm auf dem Computer und übertrage es auf das Robot Brain:

```
1 turnRight ( 0.5 , rotations , 50 );
2
```

2. Lass den Roboter das Programm dreimal durchlaufen und miss jeweils den Winkel, den sich der Roboter gedreht hat. Trage das Ergebnis in die Tabelle unten ein und berechne den Durchschnitt.
3. Wiederhole Schritte 1 und 2 jeweils mit 1 und 1,5 Umdrehungen.

Umdrehungen	Fahrt 1	Fahrt 2	Fahrt 3	Durchschnitt
0,5				
1				
1,5				

Aufgabe 2: „turnRight“ Befehl mit Zeit

1. Erstelle folgendes Programm auf dem Computer und übertrage es auf das Robot Brain:

```
1 turnRight ( 0.5 , seconds , 50 );
2
```

2. Lass den Roboter das Programm dreimal durchlaufen und miss jeweils den Winkel, den sich der Roboter gedreht hat. Trage das Ergebnis in die Tabelle unten ein und berechne den Durchschnitt.
3. Wiederhole Schritte 1 und 2 jeweils mit 1 und 1,5 Sekunden.

Zeit	Fahrt 1	Fahrt 2	Fahrt 3	Durchschnitt
0,5 s				
1 s				
1,5 s				

Was sind deine Schlussfolgerungen?

Was geschieht, wenn in Aufgabe 1 und in Aufgabe 2 die Leistung erhöht wird?

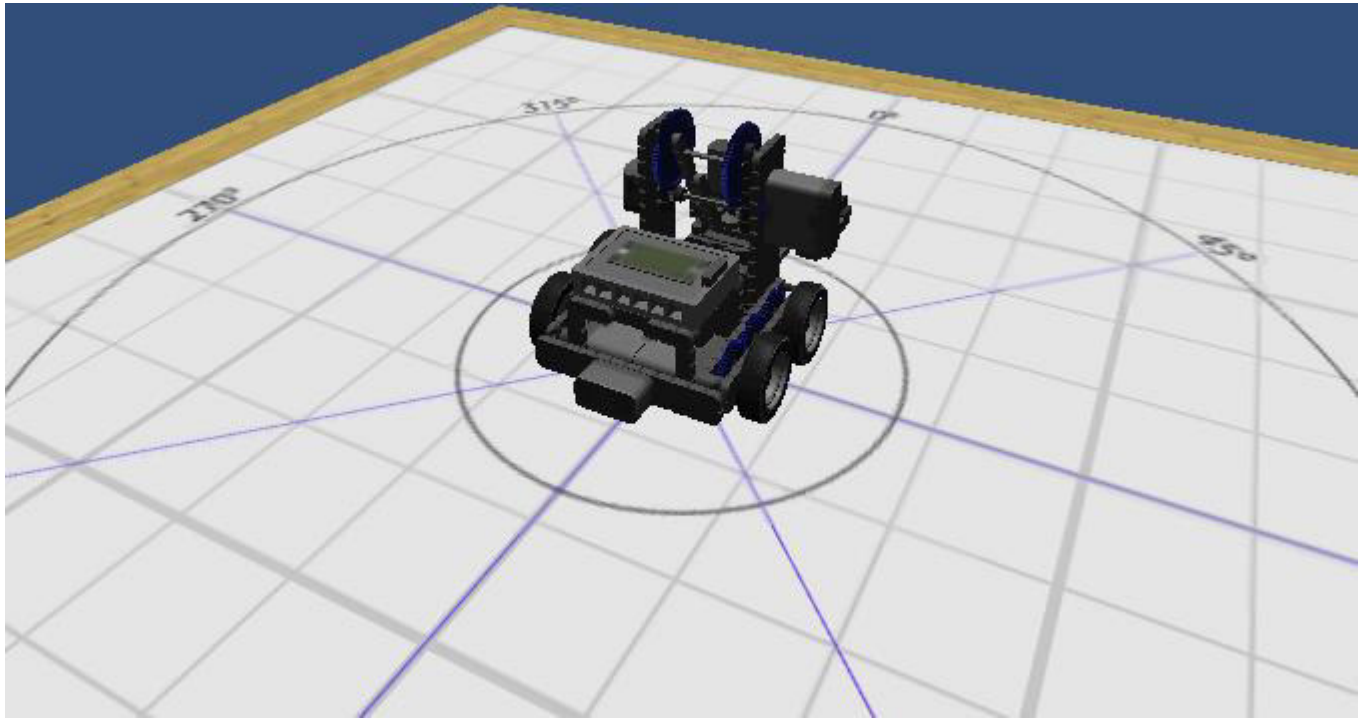
Wird der Drehwinkel des Roboters vom Ladezustand der Batterie beeinflusst?

BY NC SA

Erstellt von: Dreibein Lehrsysteme GmbH

Auf der Stelle drehen

In dieser Übung wirst du mit den Befehlen *“turnLeft”* und *“turnRight”* einfache Drehbewegungen programmieren.

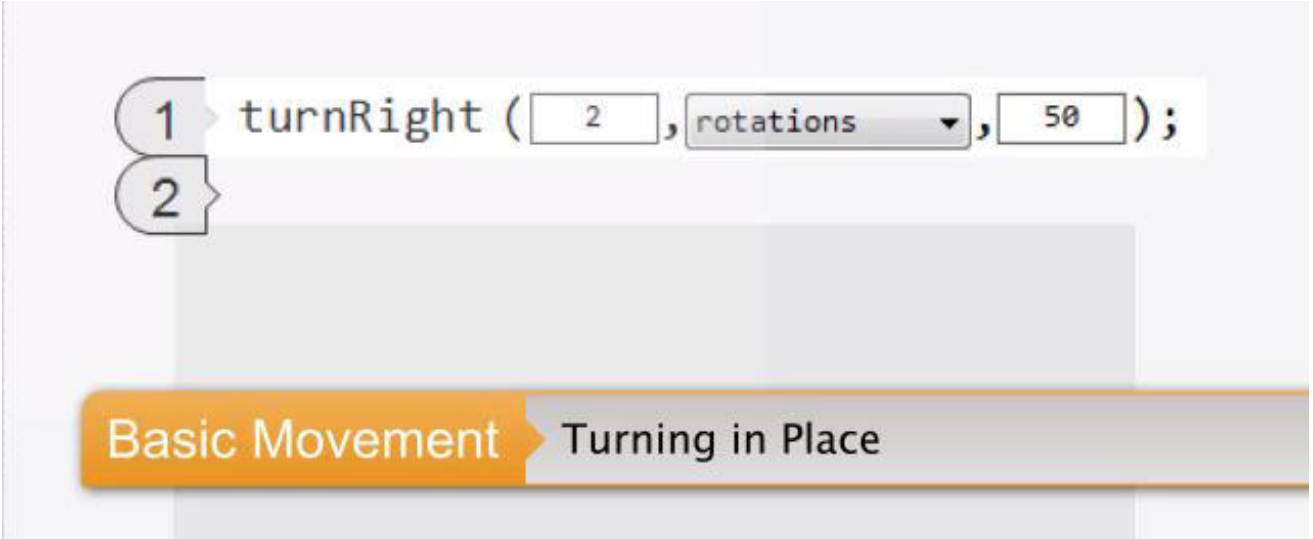


Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Programmieren mit ROBOTC

Lernziel

Die Schüler*innen können mithilfe einfacher Befehle in ROBOTC Graphical die Fahrrichtung des Roboters ändern.



Durch Anklicken der Abbildung wird ein englischsprachiges Video im Browser geöffnet
Link: http://www.education.rec.ri.cmu.edu/products/teaching_robotc_vexiq/lesson/media_videos/1_3-3_turning.mp4

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Drehen nach Zeit und Umdrehung

Lernziel

Die Schüler*innen können bei den Befehlen *„turnRight“* und *„turnLeft“* zwischen der Zeit- und der Umdrehungsvorgabe unterscheiden und setzen diese bei der Programmierung selbstständig ein.

Aufgabe 1: ‘turnRight’ Befehl mit Umdrehungen

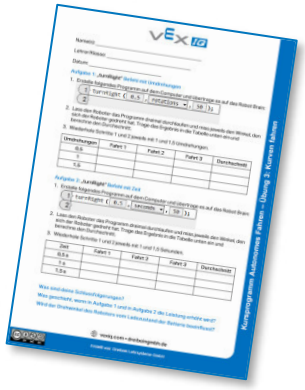
```
1 turnRight ( 1 , rotations , 50 );
2
```

Programmiere eine Drehbewegung mit den im Arbeitsblatt angegebenen Umdrehungen (“rotations”) bei einer Leistung von 50%.

Aufgabe 2: ‘turnRight’ Befehl mit Zeit

```
1 turnRight ( 1 , seconds , 50 );
2
```

Programmiere eine Drehbewegung mit den im Arbeitsblatt angegebenen Zeiten bei einer Leistung von 50%.



Was sind deine Schlussfolgerungen?

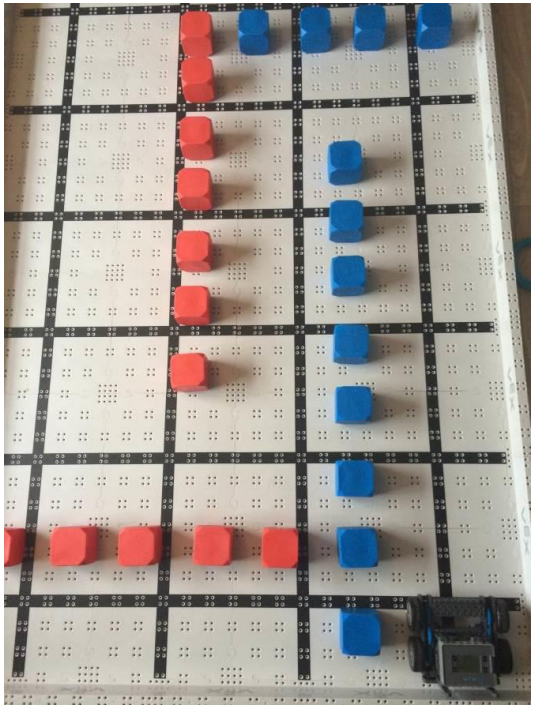
Was geschieht, wenn in den Aufgaben 1 und 2 die Leistung erhöht wird?

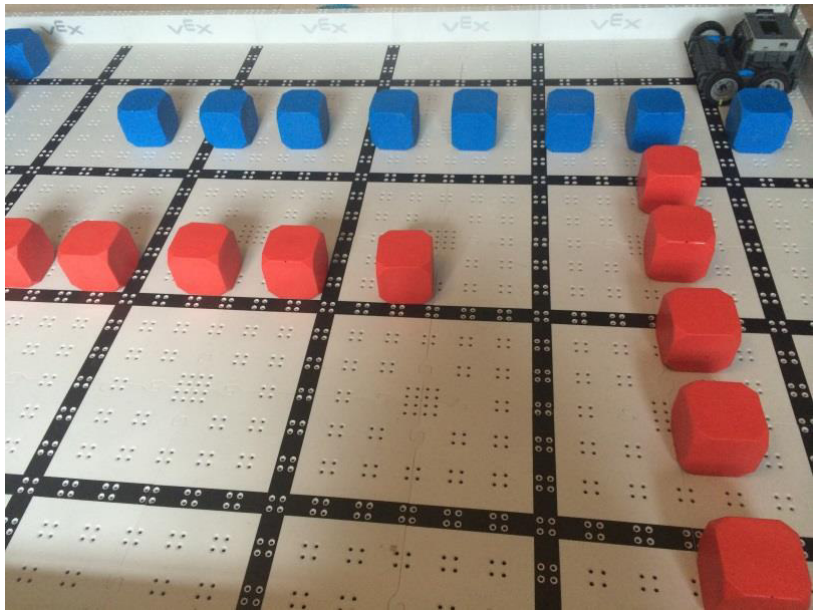
Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Labyrinth-Fahrt

Aufgabe 3: Baue einen Labyrinth-Parcours auf einer ebenen Oberfläche auf. Schreibe ein Programm mit den Befehlen *„forward“*, *„turnLeft“* und *„turnRight“*, sodass der Roboter ohne Berühren der Würfel selbstständig durch den Parcours fährt.





Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Labyrinth-Fahrt - Video

Link zur Online-Version auf Youtube: https://youtu.be/W-5TMyqZHZk

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Verständnisfragen

Lasst uns gemeinsam folgende Fragen beantworten:

1. Was war der wesentliche Unterschied zwischen dem „rotations“ und dem „seconds“ Befehl?

2. War der Drehwinkel des Roboters bei aller Versuchen gleich?

3. Kann die Änderung der Leistung den Drehwinkel beeinflussen?

4. Kann der Drehwinkel vom Ladezustand der Batterie beeinflusst werden?

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Zusammenfassung

Lernziel

Die Schüler*innen können:

- mithilfe einfacher Befehle in ROBOTC Graphical die Fahrrichtung des Roboters ändern
- bei den Befehlen *„turnRight“* und *„turnLeft“* zwischen der Zeit- und der Umdrehungsvorgabe unterscheiden
- die Befehle *„turnRight“* und *„turnLeft“* bei der Programmierung selbstständig mit den passenden Vorgabewerten einsetzen

Heute hast du:

- *gelernt, wie du die Richtung eines Fahrroboter ändern kannst*
- *den Fahrroboter mit dem „turnLeft“ und „turnRight“ Befehl programmierst*
- *untersucht, wie sich der Zeit- und Umdrehungsbefehl auf den Roboter auswirken*
- *den Labyrinth-Parcours mit dem Fahrroboter absolviert*

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Übung 4

Verwendung des Kontaktschalters

Übung 4 – Kontaktschalter

Unterrichtsinhalte: In dieser Übung lernen die Schüler*innen die Funktionsweise des Kontaktschalters kennen und befassen sich in diesem Zusammenhang mit der booleschen Logik. Durch den Einbau des Kontaktschalters in den Roboter und das Einbinden in ein selbstgeschriebenes Programm kann die Bewegung des Roboters durch Umweltbedingungen beeinflusst werden.

Themen	Altersgruppe	Dauer
Boolescher Logik, Schalter und Taster, Wahrnehmung der Umwelt, Sensoren	10-16	60 min

Lernziele

1. Die Schüler*innen können die Funktion eines Kontaktschalters erklären und erschließen sich den Nutzen zur Steuerung eines Roboters.
2. Die Schüler*innen können den Kontaktschalter selbstständig an den Fahrroboter anbauen, anschließen und die Konfiguration in der Programmierumgebung anpassen.
3. Die Schüler*innen können die Bedeutung von der Begriffe „Wahrheitswerte“ und „Vergleichsoperatoren“ in der Booleschen Logik erklären.
4. Die Schüler*innen können den Kontaktschalter in ein selbsterstelltes Programm einbinden und damit die Bewegung des Roboters steuern.
5. Die Schüler*innen können den Roboter nach einer Berührung automatisch stoppen oder die Fahrrichtung ändern lassen.

Dauer	Aktivitäten	Verbunden mit Lernziel
5 min	Einleitung	1, 2, 3
5 min	Funktionsweise und Verwendung des Kontaktschalters; Einbau in den Roboter.	1, 2
10 min	Bearbeitung des Arbeitsblattes zur booleschen Logik.	2
10 min	Einbinden des Kontaktschalters in die Programmierung mit Hilfe des *‘waitUntil‘* Befehls.	3
25 min	Selbstständiges Lösen einer Programmieraufgabe zur Verwendung des Kontaktschalters zur Richtungsänderung des Roboters.	1, 2, 3
5 min	Reflexion der Übung anhand offener Fragen.	1, 2, 3

Materialien

PowerPoint Übung 4
Arbeitsblatt “Boolesche Logik”
VEX IQ Super Kit
VEX IQ Wettbewerbsfeld und Würfelset
Computer mit RobotC Programmierumgebung

BY NC SA

Name(n): _____

Lehrer/Klasse: _____

Datum: _____

Boolesche Logik - Wahrheitswerte

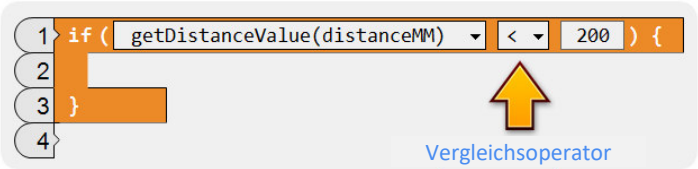
Roboter müssen ganz exakt wissen, welche Entscheidung sie unter bestimmten Bedingungen treffen müssen. Daher geht eine Entscheidung immer auf eine Frage zurück, die genau zwei Antwortmöglichkeiten hat: **ja** oder **nein** bzw. **wahr** oder **falsch**.

Aussagen, die nur wahr oder falsch sein können, nennt man boolesche Aussagen.
Die Zustand bzw. Wert wahr-oder-falsch dieser Aussagen nennt man Wahrheitswert.

Hier zwei Beispiele, wie ein Roboter auf eine boolesche Aussage antwortet:

Boolesche Aussage	Antwort
Der Himmel ist blau.	
Es schneit im Sommer.	

Boolesche Logik – Vergleichsoperatoren



Der Vergleich von zwei Werten ist ein wesentlicher Grundsatz im booleschen Entscheidungsprozess. Im nebenstehenden Beispiel wird der Messwert des Abstandssensors mit dem Wert 200 verglichen. Der Vergleichsoperator ist…

Aufgabe: **Ordne jedem Vergleichsoperator aus der Programmieroberfläche von ROBOTC eine der unten genannten Beschreibungen zu**

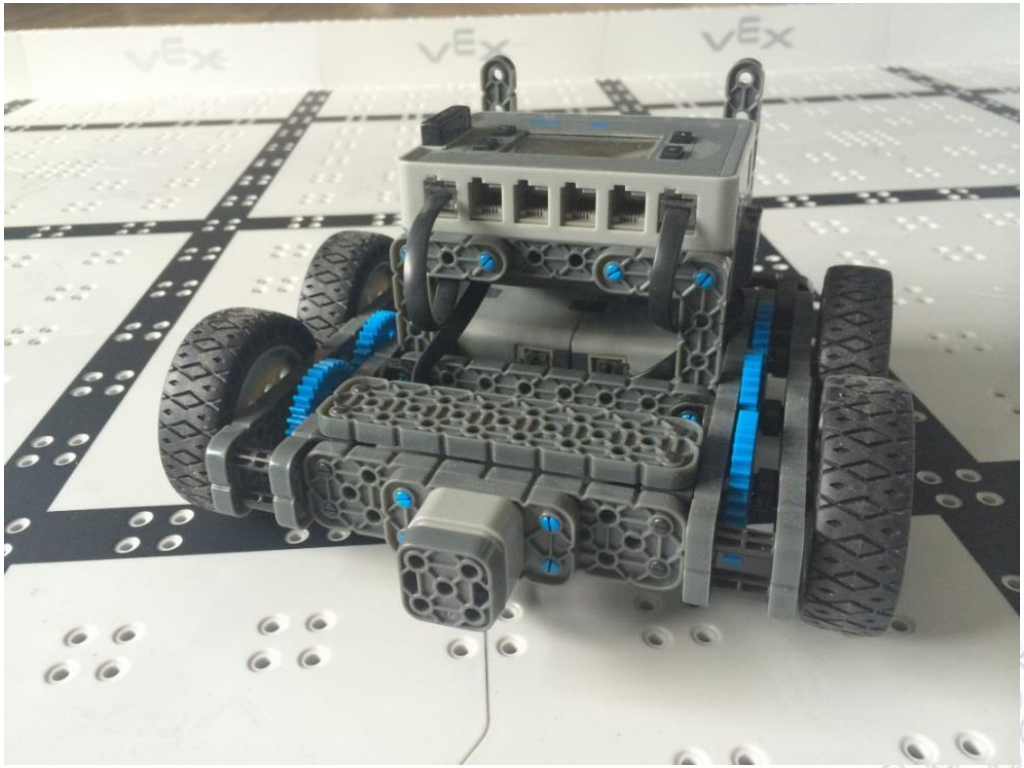
ROBOTC Symbol	Beschreibung	ROBOTC Symbol	Beschreibung
==		!=	
<=		>	
>=		<	

größer gleich **größer als** **gleich** **kleiner gleich** **kleiner als** **ungleich**

BY NC SA

Der Kontaktschalter

In dieser Übung wirst Du den Kontaktschalter kennenlernen und erfahren, wie man damit den Roboter steuern kann.



Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Funktionsweise des Kontaktschalters

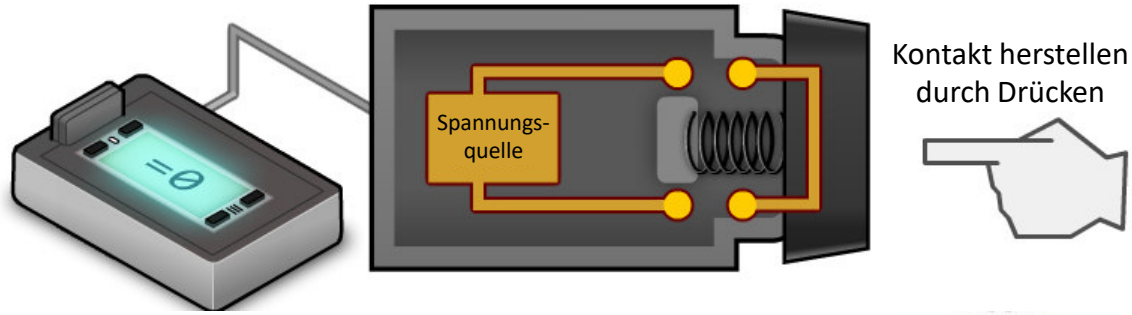
Lernziel

Die Schüler*innen können die Funktion eines Kontaktschalters erklären und erschließen sich den Nutzen zur Steuerung eines Roboters.

Durch Drücken des Kontaktschalters wird ein elektrischer Kreis geschlossen, wodurch ein elektrischer Strom fließt.

Beim Loslassen des Kontaktschalters wird durch Federkraft der elektrische Kreis und damit der Strom unterbrochen.

Das Robot Brain erkennt, ob ein elektrischer Strom fließt und kann damit feststellen, ob der Schalter geschlossen oder geöffnet ist.



Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Kontaktschalter einbauen

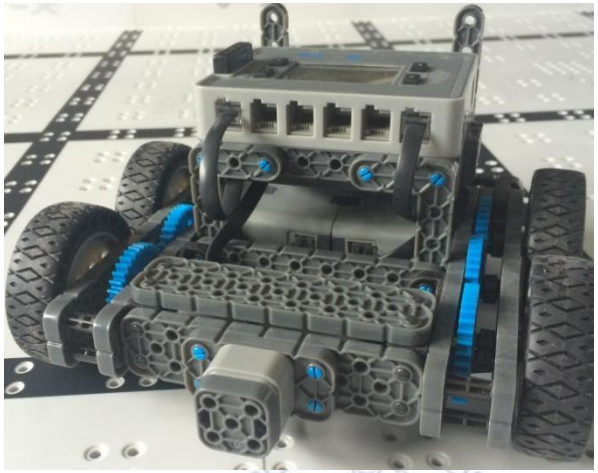
Lernziel

Die Schüler*innen können den Kontaktschalter selbstständig an den Fahrroboter anbauen, anschließen und die Konfiguration in der Programmierumgebung anpassen.

Aufgabe 1: Einbau des Kontaktschalters

Befestige den Kontaktschalter wie auf dem Bild rechts.

Verwende dazu vier 1x1 Stifte.



Verwende anschließend ein Sensorkabel in der passenden Länge, um den Kontaktschalter mit dem Robot Brain zu verbinden.

Schließe den Kontaktschalter an Port 8 des Robot Brains an.

Vergiss nicht, den Kontaktschalter in die Roboter-Konfiguration in ROBOTC einzutragen!

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

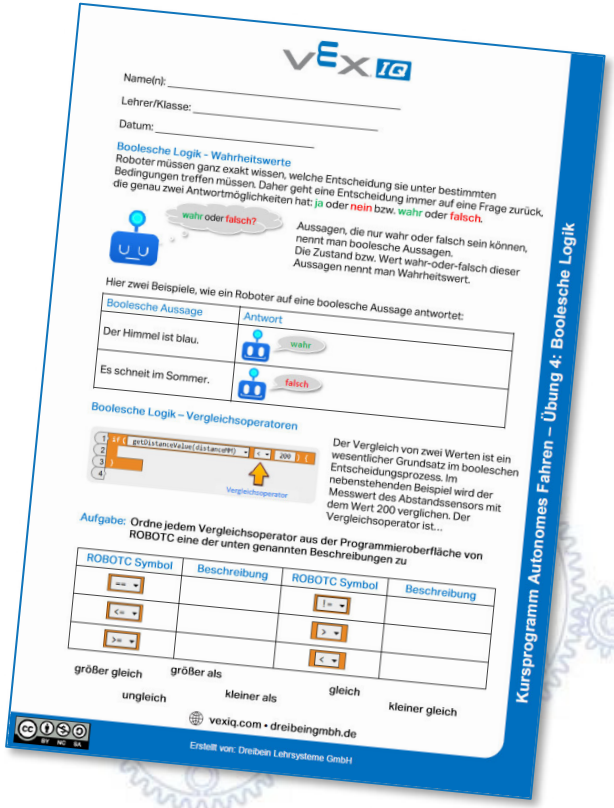
Arbeitsblatt

Lernziel

Die Schüler*innen können die Bedeutung von der Begriffe „Wahrheitswerte“ und „Vergleichsoperatoren“ in der Booleschen Logik erklären.

Aufgabe 2: **Ordne richtig zu!**

Ordne jedem Vergleichsoperator aus der Programmieroberfläche von ROBOTC eine der auf dem Arbeitsblatt genannten Beschreibungen zu.



Name(n): _____

Lehrer/Klasse: _____

Datum: _____

Boolesche Logik - Wahrheitswerte

Roboter müssen ganz exakt wissen, welche Entscheidung sie unter bestimmten Bedingungen treffen müssen. Daher geht eine Entscheidung immer auf eine Frage zurück, die genau zwei Antwortmöglichkeiten hat: ja oder nein bzw. wahr oder falsch.

wahr oder falsch?

Aussagen, die nur wahr oder falsch sein können, nennt man boolesche Aussagen. Die Zustand bzw. Wert wahr-oder-falsch dieser Aussagen nennt man Wahrheitswert.

Hier zwei Beispiele, wie ein Roboter auf eine boolesche Aussage antwortet:

Boolesche Aussage	Antwort
Der Himmel ist blau.	wahr
Es schneit im Sommer.	falsch

Boolesche Logik – Vergleichsoperatoren

```
1 if ( getDistanceValue(distanceMM) < 200 ) {
2
3 }
4
```

Vergleichsoperator

Der Vergleich von zwei Werten ist ein wesentlicher Grundsatz im boolescheen Entscheidungsprozess. Im nebenstehenden Beispiel wird der Messwert des Abstandssensors mit dem Wert 200 verglichen. Der Vergleichsoperator ist...

Aufgabe: Ordne jedem Vergleichsoperator aus der Programmieroberfläche von ROBOTC eine der unten genannten Beschreibungen zu

ROBOTC Symbol	Beschreibung	ROBOTC Symbol	Beschreibung
==		!=	
<=		>	
>=		<	

größer gleich, größer als, ungleich, kleiner als, gleich, kleiner gleich

vexiq.com • dreibeingmbh.de

Erstellt von: Dreibein Lehrsysteme GmbH

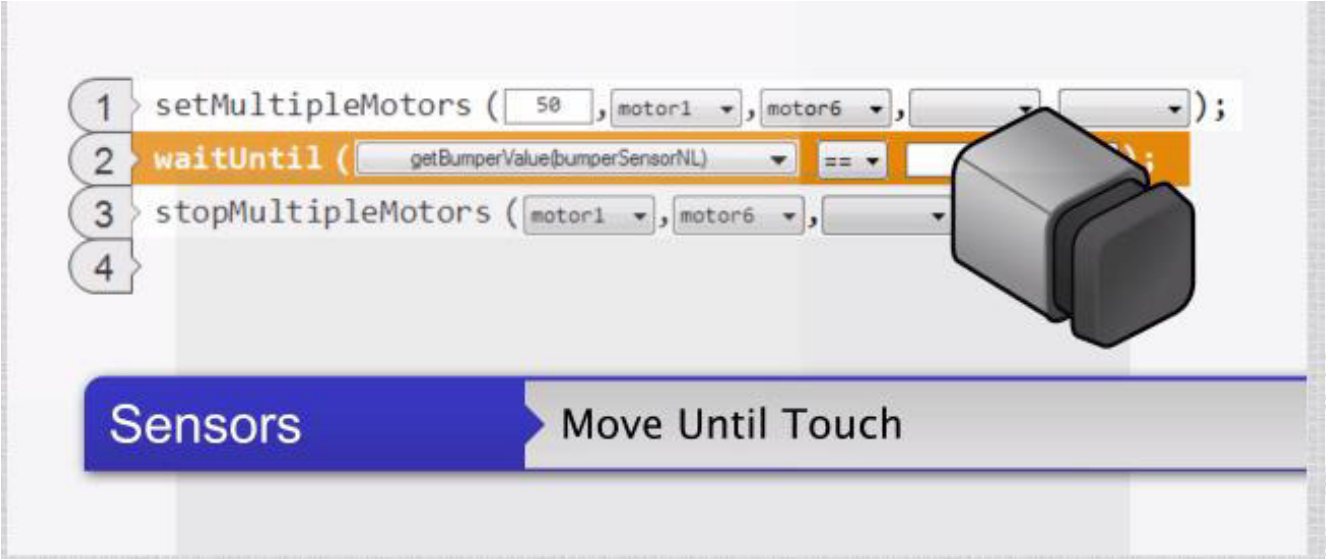
Kursprogramm Autonomes Fahren – Übung 4: Boolesche Logik

CC BY NC SA

Programmieren mit ROBOTC

Lernziel

Die Schüler*innen können den Kontaktschalter in ein selbsterstelltes Programm einbinden und damit die Bewegung des Roboters steuern.



