

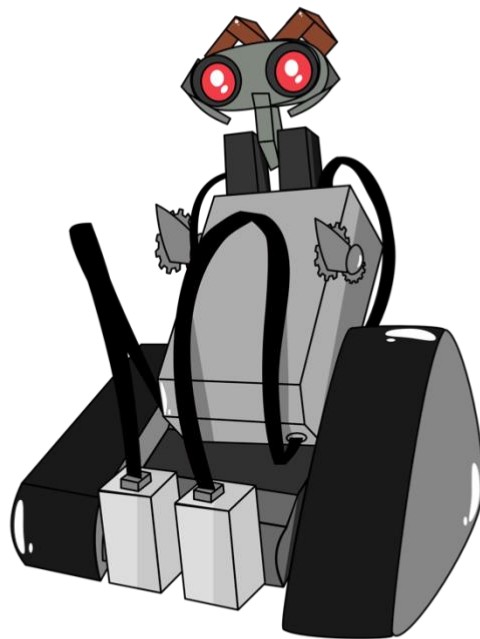
Aufgabenblatt 2 – Kontroll-Blöcke

Im zweiten Aufgabenblatt geht es darum, die Kontroll-Blöcke, sowie den Farbsensor kennenzulernen. Dazu lernt ihr zunächst, welche Kontroll-Blöcke es gibt und wie ihr den Farbsensor verwenden könnt. Das neu erlernte Wissen könnt ihr dann in der ersten Line-Following Aufgabe anwenden.

Bei Fragen meldet euch gerne bei robotikwettbewerb@in.tum.de.

Abgabefrist für alle Blätter ist der **09.05.2021, 20:00**.

Viel Spaß und Erfolg!



Theorie

Farbsensor

Der Farbsensor (oder: Color Sensor) ist ein optischer Sensor und kann verschiedene Farben erkennen, indem er Licht (rot-, grün-, blau-farbig) auf das Objekt strahlt, welches reflektiert wird. In Open Roberta hat der Farbsensor vier verschiedene Modi (Farbe, Licht, Umgebungslicht und RGB).

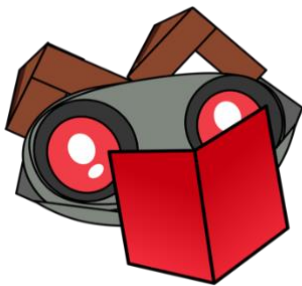
Im Modus *Farbe* kann der Farbsensor acht verschiedene Grundfarben erkennen. Dabei ist zu beachten, dass es dabei sein kann, dass ein anderer Farbton nicht als die vorgegebene Farbe erkannt wird. Zum Beispiel wird ein nicht ganz so kräftiges Grün nicht als das verglichene Grün erkannt.

Im *RGB*-Modus erkennt der Sensor die Farbe, indem er die Farbwerte von Rot, Grün und Blau auf einmal misst und das reflektierte Licht entsprechend dieser Anteile wiedergibt. Es wird also eine Liste mit drei Werten, jeweils auf einer Skala von 0 bis 255 ausgegeben.

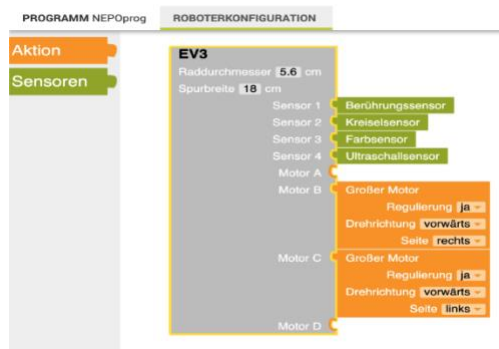
Im *Licht*-Modus sendet der Sensor mit Hilfe seiner roten LED, Licht aus und misst die reflektierte Lichtstärke von einer Skala von 0 bis 100.

Schließlich gibt es den *Umgebungslicht*-Modus, welcher das Umgebungslicht von externen Quellen wie Sonnenlicht misst.

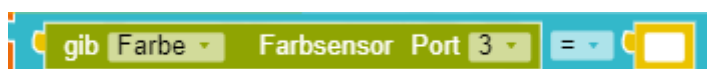
Mit dem Farbsensor kann der Roboter also zum Beispiel auf einer Straße fahren, indem er überprüft, ob er auf einer grauen Straße oder auf einer grünen Wiese fährt.



Um den Sensor verwenden zu können müsst ihr sicherstellen, dass er in der Roboterkonfiguration aktiviert ist. Das ist standardmäßig bereits der Fall.



