Selbstlernanleitung

Simulation
eines Lego Mindstorms EV3-Roboters
unter Nutzung des Open Roberta Lab

Alle mit [1)] gekennzeichneten Grafiken sind Screenshots der Seite: https://lab.open-roberta.org

Technische Hilfsmittel

Lego Mindstorms EV3	Mindstorms ist eine Produktlinie des Spielwarenherstellers Lego. Seit 2013 ist die Serie EV3 erhältlich und an Schulen weit verbreitet. Sie kombiniert Lego Technic Bausteine mit Computertechnologie und gestattet es autonome Robotersysteme herzustellen. Lego bietet eine eigene, offline nutzbare Software zur Controller-Programmierung an. Um andere Programmiersprachen zu nutzen, besitzt der Controller einen Micro-SD-Card-Slot. In diesem Kurs wird Java (leJOS EV3 0.9.1) verwendet. Wenn der Controller entsprechend vorbereitet ist, kann er direkt mit dem Open Roberta Lab verbunden werden.
Open Roberta Lab	Das Open Roberta Lab ist eine Open-Source-Plattform des Frauenhofer IAIS, die es gestattet, verschiedenste Hardware mit der ikonischen Programmiersprache NEPO zu programmieren. Die Plattform ist browserbasiert, online nutzbar und erfordert einen beständigen Internetzugang. Sollte kein Internet verfügbar sein, wird ein Image für den Raspberry Pi angeboten, welches auch weniger bewanderten Anwendern eine Offlinenutzung ermöglicht. Wenn keine eigene Hardware vorhanden ist, kann die Simulationsumgebung genutzt werden. Es stehen eine Vielzahl, für den privaten Gebrauch kostenfreie Materialien zur Verfügung. Die Plattform wird ständig erweitert und verbessert, es existiert eine aktive Community. In Abgrenzung zum Layout der Lego-Software ist hier ein engerer Bezug zur Informatik erkennbar. Der Schritt zu einer textbasierten Programmiersprache fällt leichter. Internetadresse: https://lab.open-roberta.org/ Rechteinhaber: Das Fraunhofer-Institut für Intelligente Analyse- und Informationssysteme Schloss Birlinghoven 53757 Sankt Augustin ist eine rechtlich nicht selbständige Einrichtung der Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e. V. Hansastraße 27 c 80686 München

Alle mit [1)] gekennzeichneten Grafiken sind Screenshots der Seite: https://lab.open-roberta.org

1. Herstellen der Arbeitsbereitschaft

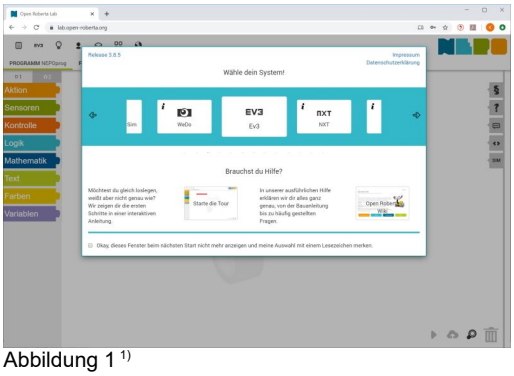


Abbildung 1 [1)]

Stellen Sie sicher, dass Ihr Endgerät über eine Internetverbindung verfügt. Öffnen Sie den Browser (z.B. Google Chrome) und rufen Sie folgende Adresse auf:

https://lab.open-roberta.org/

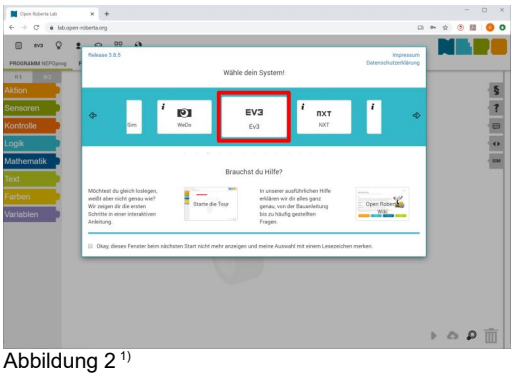


Abbildung 2 [1)]

Wählen Sie durch Linksklick als System EV3.

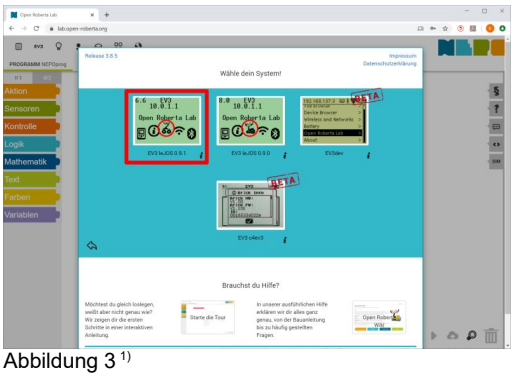


Abbildung 3 [1)]

Wählen Sie als Systemsprache leJOS 0.9.1.

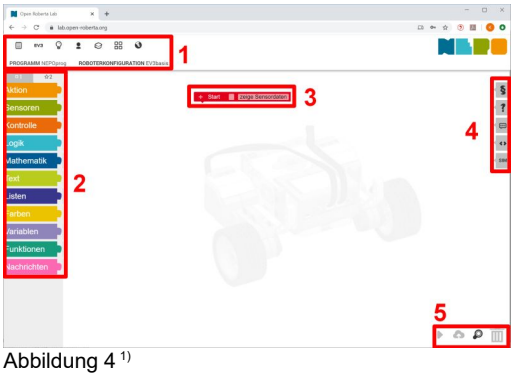


Abbildung 4 [1)]

Das Open-Roberta-Lab besteht aus verschiedenen Bereichen. Für die Lösung Ihrer Aufgaben benötigen Sie den:

1. Menübereich
2. Blockbereich
3. Programmierbereich
4. Wahlbereich
5. Aktionsbereich

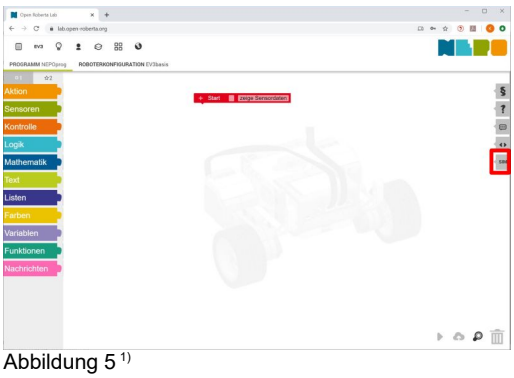


Abbildung 5 [1)]

Öffnen Sie bitte im Wahlbereich durch Klick auf „SIM“ die Simulationsansicht.