Durch Anklicken der Abbildung wird ein englischsprachiges Video im Browser geöffnet
Link: http://www.education.rec.ri.cmu.edu/products/teaching_robotc_vexiq/lesson/media_videos/2_1-3_touch.mp4

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Stoppen oder Drehen nach Berührung

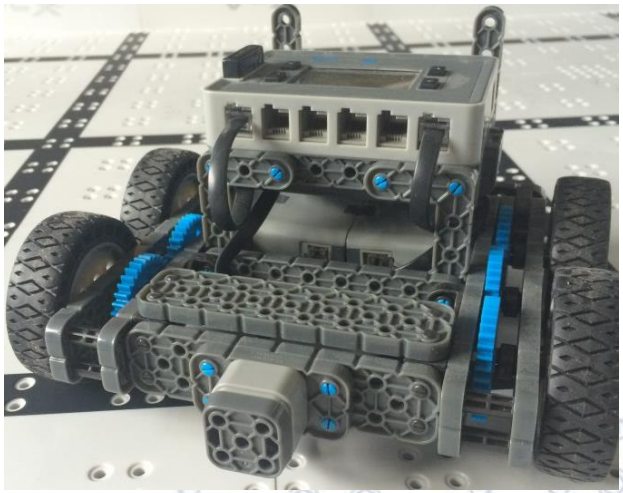
Lernziel

Die Schüler*innen können den Roboter nach einer Berührung automatisch stoppen oder die Fahrrichtung ändern lassen.

Aufgabe 3: ‘waitUntil’ Befehl zum Stoppen

```
1 setMultipleMotors ( 50 , motor1 , motor6 , , );
2 waitUntil ( getBumperValue(bumpSwitchLeft) == 1 );
3 stopMultipleMotors ( motor1 , motor6 , , );
```

Schreibe ein Programm, bei dem der Roboter stoppt, sobald er ein Hindernis berührt. Der Kontaktschalter ist robust genug, um den Roboter gegen die Begrenzung des Wettbewerbsfeldes oder gegen einen Holzblock fahren zu lassen.



Aufgabe 4: ‘waitUntil’ Befehl zur Richtungsänderung

```
1 setMultipleMotors ( 50 , motor1 , motor6 , , );
2 waitUntil ( getBumperValue(bumpSwitchLeft) == 1 );
3 backward ( .5 , rotations , 50 );
4 turnRight ( .7 , rotations , 50 );
```

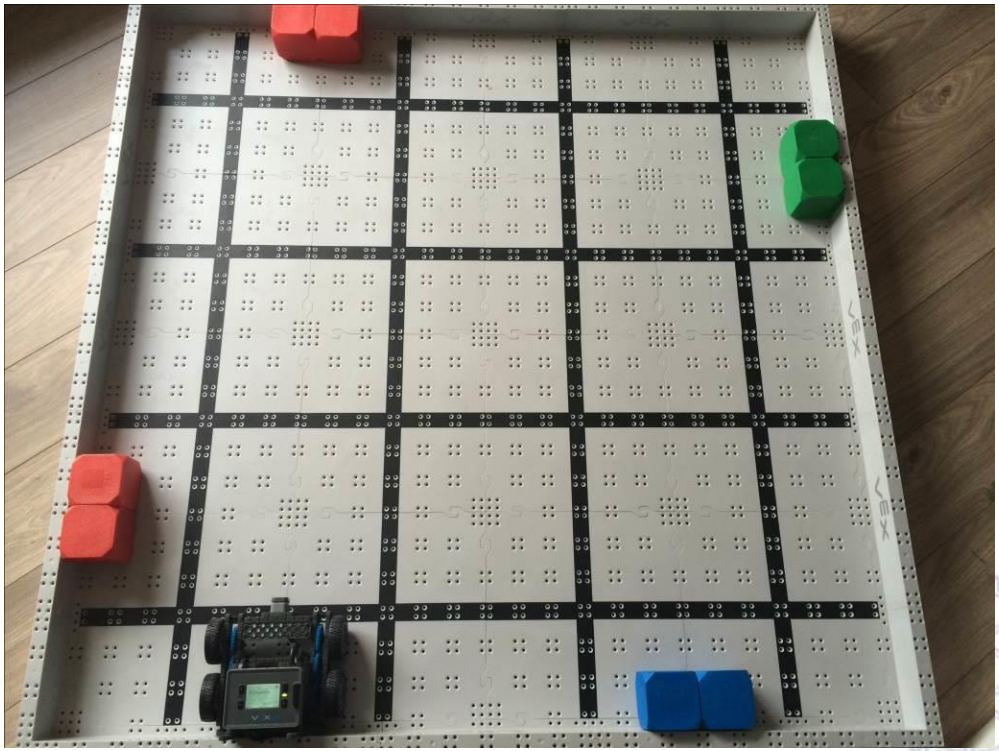
Schreibe ein Programm, bei dem der Roboter die Richtung ändert, sobald er ein Hindernis berührt.

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Berührungsfahrt

Aufgabe 5: Baue das Wettbewerbsfeld mit den farbigen Würfel wie abgebildet auf und schreibe mit den Dir bekannten Befehlen ein Programm, bei dem der Roboter jeden Würfelblock genau einmal berührt. Wer schafft den schnellsten Durchlauf?



Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Berührungsfahrt

Link zur Online-Version auf Youtube: https://youtu.be/MRkTQE0Mtf0

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Berührungsfahrt

Lösungsvorschlag

```
1  setMultipleMotors ( 50 , motor1 , motor6 , , , );
2  waitUntil ( getBumperValue(bumpSwitchLeft) == 1 );
3  backward ( .5 , rotations , 50 );
4  turnRight ( .69 , rotations , 50 );
5  setMultipleMotors ( 50 , motor1 , motor6 , , , );
6  waitUntil ( getBumperValue(bumpSwitchLeft) == 1 );
7  backward ( .5 , rotations , 50 );
8  turnRight ( .69 , rotations , 50 );
9  setMultipleMotors ( 50 , motor1 , motor6 , , , );
10 waitUntil ( getBumperValue(bumpSwitchLeft) == 1 );
11 backward ( .5 , rotations , 50 );
12 turnRight ( .69 , rotations , 50 );
13 setMultipleMotors ( 50 , motor1 , motor6 , , , );
14 waitUntil ( getBumperValue(bumpSwitchLeft) == 1 );
15 stopMultipleMotors ( motor1 , motor6 , , , );
```


Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Verständnisfragen

Lasst uns gemeinsam folgende Fragen beantworten:

1. Welche Art von Sensor ist der Kontaktschalter?

2. Welche Geräte oder Maschinen könnten einen Kontaktschalter verwenden?

3. Welche Probleme könnten auftreten, wenn nur ein Kontaktschalter verwendet wird?

4. Wie könnten die bei Frage 3 genannten Probleme gelöst werden?

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Zusammenfassung

Lernziel

Die Schüler*innen können:

- die Funktion eines Kontaktschalters erklären und erschließen sich den Nutzen zur Steuerung eines Roboters
- den Kontaktschalter selbstständig an den Fahrroboter anbauen, anschließen und die Konfiguration in der Programmierumgebung anpassen
- die Bedeutung von der Begriffe „Wahrheitswerte“ und „Vergleichsoperatoren“ in der Booleschen Logik erklären
- den Kontaktschalter in ein selbsterstelltes Programm einbinden und damit die Bewegung des Roboters steuern
- den Roboter nach einer Berührung automatisch stoppen oder die Fahrrichtung ändern lassen

Heute hast du:

- *gelernt, wie der Kontaktschalter funktioniert*
- *den Kontaktschalter in den Fahrroboter eingebaut*
- *den ‘waitUntil‘ Befehl bei der Programmierung des Roboters kennengelernt*
- *den Berührungsparcour absolviert*

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Übung 5

Verwendung der Touch LED

Übung 5 – Touch LED

Unterrichtsinhalte: In dieser Übung befassen sich die Schüler*innen mit den Funktionen der Touch LED und lernen in diesem Zusammenhang den Unterschied zwischen Sensoren und Aktoren. Durch den Einbau der Touch LED in den Fahrroboter und das Einbinden in ein selbstgeschriebenes Programm wird die mehrfarbige LED zur Richtungsanzeige verwendet.

Themen	Altersgruppe	Dauer
Sensoren und Aktoren, Sensortasten, LED	10-16	60 min

Lernziele

1. Die Schüler*innen können den Unterschied zwischen Sensoren und Aktoren erklären und die Komponenten des Bausatzes entsprechend zuordnen.
2. Die Schüler*innen können die Funktionsweise einer kapazitiven Sensortaste wiedergeben und beurteilen, welche Materialien sich zum Berühren eignen.
3. Die Schüler*innen können die Touch LED in den Fahrroboter einbauen und anschließen.
4. Die Schüler*innen können mit verschiedenen Befehlen die Farbe der LED vorgeben.
5. Die Schüler*innen können selbstständig ein Programm erstellen, in dem sich durch Berühren der Sensortaste die Farbe der LED ändert.

Dauer	Aktivitäten	Verbunden mit Lernziel
5 min	Einleitung	1, 2, 3, 4, 5
10 min	Funktionsweise und Verwendung Touch LED; Einbau in den Roboter.	1, 2, 3
10 min	Programmierung der LED Farben mit den Befehlen *'setTouchLEDHue'* und *'setTouchLEDRGB'*.	3, 4
10 min	Erstellen eines einfachen Programms mit Hilfe der Befehle *'waitUntil'* und *'setTouchLEDColor'*.	4, 5
20 min	Selbstständiges Lösen einer Programmieraufgabe zur Verwendung der Touch LED als Richtungsanzeige des Fahrroboters.	4, 5
5 min	Reflexion der Übung anhand offener Fragen.	1, 2, 3, 4, 5

Materialien

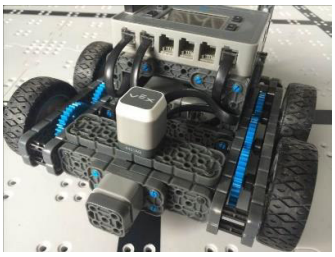
PowerPoint Übung 5
Arbeitsblatt "Verwendung der Touch LED"
VEX IQ Super Kit
VEX IQ Wettbewerbsfeld und Würfelset
Computer mit RobotC Programmierumgebung

BY NC SA

Name(n): _____

Lehrer/Klasse: _____

Datum: _____



Aufgabe 1: Einbau der Touch LED

Befestige die Touch LED wie auf dem Bild mit vier 1x1 Stiften.
Verwende anschließend ein Sensorkabel in der passenden Länge, um die Touch LED mit dem Robot Brain zu verbinden.
Schließe die Touch LED an Port 5 des Robot Brains an.

Vergiss nicht, die Touch-LED in die Roboter-Konfiguration in RobotC einzutragen!

Aufgabe 2: 'setTouchLEDHue' Befehl zur Farbsteuerung

Erstelle folgendes Programm in RobotC und übertrage es auf den Roboter:

```
1 setTouchLEDHue ( touchLED , 0 );
```

Verwende folgende Werte für die Farbtöne (engl. Hue) und trage die LED Farbe in die Tabelle ein.

Farbton	LED Farbe	Farbton	LED Farbe	Farbton	LED Farbe
0		127		255	
42		170		30	
85		212		230	

Aufgabe 3: 'setTouchLEDRGB' Befehl zur Farbsteuerung

Erstelle folgendes Programm in RobotC und übertrage es auf den Roboter:

```
1 setTouchLEDRGB ( touchLED , 0 , 0 , 0 );
```

Entnimm die RGB-Werte aus der folgenden Tabelle und trage die jeweilige Farbe der LED ein.

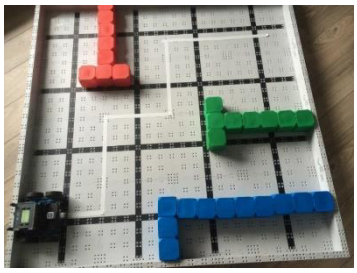
RGB-Wert	LED Farbe	RGB-Wert	LED Farbe	RGB-Wert	LED Farbe
(0,0,0)		(0,255,0)		(0,255,255)	
(255,255,255)		(0,0,255)		(255,0,255)	
(255,0,0)		(255,255,0)		(192,192,192)	

Aufgabe 4: Änderung der LED-Farbe durch Berühren der Sensortaste

Schreibe ein Programm, bei dem durch wiederholtes Berühren der Sensortaste nacheinander die LED in den Farben Gelb, Blau, Rot und Grün leuchtet.
Verwende dazu die Befehle 'setTouchLEDColor' und 'waitUntil'.

Aufgabe 5: Richtungsanzeige

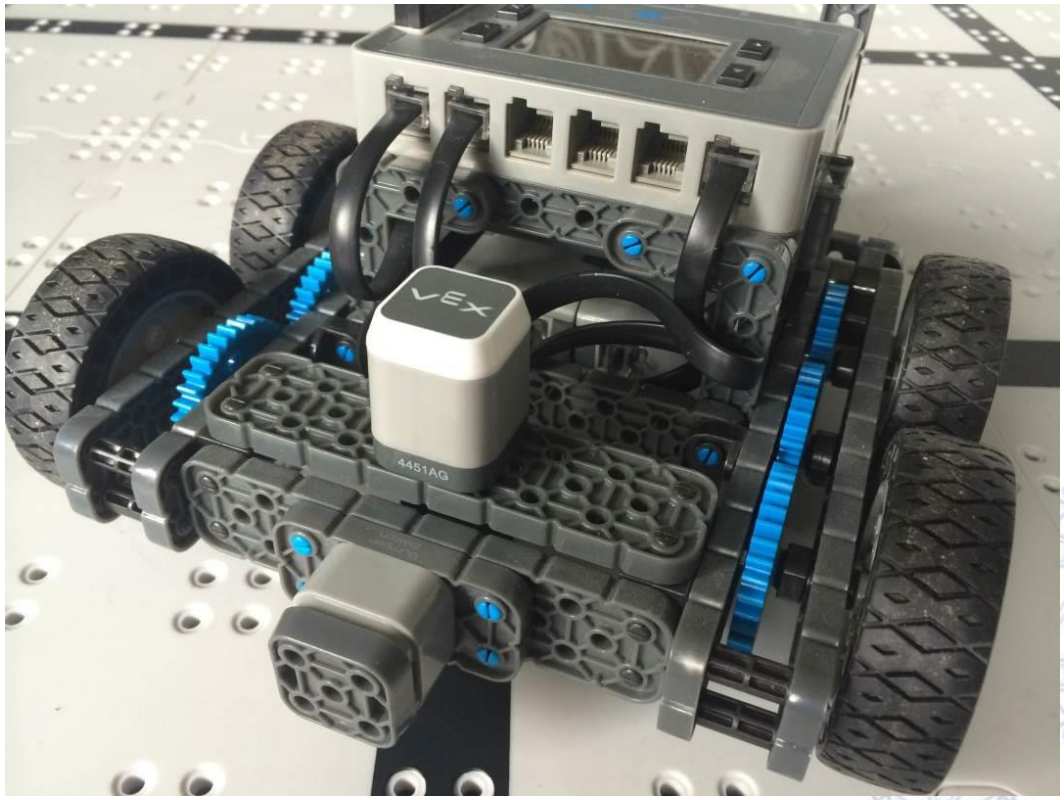
Baue das Wettbewerbsfeld mit den farbigen Würfel wie abgebildet auf und schreibe mit den Dir bekannten Befehlen ein Programm, bei dem der Roboter den weiß markierten Weg durch den Parcours fährt. Der Roboter soll jeweils nach dem Berühren der Würfelblöcke seine Richtung ändern. Durch unterschiedliche Farben der Touch LED soll ersichtlich sein, in welche Richtung sich der Roboter gerade bewegt (links, rechts, vorwärts, rückwärts).



BY NC SA

Die Touch LED

In dieser Übung wirst Du die Touch LED kennenlernen und erfahren, wie diese im Roboter eingesetzt werden kann.



Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Funktionen der Touch LED

Lernziel

Die Schüler*innen können den Unterschied zwischen Sensoren und Aktoren erklären und die Komponenten des Bausatzes entsprechend zuordnen.

Sensor und Aktor

Die Touch LED ist ein Sensor und Aktor zugleich:

Mit der kapazitiven Sensortaste kann der Roboter Informationen aus der Umwelt aufnehmen und ist damit ein Sensor.

Mit der integrierten mehrfarbigen LED kann der Roboter Informationen an die Umwelt ausgeben und ist damit ein Aktor.

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Funktionsweise der Sensortaste

Lernziel

Die Schüler*innen können die Funktionsweise einer kapazitiven Sensortaste wiedergeben und beurteilen, welche Materialien sich zum Berühren eignen.

Hast du schon mal beim Berühren einer anderen Person einen elektrischen Schlag bekommen?

Das war eine elektrostatische Entladung. Diese tritt auf, weil der menschliche Körper wie ein Kondensator Energie speichern kann. Diese Eigenschaft wird für Touchscreens bei Smartphones oder Tablets genutzt – genauso wie bei der VEX IQ Touch LED.

Die Sensortaste besitzt einen eingebauten Kondensator. Wenn der Sensor berührt wird, kommt ein zusätzlicher Kondensator (der Finger) parallel hinzu – die Gesamtkapazität ändert sich und das Robot Brain erkannt das.

Versuche, die Sensortaste mit anderen Materialien zu schalten – was funktioniert, was nicht?

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Touch LED einbauen

Lernziel

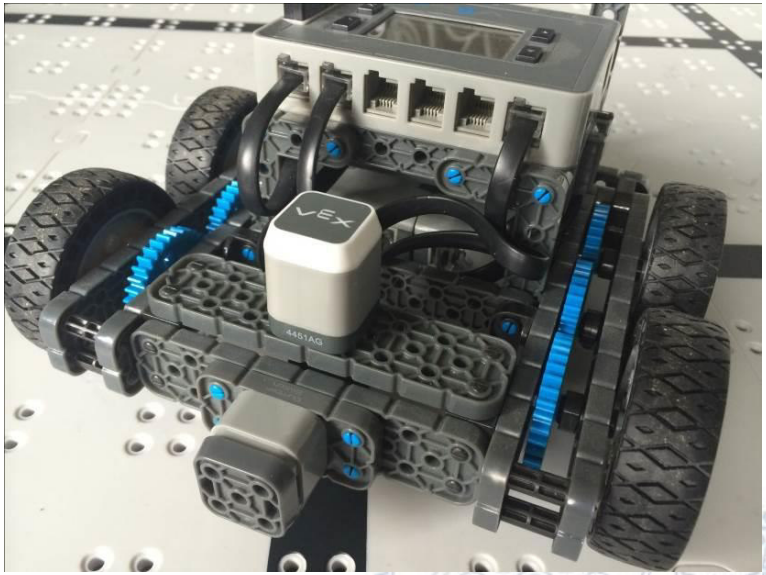
Die Schüler*innen können die Touch LED in den Fahrroboter einbauen und anschließen.

Aufgabe 1: Einbau der Touch LED

Befestige die Touch LED wie auf dem Bild mit vier 1x1 Stiften.

Verwende anschließend ein Sensorkabel in der passenden Länge, um die Touch LED mit dem Robot Brain zu verbinden.

Schließe die Touch LED an Port 5 des Robot Brains an.



Vergiss nicht, die Touch-LED in die Roboter-Konfiguration in RobotC einzutragen!

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Farbsteuerung der LED

Lernziel

Die Schüler*innen können mit verschiedenen Befehlen die Farbe der LED vorgeben.

Aufgabe 2: ‘setTouchLEDHue’ Befehl zur Farbsteuerung

```
1 setTouchLEDHue ( touchLED , 0 );
```

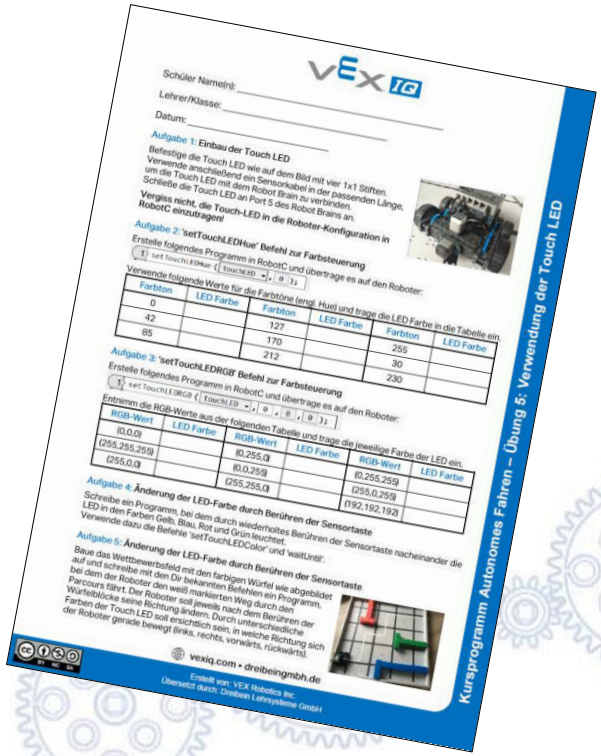
Schreibe ein Programm, wie oben abgebildet und verändere den Farbton (engl. Hue) entsprechend der Werte auf dem Arbeitsblatt.

Aufgabe 3: ‘setTouchLEDRGB’ Befehl zur Farbsteuerung

```
1 setTouchLEDRGB ( touchLED , 0 , 0 , 0 );
```

Schreibe ein Programm, wie oben abgebildet und verändere die RGB-Werte entsprechend der Tabelle auf dem Arbeitsblatt.

Was bedeutet die Abkürzung RGB?


Schüler Name(n): _____
Lehrer/Klasse: _____
Datum: _____

Aufgabe 1: Einbau der Touch LED

Befestige die Touch LED wie auf dem Bild mit vier 1x1 Stiften. Verwende anschließend ein Sensorkabel in der passenden Länge, um die Touch LED mit dem Robot Brain zu verbinden. Schließe die Touch LED an Port 5 des Robot Brains an.

Vergiss nicht, die Touch-LED in die Roboter-Konfiguration in RobotC einzutragen!

Aufgabe 2: 'setTouchLEDHue' Befehl zur Farbsteuerung

Erstelle folgendes Programm in RobotC und übertrage es auf den Roboter:

```
1 setTouchLEDHue ( touchLED , 0 );
```

Verwende folgende Werte für die Farbtöne (engl. Hue) und trage die LED Farbe in die Tabelle ein.

Farbton	LED Farbe	Farbton	LED Farbe	Farbton	LED Farbe
0		127		255	
42		170		30	
85		212		230	

Aufgabe 3: 'setTouchLEDRGB' Befehl zur Farbsteuerung

Erstelle folgendes Programm in RobotC und übertrage es auf den Roboter:

```
1 setTouchLEDRGB ( touchLED , 0 , 0 , 0 );
```

Entnimm die RGB-Werte aus der folgenden Tabelle und trage die jeweilige Farbe der LED ein.

RGB-Wert	LED Farbe	RGB-Wert	LED Farbe	RGB-Wert	LED Farbe
(0,0,0)		(0,255,0)		(0,255,255)	
(255,255,255)		(0,0,255)		(255,0,255)	
(255,0,0)		(255,255,0)		(192,192,192)	

Aufgabe 4: Änderung der LED-Farbe durch Berühren der Sensortaste

Schreibe ein Programm, bei dem durch wiederholtes Berühren der Sensortaste nacheinander die LED in den Farben Gelb, Blau, Rot und Grün leuchtet.
Verwende dazu die Befehle 'setTouchLEDColor' und 'waitUntil'.

Aufgabe 5: Änderung der LED-Farbe durch Berühren der Sensortaste

Baue das Wettbewerbsfeld mit den farbigen Würfel wie abgebildet auf und schreibe mit den Dir bekannten Befehlen ein Programm, bei dem der Roboter den weiß markierten Befehlen ein Programm. Parcours fährt. Der Roboter soll jeweils nach dem Berühren der Würfelblöcke seine Richtung ändern. Durch unterschiedliche Farben der Touch LED soll ersichtlich sein, in welche Richtung sich der Roboter gerade bewegt (links, rechts, vorwärts, rückwärts).

vexiq.com • dreibeingmbh.de

Erstellt von: VEX Robotics Inc.
Übersetzt durch: Dreibein Lehrsysteme GmbH

Kursprogramm Autonomes Fahren – Übung 5: Verwendung der Touch LED

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Farbsteuerung der LED

Lernziel

Die Schüler*innen können selbstständig ein Programm erstellen, in dem sich durch Berühren der Sensortaste die Farbe der LED ändert.

Aufgabe 4: Änderung der LED-Farbe durch Berühren der Sensortaste

Schreibe ein Programm, bei dem durch wiederholtes Berühren der Sensortaste nacheinander die LED in den Farben Gelb, Blau, Rot und Grün leuchtet.
Verwende dazu die Befehle ‘setTouchLEDColor‘ und ‘waitUntil‘.

```
1  waitUntil ( getTouchLEDValue(touchLED) == 1 );
2  setTouchLEDColor ( touchLED , colorYellow );
3  waitUntil ( getTouchLEDValue(touchLED) == 0 );
4  waitUntil ( getTouchLEDValue(touchLED) == 1 );
5  setTouchLEDColor ( touchLED , colorBlue );
6  waitUntil ( getTouchLEDValue(touchLED) == 0 );
7  waitUntil ( getTouchLEDValue(touchLED) == 1 );
8  setTouchLEDColor ( touchLED , colorRed );
9  waitUntil ( getTouchLEDValue(touchLED) == 0 );
10 waitUntil ( getTouchLEDValue(touchLED) == 1 );
11 setTouchLEDColor ( touchLED , colorGreen );
```

Was passiert, wenn der Befehl ‘waitUntil(getTouchLEDValue(touchLED)=0)‘ in den Zeilen 3, 6 und 9 weggelassen wird?

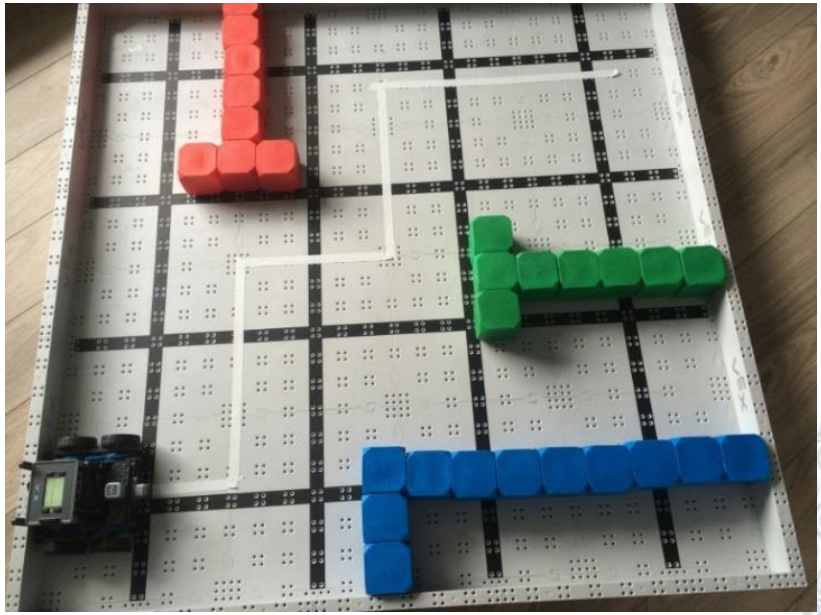
Wieso wird dieser Befehl bei Verwendung des Kontaktschalters nicht benötigt?

Nutze deine Erfahrungen und die verwendeten Befehle aus dieser Programmierübung zur Lösung der nächsten Aufgabe.

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Richtungsanzeige

Aufgabe 5: Baue das Wettbewerbsfeld mit den farbigen Würfel wie abgebildet auf und schreibe mit den Dir bekannten Befehlen ein Programm, bei dem der Roboter den weiß markierten Weg durch den Parcours fährt. Der Roboter soll jeweils nach dem Berühren der Würfelblöcke seine Richtung ändern. Durch unterschiedliche Farben der Touch LED soll ersichtlich sein, in welche Richtung sich der Roboter gerade bewegt (links, rechts, vorwärts, rückwärts).



Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Richtungsanzeige

Link zur Online-Version auf Youtube: https://youtu.be/65VaZSLaGI8

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Richtungsanzeige

Lösungsvorschlag

```
1  setMultipleMotors ( 50 , motor1 , motor6 , , );
2  setTouchLEDColor ( port5 , colorGreen );
3  waitUntil ( getBumperValue(bumpSwitchLeft) == 1 );
4  setTouchLEDColor ( port5 , colorRed );
5  backward ( .5 , rotations , 50 );
6  setTouchLEDColor ( port5 , colorBlue );
7  turnLeft ( .69 , rotations , 50 );
8  setTouchLEDColor ( port5 , colorGreen );
9  setMultipleMotors ( 50 , motor1 , motor6 , , );
10 waitUntil ( getBumperValue(bumpSwitchLeft) == 1 );
11 setTouchLEDColor ( port5 , colorRed );
12 backward ( .5 , rotations , 50 );
13 setTouchLEDColor ( port5 , colorYellow );
14 turnRight ( .69 , rotations , 50 );
15 setTouchLEDColor ( port5 , colorGreen );
16 setMultipleMotors ( 50 , motor1 , motor6 , , );
17 waitUntil ( getBumperValue(bumpSwitchLeft) == 1 );
18 setTouchLEDColor ( port5 , colorRed );
19 backward ( .5 , rotations , 50 );
20 setTouchLEDColor ( port5 , colorBlue );
```

```
21 turnLeft ( .69 , rotations , 50 );
22 setTouchLEDColor ( port5 , colorGreen );
23 setMultipleMotors ( 50 , motor1 , motor6 , , );
24 waitUntil ( getBumperValue(bumpSwitchLeft) == 1 );
25 setTouchLEDColor ( port5 , colorRed );
26 backward ( .5 , rotations , 50 );
27 setTouchLEDColor ( port5 , colorYellow );
28 turnRight ( .69 , rotations , 50 );
29 setTouchLEDColor ( port5 , colorGreen );
30 setMultipleMotors ( 50 , motor1 , motor6 , , );
31 waitUntil ( getBumperValue(bumpSwitchLeft) == 1 );
32 setTouchLEDColor ( port5 , colorOrange );
33 stopMultipleMotors ( motor1 , motor6 , , );
```


Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Verständnisfragen

Lasst uns gemeinsam folgende Fragen beantworten:

1. Welche Funktionen besitzt die Touch LED und zu welcher Gruppe von Bauteilen lässt sie sich zuordnen?

2. In welchen Geräten oder Bauteilen könnten LEDs verwendet werden?

3. Wie könnte ein Roboter außerdem eine Bewegung oder Richtungsänderung signalisieren?

4. Wie könnte der Roboter ohne Berührung ein Hindernis erkennen und seine Richtung ändern?

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Zusammenfassung

Lernziel

Die Schüler*innen können:

- den Unterschied zwischen Sensoren und Aktoren erklären und die Komponenten des Bausatzes entsprechend zuordnen
- die Funktionsweise einer kapazitiven Sensortaste wiedergeben und beurteilen, welche Materialien sich zum Berühren eignen
- die Touch LED in den Fahrroboter einbauen und anschließen
- mit verschiedenen Befehlen die Farbe der LED vorgeben
- selbstständig ein Programm erstellen, in dem sich durch Berühren der Sensortaste die Farbe der LED ändert

Heute hast du:

- *den Unterschied zwischen Sensoren und Aktoren kennengelernt*
- *gelernt, welche Funktionen die Touch LED hat*
- *die Touch LED in den Fahrroboter eingebaut*
- *die Befehle ‘setTouchLEDColor‘, ‘waitUntil‘ und ‘multipleMotors‘ zur Programmierung des Roboters verwendet*

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Übung 6

Verwendung des Abstandssensors

Übung 6 – Abstandssensor

Unterrichtsinhalte: In dieser Übung erlernen die Schüler*innen die Funktionsweise des Abstandssensors und befassen sich in diesem Zusammenhang mit möglichen Abweichen von Soll zu Istwerten. Durch den Einbau des Abstandssensors in den Fahrroboter und das Einbinden in ein selbstgeschriebenes Programm wird eine berührungsfreie Fahrt durch ein Labyrinth erreicht.

Themen	Altersgruppe	Dauer
Schall, Schallwellen, Schallreflexion, Schallabsorption, Soll- und Istwerte, Messtoleranzen	10-16	60 min

Lernziele

1. Die Schüler*innen können die Funktionsweise eines Ultraschall-Abstandssensors erklären und beurteilen, welche Gegenstände zur Abstandsmessung geeignet sind.
2. Die Schüler*innen können den Abstandssensor in den Fahrroboter einbauen und anschließen.
3. Die Schüler*innen können den Abstandssensor in ein selbst erstelltes Programm einbinden und durch Auslesen des Abstandsmesswertes die Bewegung des Roboters beeinflussen.
4. Die Schüler*innen können die Abweichungen zwischen Soll- und Istwerten in einem Steuerungssystem analysieren und die Erkenntnisse in die Programmierung implementieren.

Dauer	Aktivitäten	Verbunden mit Lernziel
5 min	Einleitung	1, 2, 3, 4
15 min	Funktionsweise und Verwendung Abstandssensor; Einbau in den Roboter	1, 2, 3
15 min	Erstellen eines einfachen Programms zum experimentellen Analysieren von Soll- und Istwerten und von Einflussgrößen auf den Abstandssensor.	3, 4
20 min	Selbstständiges Lösen einer Programmieraufgabe zur Verwendung des Abstandssensors zum berührungsfreien Umfahren von Hindernissen.	3, 4
5 min	Reflexion der Übung anhand offener Fragen.	1, 2, 3, 4

Materialien

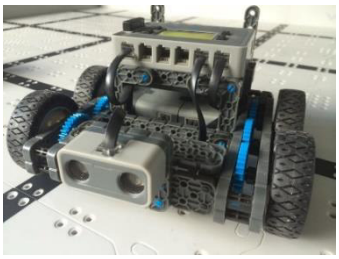
PowerPoint Übung 6
Arbeitsblatt “Verwendung des Abstandssensors”
VEX IQ Super Kit
VEX IQ Wettbewerbsfeld und Würfelset
Computer mit RobotC Programmierumgebung

BY NC SA

Name(n): _____

Lehrer/Klasse: _____

Datum: _____



Aufgabe 1: Einbau des Abstandssensors

Befestige den Abstandssensor wie auf dem Bild gezeigt. Der Aufbau entspricht den Schritten 110 und 111 in der Montageanleitung.

Verwende anschließend ein Sensorkabel in der passenden Länge, um den Abstandssensor mit Port 2 des Robot Brains zu verbinden.

Vergiss nicht, den Abstandssensor in die Roboter-Konfiguration in RobotC einzutragen!

Aufgabe 2: Abstand: Sollwert versus Istwert

1. Erstelle folgendes Programm in RobotC und übertrage es auf den Roboter:

```
1 setMultipleMotors ( 50 , motor1 , motor6 , , );
2 waitUntil ( getDistanceValue(distanceMM) < 150 );
3 stopMultipleMotors ( motor1 , motor6 , , );
```

2. Lass den Roboter dreimal auf einer ebenen Oberfläche auf einen Gegenstand oder eine Wand zufahren und miss jeweils den Abstand zum Gegenstand bzw. zur Wand. Trage das Ergebnis in die Tabelle unten ein und berechne den Durchschnitt.
3. Wiederhole Schritte 1 und 2 jeweils mit den Sollwerten 25 cm und 35 cm.

Sollwert	Istwert 1	Istwert 2	Istwert 3	Durchschnitt
15 cm				
25 cm				
35 cm				

Aufgabe 3: Einfluss der Motorleistung

1. Erstelle folgendes Programm in RobotC und übertrage es auf den Roboter:

```
1 setMultipleMotors ( 75 , motor1 , motor6 , , );
2 waitUntil ( getDistanceValue(distanceMM) < 150 );
3 stopMultipleMotors ( motor1 , motor6 , , );
```

2. Lass den Roboter dreimal auf einer ebenen Oberfläche auf einen Gegenstand oder eine Wand zufahren und miss jeweils den Abstand zum Gegenstand bzw. zur Wand. Trage das Ergebnis in die Tabelle unten ein und berechne den Durchschnitt.
3. Wiederhole Schritte 1 und 2 jeweils mit den Motorleistungen 50 und 25.

Motorleistung	Abstand 1	Abstand 2	Abstand 3	Durchschnitt
75				
50				
25				

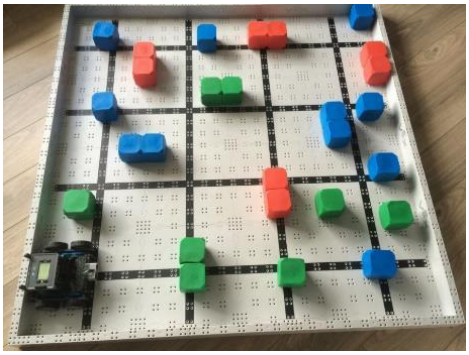
BY NC SA

Was sind Deine Schlussfolgerungen aus den Aufgaben 2 und 3?

Aufgabe 4: Marsmission

Baue das Wettbewerbsfeld mit den farbigen Würfeln wie abgebildet auf – die Würfel sind Felsbrocken auf dem Mars und müssen umfahren werden!

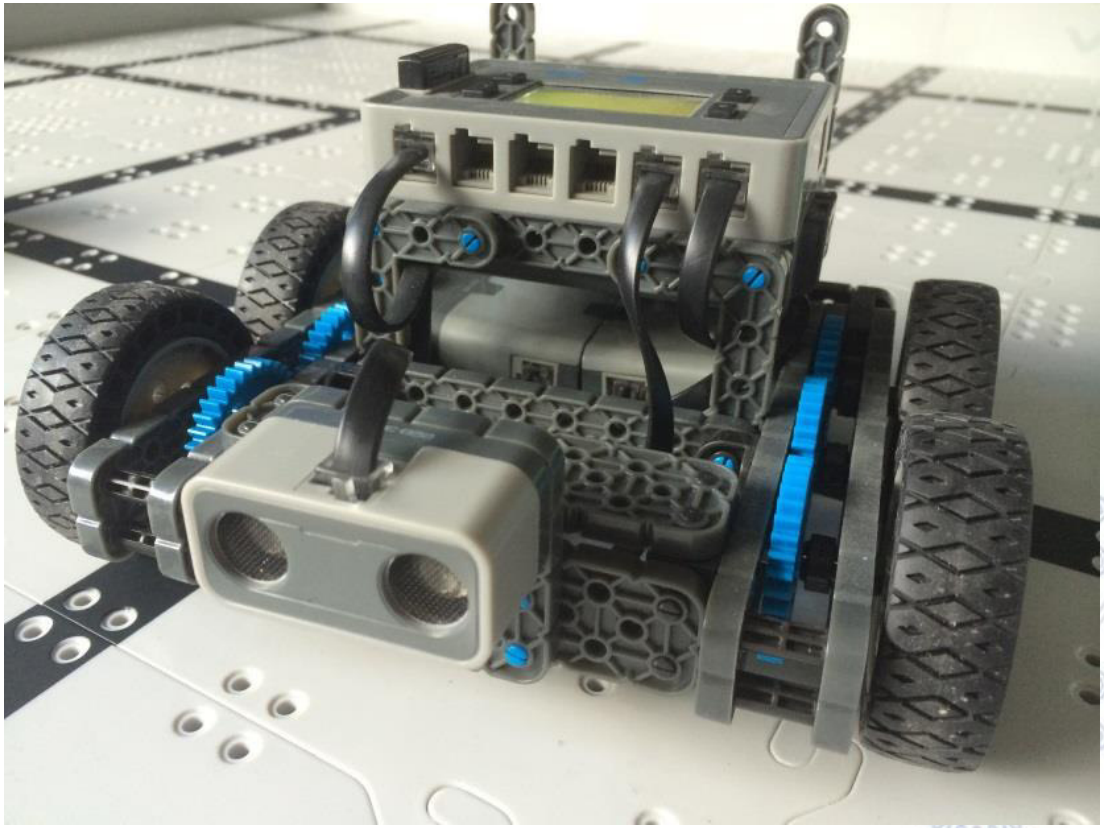
Schreibe mit den Dir bekannten Befehlen ein Programm, bei dem sich der Roboter durch den Parcours bewegt, ohne einen Felsbrocken zu berühren. Der Fahrroboter soll mindestens einmal diagonal über das Feld fahren.



BY NC SA

Der Abstandssensor

In dieser Übung wirst Du den Abstandssensor kennenlernen und erfahren, wie dieser im Roboter eingesetzt werden kann.



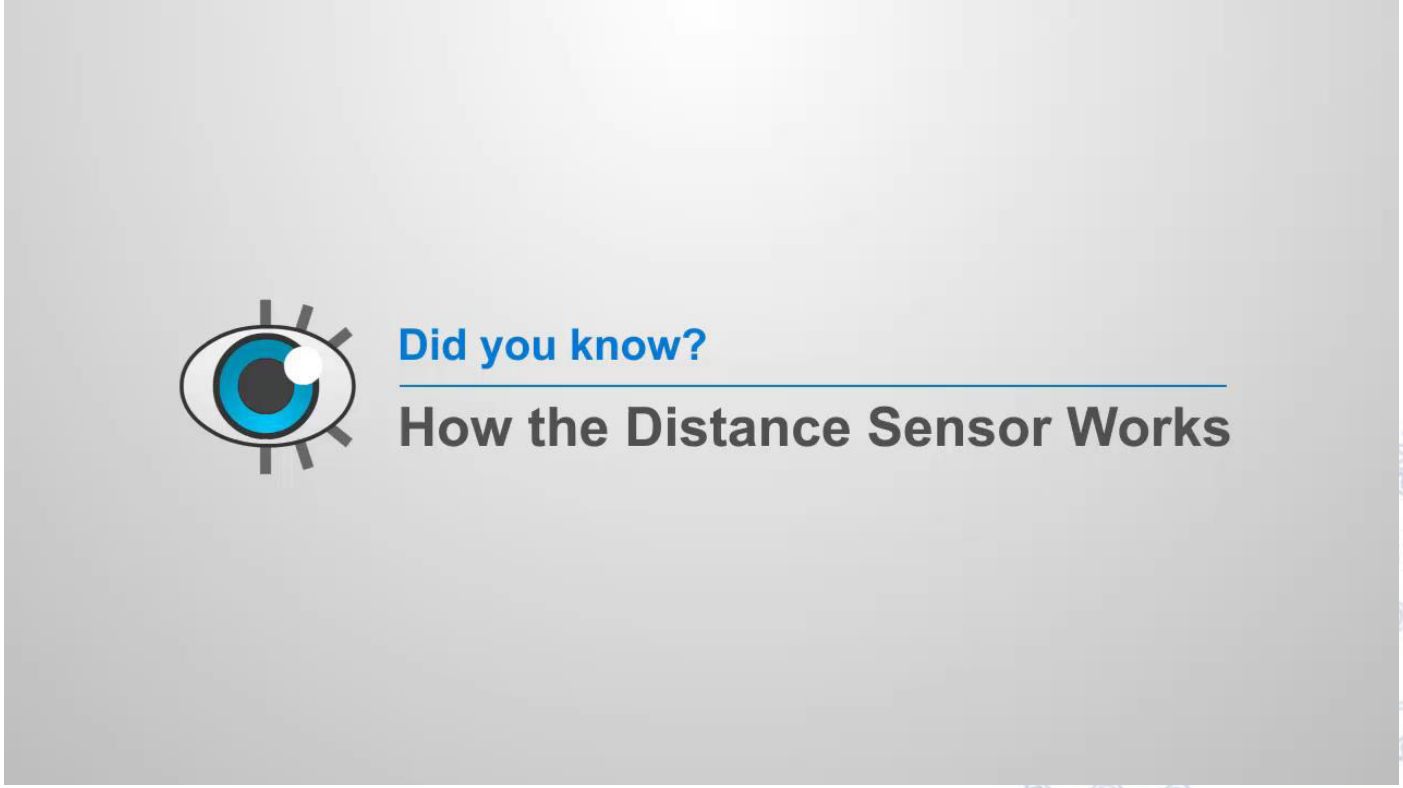
Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Funktionsweise des Abstandssensors

Lernziel

Die Schüler*innen können die Funktionsweise eines Ultraschall-Abstandssensors erklären und beurteilen, welche Gegenstände zur Abstandsmessung geeignet sind.



Link zur Online-Version: http://cmra.rec.ri.cmu.edu/products/teaching_robotc_vexiq/lesson/media_videos/howItWorks_distance.mp4

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Abstandssensor einbauen

Lernziel

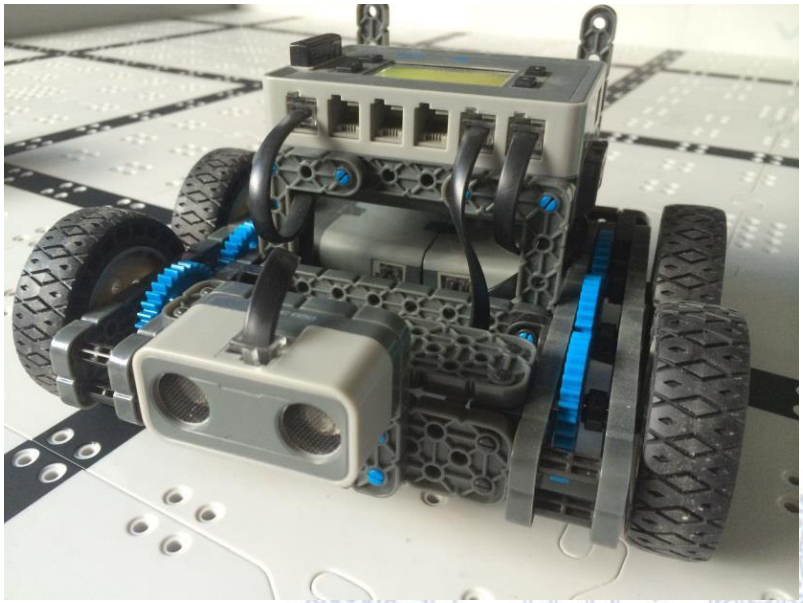
Die Schüler*innen können den Abstandssensor in den Fahrroboter einbauen und anschließen.

Aufgabe 1: Einbau des Abstandssensors

Befestige den Abstandssensor wie auf dem Bild gezeigt. Der Aufbau entspricht den Schritten 110 und 111 in der Montageanleitung.

Verwende anschließend ein Sensoranschluss-kabel in der passenden Länge, um den Abstandssensor mit dem Robot Brain zu verbinden.

Schließe den Abstandssensor an Port 2 des Robot Brains an.



Vergiss nicht, den Abstandssensor in die Roboter-Konfiguration in RobotC einzutragen!

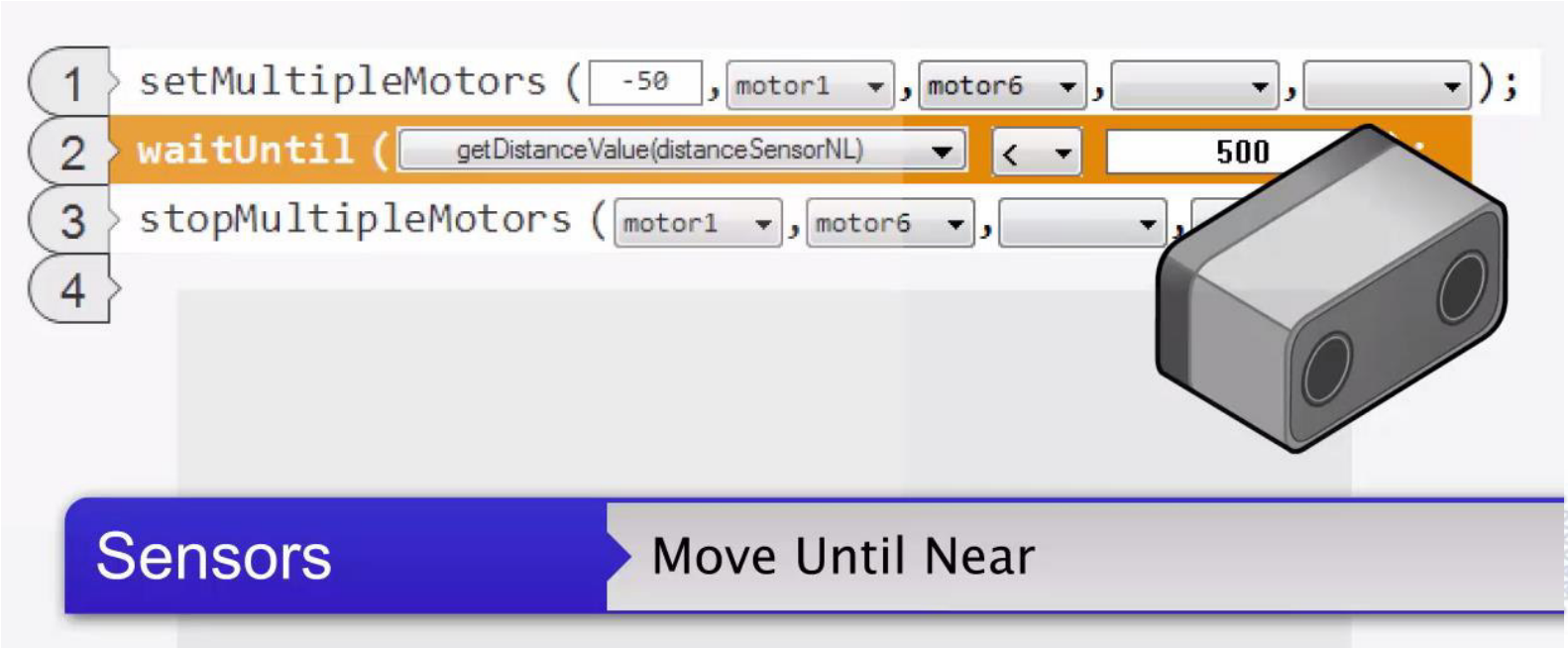
Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Programmieren mit ROBOTC

Lernziel

Die Schüler*innen können den Abstandssensor in ein selbst erstelltes Programm einbinden und durch Auslesen des Abstandsmesswertes die Bewegung des Roboters beeinflussen.



Durch Anklicken der Abbildung wird ein englischsprachiges Video im Browser geöffnet
Link: http://www.education.rec.ri.cmu.edu/products/teaching_robotc_vexiq/lesson/media_videos/2_2-3_ultrasonic.mp4

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

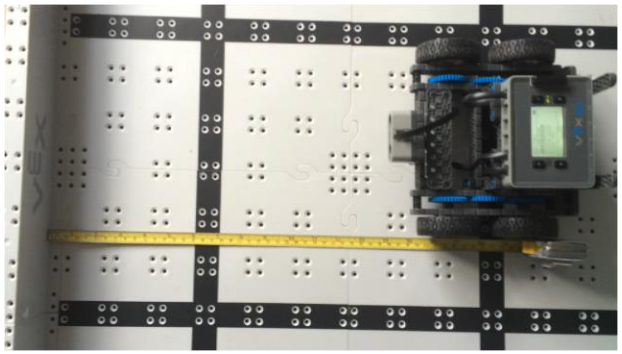
Abstandssensor testen

Lernziel

Die Schüler*innen können die Abweichungen zwischen Soll- und Istwerten in einem Steuerungssystem analysieren und die Erkenntnisse in die Programmierung implementieren.

Aufgabe 2: Abstand: Sollwert versus Istwert

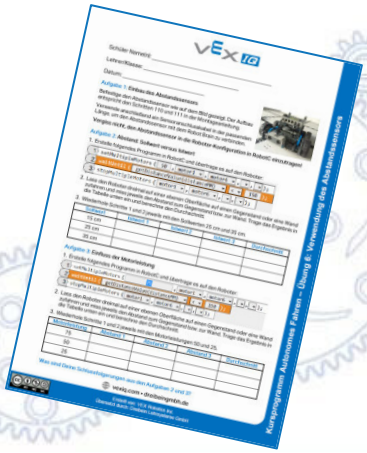
```
1 setMultipleMotors ( 50 , motor1 , motor6 , , );
2 waitUntil ( getDistanceValue(distanceMM) < 150 );
3 stopMultipleMotors ( motor1 , motor6 , , );
```



Variiere den Sollwert des Abstands in der Programmierung und bestimme die Istwerte wie auf dem Arbeitsblatt beschrieben.

Aufgabe 3: Einfluss der Motorleistung

```
1 setMultipleMotors ( 75 , motor1 , motor6 , , );
2 waitUntil ( getDistanceValue(distanceMM) < 150 );
3 stopMultipleMotors ( motor1 , motor6 , , );
```



Variiere den Sollwert der Motorleistung in der Programmierung und bestimme den Abstand wie auf dem Arbeitsblatt beschrieben.

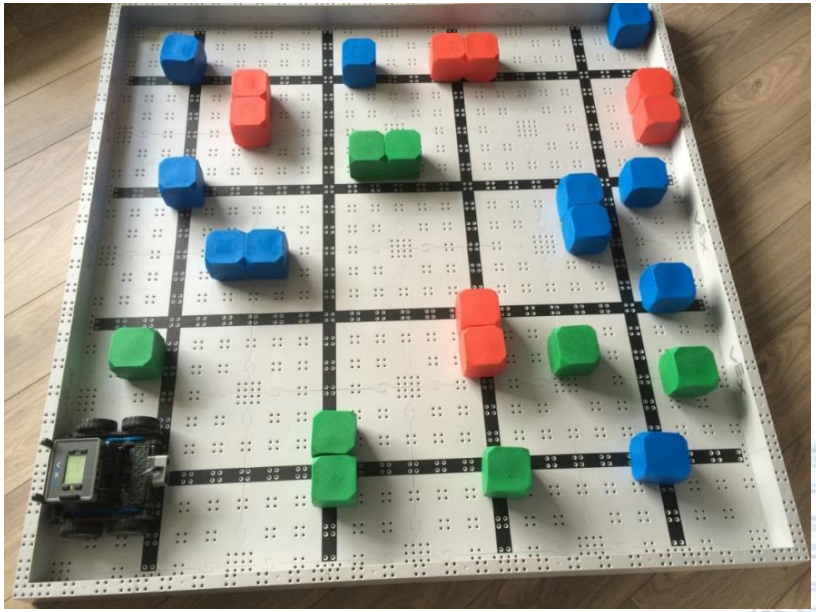
Was sind Deine Schlussfolgerungen aus den Aufgaben 2 und 3?

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Marsmission

Aufgabe 4: Baue das Wettbewerbsfeld mit den farbigen Würfeln wie abgebildet auf – die Würfel sind Felsbrocken auf dem Mars und müssen umfahren werden!
Schreibe mit den Dir bekannten Befehlen ein Programm, bei dem sich der Roboter durch den Parcours bewegt, ohne einen Felsbrocken zu berühren. Der Fahrroboter soll mindestens einmal diagonal über das Feld fahren.



Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Marsmission

Link zur Online-Version auf Youtube: https://youtu.be/clUXipYmCXM

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Marsmission

Lösungsvorschlag

```
1  setMultipleMotors ( 50 , motor1 , motor6 , , , );
2  waitUntil ( getDistanceValue(distanceMM) < 75 );
3  turnLeft ( .69 , rotations , 50 );
4  setMultipleMotors ( 50 , motor1 , motor6 , , , );
5  waitUntil ( getDistanceValue(distanceMM) < 75 );
6  turnRight ( .69 , rotations , 50 );
7  setMultipleMotors ( 50 , motor1 , motor6 , , , );
8  waitUntil ( getDistanceValue(distanceMM) < 75 );
9  turnLeft ( .69 , rotations , 50 );
10 setMultipleMotors ( 50 , motor1 , motor6 , , , );
11 waitUntil ( getDistanceValue(distanceMM) < 75 );
12 turnRight ( .69 , rotations , 50 );
13 setMultipleMotors ( 50 , motor1 , motor6 , , , );
14 waitUntil ( getDistanceValue(distanceMM) < 75 );
15 turnLeft ( .69 , rotations , 50 );
16 setMultipleMotors ( 50 , motor1 , motor6 , , , );
17 waitUntil ( getDistanceValue(distanceMM) < 75 );
18 turnRight ( .69 , rotations , 50 );
19 setMultipleMotors ( 50 , motor1 , motor6 , , , );
20 waitUntil ( getDistanceValue(distanceMM) < 75 );
21 turnLeft ( .69 , rotations , 50 );
22 setMultipleMotors ( 50 , motor1 , motor6 , , , );
23 waitUntil ( getDistanceValue(distanceMM) < 75 );
24 stopMultipleMotors ( motor1 , motor6 , , , );
25
```


Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Verständnisfragen

Lasst uns gemeinsam folgende Fragen beantworten:

1. Wie funktioniert der Abstandssensor?

2. Hatte der Vorgabewert der Motorleistung eine Auswirkung auf den Abstand zum Hindernis beim Anhalten?

3. Hat der Abstandssensor den Fahrroboter immer mit dem selben Abstand zum stehen gebracht?

4. Wo könnte der Abstandssensor ähnlich wie in der Marsmission aus Aufgabe 4 in unserer Umgebung auf der Erde verwendet werden?

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Zusammenfassung

Lernziel

Die Schüler*innen können:

- die Funktionsweise eines Ultraschall-Abstandssensors erklären und beurteilen, welche Gegenstände zur Abstandsmessung geeignet sind
- den Abstandssensor in den Fahrroboter einbauen und anschließen
- den Abstandssensor in ein selbst erstelltes Programm einbinden und durch Auslesen des Abstandsmesswertes die Bewegung des Roboters beeinflussen
- die Abweichungen zwischen Soll- und Istwerten in einem Steuerungssystem analysieren und die Erkenntnisse in die Programmierung implementieren

Heute hast du:

- *gelernt, wie ein Ultraschall-Abstandssensor funktioniert*
- *die Abweichungen von Soll- und Istwerten in einem Steuerungssystem analysiert*
- *festgestellt, welchen Einfluss die Motorleistung auf die Verwendung des Abstandssensors hat*
- *den Fahrroboter mit dem Befehl ‘getDistanceValue‘ programmiert*

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Übung 7

Verwendung des Farbsensors

Übung 7 – Farbsensor

Unterrichtsinhalte: In dieser Übung lernen die Schüler*innen die Funktionen des VEX IQ Farbsensor-Moduls kennen. Dazu wird die Funktionsweise der drei integrierten Sensoren (RGB-Farbsensor, Graustufen-Sensor, Infrarot-Näherungssensor) in Zusammenhang mit den verschiedenen Mess-Modi erläutert. In zwei Aufgaben werden die verschiedenen Einflussgrößen auf die Farbmessung experimentell untersucht und ausgewertet. Als abschließende Aufgabe soll der Farbsensor genutzt werden, um den Fahrroboter automatisch in eine farbige Garage einparken zu lassen.

