


1 Einführung

1.1 Aufbau des Lernpfads

Der Lernpfad ist in mehrere Kapitel unterteilt:

1. Einführung
 2. Sequenz
 3. Schleifen
 4. Fallunterscheidung
 5. Variablen
 6. Probleme Lösen
- Jedes Thema hat mehrere kleine **Kapitel**, in denen ihr **Videos** seht. Hier werden euch neue **Funktionen** erklärt oder **Aufgaben** gegeben.
 - Die **Reihenfolge** ist wichtig, weil alle Videos aufeinander aufbauen.
 - Alles, was ihr für die Aufgaben wissen müsst, wird vorher in den Videos  erklärt.

1.2 Strategien für den Lernpfad

Videos richtig nutzen

- **Reihenfolge beachten**
Schaut die Videos in der vorgegebenen Reihenfolge an, denn sie bauen aufeinander auf. So versteht ihr alles Schritt für Schritt.
- **Notizen machen**
Schreibt euch die wichtigsten Infos aus den Videos auf, dann könnt ihr später schnell nachlesen.
- **Videos mehrmals ansehen**
Ein Video noch mal ansehen ist immer erlaubt! Wenn ihr aber direkt mitschreibt, seid ihr schneller.

Arbeiten im Team

- **Helft euch gegenseitig**
Manchmal versteht eine Person etwas besser und kann es der anderen erklären. Dafür seid ihr ein Team - gemeinsam geht's leichter!

Das Programmieren

- **Notizen für die Aufgaben - Steps**

Schreibt euch bei den Aufgaben die einzelnen Schritte (Steps) auf, die gezeigt werden. Dann versteht ihr besser, was der Roboter machen soll.

- **Wenn etwas nicht klappt**

- Schaut euch den Block an, bei dem es hakt - mit den Steps geht das ganz leicht.
- Überlegt: Was macht der Block? Was bedeuten die Einstellungen und Zahlen darin?
- Probiert aus, was passiert, wenn ihr etwas ändert.
- Findet heraus, wie ihr den Block so verändern könnt, dass der Roboter das tut, was ihr möchtet.
- Schaut am Ende dieses Lernpfades nach. Dort werden Ursachen und Lösungen für häufige Probleme erklärt.

Denken, Ausprobieren, Verstehen.

Dann fällt euch das Programmieren von Mal zu Mal leichter!

Es muss nicht alles beim ersten Mal klappen.

Die Informationen aus diesem ersten Abschnitt findet ihr nochmal in eurem ersten Video:

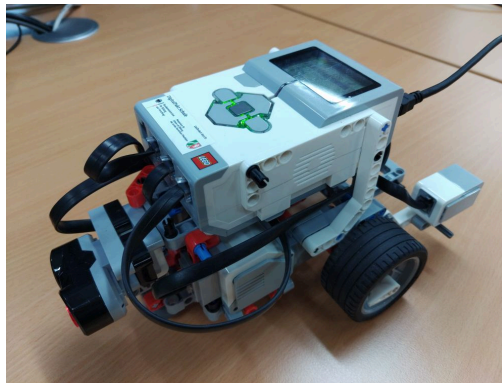
📺 [So funktioniert der Roboter-Lernpfad](#)

2 Sequenz

In einem einfachen Programm werden verschiedene Anweisungen hintereinander ausgeführt. Eine solche Folge von Anweisungen nennen wir **Sequenz**. Im ersten Kapitel lassen sich alle Programme durch eine einfache Sequenz programmieren.

Bevor wir mit dem Programmieren loslegen, müssen wir allerdings noch die Werkzeuge einführen, die euch zur Verfügung stehen. Das sind der Roboter und die Entwicklungsumgebung auf dem Computer.

2.1 Den Roboter verstehen



In der echten Welt ist ein Roboter eine programmierbare Maschine, die ähnlich zu uns Menschen die Umwelt wahrnehmen und durch programmierte Aktionen darauf Einfluss nehmen kann.