Alle mit [1)] gekennzeichneten Grafiken sind Screenshots der Seite: https://lab.open-roberta.org

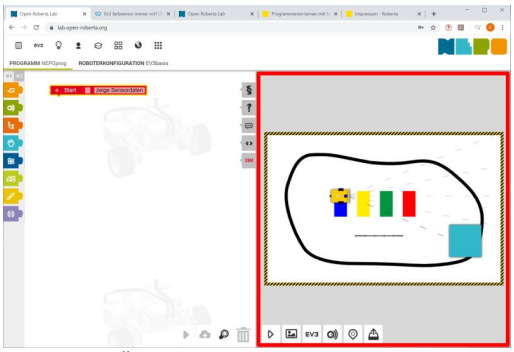


Abbildung 6 [1)]

Die Simulationsansicht besteht aus:

1. Hintergrundbild
2. Roboter (beweglich)
3. Hindernis (beweglich)
4. Aktionsbereich

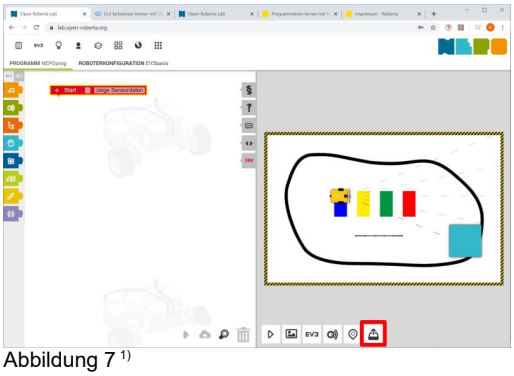


Abbildung 7 [1)]

Laden Sie das Bild „Krankenstation.png“ als Hintergrundbild in die Simulation. Nutzen Sie den Button „Hintergrundbild hochladen“ im Aktionsbereich der Simulationsansicht. Die Datei wird von der Lehrkraft bereitgestellt.

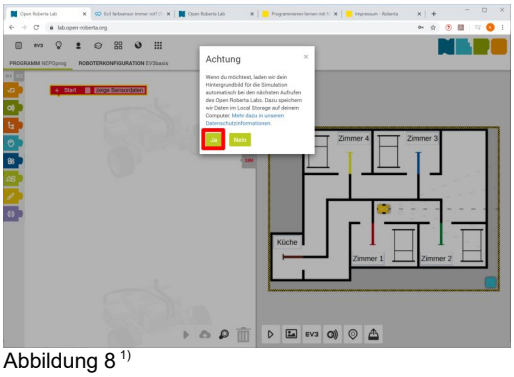


Abbildung 8 [1)]

Wenn Sie die nachfolgende Abfrage mit „Ja“ beantworten, wird das Bild lokal auf dem Endgerät gespeichert.

Hinweis:

- Wenn eine Speicherung aus administrativen Gründen nicht möglich ist, muss das Hintergrundbild bei bei jeder neuen Nutzung von Open Roberta Lab hochgeladen werden.

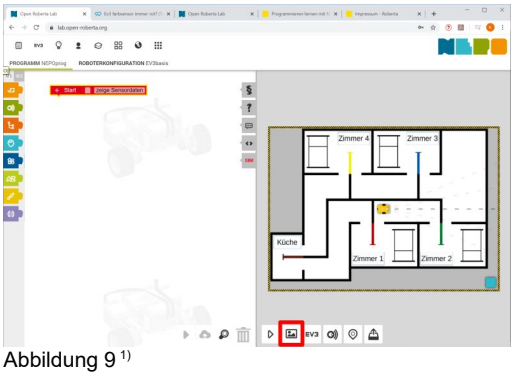


Abbildung 9 [1)]

Das neue Hintergrundbild steht bei nachfolgenden Nutzungen von Open Roberta Lab ohne erneutes Hochladen zur Verfügung und kann über den Hintergrundwahlschalter im Aktionsmenü der Simulationsansicht eingestellt werden.

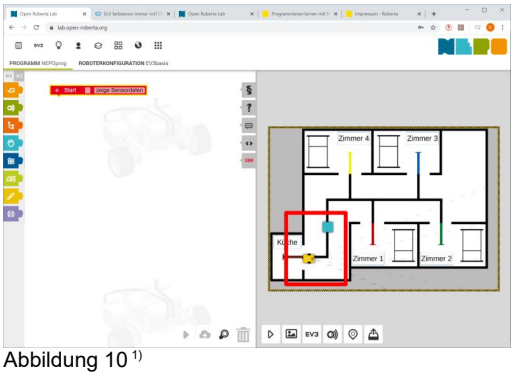
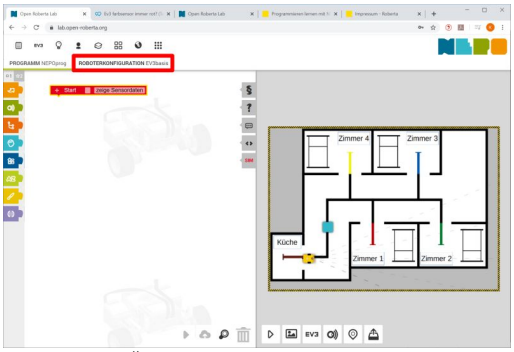
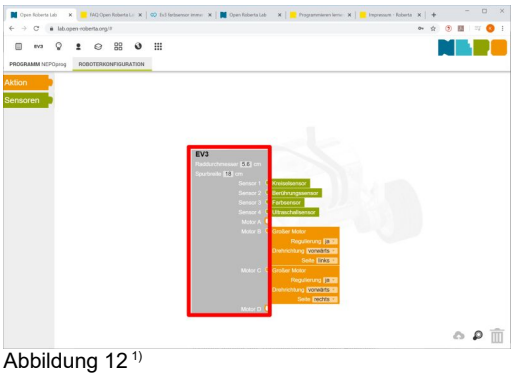
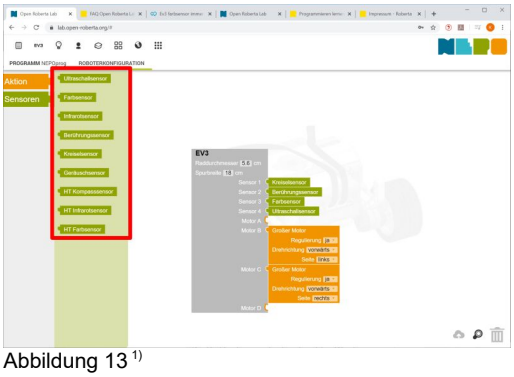
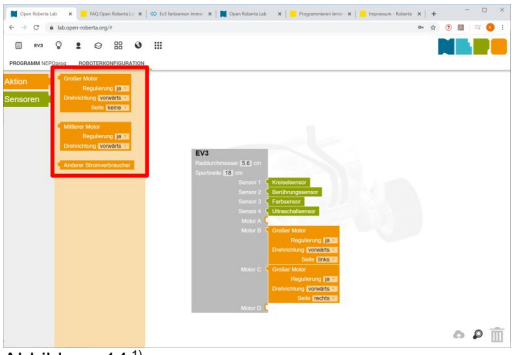
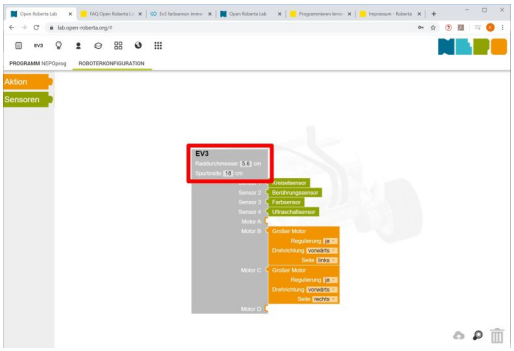


Abbildung 10 [1)]

Platzieren Sie den Roboter und das Hindernis entsprechend der Abbildung 10.