Themen	Altersgruppe	Dauer
Farbsensor, Graustufensensor, Infrarot-Näherungs-sensor, RGB-Skala, Lichtfarbe, Farbtemperatur	10-16	60 min

Lernziele

1. Die Schüler*innen können die Funktionsweise der integrierten Sensoren (RGB-Farbsensor, Graustufensensor, Infrarot-Näherungssensor) wiedergeben und differenzieren.
2. Die Schüler*innen können den Farbsensor in den Fahrroboter einbauen und anschließen.
3. Die Schüler*innen können die verschiedenen Mess-Modi des Farbsensors erklären und den richtigen Modus für eine Programmieraufgabe auswählen.
4. Die Schüler*innen können mit einem selbst erstellten Programm Messwerte eines Farbsensors auslesen und verarbeiten.
5. Die Schüler*innen können den Einfluss von Farbtönen, Oberflächen und Beleuchtung auf den Messwert bestimmen.

Dauer	Aktivitäten	Verbunden mit Lernziel
10 min	Einleitung & Funktionen des VEX IQ Farbsensor-Moduls	1, 3
10 min	Einbau in den Roboter und Konfiguration in der Programmieroberfläche	1, 2, 3
20 min	Experimentelles Analysieren der verschiedenen Einflussgrößen auf die Farbmessung	3, 4, 5
15 min	Selbstständiges Lösen einer Programmieraufgabe zur Verwendung des Farbsensors zum automatischen Einparken	3, 4, 5
5 min	Reflexion der Übung anhand offener Fragen	1, 2, 3, 4, 5

Materialien

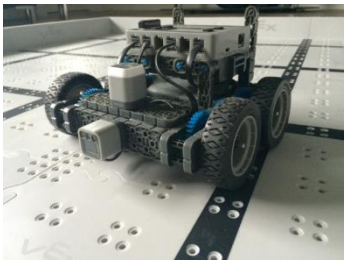
PowerPoint Präsentation Übung 7
Arbeitsblatt “Verwendung des Farbsensors”
VEX IQ Super Kit
VEX IQ Wettbewerbsfeld und Würfelset
Computer mit RobotC Programmierumgebung
Beleuchtung mit unterschiedlicher Farbtemperatur

BY NC SA
Erstellt von: Dreibein Lehrsysteme GmbH

Name(n): _____

Lehrer/Klasse: _____

Datum: _____



Aufgabe 1: Einbau des Farbsensors

Befestige den Farbsensor am Fahrroboter mit vier 1x1 Verbindungs-stiften wie auf dem Bild gezeigt.

Verwende anschließend ein Sensoranschlusskabel in der passenden Länge, um den Farbsensor mit Port 3 des Robot Brains zu verbinden.

Vergiss nicht, den Farbsensor in die Roboter-Konfiguration in RobotC einzutragen!

Aufgabe 2: Erkennen von Farben

1. Erstelle folgendes Programm in RobotC und übertrage es auf den Roboter:

```
1 setTouchLEDColor ( touchLED , colorNone );
2 waitUntil ( getColorName(colorDetector) == colorGreen );
3 setTouchLEDColor ( touchLED , colorGreen );
```

2. Starte das Programm und halte drei Gegenstände mit unterschiedlichen Grüntönen und verschiedenen Oberflächen vor den Farbsensor.
3. Wiederhole Schritte 1 und 2 mit jeweils anderen Farben (rot, gelb, blau, …).

Farbe	Versuch 1	Versuch 2	Versuch 3	Wie viele richtig erkannt?
grün				

Aufgabe 3: Einfluss der Beleuchtung

1. Erstelle folgendes Programm in RobotC und übertrage es auf den Roboter:

```
1 repeat (forever ) {
2   displaySensorValues ( line1 , colorDetector );
3 }
```

2. Starte das Programm und halte einen grünen Gegenstand jeweils bei Tageslicht, kalt-weißer und warm-weißer Beleuchtung vor den Farbsensor und notiere den Messwert auf der Anzeige des Robot Brains. (Hinweis: der Farbwert ist der erste der beiden angezeigten Werte.)
3. Wiederhole Schritt 2 mit jeweils anderen Farben (rot, gelb, blau, …).

Farbe	Farbwert bei Tageslicht	Farbwert bei kalt-weißer Beleuchtung	Farbwert bei warm-weißer Beleuchtung
grün			

BY NC SA

Verwende folgende Farbskala, um den gemessenen Farbwert mit dem realen Farbwert zu vergleichen:

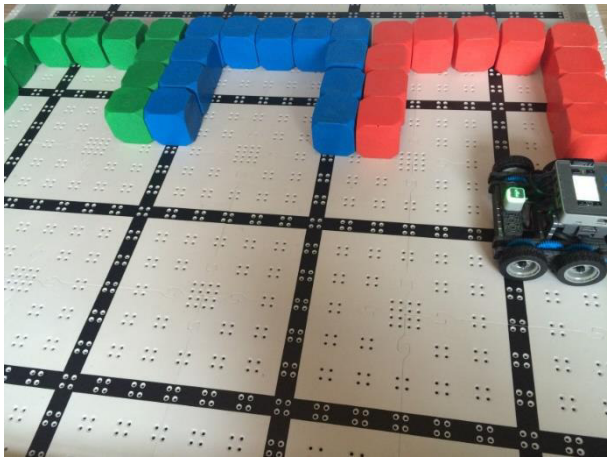
RGB-Skala	0	60	120	180	240	300	360
VEX IQ-Farbwert	0	42	85	127	170	212	255

Was sind Deine Schlussfolgerungen aus den Aufgaben 2 und 3?

Aufgabe 4: Automatisches Einparken

Baue auf dem Wettbewerbsfeld mit den farbigen Würfeln drei farbige Garagen auf – der Fahrroboter muss darin genügend Platz zum Einparken haben!

Schreibe mit den Dir bekannten Befehlen ein Programm, sodass der Roboter zunächst an den Garagen vorbeifährt, um dann in die grüne Garage einzuparken.

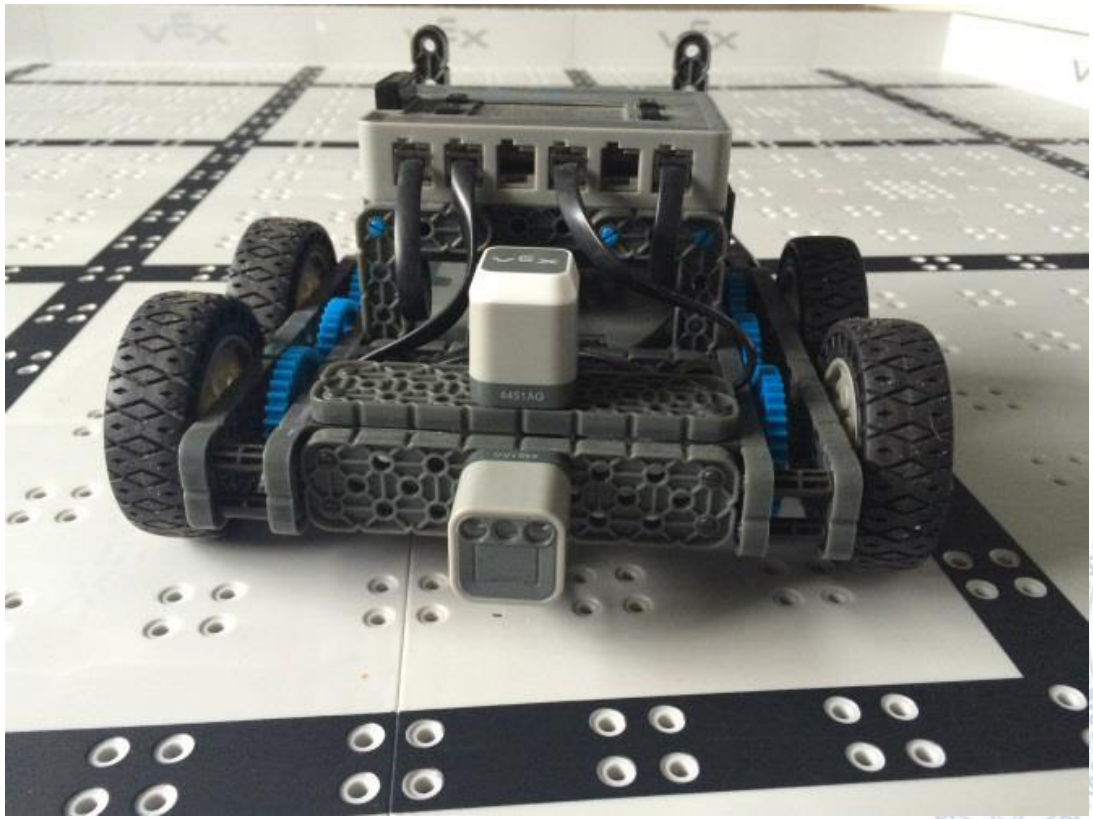


BY NC SA

Erstellt von: Dreibein Lehrsysteme GmbH

Der Farbsensor

In dieser Übung wirst Du den Farbsensor kennenlernen und erfahren, wie dieser im Roboter eingesetzt werden kann.



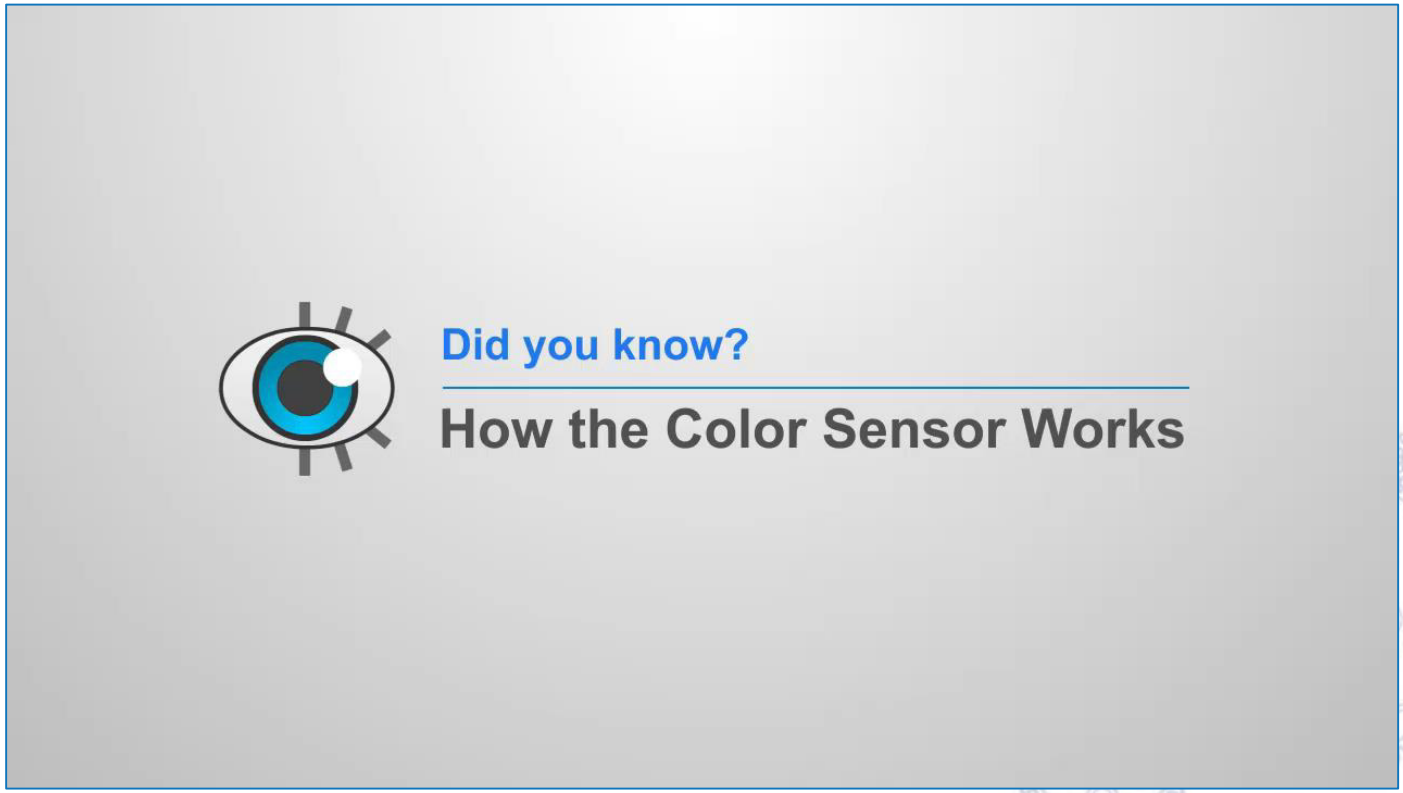
Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Funktionsweise des Farbsensors

Lernziel

Die Schüler*innen können die Funktionsweise der integrierten Sensoren (RGB-Farbsensor, Graustufensensor, Infrarot-Näherungssensor) wiedergeben und differenzieren.



Link zur Online-Version: http://cmra.rec.ri.cmu.edu/products/teaching_robotc_vexiq/lesson/media_videos/howItWorks_color.mp4

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Farbsensor einbauen

Lernziel

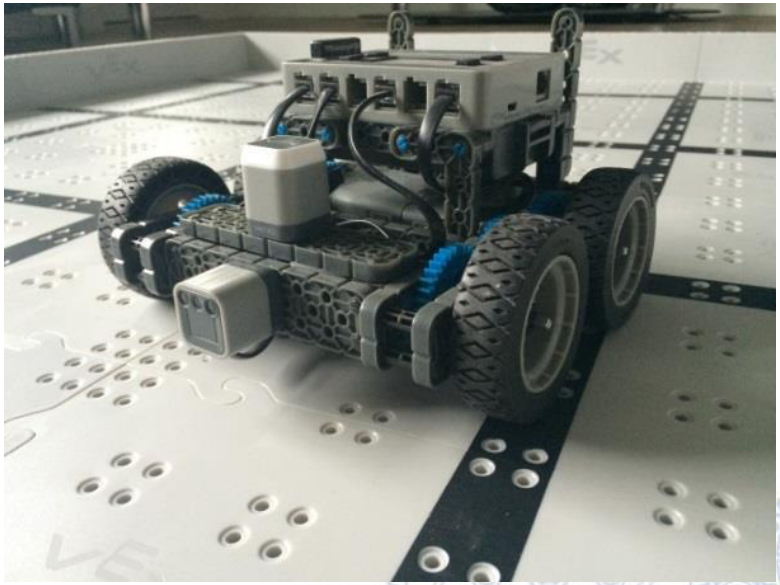
Die Schüler*innen können den Farbsensor in den Fahrroboter einbauen und anschließen.

Aufgabe 1: Einbau des Farbsensors

Befestige den Farbsensor am Fahrroboter mit vier 1x1 Verbindungsstiften wie auf dem Bild gezeigt.

Verwende anschließend ein Sensoranschluss-kabel in der passenden Länge, um den Farbsensor mit dem Robot Brain zu verbinden.

Schließe den Farbsensor an Port 3 des Robot Brains an.



Vergiss nicht, den Farbsensor in die Roboter-Konfiguration in RobotC einzutragen!

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

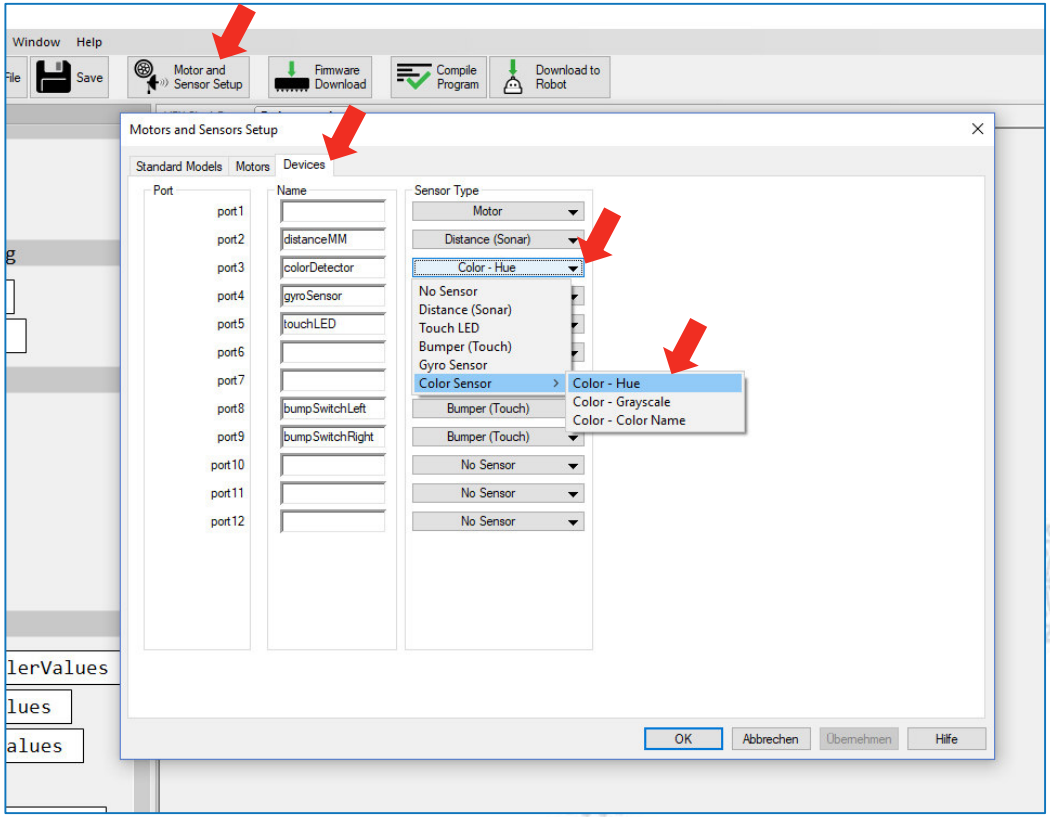
Programmieren mit ROBOTC

Lernziel

Die Schüler*innen können die verschiedenen Mess-Modi des Farbsensors erklären und den richtigen Modus für eine Programmieraufgabe auswählen.

Vor der Verwendung des Farbsensors muss der richtige Mess-Modus eingestellt werden. In der Roboter-Konfiguration von RobotC kannst du dazu 3 verschiedene Modi einstellen:

- Farbwert (Hue)
- Graustufen (Grayscale)
- Farbname (Color Name)



Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Farbwert-Modus

Bei Auswahl des Farbwert-Modus gibt der Farbsensor je nach erkannter Farbe Werte von 0…255 zurück. Die zugehörigen Farben entsprechen der unten abgebildeten Skala und können mithilfe des Dreisatzes in die RGB-Skala umgerechnet werden.

RGB-Skala	0	60	120	180	240	300	360
VEX IQ-Farbwert	0	42	85	127	170	212	255

Der Farbwert-Modus muss in der Roboter-Konfiguration auch gewählt werden, wenn der Infrarot-Näherungssensor verwendet werden soll. Im Programm werden dann unterschiedliche Befehle verwendet, um den Farbsensor bzw. den Näherungssensor auszulesen.

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Graustufen-Modus

Bei Auswahl des Graustufen-Modus gibt der Farbsensor je nach erkannter Helligkeit des Objektes typischer Weise Werte von 0…400 zurück.

- Dunkle Objekte geben einen niedrigen Wert zurück
- Helle Objekte geben einen hohen Wert zurück

Die Messwerte können von verschiedenen Faktoren beeinflusst werden:

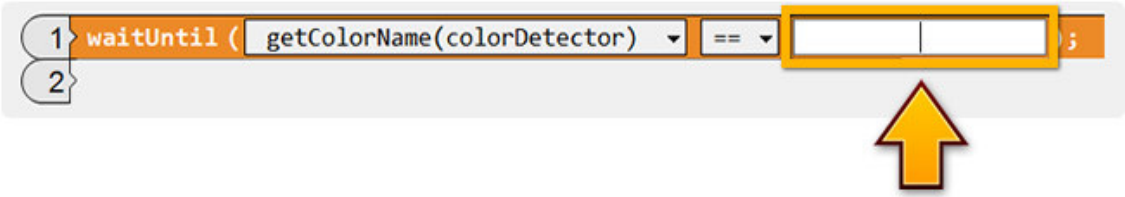
- Oberfläche des Objektes
- Beleuchtung des Objektes
- Beleuchtung der Umgebung

Der Graustufen-Modus wird zumeist für den Linienfolger („line tracker“) verwendet.

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Farbnamen-Modus



Folgende Farben können bei Auswahl des Farbnamen-Modus vom Sensor erkannt werden:

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

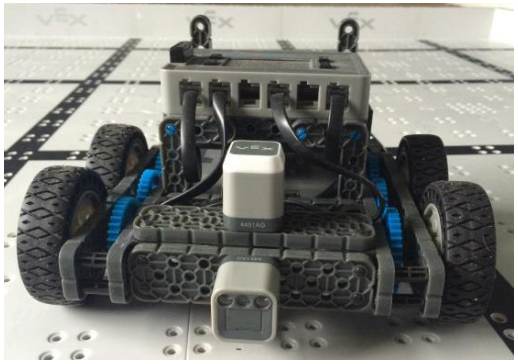
Farbsensor testen

Lernziel

Die Schüler*innen können mit einem selbst erstellten Programm Messwerte eines Farbsensors auslesen und verarbeiten. Darüber hinaus können sie den Einfluss von Farbtönen, Oberflächen und Beleuchtung auf den Messwert bestimmen.

Aufgabe 2: Erkennen von Farben

```
1 setTouchLEDColor ( touchLED , colorNone );
2 waitUntil ( getColorName(colorDetector) == colorGreen );
3 setTouchLEDColor ( touchLED , colorGreen );
```



Übertrage dieses Programm auf den Roboter und versuche mit verschiedenen grünen Objekten die Touch LED zum Leuchten zu bringen.

Führe das selbe Experiment auch mit anderen Farben durch.

Aufgabe 3: Einfluss der Beleuchtung

```
1 repeat (forever ) {
2     displaySensorValues ( line1 , colorDetector );
3 }
```

Verwende dieses Programm zur Anzeige des Farbwertes auf dem Display des Robot Brains. Untersuche, wie sich der Wert bei Veränderung der Beleuchtung verhält!

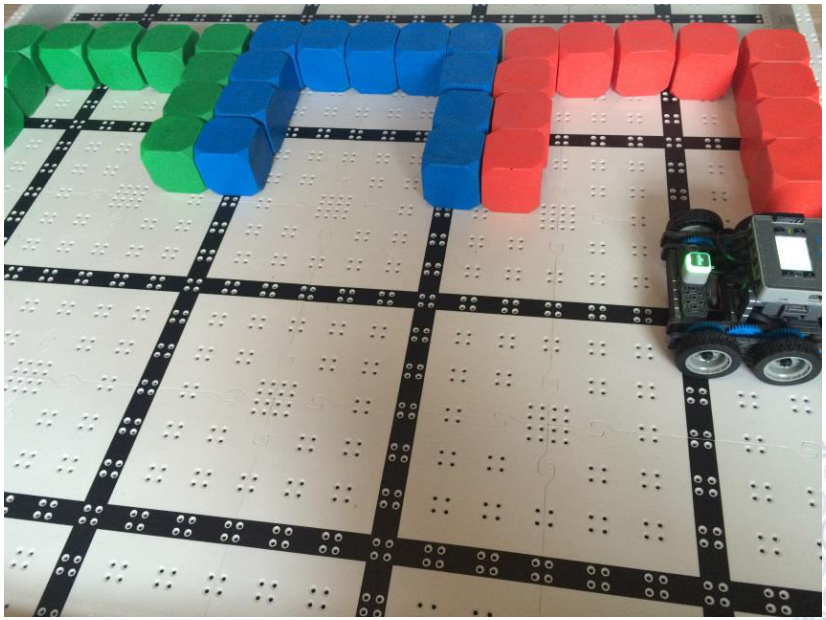
Wie kann die Farberkennung aus Aufgabe 2 sinnvoll eingesetzt werden?

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Automatisches Einparken

Aufgabe 4: Baue auf dem Wettbewerbsfeld mit den farbigen Würfeln drei farbige Garagen auf – der Fahrroboter muss darin genügend Platz zum Einparken haben!
Schreibe mit den Dir bekannten Befehlen ein Programm, sodass der Roboter zunächst an den Garagen vorbeifährt, um dann in die grüne Garage einzuparken.



Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Automatisches Einparken

Link zur Online-Version auf Youtube: https://youtu.be/fsrLbyy4K_U

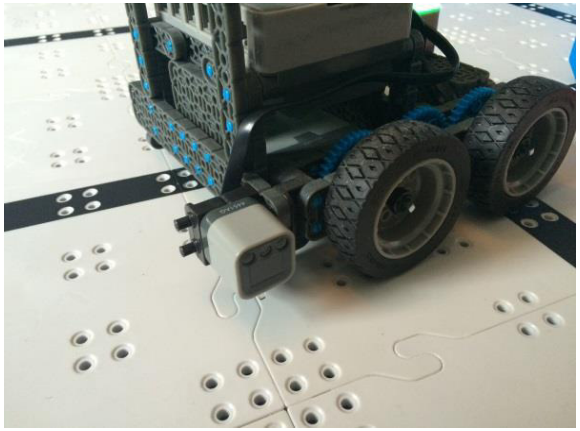
Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Automatisches Einparken

Lösungsvorschlag

```
1 setTouchLEDColor ( touchLED , colorNone );
2 setMultipleMotors ( 50 , leftMotor , rightMotor , , );
3 waitUntil ( getColorName(colorDetector) == colorGreen );
4 setTouchLEDColor ( touchLED , colorGreen );
5 stopMultipleMotors ( leftMotor , rightMotor , , );
6 forward ( .55 , rotations , 50 );
7 turnLeft ( .69 , rotations , 50 );
8 backward ( 1.25 , rotations , 50 );
```



Zum Erkennen der Garagenfarbe musste der Farbsensor versetzt werden. Auf dem Bild ist ein möglicher Aufbau zu sehen.

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Verständnisfragen

Lasst uns gemeinsam folgende Fragen beantworten:

1. Wie funktioniert der Farbsensor?

2. Wie viele Farben kann der Farbsensor erkennen?

3. Hat der Farbsensor im Test alle Farben korrekt erkannt?
   Welche Einflüsse gab es?

4. In welcher realen Anwendung könnte der Farbsensor sinnvoll eingesetzt werden?

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Zusammenfassung

Lernziel

Die Schüler*innen können:

- die Funktionsweise der integrierten Sensoren (RGB-Farbsensor, Graustufensensor, Infrarot-Näherungssensor) wiedergeben und differenzieren
- den Farbsensor in den Fahrroboter einbauen und anschließen
- die verschiedenen Mess-Modi des Farbsensors erklären und den richtigen Modus für eine Programmieraufgabe auswählen
- mit einem selbst erstellten Programm Messwerte eines Farbsensors auslesen und verarbeiten
- den Einfluss von Farbtönen, Oberflächen und Beleuchtung auf den Messwert bestimmen

Heute hast du:

- *gelernt, wie ein Farbsensor funktioniert*
- *den Unterschied zwischen Farb-, Graustufen- und Näherungssensor kennengelernt*
- *den Farbsensor in ein eigenes Programm integriert und damit den Roboter gesteuert*
- *herausgefunden, welche Effekte die Farbmessung beeinflussen*

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Übung 8

Verwendung des Gyrosensors

Übung 8 – Gyrosensor

Unterrichtsinhalte: In dieser Übung lernen die Schüler*innen die Funktionen des VEX IQ Gyrosensors kennen. Dazu werden Aufbau und Funktionsweise des Sensors erläutert. In zwei Aufgaben wird untersucht, wie genau der Sensor unter Einfluss der Drehgeschwindigkeit misst. Darauf basierend kann eine Sollwertkorrektur für die abschließende Aufgabe vorgenommen werden. Dabei soll der Gyrosensor genutzt werden, um den Fahrroboter selbstständig durch einen Zickzack-Parcours fahren zu lassen.

Themen	Altersgruppe	Dauer
Gyrosensor, Drehpunkt, Drehgeschwindigkeit, Winkelmessung, Sollwert, Istwert, Sollwertkorrektur	10-16	60 min

Lernziele

1. Die Schüler*innen können den Aufbau und die Funktionsweise des Gyrosensors erklären.
2. Die Schüler*innen können den Gyrosensor in den Fahrroboter einbauen und anschließen.
3. Die Schüler*innen können Messwerte eines Gyrosensors auslesen und in den Programmablauf einbinden.
4. Die Schüler*innen können den Einfluss der Drehgeschwindigkeit auf den Messwert bestimmen und eine Sollwertkorrektur durchführen.