- **Sensoren** sind dabei vergleichbar zu unseren Sinnesorganen, wie z.B. den Augen. Mit den Augen können wir erkennen, ob es hell oder dunkel ist. Mit der Haut können wir erkennen, ob es warm oder kalt ist. Mit unseren Ohren können wir hören, ob es laut oder leise ist.
- **Aktoren** sind vergleichbar zu den Muskeln, die unseren Körper bewegen. Mit unseren Muskeln können wir uns in der Welt fortbewegen oder unserem Tischnachbar winken. Unsere Stimme ist auch ein Aktor, denn durch Sprechen verändern wir die Lautstärke unserer Umgebung.

Fallen euch noch weitere Beispiele zu Sensoren und Aktoren ein?

Unser Roboter hat auch **Sensoren** und **Aktoren**. Diese müssen wir in unserem Programm geschickt nutzen, um das gewünschte Verhalten der Aufgaben zu erzielen. Im Laufe des Lernpfads lernen wir schrittweise die Sensoren und Aktoren des Roboters kennen.

Dieses kurze Video erklärt in Kürze die wichtigsten Aspekte:

 [Den Roboter verstehen](#)

2.2 Die Entwicklungsumgebung kennenlernen



Jetzt, da wir den Roboter verstehen, können wir lernen, wie man ihn programmiert. Wir erstellen unsere Programme am Computer und können diese dann auf den Roboter übertragen und ausführen. Dazu brauchen wir ein spezielles Programm, eine **Entwicklungsumgebung**, welche wir über den gesamten Lernpfad benutzen werden.

In dieser Entwicklungsumgebung können wir in einem benannten **Projekt**, benannte Programme erstellen, die wir auf den Roboter übertragen können. Das Bild **oben links** zeigt das Programm „Sequenz 1“ als Teil des „Sequenz.ev3“-Projekts.

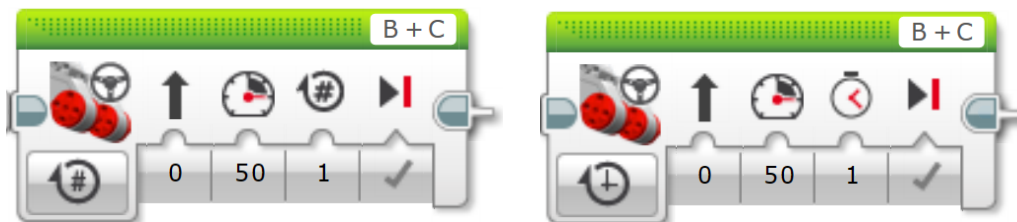
Erstellt euch in eurem Gruppenordner einen **neuen Ordner**, in dem ihr eure Projekte speichern möchtet. Die Programme werden immer Teil der einzelnen Projekte sein. Benennt den Ordner nach den Namen eurer Gruppenmitglieder, z.B. marie_max .

Das folgende Video zeigt euch, wie ihr jetzt ein Projekt anlegen könnt und darin Programme erstellt. Außerdem werden euch die verschiedenen Optionen und Ansichten innerhalb der Entwicklungsumgebung vorgestellt.

📺 [Einführung in die Entwicklungsumgebung](#)

Jetzt versteht ihr den Roboter und könnt die Entwicklungsumgebung bedienen.

2.3 Motoren ansteuern für Umdrehungen oder Sekunden



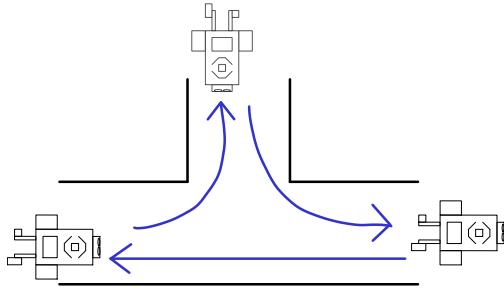
Für die erste Aufgabe müssen wir lernen, die Motoren des Roboters anzusteuern. Das folgende Video führt den Motor und dessen Optionen ein:

📺 [Motoren Teil 1 - Umdrehungen und Sekunden](#)

2.4 Aufgabe 1: „Wende in drei Zügen“

Allgemeiner Hinweis: Erstellt für jedes Kapitel und die **Wiederholungsaufgaben** ein eigenes Projekt in eurem Ordner.

Für jede Programmieraufgabe erstellt ihr ein eigenes Programm in dem Projekt, zu dem die Aufgabe gehört.