Alle mit [1)] gekennzeichneten Grafiken sind Screenshots der Seite: https://lab.open-roberta.org

2. Konfiguration des Roboters

 Abbildung 11 [1)]	Öffnen Sie die Roboterkonfiguration.
 Abbildung 12 [1)]	Hier können der Lego EV3-Controller mit seinen Anschlussmöglichkeiten für Sensoren und Aktoren sowie grundlegende Eigenschaften (Raddurchmesser, Spurweite) des Roboters konfiguriert werden.
 Abbildung 13 [1)]	Sensoren liefern dem Roboter Informationen über seinen eigenen Zustand und die ihn umgebende Welt. Hier stehen verschiedene Sensoren zur Wahl. Bei realer Hardware können alle (max. 4) verbaut werden. Für den Roboter in der Simulationsansicht sind Anzahl und Art beschränkt. Zur Verfügung stehen jeweils ein: • Kreiselsensor • Berührungssensor • Farbsensor • Ultraschallsensor
 Abbildung 14 [1)]	Aktoren dienen der Zustandsänderung des Roboters oder der Veränderung der Welt. Hier stehen verschiedene Aktoren zur Wahl. Bei realer Hardware können alle in unterschiedlicher Zahl und Kombination (max. 4) verbaut werden. Für den Roboter in der Simulationsansicht dienen die Aktoren der Mobilität. Es stehe zwei große Motoren zur Verfügung, die wie ein Differential arbeiten.
 Abbildung 15 [1)]	Die grundlegenden Eigenschaften (Raddurchmesser, Spurweite) des Roboters haben Einfluss auf das Fahrverhalten. Der Raddurchmesser bestimmt, welcher Weg bei einer Umdrehung zurückgelegt wird. Die Spurweite hat Einfluss auf das Drehverhalten. Je größer die Spurweite ist, desto mehr Weg müssen die Räder für eine 360°-Drehung zurücklegen.

Alle mit [1)] gekennzeichneten Grafiken sind Screenshots der Seite: https://lab.open-roberta.org

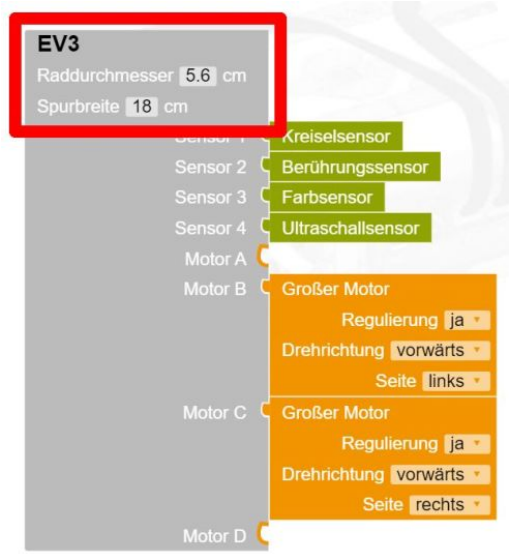


Abbildung 16 [1)]

Stellen Sie für die Simulationsansicht

- die Spurweite auf 18 cm und
- den Raddurchmesser auf 5,6 cm.

Hinweis:

- Es existieren im Deutschen und Englischen unterschiedliche Dezimaltrennzeichen.
  - Deutsch: 5,6 cm
  - Englisch: 5.6 cm

  Da in der Informatik die englische Schreibweise verwendet wird, findest du dies auch im Open Roberta Lab.

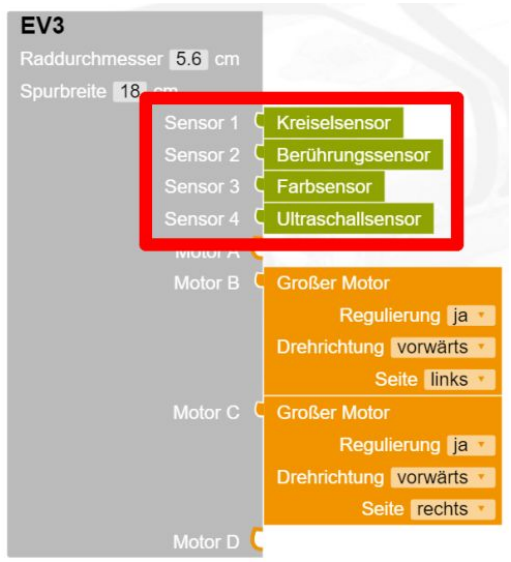


Abbildung 17 [1)]

Bei realer Hardware können alle Ports mit allen Sensoren bestückt werden. Diese können im Open Roberta Lab auch angesprochen werden.

In der Simulationsansicht können folgende Sensoren (von jeder Sorte einer) beliebig angeschlossen werden:

- Kreiselsensor
- Berührungssensor
- Farbsensor
- Ultraschallsensor

Schließen Sie die Sensoren entsprechend der Abbildung 17 durch Öffnen des Sensor-Depots mittels Drag & Drop an.

Hinweis:

- Im Programm müssen die Sensoren den korrekten Steckplätzen zugeordnet sein. Wenn Programme geteilt werden sollen bzw. reale Hardware benutzt wird, kann eine Nichtbeachtung problematisch sein.

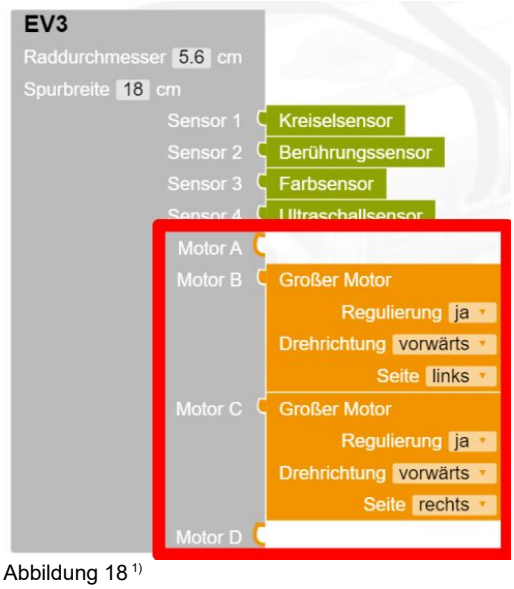


Abbildung 18 [1)]

Bei realer Hardware können alle Ports mit allen Aktoren bestückt werden.

Für den Roboter in der Simulationsansicht stehen jedoch nur 2 große Motoren zur Verfügung. Diese können beliebig angeschlossen werden.

Schließen Sie die Aktoren entsprechen der Abbildung 18 durch Öffnen des Aktor-Depots per Drag & Drop an und konfigurieren Sie die Parameter.

Hinweis:

- Im Programm müssen die Aktoren den korrekten Steckplätzen zugeordnet sein. Wenn Programme geteilt werden sollen bzw. reale Hardware benutzt wird, kann eine Nichtbeachtung problematisch sein.

3. Optional: Benutzerkonto anlegen

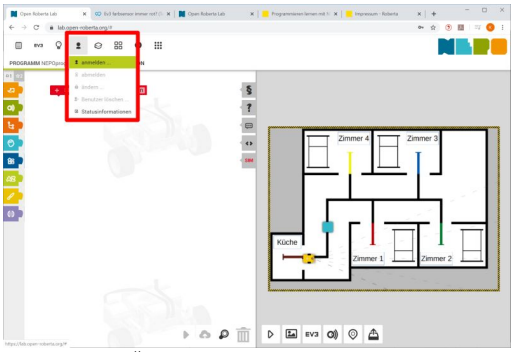


Abbildung 19 [1)]

Das Anlegen eines Benutzerkontos ist nicht zwingend notwendig aber komfortabel.

Durch Klick auf den Button „Benutzer“ und anschließend auf „anmelden“ gelangen Sie in den Anmeldedialog.

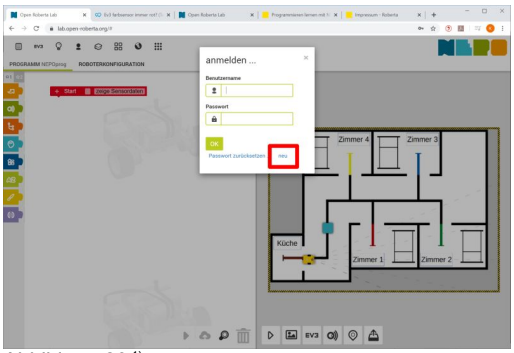


Abbildung 20 [1)]

Klicken Sie hier auf „neu“.