Dauer	Aktivitäten	Verbunden mit Lernziel
5 min	Einleitung Funktionen des VEX IQ Farbsensor-Moduls	1, 2, 3, 4
10 min	Aufbau und Funktionsweise des VEX IQ Gyrosensors	1
5 min	Einbau in den Roboter und Konfiguration in der Programmieroberfläche	2
20 min	Programmier- und Testaufgaben zur Drehwinkelmessung	3, 4
15 min	Selbstständiges Programmieren zur Fahrt durch einen Zickzack-Parcours	3, 4
5 min	Reflexion der Übung anhand offener Fragen	1, 2, 3, 4

Materialien

PowerPoint Präsentation Übung 8
Arbeitsblatt “Verwendung des Gyrosensors”
VEX IQ Super Kit
VEX IQ Wettbewerbsfeld und Würfelset
Computer mit RobotC Programmierumgebung

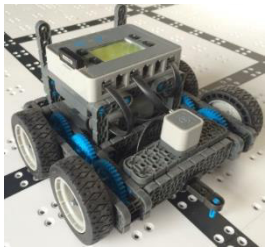
BY NC SA

Name(n): _____

Lehrer/Klasse: _____

Datum: _____

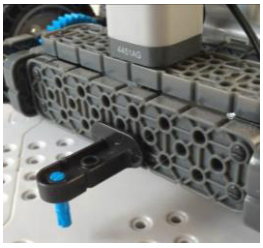
Aufgabe 1: Einbau des Gyrosensors



Befestige den Gyrosensor am Fahrroboter mit vier 1x1 Verbindungs-stiften wie auf dem Bild gezeigt.

Verwende anschließend ein Sensoranschlusskabel in der passenden Länge, um den Gyrosensor mit Port 4 des Robot Brains zu verbinden.

Vergiss nicht, den Gyrosensor in die Roboter-Konfiguration in RobotC einzutragen!



Für Aufgabe 2 und 3 benötigst Du einen Zeiger, den Du wie auf dem unteren Bild mit einem breiten Winkelverbinder (2x1 mit Versatz) und einem 1x2 Verbindungsstift aufbaust.

Aufgabe 2: Einfluss der Geschwindigkeit auf den Drehwinkel

1. Erstelle folgendes Programm in RobotC und übertrage es auf den Roboter:

```
1 resetGyro ( gyroSensor );
2 setMotor ( motor1 , -50 );
3 setMotor ( motor6 , 50 );
4 waitUntil ( getGyroDegrees(gyroSensor) > 90 );
5 stopMultipleMotors ( motor1 , motor6 , , );
```

2. Lass das Programm dreimal durchlaufen und notiere jeweils den exakten Winkel, den sich der Fahrroboter gedreht hat.
3. Wiederhole Schritte 1 und 2 mit den in der Tabelle angegebenen Motorleistungen.

Drehgeschwindigkeit (Motorleistung)	Drehwinkel Versuch 1	Drehwinkel Versuch 2	Drehwinkel Versuch 3	Drehwinkel Durchschnitt
50				
40				
30				

BY NC SA
Erstellt von: Dreibein Lehrsysteme GmbH

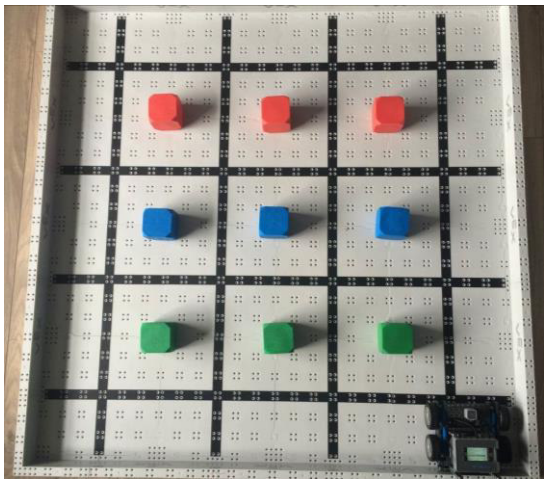
Aufgabe 3: Drehwinkel: Änderung des Sollwertes

1. Verwende das ursprüngliche Programm aus Aufgabe 2.
2. Lass das Programm dreimal durchlaufen und notiere jeweils den exakten Winkel, den sich der Fahrroboter gedreht hat.
3. Wiederhole Schritte 1 und 2 mit den in der Tabelle angegebenen Sollwerten.

Drehwinkel Sollwert	Istwert 1	Istwert 2	Istwert 3	Istwert Durchschnitt
90°				
85°				
80°				

Welche Erkenntnisse kannst Du aus Aufgabe 2 und 3 für die Verwendung des Gyrosensors gewinnen?

Aufgabe 4: Rasenmäher Parcours



Menschen suchen immer nach Möglichkeiten, zeitaufwändige Aufgaben abzugeben. Eine solche Aufgabe ist auch das Mähen großer Wiesen.

Baue für diese Aufgabe wie auf dem Bild ein Gitter aus neun Würfel auf. Du musst nun unter Verwendung des Gyrosensors und den Dir bekannten Befehlen ein Programm schreiben, mit dem der Fahrroboter im Zickzack zwischen den Würfeln hindurch fährt, ohne diese zu berühren.

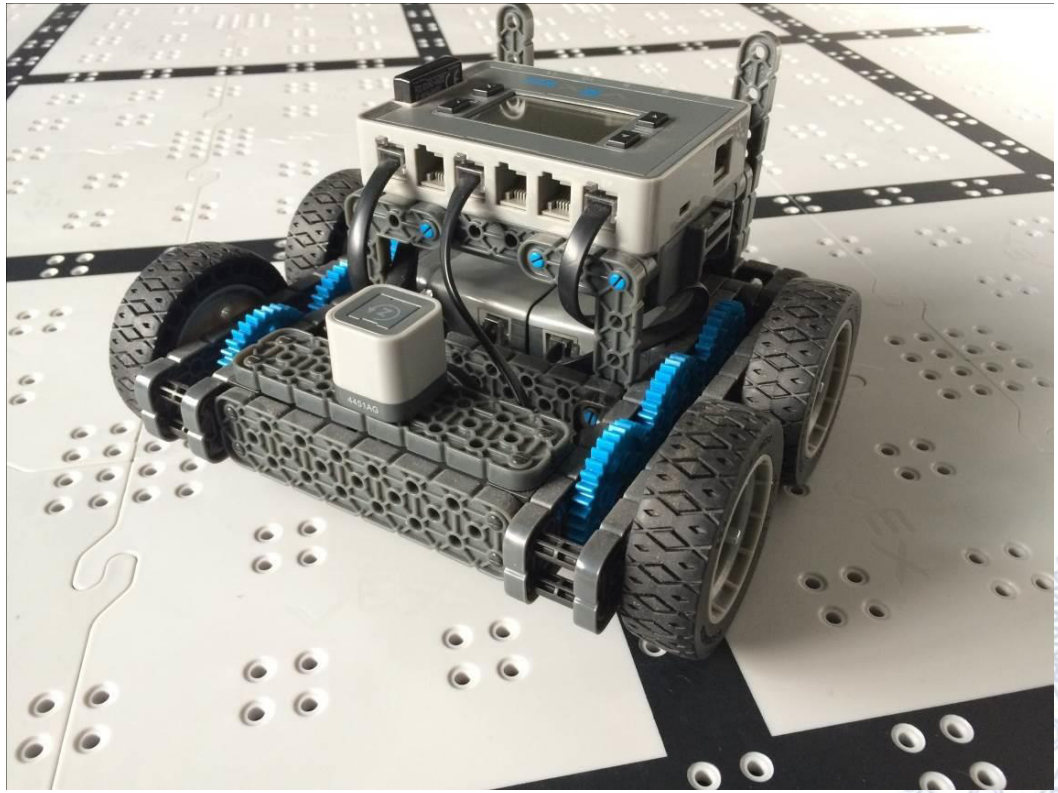
Das schnellste Team gewinnt den Wettbewerb. Für jede Würfelberührung gibt es allerdings 15 Strafsekunden.

BY NC SA

Erstellt von: Dreibein Lehrsysteme GmbH

Der Gyrosensor

In dieser Übung wirst Du den Gyrosensor kennenlernen und erfahren, wie dieser im Roboter eingesetzt werden kann.



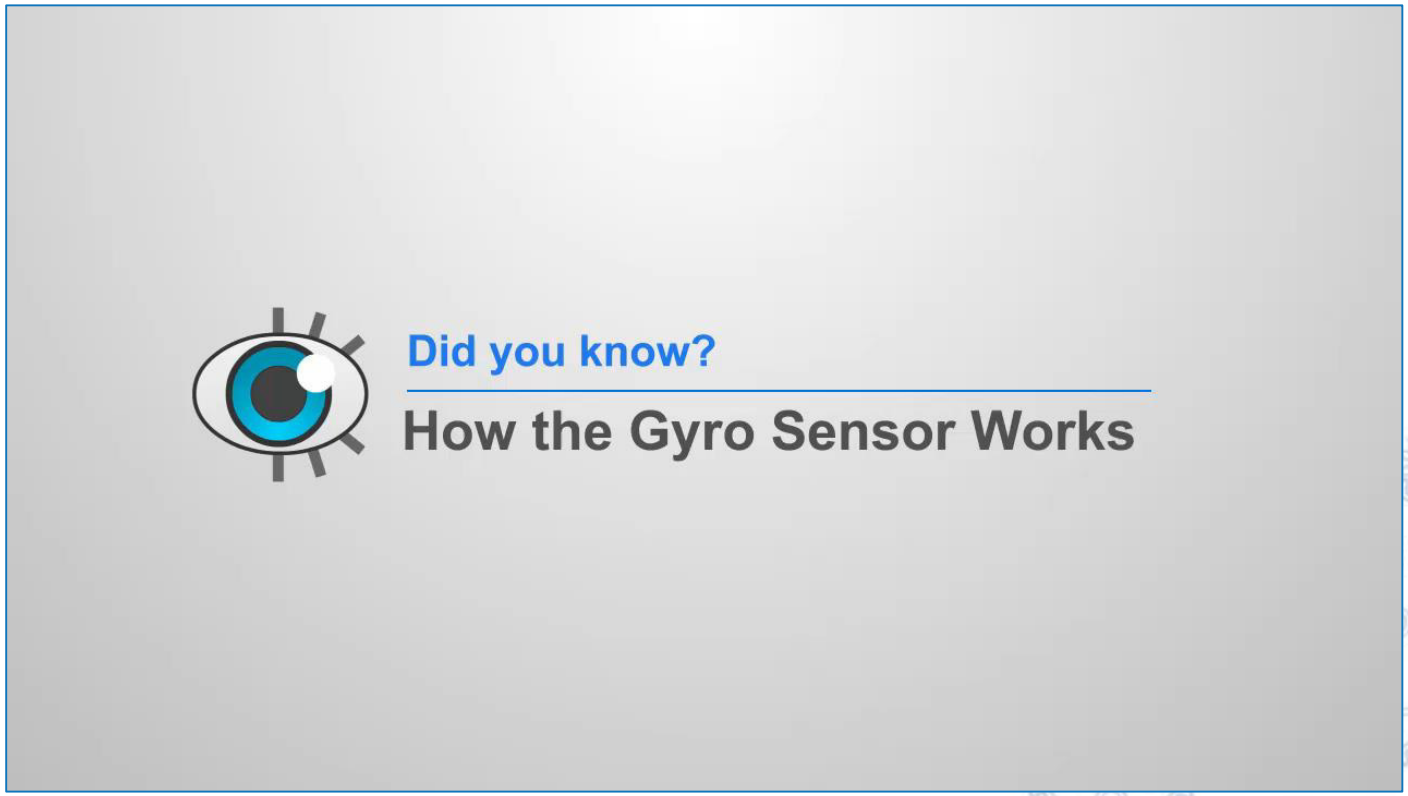
Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Funktionsweise des Gyrosensors

Lernziel

Die Schüler*innen können den Aufbau und die Funktionsweise des Gyrosensors erklären.



Link zur Online-Version: http://cmra.rec.ri.cmu.edu/products/teaching_robotc_vexiq/lesson/media_videos/howItWorks_gyro.mp4

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Gyrosensor einbauen

Lernziel

Die Schüler*innen können den Gyrosensor in den Fahrroboter einbauen und anschließen.

Aufgabe 1: Einbau des Gyrosensors

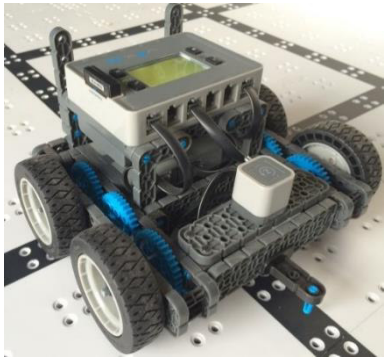
Befestige den Gyrosensor am Fahrroboter mit vier 1x1 Verbindungs-stiften wie auf dem Bild gezeigt.

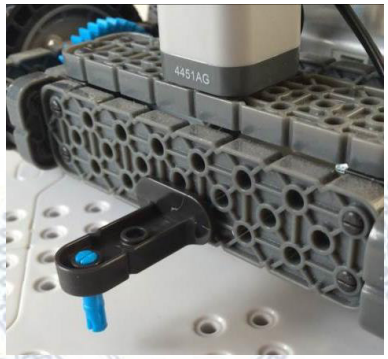
Verwende anschließend ein Sensoranschlusskabel in der passenden Länge, um den Gyrosensor mit dem Robot Brain zu verbinden.

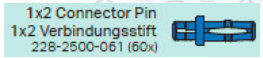
Schließe den Gyrosensor an Port 4 des Robot Brains an.

Für Aufgabe 2 und 3 benötigst Du einen Zeiger, den Du wie auf dem unteren Bild mit einem breiten Winkelverbinder (2x1 mit Versatz) und einem 1x2 Verbindungsstift aufbaust.

Vergiss nicht, den Gyrosensor in die Roboter-Konfiguration in RobotC einzutragen!







Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

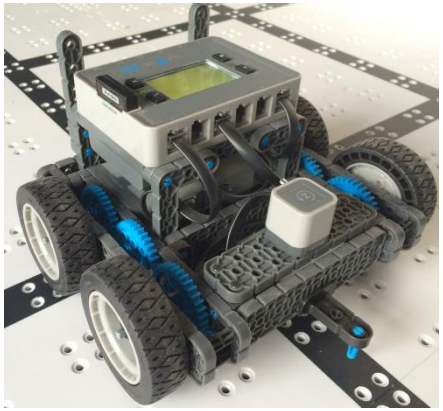
Gyrosensor testen

Lernziel

Die Schüler*innen können Messwerte eines Gyrosensors auslesen und in den Programmablauf einbinden. Darüber hinaus können sie den Einfluss der Drehgeschwindigkeit auf den Messwert bestimmen und eine Sollwertkorrektur durchführen.

Aufgabe 2: Einfluss der Geschwindigkeit auf den Drehwinkel

```
1 resetGyro ( gyroSensor );
2 setMotor ( motor1 , -50 );
3 setMotor ( motor6 , 50 );
4 waitUntil ( getGyroDegrees(gyroSensor) > 90 );
5 stopMultipleMotors ( motor1 , motor6 , , );
```



Übertrage dieses Programm auf den Roboter und finde heraus, welchen Einfluss die Drehgeschwindigkeit (Motorleistung) auf den Drehwinkel hat.

Aufgabe 3: Drehwinkel: Änderung des Sollwertes

Halte nun die Motorleistung konstant und ändere den Vorgabewert des Drehwinkels.

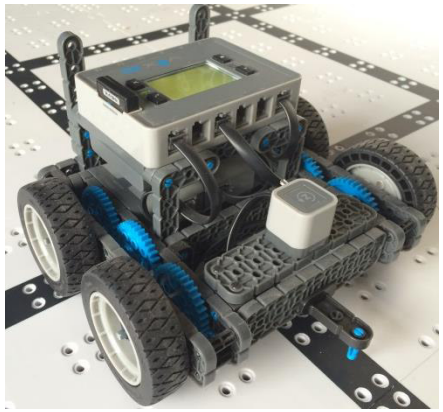
Welche Erkenntnisse kannst Du aus Aufgabe 2 und 3 für die Verwendung des Gyrosensors gewinnen?

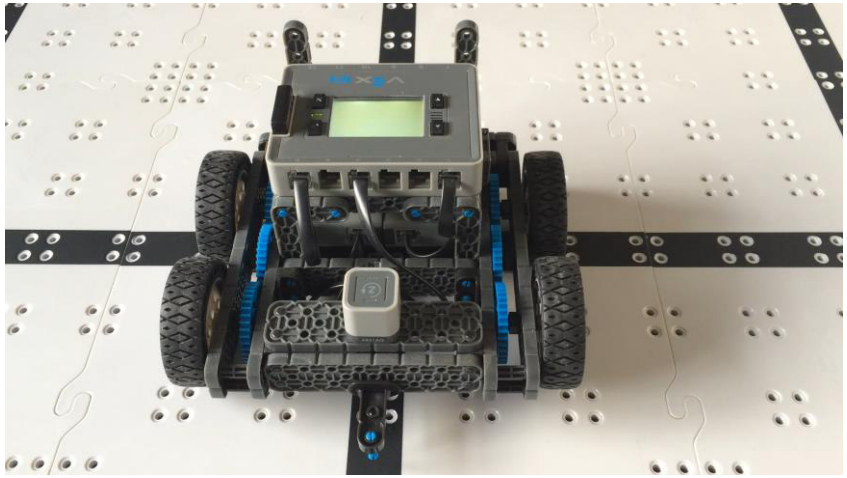
Erstellt von: Dreibein Lehrsysteme GmbH

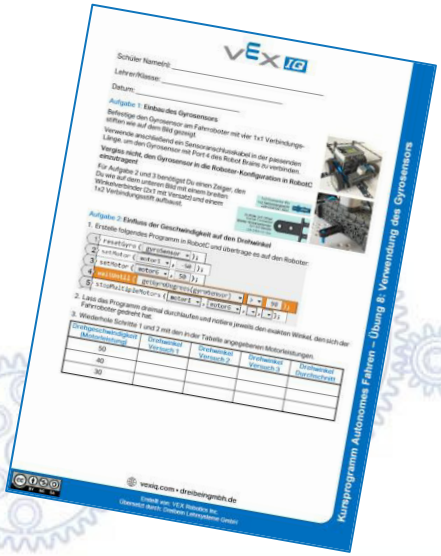
CC BY NC SA

Gyrosensor testen

Testaufbau: Nutze das Wettbewerbsfeld und positioniere die Hinterräder des Fahrroboters direkt über einem Kreuz, sodass beide Räder auf der schwarzen Linie stehen. Der Zeiger sollte dann genau auf die dazu rechtwinklig stehende Linie zeigen.





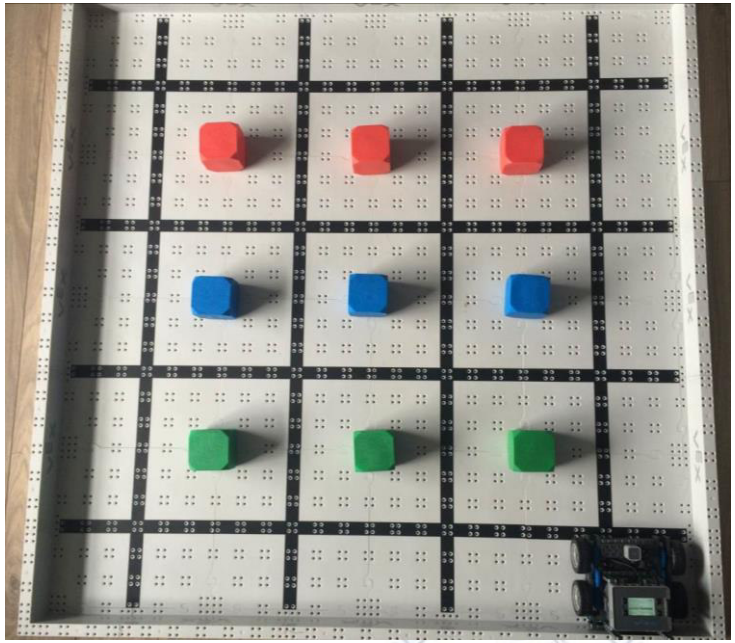


Führe die Aufgaben wie auf dem Arbeitsblatt beschrieben durch und miss den Drehwinkel immer von der Mitte des Kreuzes.

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Rasenmäher Parcours

Aufgabe 4: Menschen suchen immer nach Möglichkeiten, zeitaufwändige Aufgaben abzugeben. Eine solche Aufgabe ist auch das Mähen großer Wiesen.



Baue für diese Aufgabe wie auf dem Bild ein Gitter aus neun Würfel auf. Du musst nun unter Verwendung des Gyrosensors und den Dir bekannten Befehlen ein Programm schreiben, mit dem der Fahrroboter im Zickzack zwischen den Würfeln hindurch fährt, ohne diese zu berühren.

Das schnellste Team gewinnt den Wettbewerb. Für jede Würfelberührung gibt es allerdings 15 Strafsekunden.

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Rasenmäher Parcours

Link zur Online-Version auf Youtube: https://youtu.be/WTBVRML1o14

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Rasenmäher Parcours

Lösungsvorschlag

```
1  resetGyro ( gyroSensor );
2  forward ( 4.75 , rotations , 40 );
3  setMotor ( motor1 , -40 );
4  setMotor ( motor6 , 40 );
5  waitUntil ( getGyroDegrees(gyroSensor) > 85 );
6  stopMultipleMotors ( motor1 , motor6 , , );
7  forward ( 1.47 , rotations , 50 );
8  setMotor ( motor1 , -40 );
9  setMotor ( motor6 , 40 );
10 waitUntil ( getGyroDegrees(gyroSensor) > 170 );
11 stopMultipleMotors ( motor1 , motor6 , , );
12 forward ( 4.50 , rotations , 50 );
13 setMotor ( motor1 , 40 );
14 setMotor ( motor6 , -40 );
15 waitUntil ( getGyroDegrees(gyroSensor) < 85 );
16 stopMultipleMotors ( motor1 , motor6 , , );
```

...und so weiter.

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Verständnisfragen

Lasst uns gemeinsam folgende Fragen beantworten:

1. Wie funktioniert der Gyrosensor?

2. Warum muss der Messwert des Gyrosensor am Anfang des Programms zurückgesetzt werden?

3. Welchen Einfluss hat die Drehgeschwindigkeit des Roboters auf den Drehwinkel?

4. In welchen anderen Anwendungen könnte der Gyrosensor sinnvoll eingesetzt werden?

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Zusammenfassung

Lernziel

Die Schüler*innen können:

- den Aufbau und die Funktionsweise des Gyrosensors erklären
- den Gyrosensor in den Fahrroboter einbauen und anschließen
- Messwerte eines Gyrosensors auslesen und in den Programmablauf einbinden
- den Einfluss der Drehgeschwindigkeit auf den Messwert bestimmen und eine Sollwertkorrektur durchführen

Heute hast du:

- *gelernt, wie ein Gyrosensor funktioniert*
- *den Gyrosensor mit dem Befehl 'get GyroDegrees' in ein eigenes Programm integriert und damit den Roboter gesteuert*
- *herausgefunden, welchen Einfluss die Drehgeschwindigkeit auf die Drehung des Roboters hat*
- *den Fahrroboter über den Rasenmäher Parcours fahren lassen*

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Projekt 1

Fahren mit Assistenzsystemen
und
Teilautonomes Fahren

Projekt 1 – Fahren mit Assistenzsystemen und teilautonomes Fahren

Unterrichtsinhalte: In dieser Übung lernen die Schüler*innen die Stufen des autonomen Fahrens kennen. Sie planen und bauen selbstständig einen Parcours, für den anschließend die Fahrroboter modifiziert werden müssen. Beim Fahren mit Assistenzsystemen müssen die Schüler*innen die Sensoren im Fahrroboter nutzen, um bestimmte Situationen beim Fahren festzustellen und den Fahrer daraufhin zu warnen oder ins Fahrgeschehen einzugreifen. Beim autonomen Fahren müssen sie die Sensoren im Fahrroboter nutzen, um bestimmte Abläufe selbständig, also ohne Eingreifen des Fahrers zu bewältigen.

Themen	Altersgruppe	Dauer
Stufen des autonomen Fahrens, Assistenzsysteme, teilautonomes Fahren, Projektplanung, Anforderungsmanagement	10-16	90…180 min

Lernziele

1. Die Schüler*innen können die Stufen des autonomen Fahrens wiedergeben und mit Beispielen erläutern.
2. Die Schüler*innen können selbstständig die Schwierigkeit der folgenden Aufgaben festlegen. Sie verstehen die Anforderungen durch die verschiedenen Parcours-Elemente, durchdenken einen Lösungsansatz für den Fahrroboter und schätzen den Aufwand ab.
3. Die Schüler*innen können selbstständig Anforderungen für den Fahrroboter aus den verschiedenen Parcours-Elementen ableiten und erarbeiten daraufhin Lösungen für die Hardware und die Programmierung zum Fahren mit Assistenzsystemen.
4. Die Schüler*innen können selbstständig Anforderungen für den Fahrroboter aus den verschiedenen Parcours-Elementen ableiten und erarbeiten daraufhin Lösungen für die Hardware und die Programmierung zum teilautonomen Fahren.

Dauer	Aktivitäten	Verbunden mit Lernziel
10 min	Einleitung & Projektbesprechung	1, 2,3, 4
15 min	Recherche zu „Stufen des autonomen Fahrens“	1
15 min	Planung und Aufbau Parcours	1, 2
ca. 60 min	Fahrzeug mit Assistenzsystemen aufbauen	3
ca. 60 min	Teilautonomes Fahrzeug aufbauen	4
10 min	Reflexion Projektes anhand offener Fragen	1, 2, 3, 4

Materialien

PowerPoint Präsentation Projekt 1
Arbeitsblatt “Fahren mit Assistenzsystemen und teilautonomes Fahren”
VEX IQ Super Kit
Computer mit RobotC Programmierumgebung
Unterrichtsraum mit Fläche für Parcours

BY NC SA

Name(n): _____

Lehrer/Klasse: _____

Datum: _____

Aufgabe 1: Welche Stufen des autonomen Fahrens gibt es?

„Autonomes Fahren“ ist nicht gleich „Autonomes Fahren“. Abhängig von den Funktionen eines autonomes Fahrzeugs unterscheidet man verschiedene Autonomiestufen. Recherchiere, welche Stufen es gibt und nenne jeweils ein Anwendungsbeispiel bzw. eine kurze Erläuterung.

Aufgabe 2: Baut Euren eigenen Parcours auf

Für die folgenden Aufgaben zum Fahren mit Assistenzsystemen und zum teilautonomen Fahren wird ein Parcours benötigt. Verwendet dazu Gegenstände aus Eurer Umgebung.
Beachtet, dass Ihr Euren Fahrroboter anschließend in begrenzter Zeit so modifizieren und programmieren müsst, dass der Parcours erfolgreich gemeistert werden kann.
Hier ein paar Ideen für Euren Parcours:

- Start & Ziel
- Rote Ampel
- Stopp-Schild
- Hindernis zum Umfahren
- Linie für Line-Tracker
- Wippe
- Tor / Einfahrt
- Parklücke (in Reihe & parallel)

Aufgabe 3: Fahrzeug mit Assistenzsystemen aufbauen

Modifiziert und programmiert Euren Fahrroboter so, dass er mit der Fernsteuerung gefahren werden kann und dabei Informationen und Warnungen an den Fahrer gibt oder in bestimmten Situationen ins Fahrgeschehen eingreift.
Hier ein paar Ideen für mögliche Assistenzsysteme:

- Warnt bei geringem Abstand
- Zeigt den Abstand als Messwert an
- Reduziert Geschwindigkeit bei geringem Abstand
- Stoppt bei zu geringem Abstand
- Erkennt Ampelfarben
- Stoppt bei roter Ampel

Aufgabe 4: Teilautonomes Fahrzeug aufbauen

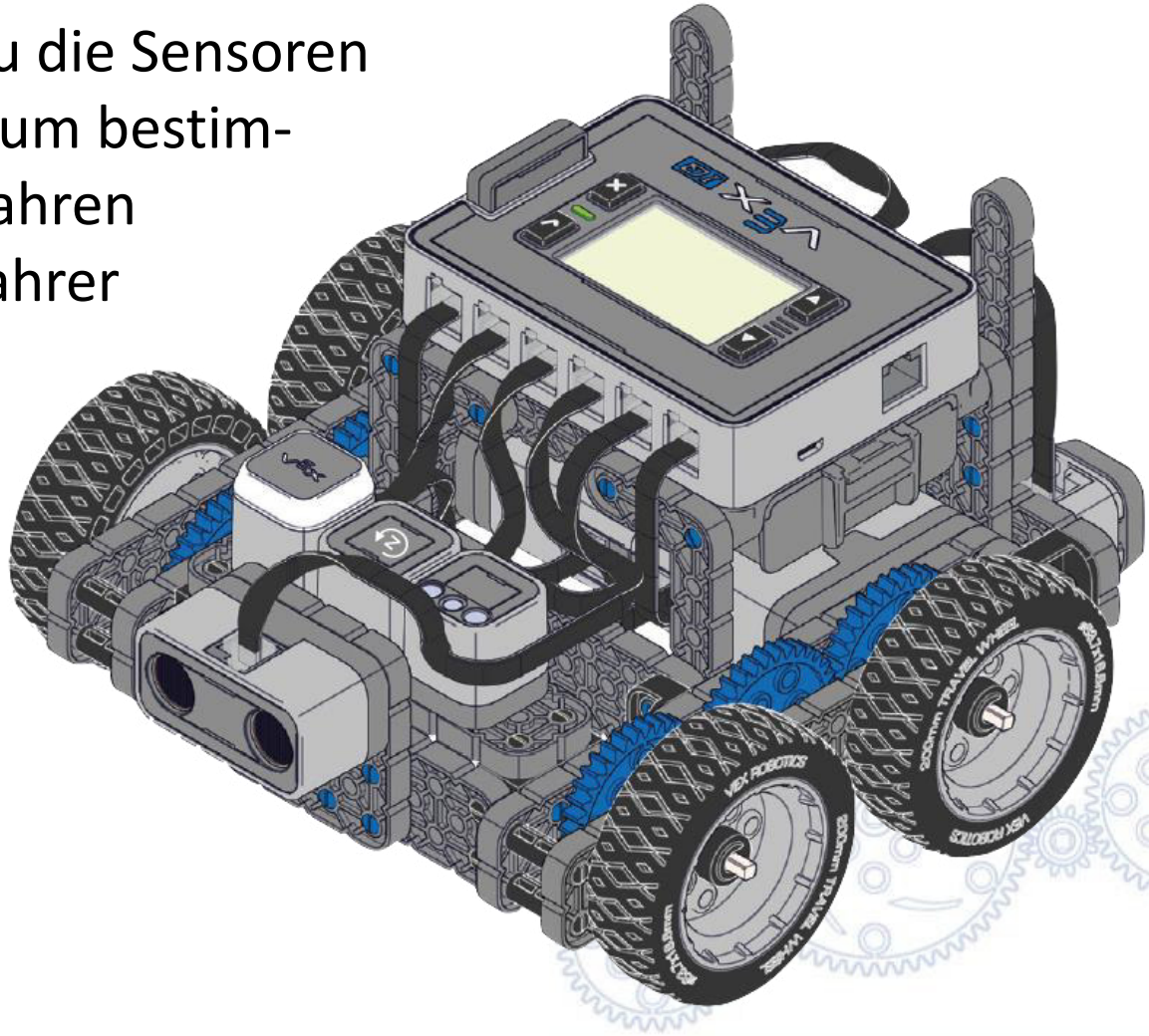
Modifiziert und programmiert Euren Fahrroboter so, dass er mit der Fernsteuerung gefahren werden kann und bestimmte Situationen auf Befehl des Fahrers selbstständig bewältigt.
Hier ein paar Ideen für teilautonome Funktionen:

- Automatisches Einparken
- Umfahren eines Hindernisses auf Knopfdruck
- Autonomes Fahren in speziell markierten Bereichen (Line-Tracker)

BY NC SA

Teil 1: Fahren mit Assistenzsystemen

In dieser Übung wirst Du die Sensoren im Fahrroboter nutzen, um bestimmte Situationen beim Fahren festzustellen und den Fahrer daraufhin zu warnen oder ins Fahrgeschehen einzugreifen.