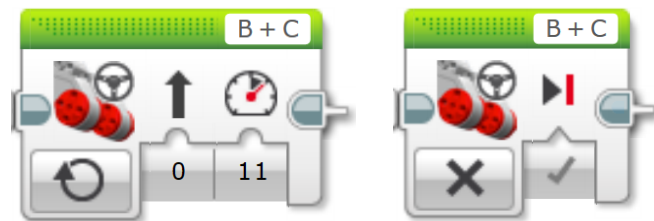


Dein 1. Programmierauftrag:

Erstelle ein Programm, damit sich dein Roboter genauso verhält wie der Roboter in diesem Video:

📺 [Wende in drei Zügen](#)

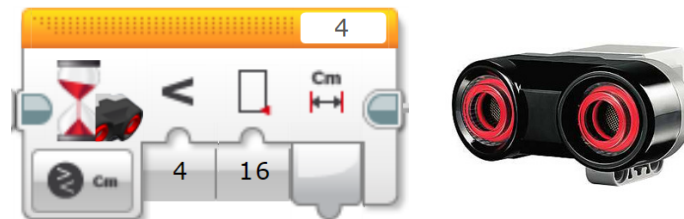
2.5 Motoren an/aus



Du hast bei deiner ersten Programmieraufgabe die Motoren für eine bestimmte Zeit (Sekunden) oder eine feste Anzahl an Umdrehungen angesteuert. Für die nächste Programmieraufgabe müssen wir aber zwei weitere Möglichkeiten (An & Aus) kennenlernen.

📺 [Motoren Teil 2 - An und Aus](#)

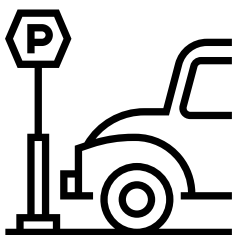
2.6 Ultraschallsensor nutzen



In der nächsten Programmieraufgabe werden wir eine Einparkhilfe programmieren. Dafür musst du mit dem Ultraschallsensor, der Entfernungen messen kann, arbeiten können.

📺 [Warten auf Zeit oder den Ultraschallsensor](#)

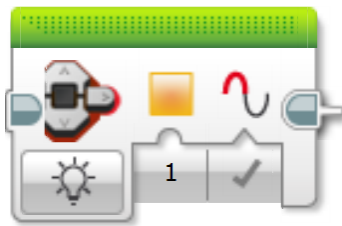
2.7 Aufgabe 2: „Einparkhilfe“



Dein 2. Programmierauftrag: Erstelle ein Programm, damit dein Roboter sich genauso verhält wie der Roboter in diesem Video:

📺 [Einparkhilfe](#)

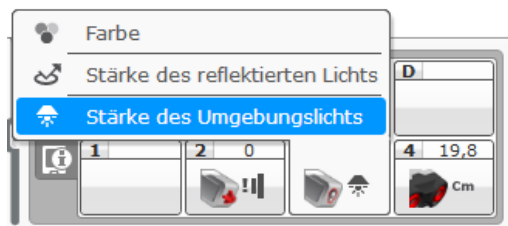
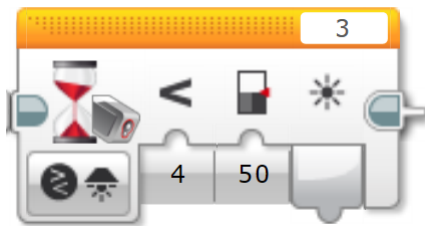
2.8 LEDs einschalten



In der nächsten Programmieraufgabe werden wir ein automatisches Einschalten der Scheinwerfer programmieren. Dafür musst du wissen, wie man die LEDs unter den Steintasten des Roboters aktiviert.

▶ [LEDs ansteuern](#)

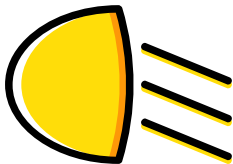
2.9 Warte auf Umgebungslicht



In der nächsten Programmieraufgabe werden wir ein automatisches Einschalten der Scheinwerfer programmieren. Wie du die LEDs einschaltest weißt du nun, aber du musst noch wissen, wie man die Stärke des Umgebungslichts vom Lichtsensor auslesen kann.