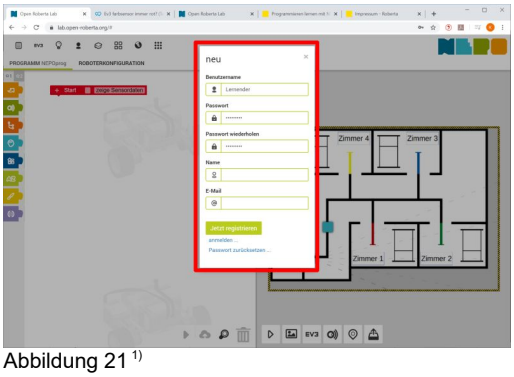


Abbildung 21 [1)]

Bei der Anmeldung ist lediglich ein Spitzname und ein Passwort nötig. Anschließend auf den Button „Jetzt registrieren“ klicken.

Damit ist ein nicht verifizierter Account angelegt, mit dem die Programme online gespeichert werden können und das Passwort geändert werden kann.

Hinweise:

- Spitzname und Passwort bitte gut merken, sonst sind die Daten verloren.
- Wenn Sie über 16 Jahre alt sind und Ihre Mailanschrift eintragen oder die Mailanschrift Ihrer Eltern eintragen, können Sie einen verifizierten Account anlegen, der weiterreichende Funktionen bietet.

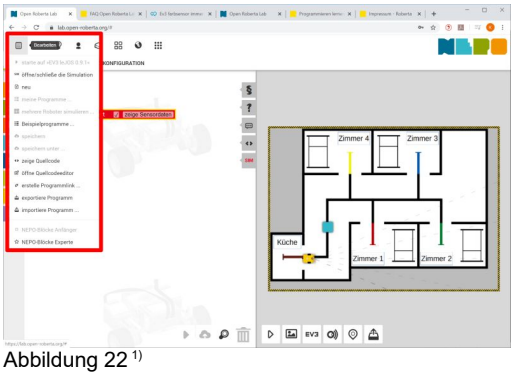


Abbildung 22 [1)]

Über den Button „Bearbeiten“ steht Ihnen das Dateimanagement zur Verfügung.

Sie können auch unangemeldet/accountlos Ihre Programme exportieren und importieren. Dazu ist ein externer Datenträger (z.B. USB-Stick) notwendig. Diese Anleitung beschreibt die Arbeitsschritte bei accountloser Nutzung des Open Roberta Lab.

Wenn Sie einen Account besitzen, können Sie Ihre Daten auch online speichern bzw. öffnen. Voraussetzung ist eine funktionierende Internetverbindung.

4. Sequenzielle Programmierung

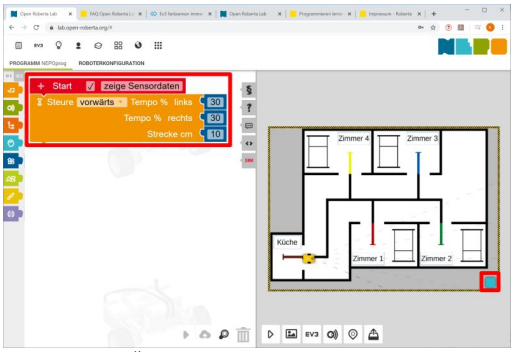


Abbildung 23 [1)]

Platzieren Sie das Hindernis in der unteren rechten Ecke der Simulationsanzeige.

Holen Sie sich aus der Blockgruppe „Aktion“ den Baustein „Steure“ per Drag & Drop und hängen Sie ihn an den Baustein „Start“ im Programmierbereich.

Stellen Sie die Parameter entsprechend der Abb. 23 ein:

- Richtung auf vorwärts
- Tempo % links auf 30
- Tempo % rechts auf 30
- Strecke cm auf 10

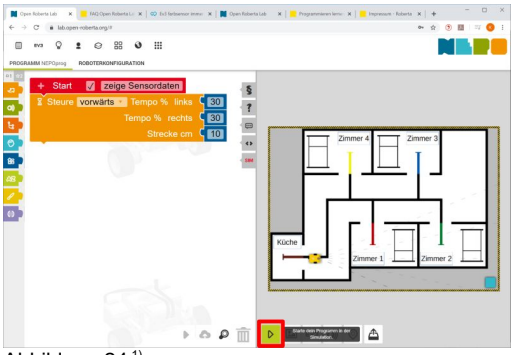


Abbildung 24 [1)]

Klicken Sie in der Simulationsanzeige auf den Button „Starte dein Programm“ und beobachten Sie den Roboter.

Der Roboter ist jetzt ein Stück vorgefahren.

Versuchen Sie es erneut:

- Platzieren Sie den Roboter an seiner Ausgangsposition,
- Verändern Sie den Wert des Parameters „Strecke cm“ so, dass der Roboter bis zur Ecke vorfährt.

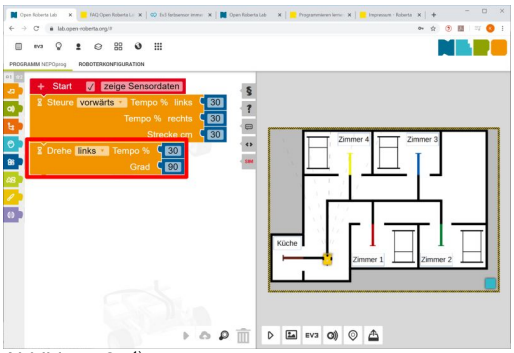


Abbildung 25 [1)]

Holen Sie sich aus der Blockgruppe „Aktion“ den Baustein „Drehe“ und hängen Sie ihn an den Baustein „Steure“ im Programmierbereich.

Stellen Sie die Parameter entsprechend der Abb. 25 ein:

- Richtung auf links
- Tempo % links auf 30
- Grad auf 90

Platzieren Sie den Roboter an seiner Ausgangsposition.

Klicken Sie in der Simulationsanzeige auf den Button „Starte dein Programm“ und beobachten Sie den Roboter.

Hinweis:

- Wenn der Roboter eine Position erreicht hat, die eine Neuausrichtung des Roboters erfordert, können Sie durch Klick auf den Button „Ausgangsposition“ im Aktionsbereich der Simulationsanzeige den Roboter wieder in seine Ausgangslage bringen.

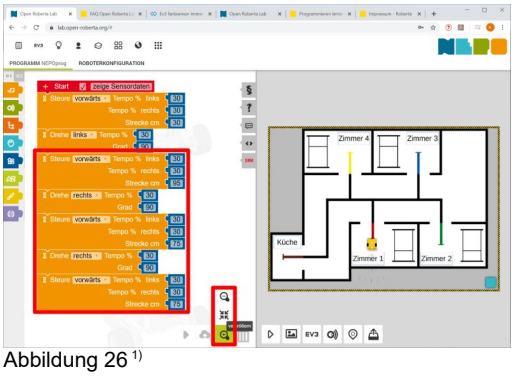


Abbildung 26 [1)]

Ergänzen Sie das Programm so, dass der Roboter bis in die Mitte des Zimmers 1 fährt.

Hinweise (Bitte unbedingt ausprobieren!):

- Wenn der Platz für das Programm knapp wird kann über den Button „Zoomen“ die Darstellung angepasst werden.
- Das Programm kann durch Halten des Linksklicks im weißen Bereich des Programmierbereiches gegriffen und geschoben werden.