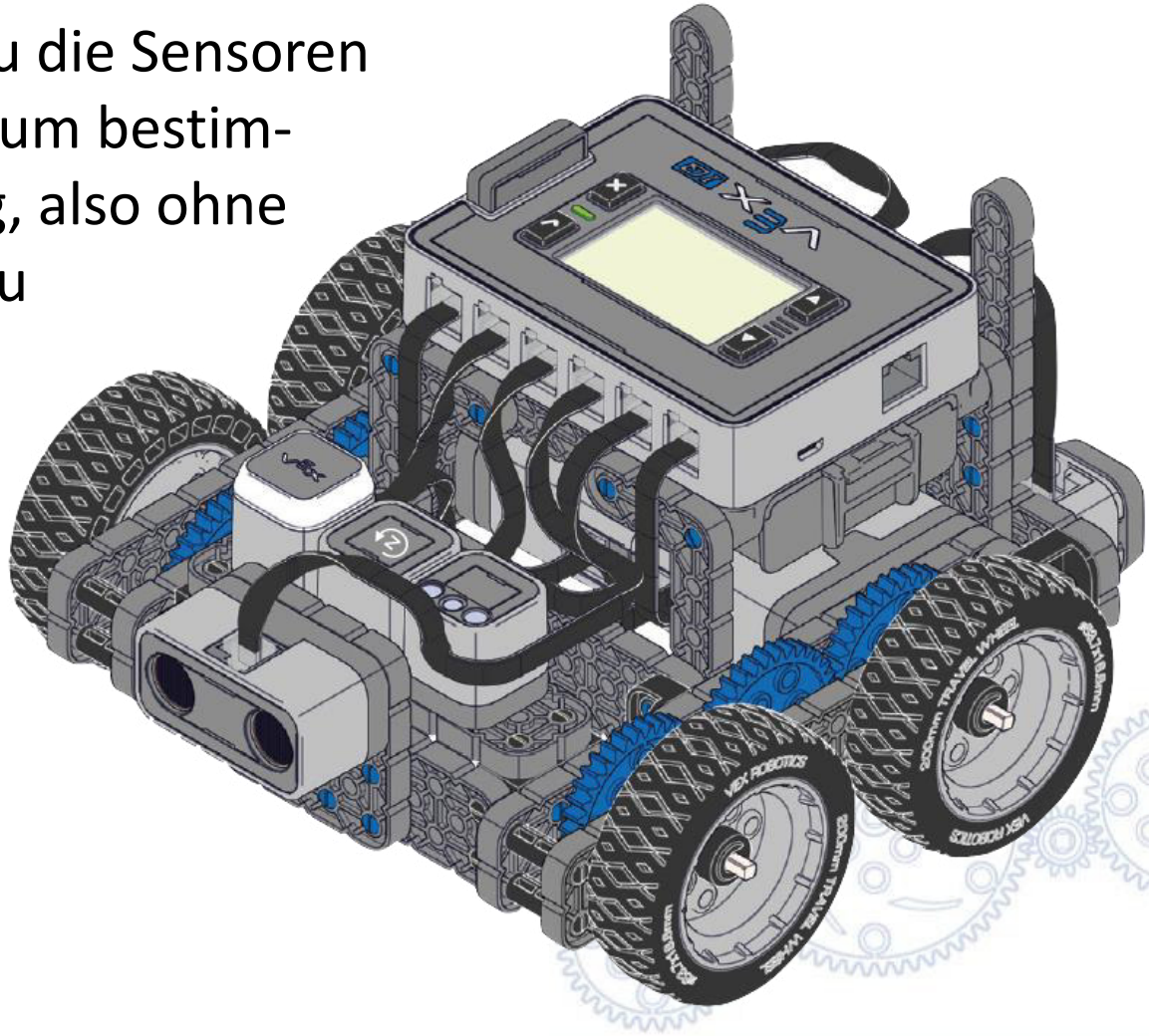


Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Teil 2: Teilautonomes Fahren

In dieser Übung wirst Du die Sensoren im Fahrroboter nutzen, um bestimmte Abläufe selbständig, also ohne Eingreifen des Fahrers zu bewältigen.



Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Autonomes Fahren

Was bedeutet eigentlich „autonom“?

- **Autonom:** selbstständig, unabhängig, eigenverantwortlich
- **Automobil:** ein Fortbewegungsmittel, das aus eigener Kraft fährt – ein Auto(mobil) macht den Fahrer autonom
- **Autonomes Fahrzeug:** ein Automobil, das eigenverantwortlich fährt – ein autonomes Fahrzeug bringt die Passagiere selbstständig zum Ziel

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Autonomes Fahren

Lernziel

Die Schüler*innen können die Stufen des autonomen Fahrens wiedergeben und mit Beispielen erläutern.

Aufgabe 1: Welche Stufen des autonomen Fahrens gibt es?

„Autonomes Fahren“ ist nicht gleich „Autonomes Fahren“. Abhängig von den Funktionen eines autonomes Fahrzeugs unterscheidet man verschiedene Autonomiestufen. Recherchiere, welche Stufen es gibt und nenne jeweils ein Anwendungsbeispiel bzw. eine kurze Erläuterung.

Lösung:

Laut Bundesanstalt für Straßenwesen gibt es folgende **Autonomiestufen**:
Level 0: „Driver only“ – der Fahrer fährt selbst
Level 1: Fahren mit Assistenzsystemen – z.B. Abstandswarner, Notbremsassistent
Level 2: Teilautomatisierung – z.B. automatisches Einparken
Level 3: Hochautomatisierung – z.B. autonomes Fahren auf Autobahnen
Level 4: Vollautomatisierung – vollständiges autonomes Fahren von A nach B mit Überwachung
Level 5: Kein Fahrer erforderlich – vollständiges autonomes Fahren von A nach B ohne Überwachung

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Parcours aufbauen

Lernziel

Die Schüler*innen können selbstständig die Schwierigkeit der folgenden Aufgaben festlegen. Sie verstehen die Anforderungen durch die verschiedenen Parcours-Elemente, durchdenken einen Lösungsansatz für den Fahrroboter und schätzen den Aufwand ab.

Aufgabe 2: Baut Euren eigenen Parcours auf

Für die folgenden Aufgaben zum Fahren mit Assistenzsystemen und zum teilautonomen Fahren wird ein Parcours benötigt. Verwendet dazu Gegenstände aus Eurer Umgebung.
Beachtet, dass Ihr Euren Fahrroboter anschließend in begrenzter Zeit so modifizieren und programmieren müsst, dass der Parcours erfolgreich gemeistert werden kann.

Hier ein paar Ideen für Euren Parcours:

- Start & Ziel
- Rote Ampel
- Stopp-Schild
- Hindernis zum Umfahren
- Linie für Line-Tracker
- Wippe
- Tor / Einfahrt
- Parklücke (in Reihe & parallel)

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Fahren mit Assistenzsystemen

Lernziel

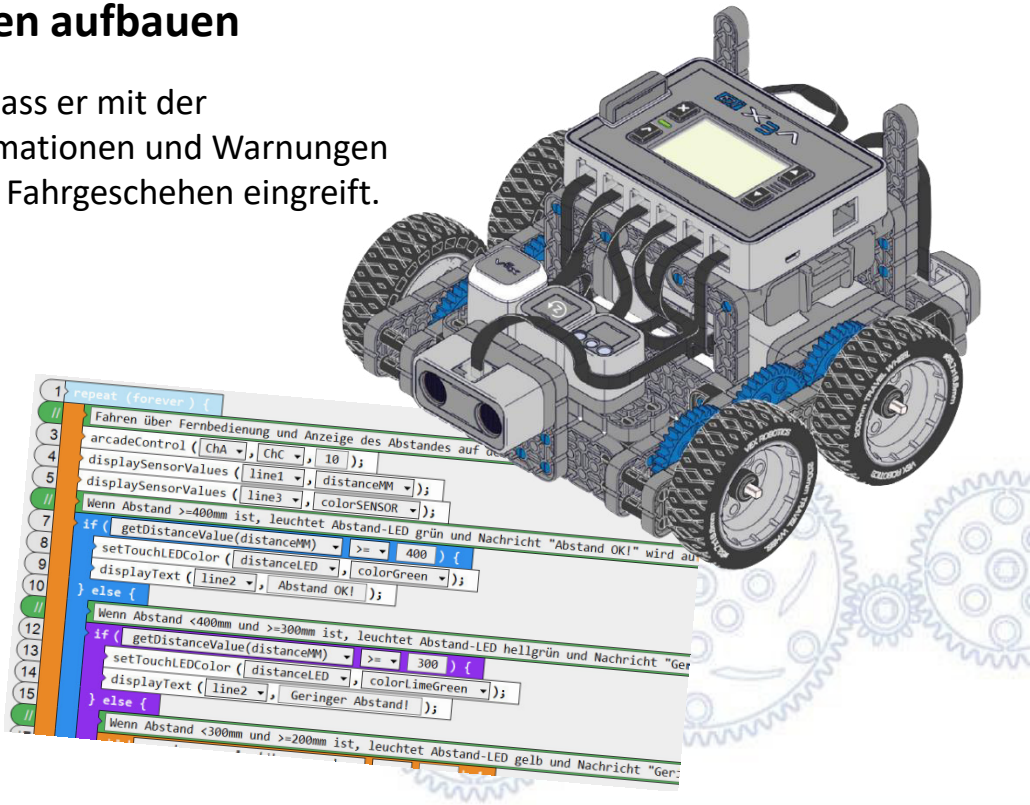
Die Schüler*innen können selbstständig Anforderungen für den Fahrroboter aus den verschiedenen Parcours-Elementen ableiten und erarbeiten daraufhin Lösungen für die Hardware und die Programmierung zum Fahren mit Assistenzsystemen.

Aufgabe 3: Fahrzeug mit Assistenzsystemen aufbauen

Modifiziert und programmiert Euren Fahrroboter so, dass er mit der Fernsteuerung gefahren werden kann und dabei Informationen und Warnungen an den Fahrer gibt oder in bestimmten Situationen ins Fahrgeschehen eingreift.

Hier ein paar Ideen für mögliche Assistenzsysteme:

- Warnt bei geringem Abstand
- Zeigt den Abstand als Messwert an
- Reduziert Geschwindigkeit bei geringem Abstand
- Stoppt bei zu geringem Abstand
- Erkennt Ampelfarben
- Stoppt bei roter Ampel



Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Fahren mit Assistenzsystemen

Lösungsansatz: Fahren mit Fernbedienung

```
1  repeat (forever ) {
//     Fahren über Fernbedienung und Anzeige des Abstandes auf dem Display
3      arcadeControl ( ChA , ChC , 10 );
4      displaySensorValues ( line1 , distanceMM );
```


Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Fahren mit Assistenzsystemen

Lösungsansatz: Abstands-Assistent

```
// Wenn Abstand >=200mm ist, leuchtet LED grün und Nachricht "Abstand OK!" wird auf dem Display angezeigt
if ( getDistanceValue(distanceMM) >= 200 ) {
  setTouchLEDColor ( distanceLED , colorGreen );
  displayText ( line2 , Abstand OK! );
} else {
  // Wenn Abstand <200mm und >=100mm ist, leuchtet LED gelb und Nachricht "Geringer Abstand!" wird auf dem Display angezeigt
  if ( getDistanceValue(distanceMM) >= 100 ) {
    setTouchLEDColor ( distanceLED , colorYellow );
    displayText ( line2 , Geringer Abstand! );
  } else {
    // Wenn Abstand <100mm und >=50mm ist, leuchtet LED orange und Nachricht "Kritischer Abstand!" wird auf dem Display angezeigt
    if ( getDistanceValue(distanceMM) >= 50 ) {
      setTouchLEDColor ( distanceLED , colorOrange );
      displayText ( line2 , Kritischer Abstand! );
    } else {
      // Wenn Abstand <50mm ist, leuchtet LED rot, Nachricht "Stopp!!!" wird auf dem Display angezeigt und alle Motoren werden gestoppt
      setTouchLEDColor ( distanceLED , colorRed );
      displayText ( line2 , Stopp!!! );
      stopAllMotors ( );
      // Um den Sicherheitsstopp zu deaktivieren und rückwärts zu fahren muss die Taste "F unten" auf der Fernbedienung gedrückt werden
      waitUntil ( getJoystickValue(BtnFDown) == 1 );
      while ( getJoystickValue(BtnFDown) == 1 ) {
        setMultipleMotors ( -20 , leftMotor , rightMotor , noMotor , noMotor );
      }
    }
  }
}
```


Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Fahren mit Assistenzsystemen

Lösungsansatz: Ampelerkennung

```
// Wenn Farbsensor rot erkennt, wird der Timer T1 zurückgesetzt, die Ampel-LED leuchtet rot und alle Motoren werden gestoppt
33  if ( getColorName(colorSENSOR) == colorRed ) {
34      resetTimer ( T1 );
35      setTouchLEDColor ( colorLED , colorRed );
36      stopAllMotors ( );
37      waitUntil ( getTimer(T1, seconds) >= 3 );
38  }
// Erkennt der Farbsensor kein rot, leuchtet die Ampel-LED wieder grün
40  if ( getColorName(colorSENSOR) != colorRed ) {
41      setTouchLEDColor ( colorLED , colorGreen );
42  }
```


Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Teilautonomes Fahren

Lernziel

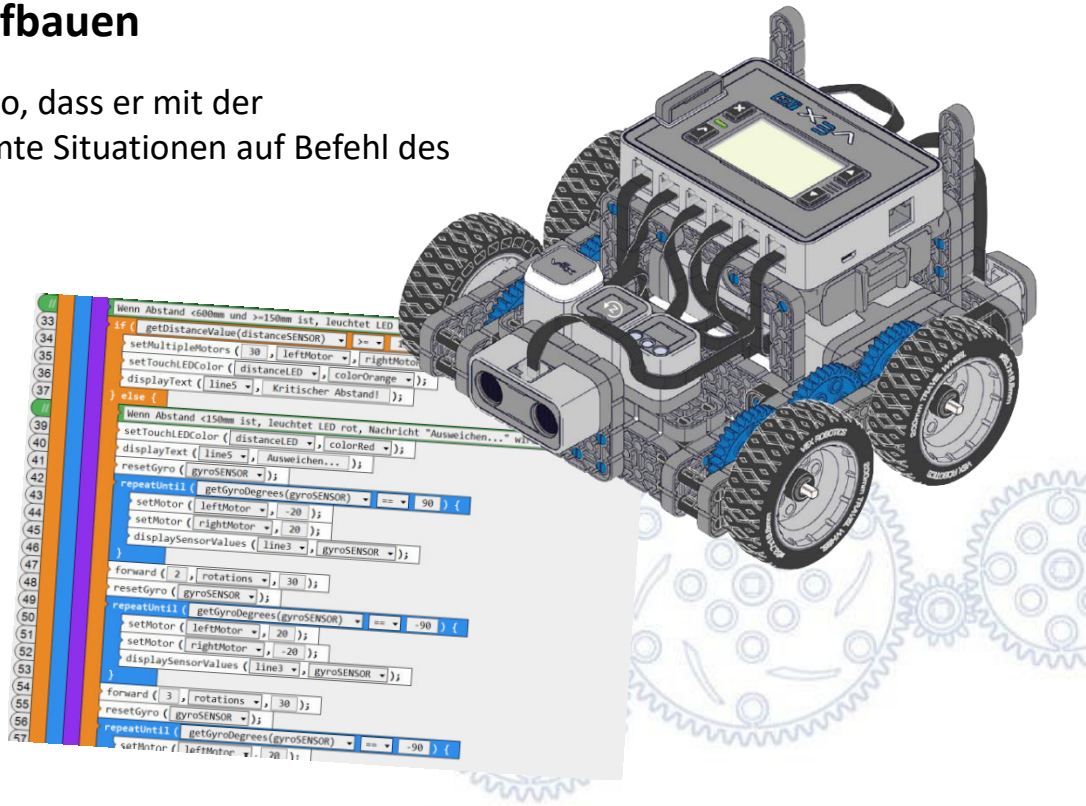
Die Schüler*innen können selbstständig Anforderungen für den Fahrroboter aus den verschiedenen Parcours-Elementen ableiten und erarbeiten daraufhin Lösungen für die Hardware und die Programmierung zum teilautonomen Fahren.

Aufgabe 4: Teilautonomes Fahrzeug aufbauen

Modifiziert und programmiert Euren Fahrroboter so, dass er mit der Fernsteuerung gefahren werden kann und bestimmte Situationen auf Befehl des Fahrers selbstständig bewältigt.

Hier ein paar Ideen für teilautonome Funktionen:

- Automatisches Einparken
- Umfahren eines Hindernisses auf Knopfdruck
- Autonomes Fahren in speziell markierten Bereichen (Line-Tracker)



Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Teilautonomes Fahren

Lösungsansatz: Umfahren eines Hindernisses

```
// Wenn Abstand <600mm und >=150mm ist, leuchtet LED orange und Nachricht "kritischer Abstand!" wird auf dem Display angezeigt
33  if ( getDistanceValue(distanceSENSOR) >= 150 ) {
34    setMultipleMotors ( 30 , leftMotor , rightMotor , noMotor , noMotor );
35    setTouchLEDColor ( distanceLED , colorOrange );
36    displayText ( line5 , Kritischer Abstand! );
37  } else {
// Wenn Abstand <150mm ist, leuchtet LED rot, Nachricht "Ausweichen..." wird auf dem Display angezeigt und alle Motoren werden gestoppt
39    setTouchLEDColor ( distanceLED , colorRed );
40    displayText ( line5 , Ausweichen... );
41    resetGyro ( gyroSENSOR );
42    repeatUntil ( getGyroDegrees(gyroSENSOR) == 90 ) {
43      setMotor ( leftMotor , -20 );
44      setMotor ( rightMotor , 20 );
45      displaySensorValues ( line3 , gyroSENSOR );
46    }
47    forward ( 2 , rotations , 30 );
48    resetGyro ( gyroSENSOR );
49    repeatUntil ( getGyroDegrees(gyroSENSOR) == -90 ) {
50      setMotor ( leftMotor , 20 );
51      setMotor ( rightMotor , -20 );
52      displaySensorValues ( line3 , gyroSENSOR );
53    }
54    forward ( 3 , rotations , 30 );
55    resetGyro ( gyroSENSOR );
56    repeatUntil ( getGyroDegrees(gyroSENSOR) == -90 ) {
57      setMotor ( leftMotor , 20 );
```

...und so weiter.

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Verständnisfragen

Lasst uns gemeinsam folgende Fragen beantworten:

1. Welche Autonomiestufen gibt es?

2. Wie unterscheiden sich „Fahren mit Assistenzsystemen“ und „Teilautonomes Fahren“?

3. Welche weiteren Funktionen fallen Euch zum „Fahren mit Assistenzsystemen“ und zum „Teilautonomes Fahren“ ein?

4. Welche technischen und gesellschaftlichen Herausforderungen könnt Ihr für das autonome Fahren aus Euren Projekten erkennen?

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Zusammenfassung

Lernziel

Die Schüler*innen können:

- die Stufen des autonomen Fahrens wiedergeben und mit Beispielen erläutern
- selbstständig die Schwierigkeit der folgenden Aufgaben festlegen. Sie verstehen die Anforderungen durch die verschiedenen Parcours-Elemente, durchdenken einen Lösungsansatz für den Fahrroboter und schätzen den Aufwand ab
- selbstständig Anforderungen für den Fahrroboter aus den verschiedenen Parcours-Elementen ableiten und erarbeiten daraufhin Lösungen für die Hardware und die Programmierung zum Fahren mit Assistenzsystemen
- selbstständig Anforderungen für den Fahrroboter aus den verschiedenen Parcours-Elementen ableiten und erarbeiten daraufhin Lösungen für die Hardware und die Programmierung zum teilautonomen Fahren

Heute hast du:

- *die Stufen des autonomen Fahrens kennengelernt*
- *selbstständig einen Parcours geplant und aufgebaut*
- *ein Fahrzeug mit Assistenzsystemen und ein teilautonomes Fahrzeug aufgebaut und programmiert*

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Projekt 2

Autonomes Fahren

Projekt 2 – Autonomes Fahren

Unterrichtsinhalte: In dieser Übung lernen die Schüler*innen anhand selbst recherchierter Praxisbeispiele, welche Umwelteinflüsse Auswirkungen auf autonome Fahrzeuge haben und wie bestimmte funktionale Beschränkungen die Anforderungen an das autonome Fahren vereinfachen. Anhand des VEX IQ Fahrroboters sollen die Schüler vorprogrammierte Fahrprogramme analysieren, in geeigneter Form dokumentieren und anschließend dieselbe Funktionalität selbst programmieren.

Themen	Altersgruppe	Dauer
Praxisbeispiele des autonomen Fahrens, Umwelteinflüsse, Funktionsanalyse, Programmablaufplan	12-16	180 min

Lernziele

1. Die Schüler*innen können sich selbstständig mit der aktuellen technischen Umwelt auseinandersetzen und bei Bedarf ihr Wissen per Internetrecherche erweitern.
2. Die Schüler*innen können den Bedarf und die Auswirkungen von bestimmten Umweltbedingungen auf autonome Fahrzeuge anhand eigener Beispiele selbstständig erarbeiten.
3. Die Schüler*innen können beobachtete Abläufe fachgerecht wiedergeben. Ursache und Wirkung können den Sensoren und Aktoren in einem Steuerungssystem zugeordnet werden.
4. Die Schüler*innen können festgelegte Abläufe und Funktionen in ein Programm überführen und können ein komplexes Programm aus mehreren Teilprogrammen zusammensetzen.

Dauer	Aktivitäten	Verbunden mit Lernziel
15 min	Einleitung & Projektbesprechung	1, 2,3, 4
15 min	Recherche zu „Praxisbeispiele für autonomes Fahren“	1
15 min	Umweltbedingungen und Begrenzungen des autonomen Fahrens	1, 2
45 min	VEX IQ Autopilot aufbauen	(3)
20 min	Analyse & Dokumentation der autonomen Fahrprogramme	3
60 min	Programmierung autonomer Fahrprogramme	4
10 min	Reflexion Projektes anhand offener Fragen	1, 2, 3, 4

Materialien

PowerPoint Präsentation Projekt 2
Arbeitsblatt “Autonomes Fahren”
VEX IQ Super Kit
Computer mit RobotC Programmierumgebung
Unterrichtsraum mit freier Fläche für autonome Roboterfahrten

BY NC SA
Erstellt von: Dreibein Lehrsysteme GmbH

Name(n): _____

Lehrer/Klasse: _____

Datum: _____

Aufgabe 1: Praxisbeispiele für Autonomes Fahren

Überlege, in welchen Bereichen oder Anwendungen es bereits autonome Fahrzeuge oder autonom bewegte Geräte gibt.

Nenne mindestens vier Beispiele.

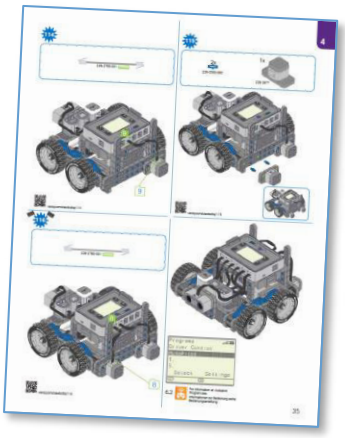
Nutze bei Bedarf das Internet, um passende Beispiele zu finden.

Aufgabe 2: Umweltbedingungen und Begrenzungen des autonomen Fahrens

Überlege, welche besonderen Umweltbedingungen bei deinen verschiedenen Beispielen vorhanden sind.

Wie und warum ist dabei das autonome Fahren begrenzt bzw. eingeschränkt?

Aufgabe 3: Analysiere die autonomen Fahrprogramme



Baue den Autopilot Fahrroboter, wie in der Montageanleitung beschrieben (Kapitel 1 und 4; Arbeitsschritte 1-19 und 102-117) auf und starte im VEX IQ Brain das Autopilot-Programm.

Beobachte und beschreibe das Verhalten des Autopilot Fahr-roboters. Gehe dabei auf äußere Einflüsse ein und erläutere, wie diese das Verhalten des Fahrroboters verändern.

Hinweis: mit dem Farbsensor und der Touch-LED lassen sich verschiedene Fahrprogramme einstellen.

Wähle geeignete Darstellungsformen für Deine Beschreibung.

Aufgabe 4: Programmiere die autonomen Fahrprogramme selbst

Wähle eins der in Aufgabe 3 analysierten autonomen Fahrprogramme und programmiere dieses selbstständig in RobotC nach.

Stimme Dich mit Deinen Mitschülern ab, sodass jedes der drei Fahrprogramme mindestens einmal programmiert wird.

Vergleiche das Verhalten Deiner Programmierung mit dem vor-programmierten Fahrprogramm und verbessere ggf. Dein Programm.

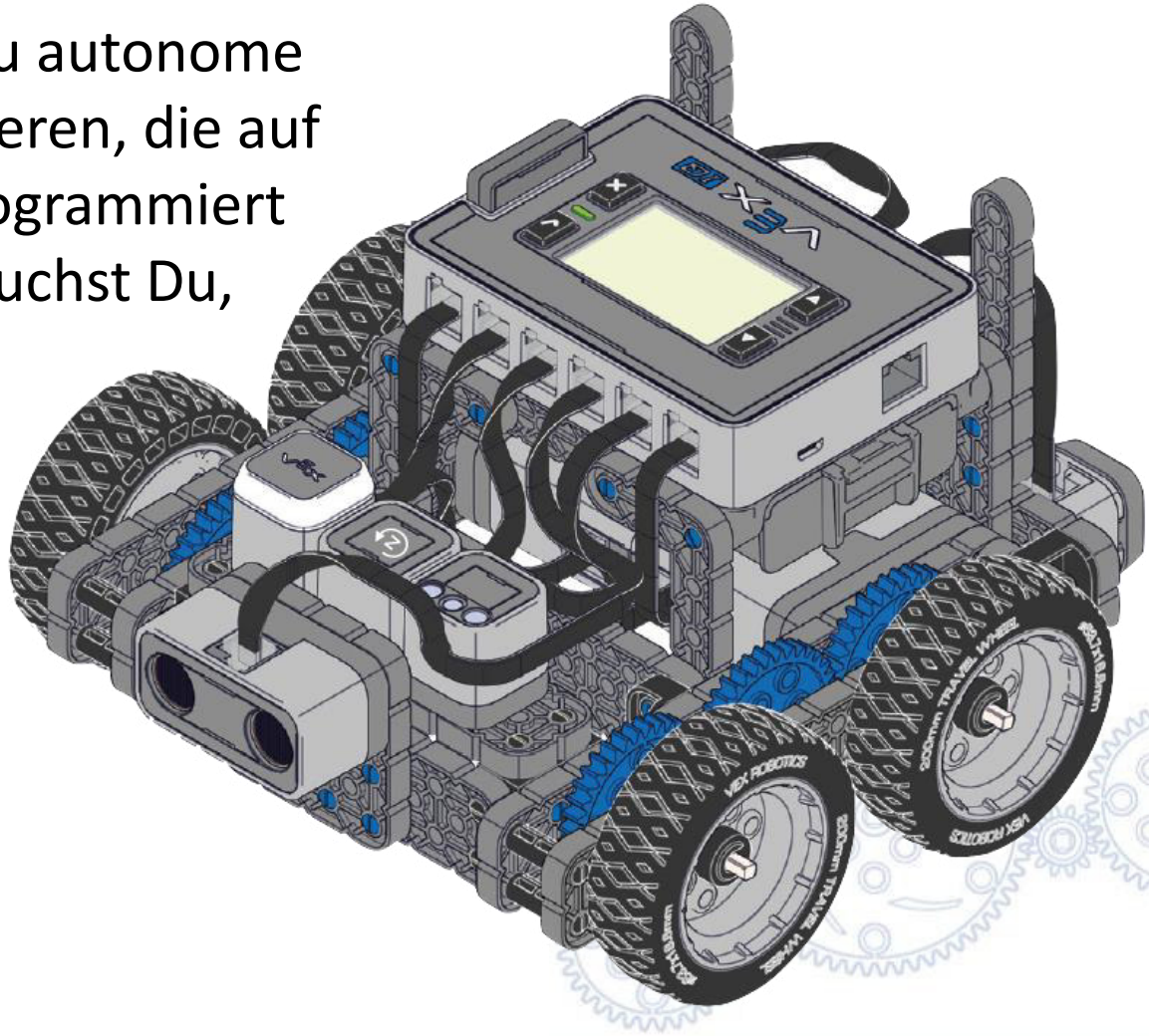
Versuche abschließend zusammen mit Deinen Mitschülern die drei Fahrprogramme in ein übergeordnetes Programm zu integrieren.