▶ [Warte auf Umgebungslicht](#)

2.10 Aufgabe 3: „Scheinwerfer anschalten“



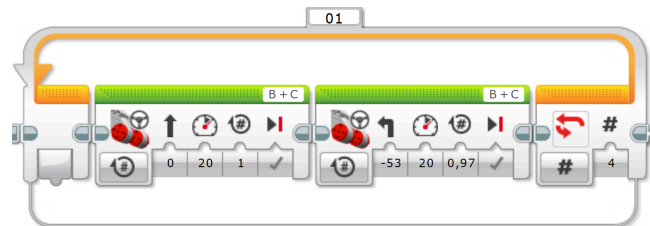
Dein 3. Programmierauftrag: Erstelle ein Programm, damit dein Roboter sich genauso verhält wie der Roboter in diesem Video:

▶ [Licht einschalten](#)

3 Schleifen

In diesem Plan lernst du Schleifen kennen. Mit ihnen kann der Roboter Befehlsfolgen wiederholen.

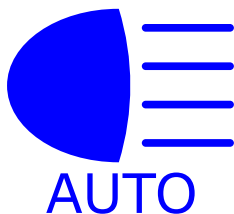
3.1 Schleifen verstehen



Bevor wir loslegen, musst du wissen, was Schleifen sind und was sie im Programm bewirken. Dieses 5-minütige Video erklärt dir in Kürze, die wichtigsten Aspekte.

[Schleifen](#)

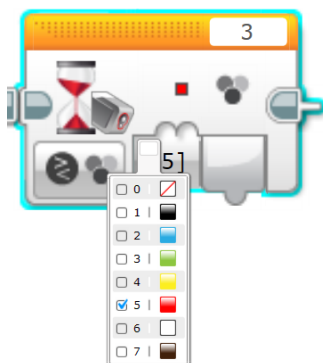
3.2 Aufgabe 4: „Automatische Scheinwerfer“



Für die Programmieraufgaben in diesem Plan zum Thema Schleifen legst du bitte ein neues Projekt mit dem Namen „Schleifen“ an. Für jede Aufgabe erstellst du bitte ein eigenes Programm in dem Projekt, damit du zu jeder Aufgabe deine Lösung langfristig gespeichert hast und immer wieder auf sie zugreifen kannst. Dein 4. Programmierauftrag: Erstelle ein Programm, damit dein Roboter sich genauso verhält wie der Roboter in diesem Video:

[Lichtautomatik](#)

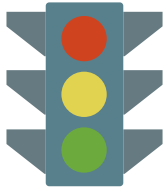
3.3 Farbe erkennen



Nachdem wir eine automatische Lichtsteuerung programmiert haben, wollen wir als nächstes einen automatischen Start- und Stoppiloten für Ampeln programmieren. Bevor wir loslegen, musst du aber noch schnell lernen, wie man mit dem Farb- / Lichtsensor auf Farbe reagieren kann. Dieses 3-minütige Video erklärt dir in Kürze, die wichtigsten Aspekte.

[Warte auf Farbe](#)

3.4 Aufgabe 5: „Start/Stop an Ampeln“



Dein 5. Programmierauftrag: Erstelle ein Programm, damit dein Roboter sich genauso verhält wie der Roboter in diesem Video:

▶ [Ampelpilot](#)

3.5 Wiederholen & Vertiefen 1

Durch folgende Wiederholungs- und Vertiefungsaufgaben sollst du deine Programmierfähigkeiten festigen. Für diese 5 Wiederholungsaufgaben hast du eine Doppelstunde Zeit (90 Minuten). Was du nicht innerhalb dieser Zeit geschafft hast, überspringst du bitte. Du benötigst dabei keine neuen Informationen zu Blöcken oder ähnlichem. Alle Aufgaben kannst du ausschließlich mit dem dir jetzt bekannten Wissen erledigen.