Alle mit [1)] gekennzeichneten Grafiken sind Screenshots der Seite: https://lab.open-roberta.org

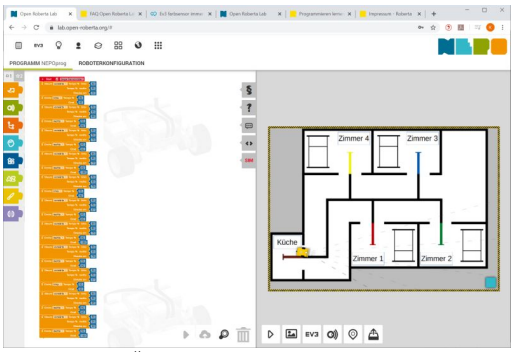


Abbildung 27 [1)]

Aufgaben:

1. Erweitern Sie Ihr Programm. Lassen Sie Ihren Roboter auf dem Rückweg zur Küche noch das Zimmer 4 besuchen, bevor er sich wieder in seiner Ausgangsposition befindet.
2. Beschleunigen Sie die Fahrt des Roboters durchgängig.
3. Exportieren Sie das fertige Programm unter dem Namen „Sequenz“ auf Ihrem externen Datenspeicher.
4. Lassen Sie Ihre Arbeit durch die Lehrkraft bewerten.

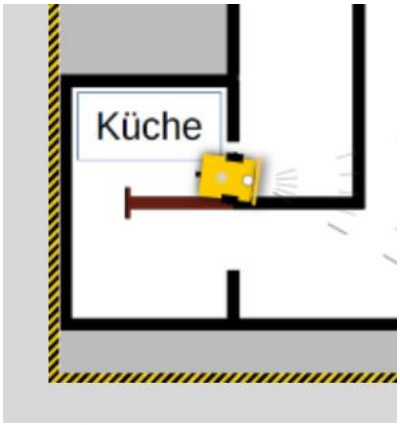


Abbildung 28 [1)]

Aufgaben (Nutzen Sie eine Textverarbeitung und speichern Sie die Datei unter dem Namen „Sequenz_Reflexion“ auf einem externen Datenspeicher.):

5. Benennen Sie Gründe dafür, warum der Roboter nie 100% richtig stehen wird.
6. Wie schätzen Sie den Programmieraufwand ein, wenn alle Zimmer beliefert werden sollen?
7. Was würde es bedeuten, wenn täglich andere Patienten mit Mittagessen versorgt werden sollen?

Alle mit [1)] gekennzeichneten Grafiken sind Screenshots der Seite: https://lab.open-roberta.org

5. Experimente mit den Sensoren

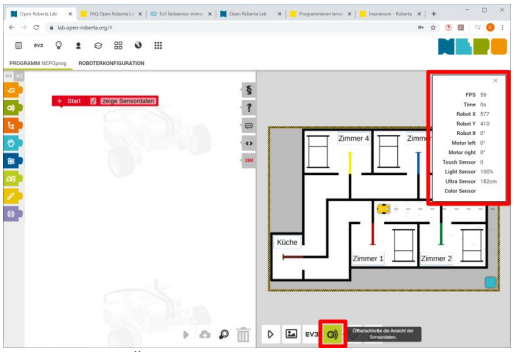


Abbildung 29 [1)]

Blenden Sie das Fenster „Sensordaten“ durch Klick auf den Button „Sensordaten“ im Aktionsmenü der Simulationsansicht ein.

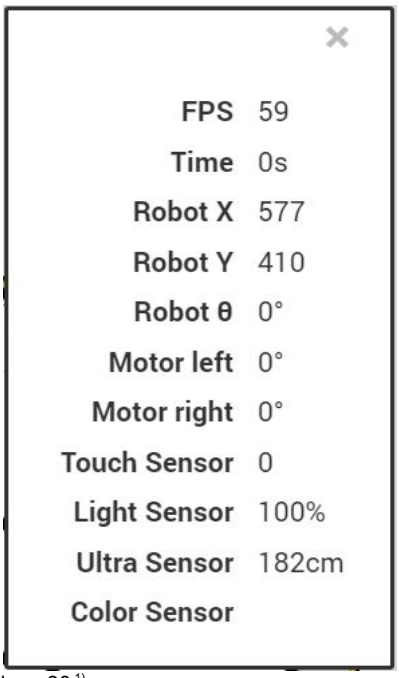


Abbildung 30 [1)]

FPS ist eine Abkürzung für „Frames Per Second“ und bedeutet, wie oft die Simulationsanzeige pro Sekunde aktualisiert wird.

Time steht für die Laufzeit Ihres Programms.

Es folgen unterschiedliche Messwerte, die einzelnen Sensoren zugeschrieben werden:

- Kreiselsensor
  (Messgenauigkeit +/- 3° bei einer 90°-Drehung)
  - **Robot X**
  - **Robot Y**
  - **Robot θ** (Theta)
- Rotationssensor (im Motor verbaut)
  (Messgenauigkeit 1°)
  - **Motor left**
  - **Motor right**
- Berührungssensor
  - **Touch Sensor**
- Farbsensor
  - **Color Sensor**
  - **Light Sensor**
- Ultraschallsensor
  - **Ultra Sensor**

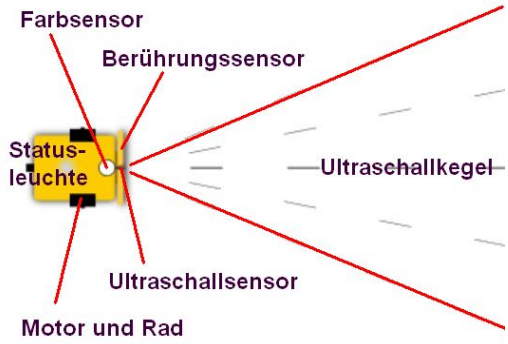


Abbildung 31

Der Roboter ist bereits aus dem Kapitel 3 bekannt. Es lohnt sich ein genauerer Blick.

Die beiden Motoren und Räder treiben den Roboter an. Die Messwerte „Motor left“ und „Motor right“ zeigen jeweils die Umdrehungen in Grad an (1 Umdrehung = 360°).

Der Berührungssensor kann nur zwei Werte messen (Touch Sensor = 0 = nicht ausgelöst, 1 = ausgelöst).

Der Farbsensor kann in verschiedenen Modi betrieben werden:

- „Color Sensor“ zeigt die bestimmte Farbe (keine Farbe = Grau, Schwarz, Blau, Grün, Gelb, Rot, Weiß, Braun)
- „Light Sensor“ zeigt die Menge des reflektierten Lichtes (ein kleiner Wert steht für dunkle Flächen)

Der Ultraschallsensor misst die Distanz zum nächstgelegenen Objekt innerhalb des Ultraschallkegels (0 bis 255 cm).

Alle mit [1)] gekennzeichneten Grafiken sind Screenshots der Seite: https://lab.open-roberta.org

Abbildung 32 [1)]

Schreiben Sie folgendes Programm und testen Sie es in der Simulationsanzeige.

Tipp:

- Die Farben der einzelnen Blöcke geben Aufschluss über die Blockgruppe

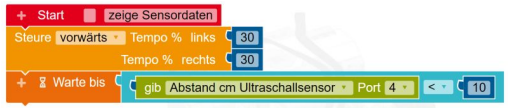


Abbildung 33 [1)]

Ändern Sie das Programm so, dass der Roboter 10 cm vor dem Hindernis zum Stehen kommt.

Beobachten Sie die Sensordaten.