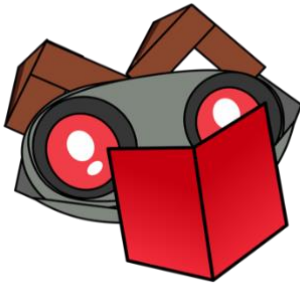
Im folgenden Beispiel vergleicht der Roboter die Farbe, welche sein Farbsensor wahrnimmt, mit der Farbe **weiß**. Das wird mithilfe eines Logik-Blocks bewerkstelligt. Das Ergebnis dieses Vergleichs ist entweder WAHR oder FALSCH. Mehr zum Thema Logik erfahrt ihr am Ende dieses Blatts.



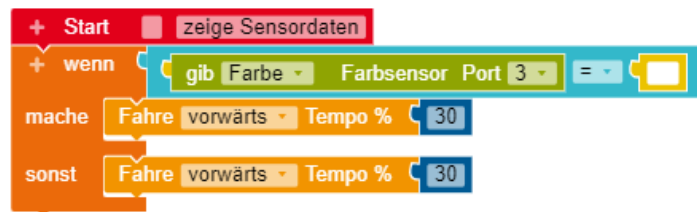
Wenn-Sonst-Blöcke

Der Wenn-Mache-Sonst-Block wird benutzt, um Unterscheidungen zu machen. Er hat eine Bedingung im Wenn-Teil, welche entweder wahr oder falsch ist. Falls die Bedingung wahr ist, so wird der Mache-Teil ausgeführt. Falls die Bedingung falsch ist, wird stattdessen der Sonst-Teil ausgeführt. Es gibt auch reine Wenn-Mache-Blöcke, die keinen Sonst-Teil haben. Bei diesen Blöcken wird einfach im Programm weiter gegangen.

Über das '+'-Symbol kann man weitere Bedingungen hinzufügen. Die Bedingungen werden nacheinander geprüft und die Aktionen der ersten Bedingung, welche erfüllt wird, werden ausgeführt. Alle nachfolgenden Bedingungen werden nicht mehr geprüft.



In diesem Beispiel überprüft der Roboter, ob der Farbsensor die Farbe **weiß** erkennt und fährt in diesem Fall vorwärts. Falls die Farbe nicht **weiß** ist, so fährt er rückwärts. Um die Farbe des



Schleifen

Schleifen können benutzt werden, um Aktionen mehrmals auszuführen. Bei Open Roberta kann man Schleifen unendlich oder x-mal ausführen lassen, wobei x eine Zahl ist, die man selbst wählt. Des Weiteren gibt es die Wiederhole-Bis-Schleifen, welche die folgenden Aktionen so lange ausführen, bis die angegebene Bedingung erfüllt wird. Dabei kann es sein, dass die Bedingung niemals erfüllt wird und die Schleife dann zu einer Unendlich-Schleife wird.



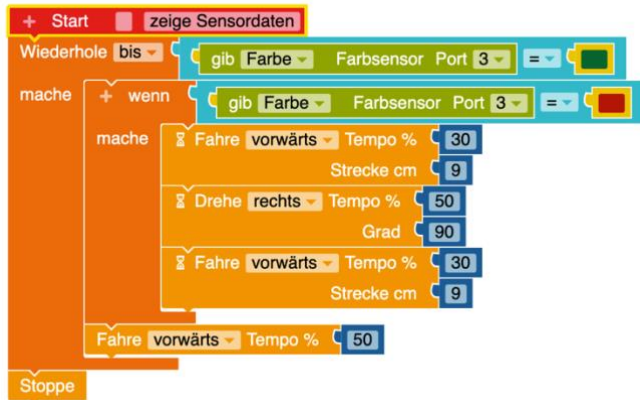
Das Beispiel unten zeigt einen Roboter, der so lange geradeaus fährt wie er **schwarz** erkennt. Sobald der Roboter eine rote Markierung erkennt, stoppt er.



Beispielaufgabe:

In diesem Beispiel seht ihr, wie ihr den Farbsensor in Kombination mit den Wenn-Sonst-Blöcke und Schleifen verwendet könnt.

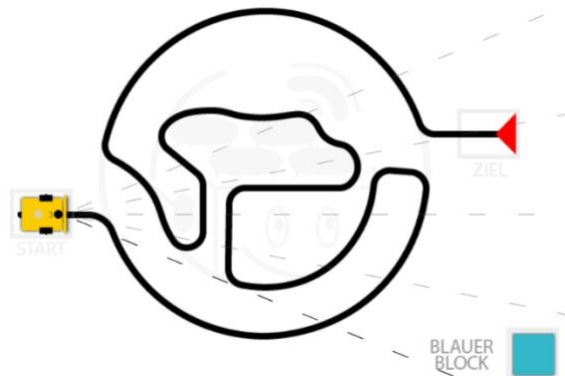
Der Roboter fährt solange geradeaus, bis er `Rot` erkennt, dreht sich dann um 90° und fährt anschließend weiter. Das wiederholt er solange, bis er `Grün` erkennt.