Aufgabe 1: ▶ [Gegenstand umfahren](#)

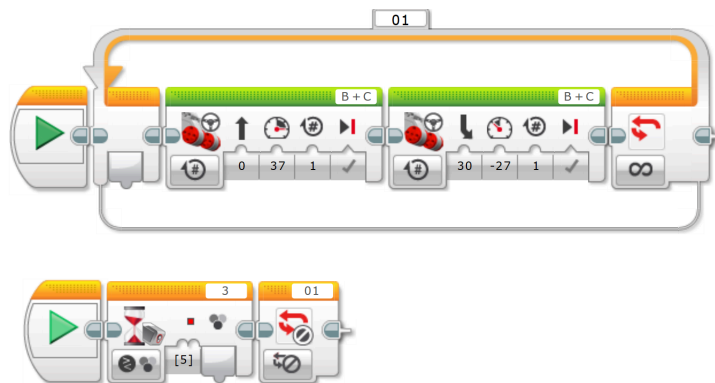
Aufgabe 2: ▶ [Endlose Kreisfahrt](#)

Aufgabe 3: ▶ [Ausweichmanöver](#)

Aufgabe 4: ▶ [Angepasste Fahrweise](#)

Aufgabe 5: ▶ [Endlos Rückwärts mit Notbremsen](#)

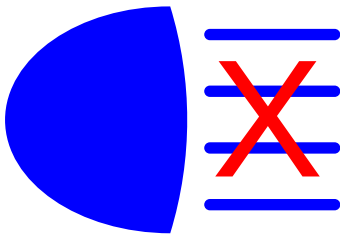
3.6 Parallele Prozesse programmieren



Du hast nun schon viel gelernt und zuletzt mehrere Übungsaufgaben zur Vertiefung und Festigung bearbeitet. Jetzt lernst du, wie man den Roboter mehrere Dinge gleichzeitig machen lässt. Dadurch vervielfältigen sich unsere Möglichkeiten in der Programmierung! :-) Dieses 6-minütige Video erklärt dir in Kürze, die wichtigsten Aspekte.

▶ [Multitasking und Interrupt](#)

3.7 Aufgabe 6: „Scheinwerfer manuell einschalten“



Dein Programmierauftrag: Erstelle ein Programm, damit dein Roboter sich genauso verhält wie der Roboter in diesem Video:

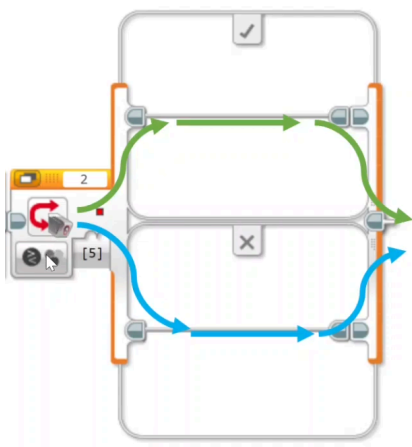
▶ [Abbruchautomatik](#)

Erweitere dazu dein Programm von Aufgabe 5.

4 Fallunterscheidung

Manchmal möchte man in der Programmierung als Programmierin oder Programmierer ausdrücken, dass eine bestimmte Anweisung nur dann ausgeführt werden soll, wenn eine bestimmte Bedingung erfüllt ist. Wir lernen nun, wie dies geht. :-)

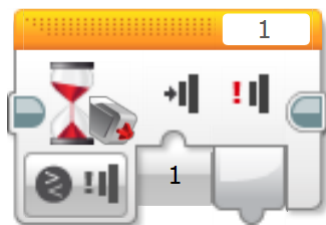
4.1 Fallunterscheidung



Bevor wir loslegen, musst du wissen, wie man Fallunterscheidungen in Programme einbaut und was sie bewirken. Dieses 5-minütige Video erklärt dir in Kürze, die wichtigsten Aspekte.

▶ [Fallunterscheidung](#)

4.2 Tastsensor



In der nächsten Aufgabe wollen wir die Rückleuchte am Auto programmieren. Als Simulation des Rückwärtsgangs wollen wir den Tastsensor (wird oft auch Berührungssensor genannt) benutzen. Dieses 1-minütige Video erklärt dir in Kürze, wie man ihn benutzt.