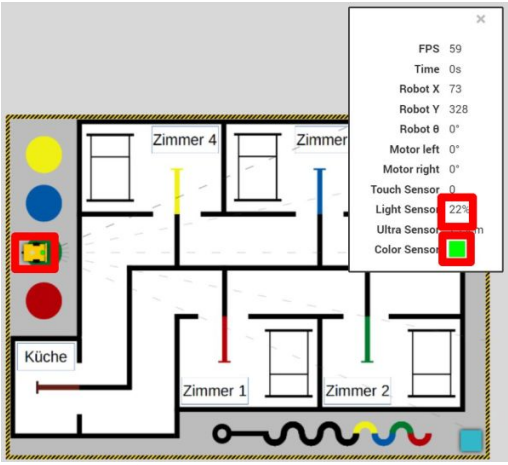


Abbildung 34 [1)]

Bewegen Sie den Roboter über die Farbkreise bzw. weitere Farbmarkierungen in der Simulationsansicht und beobachten Sie den Farbsensor am Roboter sowie die Sensordaten.

Öffnen Sie eine externe Textverarbeitung und erstellen Sie sich nachfolgende Tabelle und ergänzen Sie das reflektierte Licht in Prozent:

Farbnummer	Farbe	Reflexion [%]
0	keine Farbe (Grau)	
1	Schwarz	
2	Blau	
3	Grün	
4	Gelb	
5	Rot	
6	Weiß	
7	Braun (Küche)	

Speichern Sie die Datei unter dem Namen „Farbsensor_Reflexion“ auf einem externen Datenspeicher.

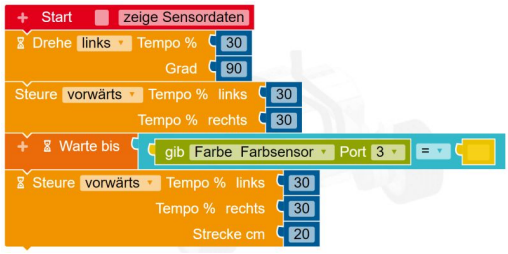


Abbildung 35 [1)]

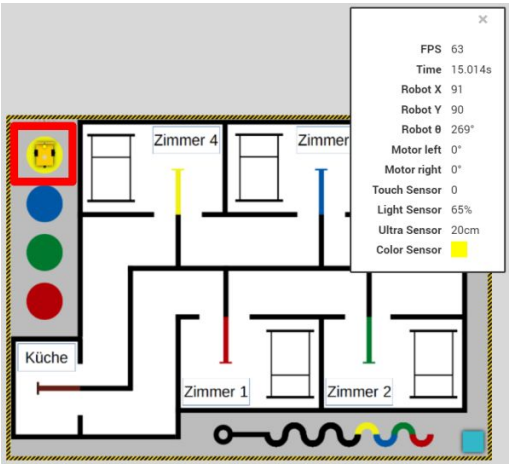


Abbildung 36 [1)]

Aufgaben:

8. Schreiben Sie ein Programm, dass den Roboter bis zur gelben Farbinsel fahren lässt.
9. Ändern Sie das Programm so, dass auch die anderen Farbinseln angefahren werden können.
10. Ergänzen Sie jetzt die notwendigen Anweisungen im Programm, damit der Roboter über jeder Farbinsel eine 360°-Drehung ausführt und danach zur nächsten Insel aufbricht, bis er die gelbe Insel erreicht.
11. Exportieren Sie das fertige Programm unter dem Namen „Farbsensor“ auf einen externen Datenspeicher.
12. Lassen Sie Ihre Arbeit durch die Lehrkraft bewerten.

6. Wiederholung - Schleife

Abbildung 37 [1)]

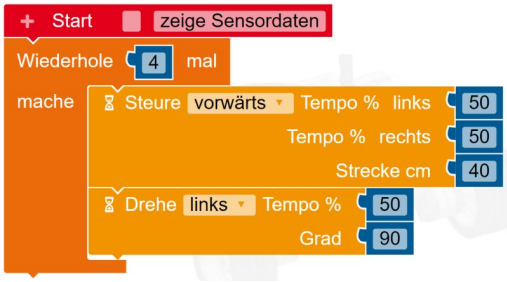


Abbildung 38 [1)]

Sollen Anweisungen mehrfach ausgeführt werden, spricht man von einer Wiederholung bzw. Schleife. Alle Schleifen haben Abbruchbedingungen. Wenn keine Abbruchbedingung formuliert ist oder die Abbruchbedingung nie erreicht wird, spricht man von Endlosschleifen. Sie können nur von außen durch Programmabbruch beendet werden.

Es gibt verschiedene Schleifen:

- die Anzahl der Wiederholungen ist im Vorfeld bekannt (Fachleute sagen for-Schleife)
- die Anzahl der Wiederholungen ist im Vorfeld nicht bekannt (Fachleute sagen while-Schleife)

Ein Beispiel:
Der Roboter soll im Quadrat mit einer Kantenlänge von 40 cm fahren.

Nach Ihrem bisherigen Kenntnisstand sieht Ihr Programm sicher so aus: Siehe Abb. 37.

Eleganter geht es so: Siehe Abb. 38.
In diesem Fall ist die Anzahl der Wiederholungen im Vorfeld bekannt, was auch aus den sich 4 Mal wiederkehrenden Anweisungen in Abb.37 ersichtlich ist. Die Abbruchbedingung ist der 4. Durchlauf der Schleife.

Aufgabe:

13. Schreiben Sie beide Programme und vergleichen Sie die Roboterfahrt.

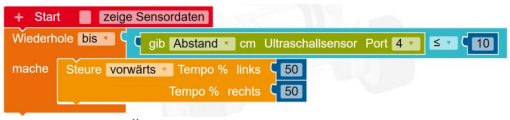


Abbildung 39 [1)]

Schleifen, bei denen die Anzahl der Wiederholungen im Vorfeld nicht bekannt ist, sind für die Nutzung von Sensoren geeignet.

Erstellen Sie das in Abb. 39 gezeigte Programm, testen Sie es in der Simulationsanzeige und finden Sie heraus, wozu es dient.

Aufgaben:

14. Schreiben Sie ein Programm, dass den Roboter anweist, an der ersten überquerten roten Fläche anzuhalten.

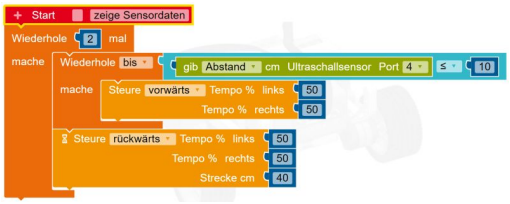


Abbildung 40 [1)]

Schleifen können auch verschachtelt werden.

Das in Abb. 40 dargestellte Programm lässt den Roboter zweimal auf das Hindernis zufahren. Testen Sie es.

Alle mit [1)] gekennzeichneten Grafiken sind Screenshots der Seite: https://lab.open-roberta.org

7. Entscheidungen: Bedingte Anweisungen – Verzweigungen - Schalter

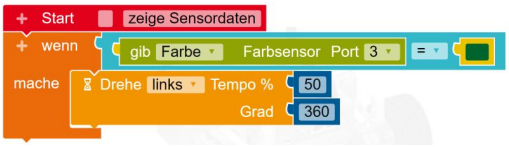


Abbildung 41 [1)]

Entscheidungen entstehen durch Fragen, die mit „Ja“ oder „Nein“ beantwortet werden können.

Beispiel bedingte Anweisung:

Sieht der Farbsensor „Grün“?
- „Ja“, der Roboter dreht 360° links,

Erzeugen Sie das in der Abb. 41 dargestellte Programm und testen Sie es in der Simulationsanzeige. Stellen Sie dabei den Roboter über unterschiedliche Farbkreise und starten Sie das Programm.