Aufgabe 2.1: Line-Following

Programmiere einen Roboter, der einer schwarzen Linie folgt und am Ende des Parcours auf der roten Haltefläche hält.

1. Ladet zunächst den Hintergrund für Aufgabe 2.1 in die Simulationsansicht und verschiebt den blauen Block und den Roboter in die vorgegebenen Felder.



2. Überlegt euch, wie ihr mithilfe von einer Schleife und Wenn-Sonst-Blöcken sicherstellen könnt, dass euer Roboter immer der schwarzen Linie folgt. Hierbei ist der Farbsensor hilfreich. Optimiert euer Programm dann so, dass auch die scharfen Kurven kein Problem mehr für den Roboter darstellen.
3. Wenn es der Roboter bis zur roten Zielfläche geschafft hat, soll er dort einfach stehen bleiben.



Wenn wir euer Programm testen, werden wir einen etwas anderen Parcours verwenden. Stellt also sicher, dass euer Roboter tatsächlich der Linie folgt und nicht nur die Kurven nachfährt.



Bonusaufgabe 2.2: Labyrinth

Programmiere einen Roboter, der aus dem Labyrinth herausfindet. Dabei stellen die weißen Flächen Wege und die schwarzen Flächen Mauern dar.

1. Ladet euch wieder zunächst den Hintergrund für die Aufgabe 2.2 in die Simulationsansicht und verschiebt den blauen Block und den Roboter in die vorgegebenen Felder.
2. Überlegt euch eine Strategie, wie der Roboter aus dem Labyrinth herausfindet. Beachtet dabei, dass der Roboter nur auf den weißen Flächen entlangfahren darf. Erkennt euer Roboter 'schwarz', so soll er rückwärtsfahren, bis er wieder auf einer weißen Fläche ist und einen anderen Weg einschlagen.
3. Wenn der Roboter es bis zur roten Fläche und damit aus dem Labyrinth geschafft hat, soll er einfach stehen bleiben.

Exkurs: Logik

Jede Aussage ist entweder wahr oder falsch. „Tomaten sind rot“ ist eine wahre Aussage, „Bananen sind blau“ hingegen ist offensichtlich falsch. Wann immer wir eine Entscheidung treffen, brechen wir diese auf einzelne wahr/falsch Aussagen herunter. Um in Open Roberta Logik nutzen zu können, gibt es Logik-Blöcke:

	<p>Der Nicht-Block macht eine wahre Aussage falsch und eine falsche Aussage wahr.</p> <p>„Tomaten sind rot“ = wahr ==> nicht(„Tomaten sind rot“) = falsch</p>
 	<p>Mit UND bzw. ODER kann man zwei Aussagen miteinander kombinieren.</p> <p>UND</p> <p>Damit eine UND-Aussage wahr wird, müssen beide Aussagen wahr sein.</p> <p>(🍷 = 🍷) UND (🍷 = 🍷) ist WAHR, da sowohl die linke als auch die rechte Seite wahr ist.</p> <p>(🍷 = 🍷) UND (🍷 = 🍷) ist FALSCH, da zwar die eine Seite wahr ist, die andere jedoch falsch.</p> <p>ODER</p> <p>Die ODER-Aussage ist immer dann wahr, wenn eine der beiden Seiten wahr ist.</p> <p>(🍷 = 🍷) ODER (🍷 = 🍷) ist WAHR</p> <p>(🍷 = 🍷) ODER (🍷 = 🍷) ist FALSCH, da keine Seite wahr ist.</p>



Man kann mehrere UND bzw. ODER auch miteinander kombinieren. Ein kleines Beispiel:

(🍷 = 🍷) ODER ((🍷 = 🍷) UND (🍷 = 🍷))
 -> (FALSCH) ODER (WAHR UND WAHR)
 -> FALSCH ODER WAHR
 -> WAHR

Dieses kleine Rätsel stellt euer Verständnis von Logik auf die Probe:

((🍷 = 🍷) ODER (🍷 = 🍷)) UND ((🍷 = 🍷) ODER (🍷 = 🍷)) UND (🍷 = 🍷)

Ist die Aussage WAHR oder FALSCH?