BY NC SA

Erstellt von: Dreibein Lehrsysteme GmbH

Autonomes Fahren

In dieser Übung wirst Du autonome Fahrprogramme analysieren, die auf dem Fahrroboter vorprogrammiert sind. Anschließend versuchst Du, diese Fahrprogramme selbst zu programmieren, um ein ähnliches Fahrverhalten zu erreichen.



Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Autonomes Fahren

Lernziel

Die Schüler*innen können sich selbstständig mit der aktuellen technischen Umwelt auseinandersetzen und bei Bedarf ihr Wissen per Internetrecherche erweitern.

Aufgabe 1: Praxisbeispiele für Autonomes Fahren

Überlege, in welchen Bereichen oder Anwendungen es bereits autonome Fahrzeuge oder autonom bewegte Geräte gibt. Nenne mindestens vier Beispiele. Nutze bei Bedarf das Internet, um passende Beispiele zu finden.

Lösungsvorschläge:

- Testfahrzeuge im Straßenverkehr (Autonomiestufe 3 & 4)
- Testfahrzeuge in abgegrenzten Bereichen (Autonomiestufe 5)
- Züge, U-Bahnen
- Transportroboter in der industriellen Fertigung
- Rasenmäher-Roboter

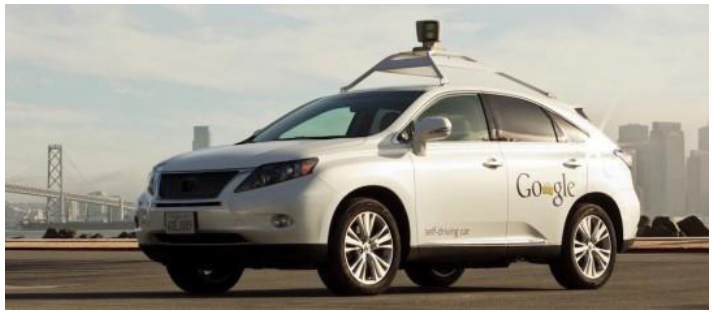
Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Autonomes Fahren - Praxisbeispiele

Testfahrzeuge im Straßenverkehr (Autonomiestufe 3 & 4)

Quelle: Waymo



Quelle: Google

Quelle: Ford

CC BY NC SA

Autonomes Fahren - Praxisbeispiele

Testfahrzeuge in abgegrenzten Bereichen (Autonomiestufe 5)

Vollautonomer Bus mit max. 9 km/h auf dem EUREF Campus Berlin

Quelle: Deutsche Bahn AG

Vollautonomer Bus für 15 Personen auf den Charité Campi Berlin Mitte und Wedding

Quelle: BVG

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Autonomes Fahren - Praxisbeispiele

Züge, U-Bahnen

Fahrerlose U-Bahn in Nürnberg

Quelle: VAG Nürnberg

Fahrerlose Skytrain in Bangkok

Quelle: Siemens

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Autonomes Fahren - Praxisbeispiele

Transportroboter in der industriellen Fertigung

Quelle: Audi

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Autonomes Fahren - Praxisbeispiele

Rasenmäher-Roboter

Quelle: Gardena

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Autonomes Fahren

Lernziel

Die Schüler*innen können den Bedarf und die Auswirkungen von bestimmten Umweltbedingungen auf Autonome Fahrzeuge anhand eigener Beispiele selbstständig erarbeiten.

Aufgabe 2: **Umweltbedingungen und Begrenzungen des autonomen Fahrens**

Überlege, welche besonderen Umweltbedingungen bei deinen verschiedenen Beispielen vorhanden sind.
Wie und warum ist dabei das autonome Fahren begrenzt bzw. eingeschränkt?

Lösungsvorschläge:

- Umweltbedingungen:
  - exklusiver Verkehrsraum (fahrerlose Schienenfahrzeuge)
  - bekannter & beruhigter Verkehrsraum (fahrerlose Busse auf festgelegten Strecken, Transportroboter)
  - Verkehrsraum mit eingebauten Leitsystemen (Transportroboter)
- Begrenzungen & Einschränkungen des autonomen Fahrens
  - begrenzte Geschwindigkeit
  - nur unter Aufsicht einer Person
  - nur in festgelegten Verkehrsräumen

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Fahrprogramme analysieren

Lernziel

Die Schüler*innen können beobachtete Abläufe fachgerecht wiedergeben. Ursache und Wirkung können den Sensoren und Aktoren in einem Steuerungssystem zugeordnet werden.

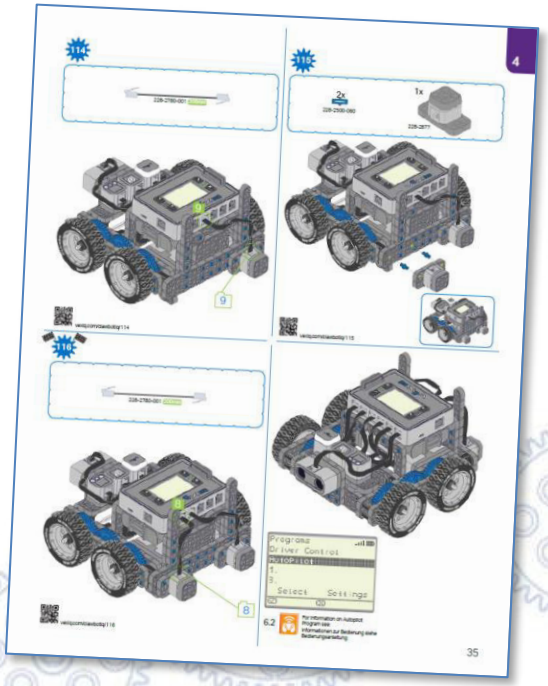
Aufgabe 3: Analysiere die autonomen Fahrprogramme

Baue den Autopilot Fahrroboter, wie in der Montageanleitung beschrieben (Kapitel 1 und 4; Arbeitsschritte 1-19 und 102-117) auf und starte im VEX IQ Brain das Autopilot-Programm.

Beobachte und beschreibe das Verhalten des Autopilot Fahrroboters. Gehe dabei auf äußere Einflüsse ein und erläutere, wie diese das Verhalten des Fahrroboters verändern.

Hinweis: mit dem Farbsensor und der Touch-LED lassen sich verschiedene Fahrprogramme einstellen.

Wähle geeignete Darstellungsformen für Deine Beschreibung.

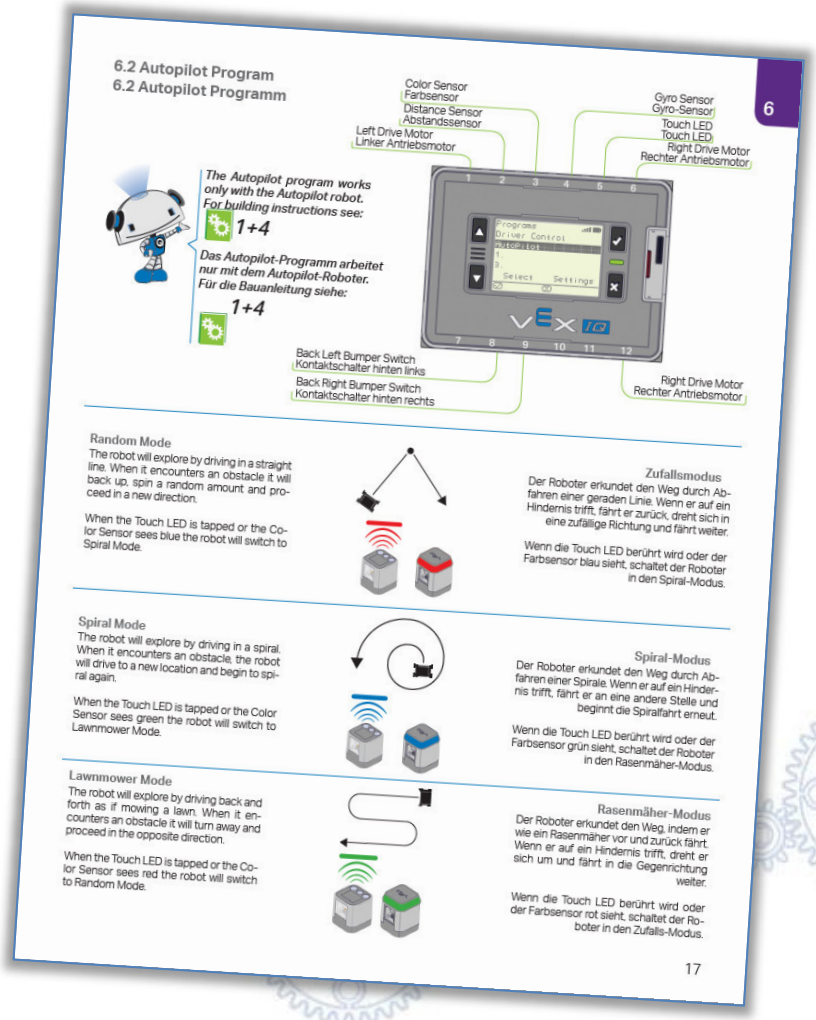


Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Fahrprogramme analysieren

Die Beschreibung der Autopilot-Programme findest du in der Bedienungsanleitung auf Seite 17.



6.2 Autopilot Program
6.2 Autopilot Programm

6

Color Sensor
Farbsensor
Distance Sensor
Abstandssensor
Left Drive Motor
Linker Antriebsmotor
Gyro Sensor
Gyro-Sensor
Touch LED
Touch LED
Right Drive Motor
Rechter Antriebsmotor

The Autopilot program works only with the Autopilot robot. For building instructions see: 1+4

Das Autopilot-Programm arbeitet nur mit dem Autopilot-Roboter. Für die Bauanleitung siehe: 1+4

Back Left Bumper Switch
Kontaktschalter hinten links
Back Right Bumper Switch
Kontaktschalter hinten rechts
Right Drive Motor
Rechter Antriebsmotor

Random Mode
The robot will explore by driving in a straight line. When it encounters an obstacle it will back up, spin a random amount and proceed in a new direction.

When the Touch LED is tapped or the Color Sensor sees blue the robot will switch to Spiral Mode.

Zufallsmodus
Der Roboter erkundet den Weg durch Abfahren einer geraden Linie. Wenn er auf ein Hindernis trifft, fährt er zurück, dreht sich in eine zufällige Richtung und fährt weiter.

Wenn die Touch LED berührt wird oder der Farbsensor blau sieht, schaltet der Roboter in den Spiral-Modus.

Spiral Mode
The robot will explore by driving in a spiral. When it encounters an obstacle, the robot will drive to a new location and begin to spiral again.

When the Touch LED is tapped or the Color Sensor sees green the robot will switch to Lawnmower Mode.

Spiral-Modus
Der Roboter erkundet den Weg durch Abfahren einer Spirale. Wenn er auf ein Hindernis trifft, fährt er an eine andere Stelle und beginnt die Spiralfahrt erneut.

Wenn die Touch LED berührt wird oder der Farbsensor grün sieht, schaltet der Roboter in den Rasenmäher-Modus.

Lawnmower Mode
The robot will explore by driving back and forth as if mowing a lawn. When it encounters an obstacle it will turn away and proceed in the opposite direction.

When the Touch LED is tapped or the Color Sensor sees red the robot will switch to Random Mode.

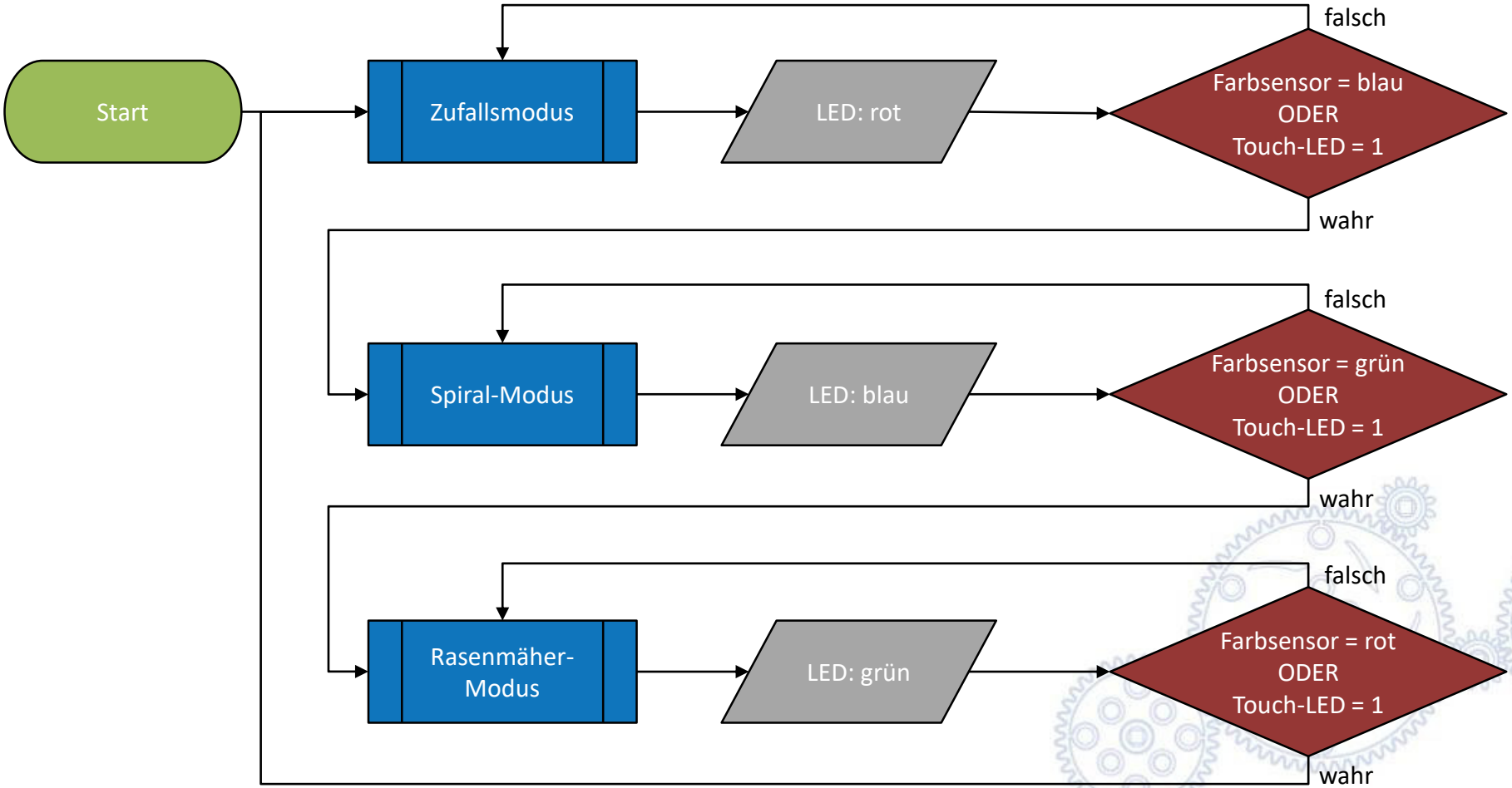
Rasenmäher-Modus
Der Roboter erkundet den Weg, indem er wie ein Rasenmäher vor und zurück fährt. Wenn er auf ein Hindernis trifft, dreht er sich um und fährt in die Gegenrichtung weiter.

Wenn die Touch LED berührt wird oder der Farbsensor rot sieht, schaltet der Roboter in den Zufalls-Modus.

17

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

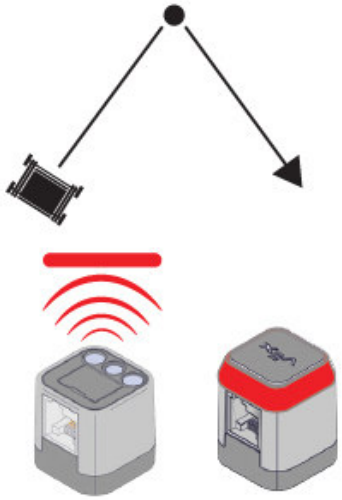
Programmablaufplan Fahrprogramme



Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Zufallsmodus

Zufallsmodus



Zufallsmodus

Der Roboter erkundet den Weg durch Abfahren einer geraden Linie. Wenn er auf ein Hindernis trifft, fährt er zurück, dreht sich in eine zufällige Richtung und fährt weiter.

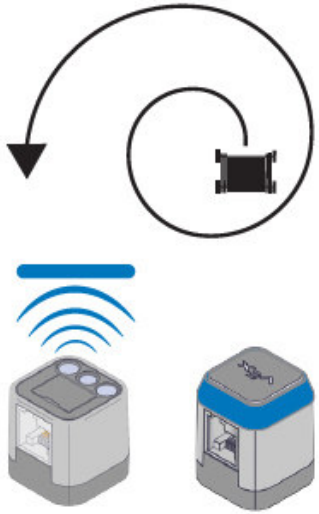
Wenn die Touch LED berührt wird oder der Farbsensor blau sieht, schaltet der Roboter in den Spiral-Modus.

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Spiral-Modus

Spiral-Modus



Spiral-Modus

Der Roboter erkundet den Weg durch Abfahren einer Spirale. Wenn er auf ein Hindernis trifft, fährt er an eine andere Stelle und beginnt die Spiralfahrt erneut.

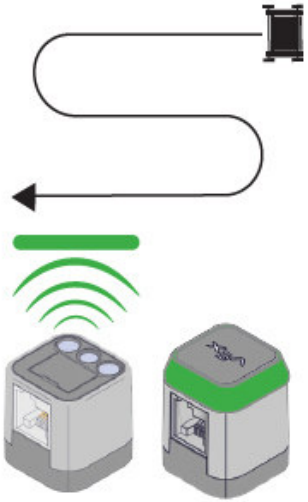
Wenn die Touch LED berührt wird oder der Farbsensor grün sieht, schaltet der Roboter in den Rasenmäher-Modus.

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Rasenmäher-Modus

Rasenmäher-Modus



Rasenmäher-Modus

Der Roboter erkundet den Weg, indem er wie ein Rasenmäher vor und zurück fährt. Wenn er auf ein Hindernis trifft, dreht er sich um und fährt in die Gegenrichtung weiter.

Wenn die Touch LED berührt wird oder der Farbsensor rot sieht, schaltet der Roboter in den Zufalls-Modus.

Erstellt von: Dreibein Lehrsysteme GmbH

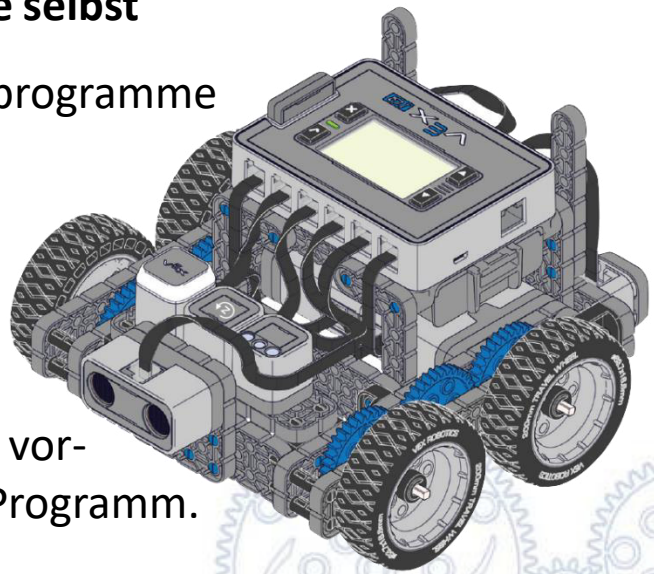
CC BY NC SA

Fahrprogramme selbst programmieren

Lernziel

Die Schüler*innen können festgelegte Abläufe und Funktionen in ein Programm überführen und können ein komplexes Programm aus mehreren Teilprogrammen zusammensetzen

Aufgabe 4: Programmiere die autonomen Fahrprogramme selbst



Wähle eins der in Aufgabe 3 analysierten autonomen Fahrprogramme und programmiere dieses selbstständig in RobotC nach.

Stimme Dich mit Deinen Mitschülern ab, sodass jedes der drei Fahrprogramme mindestens einmal programmiert wird.

Vergleiche das Verhalten Deiner Programmierung mit dem vor-programmierten Fahrprogramm und verbessere ggf. Dein Programm.

Versuche abschließend zusammen mit Deinen Mitschülern die drei Fahrprogramme in ein übergeordnetes Programm zu integrieren.

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Verständnisfragen

Lasst uns gemeinsam folgende Fragen beantworten:

1. Welche realen Beispiele gibt es für das autonome Fahren?
2. Welche Umweltbedingungen haben wesentlichen Einfluss auf das autonome Fahren?
3. Welche Beschränkungen können die Anforderungen an das autonome Fahren vereinfachen?
4. Wie lassen sich Abläufe und Funktionen eindeutig darstellen?

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Zusammenfassung

Lernziel

Die Schüler*innen können:

- sich selbstständig mit der aktuellen technischen Umwelt auseinandersetzen und bei Bedarf ihr Wissen per Internetrecherche erweitern
- den Bedarf und die Auswirkungen von bestimmten Umweltbedingungen auf autonome Fahrzeuge anhand eigener Beispiele selbstständig erarbeiten
- beobachtete Abläufe fachgerecht wiedergeben. Ursache und Wirkung können den Sensoren und Aktoren in einem Steuerungssystem zugeordnet werden.
- festgelegte Abläufe und Funktionen in ein Programm überführen und können ein komplexes Programm aus mehreren Teilprogrammen zusammensetzen

Heute hast du:

- *praktische Beispiele für das autonome Fahren kennengelernt*
- *den Bedarf und die Auswirkungen von Umweltbedingungen auf autonome Fahrzeuge erarbeitet*
- *vorhandene Fahrprogramme analysiert, dokumentiert und selbst programmiert*

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Projekt 3

Fahrzeug mit Lenkgetriebe

Projekt 3 – Fahrzeug mit Lenkgetriebe

Unterrichtsinhalte: In dieser Übung lernen die Schüler*innen anhand selbst recherchierter Informationen, welche Arten von Lenkungen es gibt und wie bestimmte Lenkungen aufgebaut sind. Mit dem VEX IQ Bausatz wird ein Straßenfahrzeug mit Achsschenkellenkung aufgebaut und die Lenkung programmiert. Darauf aufsetzend wird eine automatische Einparkfunktion in die Programmierung integriert. Optional können die Schüler*innen Verbesserungspotentiale beim Fahrzeug aufzeigen und anschließend optimieren. Die Erstellung eines CAD-Modells mit der Software SnapCAD ist eine weitere optionale Aufgabe

Themen	Altersgruppe	Dauer
Lenkung, Lenkarten, Lenkgetriebe, Achsschenkellenkung, Park-Pilot, automatisches Einparken, Optimierung, Verbesserung, Anforderungen, CAD, SnapCAD, Baugruppe, Montageanleitung erstellen	12-16	3 – 10 h

Lernziele

1. Die Schüler*innen können sich selbstständig mit einem komplexen mechanischen Aufbau auseinandersetzen und ihr Wissen per Internetrecherche oder alternativen Quellen erweitern.
2. Die Schüler*innen können den zuvor theoretisch erarbeiteten Aufbau der Achsschenkellenkung anhand des VEX IQ Bausatzes in ein funktionsfähiges Modell überführen.
3. Die Schüler*innen können zwei mechanische Baugruppen aneinander anpassen, sodass eine neue Funktionseinheit entsteht. Darüber hinaus können die Schüler*innen eine Ansteuerung des Radeinschlages per Joystick-Fernbedienung programmieren.
4. Die Schüler*innen können von der neu zu entwickelnden Produktfunktion „Park-Pilot“ technische Anforderungen ableiten und diese durch Hard- und Softwareanpassungen realisieren.
5. Die Schüler*innen können technische Funktionen analysieren und Verbesserungspotenzial aufzeigen. Sie können darüber hinaus Optimierungen umsetzen und den Erfolg bewerten.
6. Die Schüler*innen können mit einem CAD-Programm umgehen und einen realen Aufbau in ein CAD-Modell überführen. Darüber hinaus können sie in Baugruppen arbeiten und eine Montageanleitung mithilfe des Programms SnapCAD erstellen.

Dauer	Aktivitäten	Verbunden mit Lernziel
20 min	Einleitung & Projektbesprechung	1 - 6
20 min	Aufgabe 1: Lenkungsarten	1
20 min	Aufgabe 2: Aufbau einer Lenkung	1
1 h	Aufgabe 3: Mechanisches Modell einer Achsschenkellenkung	1, 2
1 h	Aufgabe 4: Integration der Lenkbaugruppe	1, 2, 3
1 h	Aufgabe 5: Park-Pilot	2, 4
1 – 3 h	Aufgabe 6: Optimierung des Fahrzeugs	1, 2, 3, 4, 5
1 – 3 h	Aufgabe 7: Erstellen eines CAD-Modells	6

Materialien

PowerPoint Präsentation Projekt 3
Arbeitsblatt “Fahrzeug mit Lenkgetriebe”
VEX IQ Super Kit
Ggf. weitere VEX IQ Bauteile
Computer mit RobotC Programmierumgebung und SnapCAD CAD-Software
Unterrichtsraum mit freier Fläche für Roboterfahrten

BY NC SA
Erstellt von: Dreibein Lehrsysteme GmbH

Name(n): _____

Lehrer/Klasse: _____

Datum: _____

Aufgabe 1: Lenkungsarten

Überlege und recherchiere, welche verschiedenen Möglichkeiten zum Lenken von Landfahrzeugen es gibt.

Nenne mindestens 3 verschiedene Lenkungsarten!

Nutze bei Bedarf das Internet, um passende Beispiele zu finden.

Aufgabe 2: Aufbau einer Lenkung

Vergleiche die Lenkung des bisher verwendeten Fahrroboters („Standard-Fahrwerk“ aus der VEX IQ Montageanleitung Seiten 4 bis 9) mit der Lenkung eines normalen PKWs.

Erkläre jeweils die Funktionsweise und Hauptbestandteile!

Nenne Vor- und Nachteile!

Aufgabe 3: Mechanisches Modell einer Achsschenkellenkung

Baue mit den vorhandenen Bauteilen aus dem VEX IQ Bausatz eine Achsschenkellenkung auf!

Die Baugruppe soll zwei freilaufende Räder besitzen, einen Lenkantrieb (Motor) und später an den Fahrroboter angebaut werden können.

Aufgabe 4: Integriere die Lenkbaugruppe in den Fahrroboter

Modifiziere den bisher verwendeten Fahrroboter („Standard-Fahrwerk“ aus der VEX IQ Montageanleitung Seiten 4 bis 9) mit Panzerlenkung so, dass Du die gerade erstellte Baugruppe mit Achsschenkellenkung einbauen kannst.

Aufgabe 5: Park-Pilot

Das Fahrzeug soll als neue Funktion den Park-Pilot bekommen. Das heißt, das Fahrzeug soll auf Knopfdruck selbstständig parallel zur Fahrbahn einparken.

Hinweise:

- Überlege, wie das Einparken Schritt für Schritt abläuft
- Führe den Ablauf mit der Fernbedienung durch
- Wie wird das automatische Einparken gestartet?
- Werden Sensoren benötigt?
- Welche Sensoren können Abstände erkennen?

BY NC SA

Aufgabe 6: Optimiere Dein Fahrzeug

Wenn die Lenkung in Dein Fahrzeug eingebaut ist und Du damit herumfahren kannst, werden Dir bald einige Dinge auffallen, die noch nicht optimal funktionieren.

Nenne konkrete Probleme und überlege Dir daraufhin Anpassungen für Dein Fahrzeug, die Verbesserungen darstellen könnten. Halte Deine Optimierungsansätze schriftlich fest.

Setze die Änderungen Schritt für Schritt in Deinem Fahrzeug um und dokumentiere anschließend, ob Dein Optimierungsansatz zu einer Verbesserung geführt hat.

Aufgabe 7: Erstelle ein CAD-Modell

Das aufgebaute Fahrzeug kann man auf dem Computer als CAD-Modell nachbauen. Die CAD-Datei kann dann einfach mit anderen geteilt und gemeinsam weitergearbeitet werden oder es auch Abbildungen erstellt werden.