Fachleute sagen dazu „if-Anweisung“.

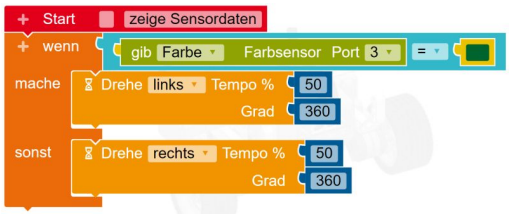


Abbildung 42 [1)]

Beispiel Verzweigung:

Sieht der Farbsensor „Grün“?
- „Ja“, der Roboter dreht 360° links,
- „Nein“ der Roboter dreht 360° rechts.

Erzeugen Sie das in der Abb. 42 dargestellte Programm und testen Sie es in der Simulationsanzeige. Stellen Sie dabei den Roboter über unterschiedliche Farbkreise und starten Sie das Programm.

Fachleute sagen dazu „if-else-Anweisung“.

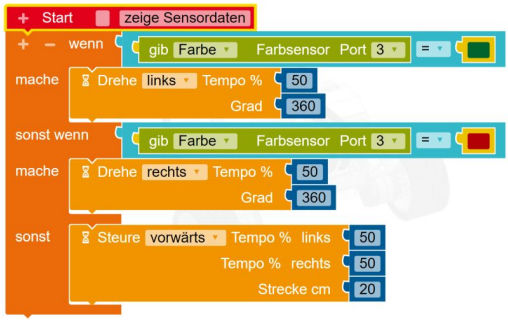


Abbildung 43 [1)]

Mehrfache Verzweigung:

Sieht der Farbsensor „Grün“?
- „Ja“, der Roboter dreht 360° links,

Sieht der Farbsensor „Rot“?
- „Ja“, der Roboter dreht 360° rechts,
- „Nein“ der Roboter fährt 20 cm vorwärts.

Erzeugen Sie das in der Abb. 43 dargestellte Programm und testen Sie es in der Simulationsanzeige. Stellen Sie dabei den Roboter über unterschiedliche Farbkreise und starten Sie das Programm.

Tipp:
- Die mehrfache Verzweigung entsteht, wenn Sie auf das „+“ der Verzweigung klicken.

Fachleute sagen dazu „Switch-Anweisung“.

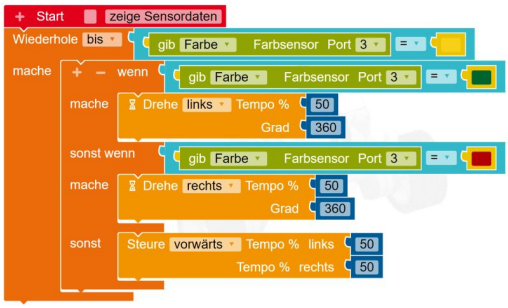


Abbildung 44 [1)]

Verzweigungen können in Schleifen gesetzt werden.

Wiederhole bis der Farbsensor gelb sieht:
- Sieht der Farbsensor „Grün“?
  - „Ja“, der Roboter dreht 360° links,
- Sieht der Farbsensor „Rot“?
  - „Ja“, der Roboter dreht 360° rechts,
  - „Nein“ der Roboter fährt 20 cm vorwärts.

Erzeugen Sie das in der Abb. 44 dargestellte Programm und testen Sie es in der Simulationsanzeige. Stellen Sie dabei den Roboter mehrfach über unterschiedliche Farbkreise, bevor Sie Ihn über Gelb stellen. Testen Sie das Verhalten ohne Neustart des Programms danach nochmals.

8. Variablen und Operatoren

Abbildung 45 [1)]

Aus der Mathematik und den Naturwissenschaften kennen Sie Variablen. Beim Programmieren werden sie auch häufig benutzt.

Variablen sind Container (Platzhalter) für Werte. Sie bestehen aus:

- Name (z.B. „Element1“)
- Typ (z.B. „Zahl“)
- Wert (z.B. „0“)

Im Unterschied zu Ihrem bisherigen Verständnis können die Variablennamen nicht nur Buchstaben sondern auch Zeichenketten sein.

Wichtige Datentypen sind:

- Ganze Zahlen, z.B. -1 oder 0 oder 13
- Logischer Wert, z.B. wahr oder falsch
- Gleitkommazahl, z.B. -0,4 oder 0,0 oder 8,26

Im Open Roberta Lab finden Sie auch:

- Farbe
- Zeichenkette
- ...

Bevor Variablen benutzt werden können, müssen sie definiert werden. Dies geschieht im Open Roberta Lab durch das „+“ im Startblock. Das Entfernen geht über das „-“ der Variablen.

Deklarieren Sie 4 Variablen entsprechend der Abb. 45.

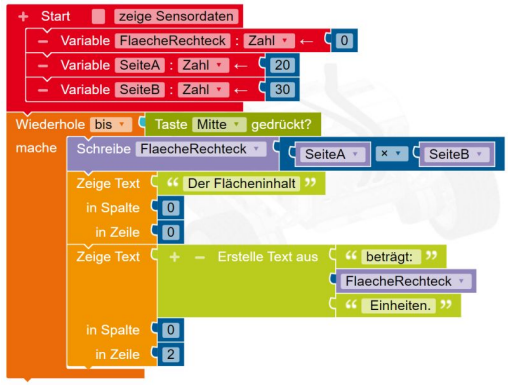


Abbildung 46 [1)]

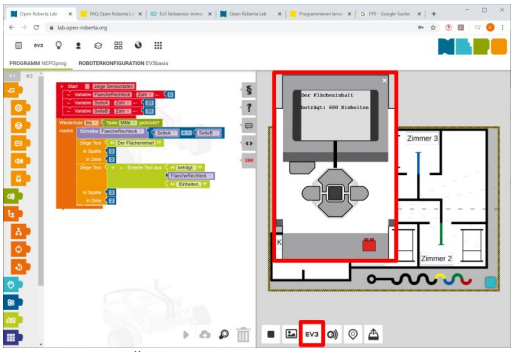


Abbildung 47 [1)]

Operatoren benötigt man für:

- Zuweisungen (=)
- Vergleiche (>, <, >=, <=, ...)
- Ausdrücke (+, -, *, /, UND, ODER, ...)

Erzeugen Sie das Programm in Abb. 46. Öffnen Sie die Simulationsanzeige und blenden Sie die Controller-Ansicht durch Klick auf „EV3“ im Aktionsbereich ein (Abb. 47). Starten Sie anschließend das Programm. Mit Druck auf die mittlere Taste am Controller wird das Programm beendet.

Aufgaben:

15. Ergänzen Sie das Programm so, dass in den Zeilen 4 und 5 die Zeichenkette „Das Ergebnis ist größer als 400 Einheiten“ angezeigt wird. Testen Sie Ihr Programm.
16. Verändern Sie die Werte von „SeiteA“ und „SeiteB“ so, dass ein Flächeninhalt kleiner 400 Einheiten entsteht. Testen Sie das Programm.
17. Verändern Sie das Programm so, dass die Meldung über ein Ergebnis größer 400 Einheiten nur gezeigt wird, wenn es „wahr“ ist.
18. Exportieren Sie das fertige Programm unter dem Namen „Operatoren“ auf einen externen Datenspeicher.

Beachten Sie, dass bei diesem Programm die Variablen nur jeweils mit einem Programmstart ausgewertet werden.