▶ [Warte auf den Tastsensor](#)

Für die Programmieraufgaben in diesem Plan zum Thema Fallunterscheidung legst du bitte ein

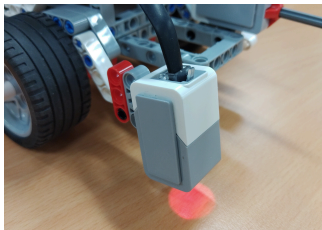
neues Projekt mit dem Namen "Fallunterscheidung" an. Für jede Aufgabe erstellst du bitte ein eigenes Programm in dem Projekt, damit du zu jeder Aufgabe deine Lösung gespeichert hast.

4.3 Aufgabe 7: „Licht im Rückwärtsgang“

Dein Programmierauftrag: Erstelle ein Programm, damit dein Roboter sich genauso verhält wie der Roboter in diesem Video:

📺 [Rückleuchte](#)

4.4 Umbau des Lichtsensors



Als nächstes wollen wir dem Roboter beibringen, einer schwarzen Linie zu folgen. Damit dies möglich ist, muss der Farbsensor nicht nach vorne, sondern nach unten auf den Boden ausgerichtet sein.

Dein Auftrag: Schau dir dieses Minivideo an, das dir zeigt, wie einfach man den Lichtsensor umbauen kann: 📺 [Umbau Licht](#)

4.5 Aufgabe 8: „Linie folgen“

Dein Programmierauftrag: Erstelle ein Programm, damit dein Roboter sich genauso verhält wie der Roboter in diesem Video:

📺 [Linienfolger](#)

4.6 Wiederholen & Vertiefen 2

Durch folgende Wiederholungs- und Vertiefungsaufgaben sollst du deine Programmierfähigkeiten festigen. Du benötigst dabei keine neue Informationen zu Blöcken oder ähnlichem. Alle Aufgaben kannst du ausschließlich mit dem dir jetzt bekannten Wissen erledigen. Für diese 3 Wiederholungsaufgaben hast du eine Doppelstunde Zeit (90 Minuten). Was du nicht innerhalb dieser Zeit geschafft hast, überspringst du bitte. Alle 3 Wiederholungsaufgaben bilden gemeinsam ein großes Projekt, welches du in diesem Video sehen kannst: 📺 [Das Projekt](#)

In dem Video, das du gerade gesehen hast, lief auf beiden Robotern dasselbe Programm. Zum Testen deines Programms kannst du dich mit einer anderen Gruppe zusammen tun. Man kann zur Not ein vor einem fahrendes Fahrzeug auch mit der Hand simulieren. Sobald ihr als Gruppe ein Programm fertiggestellt habt, meldet euch bei der Lehrkraft. Sie wird sich notieren, dass ihr als Gruppe diese Aufgabe erledigt habt. Programmieren dazu folgende Teilaufgaben und erweitere somit schrittweise dein Programm zum Linienfolger:

Aufgabe 1: 📺 [Abstand beim Linienfolgen](#)

Aufgabe 2: 📺 [Stopp auf Knopfdruck beim Linienfolgen](#)

Aufgabe 3: 📺 [Blinke, wenn gestoppt](#)

5 Variablen

An manchen Stellen muss man sich in einem Programm Dinge merken können, damit man auf diese Informationen später im Programm wieder zugreifen kann. Jetzt lernen wir, wie dies geht. :-)

5.1 Variablen



Bevor wir mit den Programmieraufgaben zu Variablen loslegen, musst du wissen, wie der Roboter sich etwas merken kann. Dieses 11-minütige Video visualisiert dir kleinschritt, die wichtigsten Aspekte.

[▶ Variablen](#)

5.2 Aufgabe 9: „Wie oft wurde gedrückt?“

Dein Programmierauftrag: Erstelle ein Programm, damit dein Roboter sich genauso verhält wie der Roboter in diesem Video:

[▶ Druck Zählen](#)

5.3 Aufgabe 10: „Zählen und Fahren“

Dein Programmierauftrag: Erstelle ein Programm, damit dein Roboter sich genauso verhält wie der Roboter in diesem Video:

[▶ Fahre wie gedrückt](#)

Erweitere dazu dein Programm von Aufgabe 9.