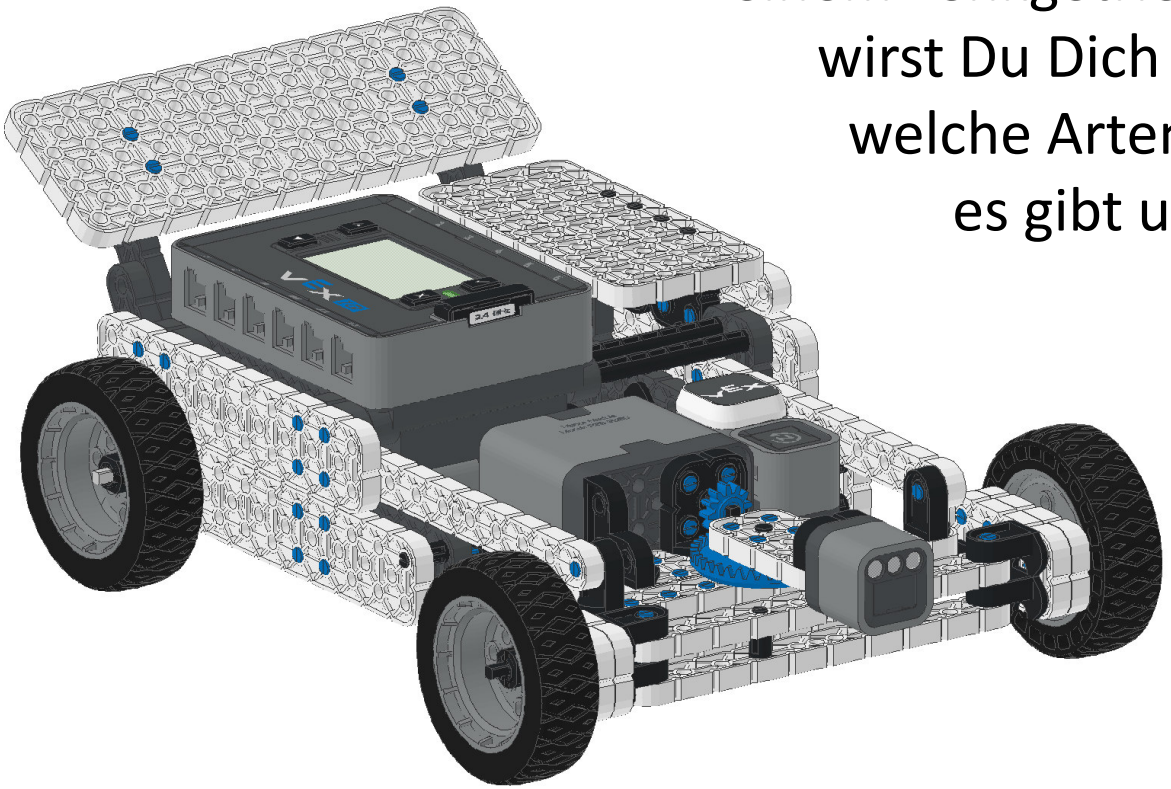
Verwende die Software SnapCAD, um ein CAD-Modell Deines Fahrzeugs zu erstellen!

Hinweis
SnapCAD ist eine kostenfreie Software. Damit kannst Du dein CAD-Modell auch zu Hause weiterbearbeiten.
https://www.vexrobotics.com/vexiq/resources/cad-snapcad

BY NC SA

Fahrzeug mit Lenkgetriebe

In dieser Übung wirst Du ein Fahrzeug mit einem Lenkgetriebe aufbauen. Dabei wirst Du Dich damit beschäftigen, welche Arten von Lenkgetrieben es gibt und wie man diese in den Fahrroboter einbauen kann.



Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Lenkungsarten

Lernziel

Die Schüler*innen können sich selbstständig mit einem komplexen mechanischen Aufbau auseinandersetzen und ihr Wissen per Recherche im Internet oder alternativen Quellen erweitern.

Aufgabe 1: Lenkungsarten

Überlege und recherchiere, welche verschiedenen Möglichkeiten zum Lenken von Landfahrzeugen es gibt. Nenne mindestens 3 verschiedene Lenkungsarten!

Lösungsvorschläge:

- Raupen- oder Panzerlenkung
- Achsschenkellenkung
- Schwenkachs- oder Drehschemellenkung
- Knicklenkung
- Einzelradlenkung
- Schienenlenkung

Siehe z.B.: https://de.wikipedia.org/wiki/Lenkung

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Aufbau einer Lenkung

Aufgabe 2: Aufbau einer Lenkung

Vergleiche die Lenkung des bisher verwendeten Fahrroboters („Standard-Fahrwerk“ aus der VEX IQ Montageanleitung Seiten 4 bis 9) mit der Lenkung eines normalen PKWs.
Erkläre jeweils die Funktionsweise und Hauptbestandteile!
Nenne Vor- und Nachteile!

Lösungsvorschläge:

Der bisher verwendete Fahrroboter besitzt eine Panzerlenkung, bei der die beiden Seiten des Fahrzeugs unterschiedlich schnell angetrieben werden. Je größer der Geschwindigkeitsunterschied beider Seiten, desto stärker die Lenkwirkung.
Hauptbestandteile: 1 separat steuerbarer Antrieb pro Fahrzeugseite
Vorteile: einfacher Aufbau, auf der Stelle drehen
Nachteile: hoher Verschleiß, schwierig bei hohen Geschwindigkeiten, 2 Antriebsmotoren notwendig

Ein normaler PKW besitzt eine Achsschenkellenkung, bei der zwei schwenkbare Räder einer Achse über eine gemeinsame Spurstange gelenkt werden. Je größer der Radeinschlag, desto stärker die Lenkwirkung.
Hauptbestandteile: Achsschenkel, Aufhängung, Spurstange, Lenkgetriebe, Lenksäule, Lenkrad
Vorteile: Spurstabilität, geschwindigkeitsunabhängig, verschleißarm
Nachteile: Aufwendiger Aufbau, begrenzter Wendekreis

Siehe z.B.: https://de.wikipedia.org/wiki/Lenkung

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Achsschenkellenkung

Lernziel

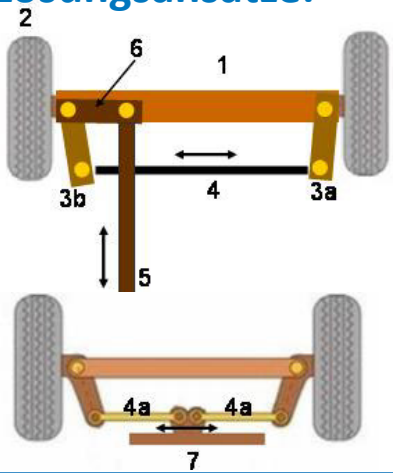
Die Schüler*innen können den zuvor theoretisch erarbeiteten Aufbau der Achsschenkellenkung anhand des VEX IQ Bausatzes in ein funktionsfähiges Modell überführen.

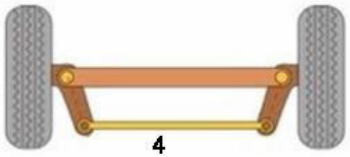
Aufgabe 3: **Mechanisches Modell einer Achsschenkellenkung**

Baue mit den vorhandenen Bauteilen aus dem VEX IQ Bausatz eine Achsschenkellenkung auf!

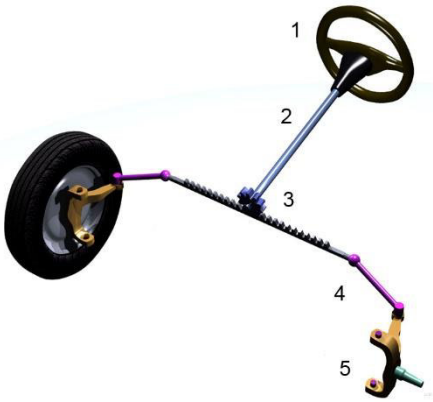
Die Baugruppe soll zwei freilaufende Räder besitzen, einen Lenkantrieb (Motor) und später an den Fahrroboter angebaut werden können.

Lösungsansätze:





1 Achskörper / Rahmen / Aufbau
2 Rad
3 Spurhebel
4 Spurstange / 4a geteilte Spurstange
5 Lenkstange
6 Lenkhebel
7 Lenkgetriebe

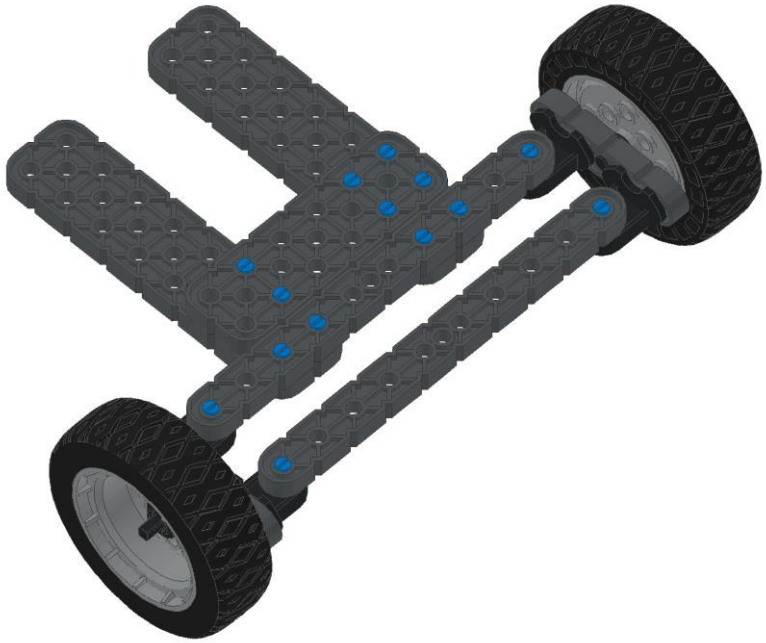


Zahnstangenlenkung
1 Lenkrad
2 Lenkstange
3 Ritzel und Zahnstange
4 Spurstangen
5 Radträger

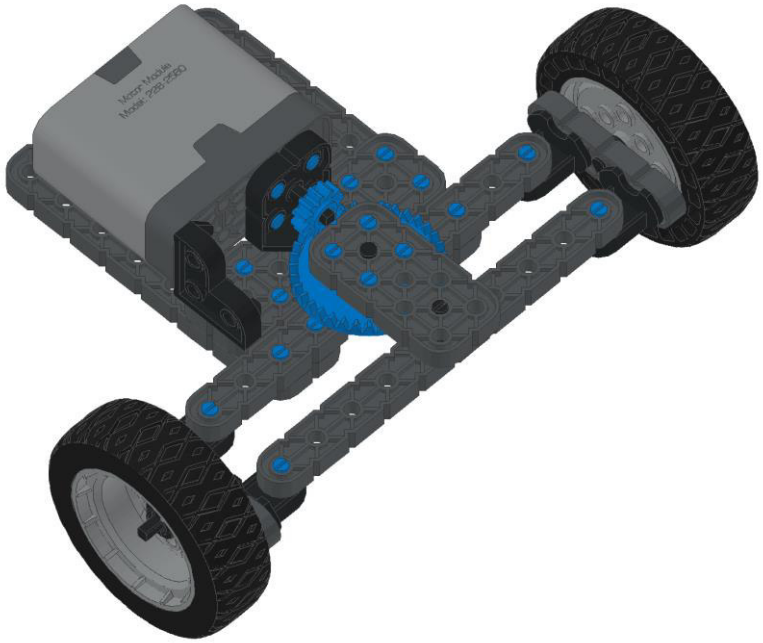
Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Lösungsvorschlag

1. Schritt: Achsschenkellenkung ohne Lenkgetriebe



2. Schritt: Lenkgetriebe und Motor



Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Aufbau Straßenfahrzeug

Lernziel

Die Schüler*innen können zwei mechanische Baugruppen aneinander anpassen, sodass eine neue Funktionseinheit entsteht. Darüber hinaus können die Schüler*innen eine Ansteuerung des Radeinschlages per Joystick-Fernbedienung programmieren.

Aufgabe 4: **Integriere die Lenkbaugruppe in den Fahrroboter**

Modifiziere den bisher verwendeten Fahrroboter („Standard-Fahrwerk“ aus der VEX IQ Montageanleitung Seiten 4 bis 9) mit Panzerlenkung so, dass Du die gerade erstellte Baugruppe mit Achsschenkellenkung einbauen kannst.

Hinweise:

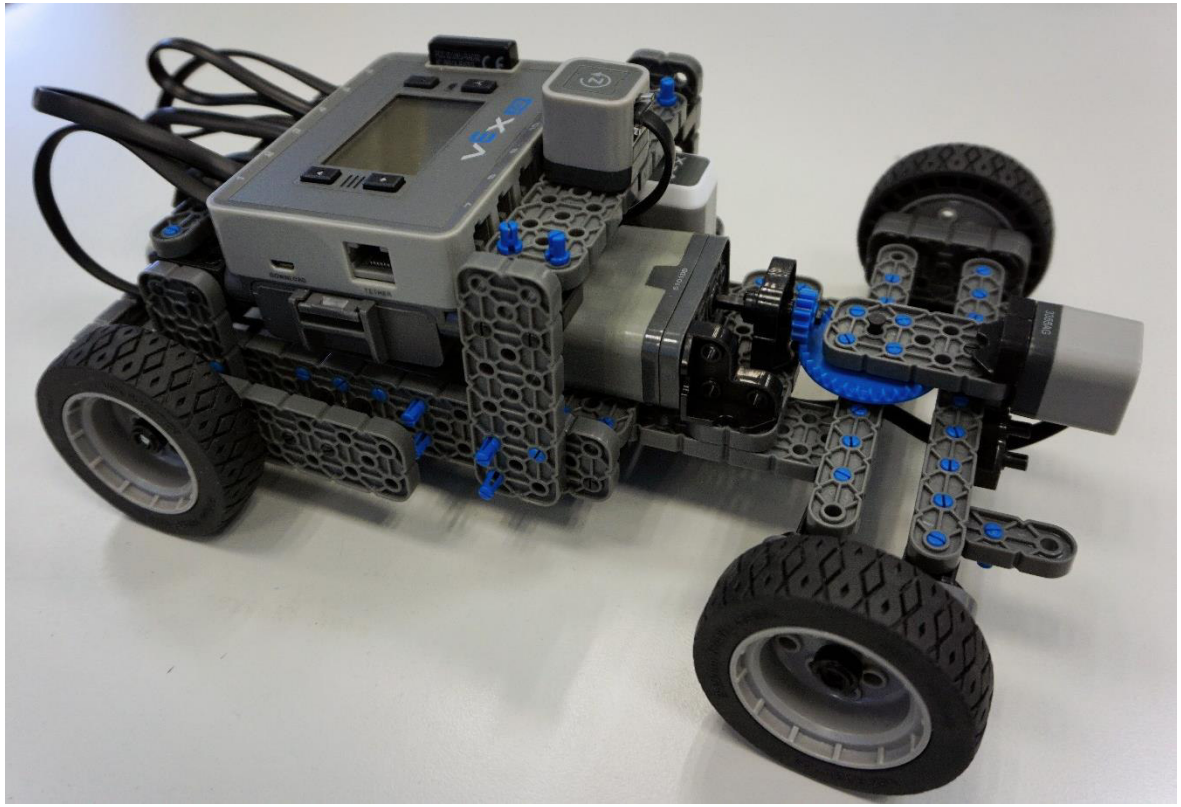
Überlege zuerst, welche Bauteile überflüssig sind.

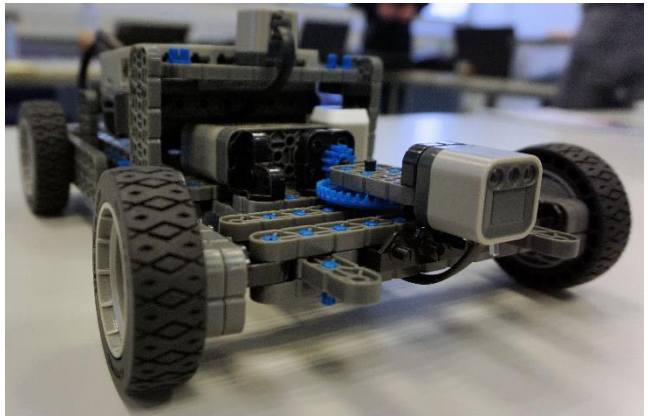
Beachte, dass die Ansteuerung des Lenkmotors über die Fernsteuerung von Dir programmiert werden muss.

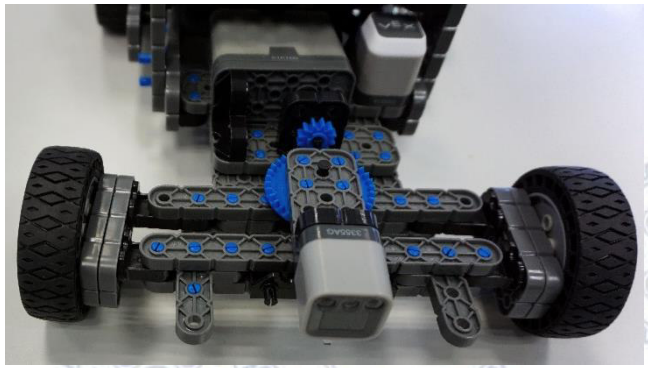
Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Lösungsvorschlag – mechanisch







Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Lösungsvorschlag – Programmierung

```
1  repeat (forever ) {
2      Antrieb_Soll = getJoystickValue(ChA);
3      Lenkung_Soll = getJoystickValue(ChC);
4      Motorstellung = getMotorEncoder(Lenkung);
5      Motorstellung1 = Motorstellung + 20;
6      Motorstellung2 = Motorstellung - 20;
7      setMotor ( Traktion , Antrieb_Soll );
8      if ( Lenkung_Soll >= Motorstellung1 ) {
9          setMotor ( Lenkung , 50 );
10     }
11     if ( Lenkung_Soll <= Motorstellung2 ) {
12         setMotor ( Lenkung , -50 );
13     }
14     wait ( 2 , milliseconds );
15     stopMotor ( Lenkung );
16 }
17
```


Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Park-Pilot

Lernziel

Die Schüler*innen können von der neu zu entwickelnden Produktfunktion „Park-Pilot“ technische Anforderungen ableiten und diese durch Hard- und Softwareanpassungen realisieren.

Aufgabe 5: Park-Pilot

Das Fahrzeug soll als neue Funktion den Park-Pilot bekommen. Das heißt, das Fahrzeug soll auf Knopfdruck selbstständig parallel zur Fahrbahn einparken.

Hinweise:

- Überlege, wie das Einparken Schritt für Schritt abläuft
- Führe den Ablauf mit der Fernbedienung durch
- Wie wird das automatische Einparken gestartet?
- Werden Sensoren benötigt?
- Welche Sensoren können Abstände erkennen?

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

Lösungsansätze Park-Pilot

Folgende Punkte können beim Erstellen des Park-Pilot erfolgreich sein:

- Die Parklücke ausreichend groß wählen
- Das Fahrzeug per Fernsteuerung an eine passende Startposition fahren
- Das automatische Einparken kann mit oder ohne Sensoren programmiert werden
- Ohne Sensoren ist die Fahrstrecke beim jeweiligen Radeinschlag relevant
- Als Abstandssensor kann sowohl der Ultraschallsensor, als auch der Lichtsensor verwendet werden

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

Weiterentwicklung

Lernziel

Die Schüler*innen können technische Funktionen analysieren und Verbesserungspotenzial aufzeigen. Sie können darüber hinaus Optimierungen umsetzen und den Erfolg bewerten.

Aufgabe 6: Optimiere Dein Fahrzeug

Wenn die Lenkung in Dein Fahrzeug eingebaut ist und Du damit herumfahren kannst, werden Dir bald einige Dinge auffallen, die noch nicht optimal funktionieren.

Nenne konkrete Probleme und überlege Dir daraufhin Anpassungen für Dein Fahrzeug, die Verbesserungen darstellen könnten. Halte Deine Optimierungsansätze schriftlich fest.

Setze die Änderungen Schritt für Schritt in Deinem Fahrzeug um und dokumentiere anschließend, ob Dein Optimierungsansatz zu einer Verbesserung geführt hat.

Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

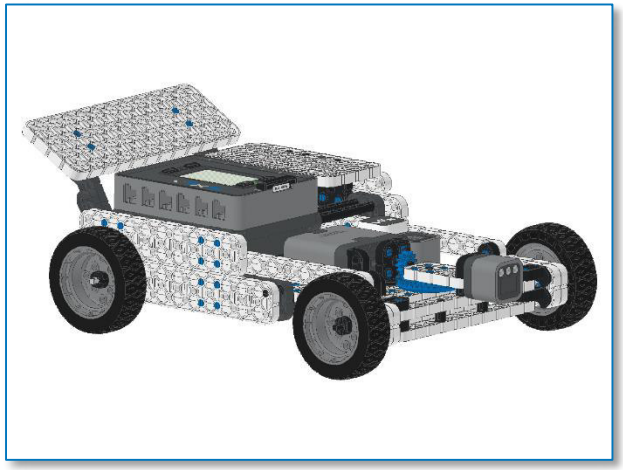
Lösungsansätze

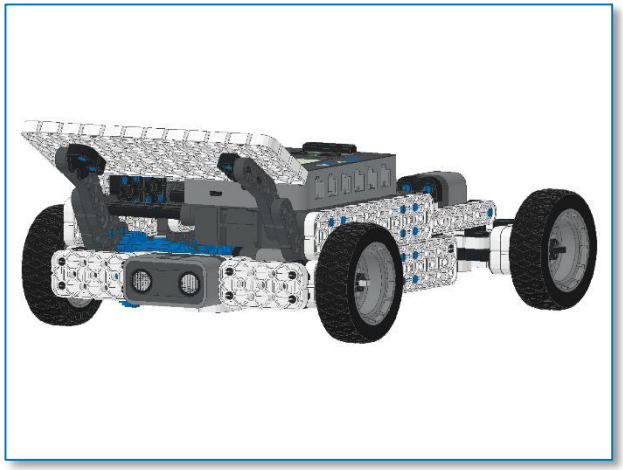
Folgende Punkte könnten Ansätze für Optimierungen am Fahrzeug sein:

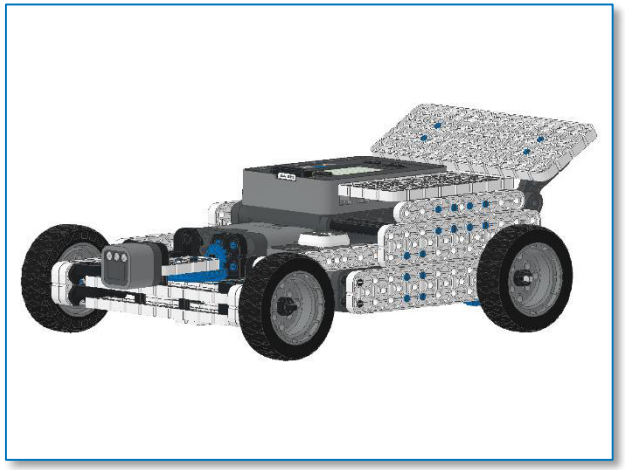
- Stabilität der Lenkbaugruppe
  Lösung: Versteifung der Achsschenkel, Aufhängung und Spurstange durch doppelte Träger
- Antriebszahnräder rutschen durch
  Lösung: stabilere Aufnahmen der Zahnräder, Wellen und des Antriebsmotors
- Auto fährt zu langsam
  Lösung: Übersetzung ändern, größere Antriebsräder
- Fahrzeug schiebt beim Lenken über die Vorderräder
  Lösung: Lenktrapez einbauen, um kurveninneres Rad stärker einzulenken
- Hinterräder drehen in Kurven leicht durch
  Lösung: Differential an der Hinterachse einbauen

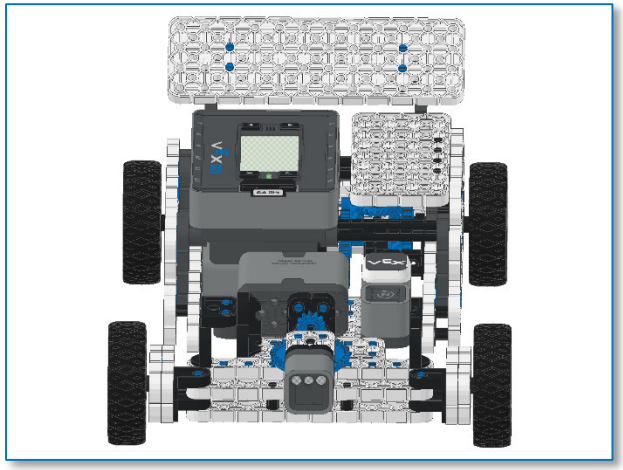
Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

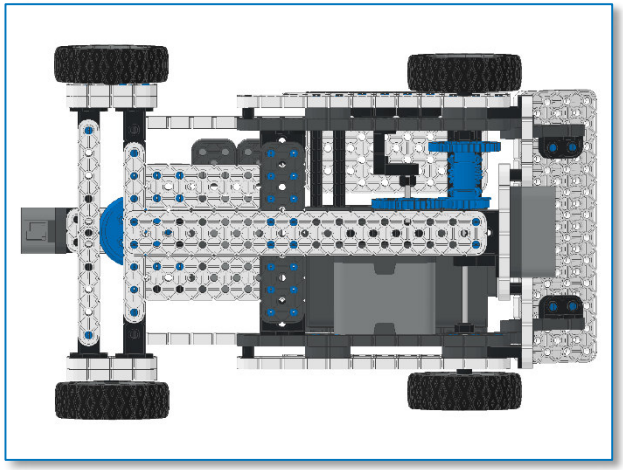
Lösungsbeispiel: Racebot

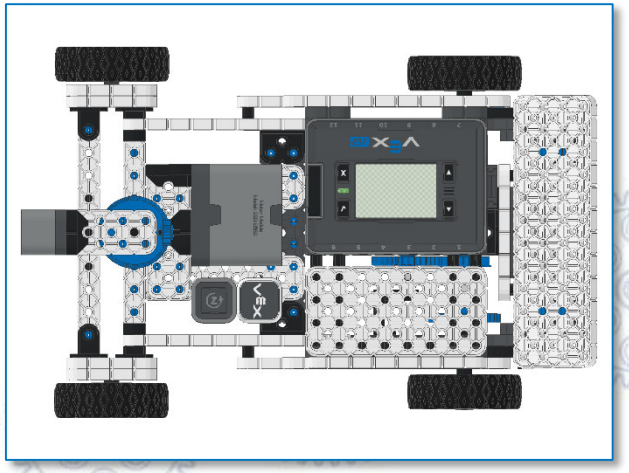












Erstellt von: Dreibein Lehrsysteme GmbH

CC BY NC SA

CAD-Modell

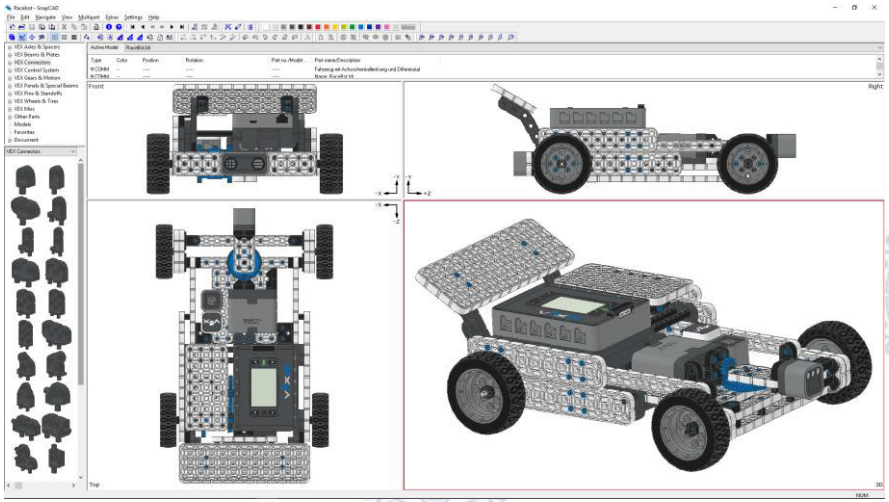
Lernziel

Die Schüler*innen können mit einem CAD-Programm umgehen und einen realen Aufbau in ein CAD-Modell überführen. Darüber hinaus können sie in Baugruppen arbeiten und eine Montageanleitung mithilfe des Programms SnapCAD erstellen.

Aufgabe 7: Erstelle ein CAD-Modell

Das aufgebaute Fahrzeug kann man auf dem Computer als CAD-Modell nachbauen. Die CAD-Datei kann dann einfach mit anderen geteilt und gemeinsam weitergearbeitet werden oder es auch Abbildungen erstellt werden.

Verwende die Software SnapCAD, um ein CAD-Modell Deines Fahrzeugs zu erstellen!



Hinweis

SnapCAD ist eine kostenfreie Software. Damit kannst Du dein CAD-Modell auch zu Hause weiterbearbeiten.
https://www.vexrobotics.com/vexiq/resources/cad-snapcad

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA

CAD-Modell Zusatzaufgaben

Aufgabe 8a: Mit Baugruppen arbeiten

Unterteile Dein CAD-Modell in Baugruppen, damit du übersichtlicher arbeiten kannst. So können auch mehrere Personen gleichzeitig an verschiedenen Baugruppen arbeiten und abschließend zusammenführen, um schneller zum Ziel zu kommen.

Aufgabe 8b: Montageanleitung erstellen

Füge Aufbauschritte in das CAD-Modell ein und erstelle mithilfe von SnapCAD eine Aufbauanleitung für Dein Fahrzeug.

Aufgabe 8c: Bauteilrecherche

Überlege, welche zusätzlichen Bauteile, die Du in der SnapCAD-Bauteilebibliothek findest, hilfreich für die Optimierung Deines Fahrzeugs sein könnten.

Erstellt von: Dreibein Lehrsysteme GmbH
CC BY NC SA