5.4 Wiederholen & Vertiefen 3

Durch folgende Wiederholungs- und Vertiefungsaufgaben sollst du deine Programmierfähigkeiten festigen. Für diese 3 Wiederholungsaufgaben hast du eine Doppelstunde Zeit (90 Minuten). Falls das Unterrichtsvorhaben zur Roboterprogrammierung noch nicht beendet ist, darfst du die Aufgaben gerne zu Ende machen und dir anschließend eigene Programmierideen überlegen und programmieren. Du benötigst dabei keine neue Informationen zu Blöcken oder ähnlichem. Alle Aufgaben kannst du ausschließlich mit dem dir jetzt bekannten Wissen erledigen. Alle 3 Wiederholungsaufgaben bilden auch hier ein gemeinsames Projekt und bauen aufeinander auf: Sobald ihr als Gruppe ein Programm fertiggestellt habt, meldet euch bei der Lehrkraft. Sie wird sich notieren, dass ihr als Gruppe diese Aufgabe erledigt habt.

Aufgabe 1: [▶ Verbrauch per Variable simulieren](#)

Aufgabe 2: [▶ Tanken per Knopfdruck simulieren](#)

Aufgabe 3: [▶ Tankvorgang noch realer simulieren](#)

6 Probleme Lösen

Problem	Mögliche Ursachen	Lösungsstrategien
Der Roboter fährt nicht so wie ich will	Zahlwerte im Motorblock müssen anders eingestellt werden	<ul style="list-style-type: none"> • Klärt, wie genau der Roboter überhaupt fahren soll <ul style="list-style-type: none"> ▸ Vorwärts oder rückwärts? ▸ Wie stark nach links oder rechts? ▸ Wie lange bzw. bis wann? • Probiert in einzelnen Parametern (die drei Zahlen unten im Motorblock) verschiedene Werte aus und erklärt euch gegenseitig, was diese bewirken <ul style="list-style-type: none"> ▸ Ändert immer nur einen Wert zur Zeit, damit ihr nachvollziehen könnt, dass verändertes Roboterverhalten von genau dieser Codeänderung ausgelöst wurde
Der Roboter fährt weiter, obwohl er wegen Sensoreingaben schon abbrechen sollte	Der Motorblock läuft für x Umdrehungen oder Sekunden und erst danach verarbeitet der Roboter Sensoreingaben	Stellt den Modus unten links im Motorblock auf „An“, dann geht der Roboter sofort zum nachfolgenden Block über.
Der Sensor reagiert nicht	<ol style="list-style-type: none"> 1. Der Schwellwert (die zweite Zahl im Warteblock) ist zu hoch oder zu niedrig 2. Im Warteblock ist der falsche Port eingestellt. Der Sensor steckt in einem anderen Port 3. Im Warteblock ist der falsche Sensor ausgewählt 	<ol style="list-style-type: none"> 1. Erzeugt unterschiedliche Sensoreingaben indem ihr z.B. vor dem Ultraschallsensor eure Hand hin- und her bewegt und beobachtet in dem Sensorübersichtsfeld unten rechts, welche Werte der Sensor misst. <ul style="list-style-type: none"> • Dadurch solltet ihr eine Vorstellung davon erhalten, welche Werte der Sensor normalerweise annimmt. Überlegt euch nun, welcher Wert mitten in diesem Wertebereich liegt und daher als Grenze zwischen hohen und niedrigen Werten geeignet ist. 2. Schaut in dem Sensorübersichtsfeld unten rechts nach, an welchem Port der Sensor angeschlossen ist. Stellt diesen Port dann im Warteblock ein. 3. Wählt im Warteblock unten links den richtigen Sensor aus. Lest euch die Sensornamen durch, die Symbole sind teilweise irreführend.
Die LED leuchtet nicht	<ol style="list-style-type: none"> 1. Nach dem Anschalten der LED wird sofort das Programm beendet und die LED geht wieder aus. 2. Der Warteblock nach dem Einschalten hat die gleiche Bedingung wie der davor. 	<ol style="list-style-type: none"> 1. Nach dem LED-Block noch einen Warteblock einfügen 2. Vergleichsoperator in einem der beiden Warteblocke ändern
Die LED geht nicht wieder aus	In der Schleife gibt es keinen Block, der die LED ausschaltet	In der Schleife am Ende einen Block einfügen, der die LED ausschaltet