Alle mit [1)] gekennzeichneten Grafiken sind Screenshots der Seite: https://lab.open-roberta.org

9. Der Linienfolger

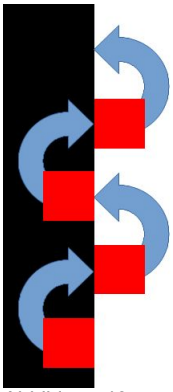


Abbildung 48

Roboter mit einem Farbsensor können an der Kante von 2 Farben fahren. Ein einfaches Funktionsprinzip ist in Abb. 48 dargestellt:

- Sieht der Farbsensor „Schwarz“, dreht sich der Roboter nach rechts.
- Sieht der Farbsensor „Weiß“, dreht sich der Roboter nach links.
- Diese Anweisungen werden ständig wiederholt.

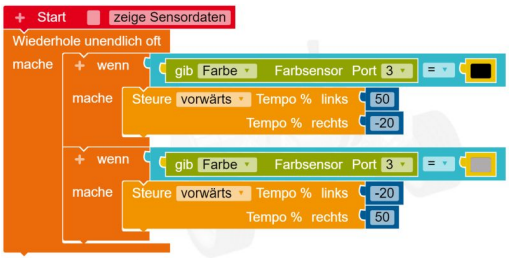


Abbildung 49 [1)]

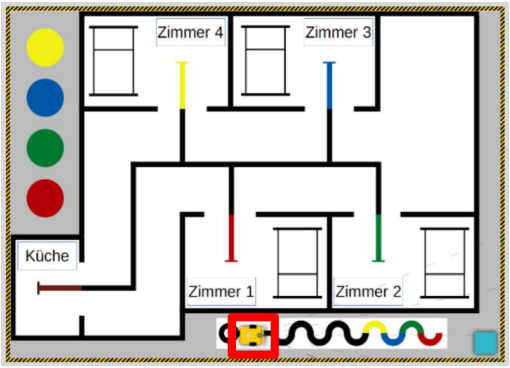


Abbildung 50 [1)]

Erstellen Sie das in Abb. 49 gezeigte Programm.

Platzieren Sie den Roboter wie in Abb. 50 gezeigt und testen Sie Ihr Programm.

Aufgaben:

19. Beobachten Sie die Fahrt des Roboters und finden Sie dafür ein Wort der Beschreibung.
20. Verändern Sie die Werte für „Tempo % links“ und „Tempo % rechts“ und versuchen Sie die Fahrt des Roboters zu optimieren.

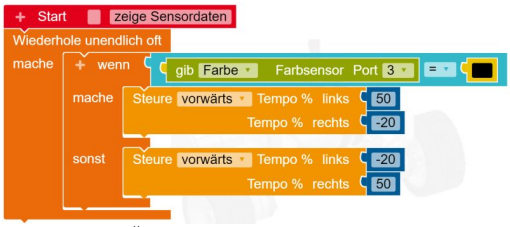


Abbildung 51 [1)]

Aufgaben:

21. Ändern Sie das Programm entsprechend der Abb. 51 ab.
22. Überlegen Sie, was sich jetzt geändert hat.
23. Ergänzen Sie das Programm so, dass der Roboter die gesamte Übungsstrecke absolviert.
24. Exportieren Sie das fertige Programm unter dem Namen „Linienfolger“ auf einen externen Datenspeicher.
25. Überlegen Sie, warum bei der mehrfachen Verzweigung der Fall „Grau“ definiert sein muss.
26. Notieren Sie Ihre Überlegungen in einer Textdatei und speichern Sie diese unter dem Namen „Linienfolger_Grau“ auf einen externen Datenspeicher.

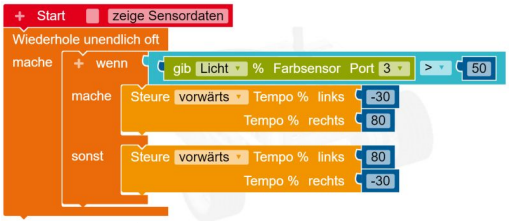


Abbildung 52 [1)]

Der Farbsensor kann auch im Modus „Licht“ arbeiten. In diesem Fall betreibt der Sensor eine eigene Lichtquelle und wertet die Menge des reflektierten Lichts aus (siehe Datei „Farbsensor“, Abb. 34). Man spricht dann auch von einem Licht- bzw. Reflexionssensor.

Um mit den Lichtwerten arbeiten zu können, wird ein Schwellwert bestimmt, der genau mittig zwischen den beiden Farben liegt.

In unserem Fall soll der Roboter der schwarzen Linie auf weißem Grund folgen. Da sich für Schwarz ein Reflexionswert (Simulationsansicht, Sensordaten, Light Sensor) von „0 %“ und für Weiß ein Wert von „100 %“ ergibt, ist der Mittelwert „50 %“ unser Schwellwert.

Aufgaben:

27. Erzeugen Sie ein Programm entsprechend der Abb. 52 und testen Sie es auf der Übungsstrecke.
28. Überlegen Sie, warum der Roboter nicht die gesamte Strecke befährt.
29. Ändern Sie das Programm so, dass die gesamte Teststrecke befahren wird.
30. Exportieren Sie das fertige Programm unter dem Namen „Linienfolger_Licht“ auf einen externen Datenspeicher.

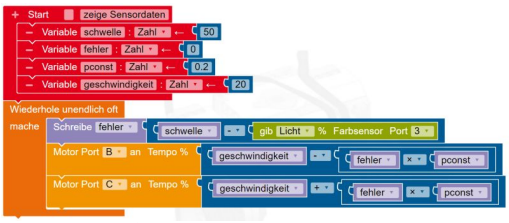


Abbildung 53 [1)]

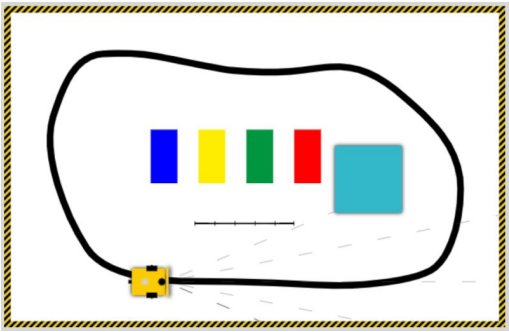


Abbildung 54 [1)]

Optional: PID-Regler (Spurhalteassistent)

Ein Nachteil der bisher beschriebenen Programme ist die „zapplige“ Fahrt des Roboters. Mit PID-Regler (3-facher Regler) wird es „geschmeidig“.

Ein Regler beeinflusst selbsttätig eine physikalische Größe, um Störungen auszugleichen.

Für das volle Verständnis des PID-Reglers wird Integral- und Differentialrechnung vorausgesetzt. Einfach erklärt:

P-Regler: (Proportional)
- ist der Hauptregler
- misst die Abweichung (Sollwert-Istwert)
- reagiert sofort

I-Regler: (Integral)
- hilft, wenn der P-Regler überfordert ist
- hat die Vergangenheit im Blick
- wirkt langfristig

D-Regler: (Differential)
- beobachtet Veränderungen des Eingangssignals
- versucht in die Zukunft zu schauen
- reagiert sofort auf Störungen

Das Beispiel zeigt einen Regler („fehler“). Der Fehler ergibt sich aus dem Sollwert („schwelle“) abzüglich dem Istwert (Sensorwert).

Aufgaben:

1. Schauen Sie sich bei YouTube das Video „Folge #6 Spurassistent/Linienfolger Lego EV3 – Programmieren feat. Daniel Jung & Sophie Charlotte“ an: https://www.youtube.com/watch?v=ykdaVHvayy8
2. Erzeugen Sie das Programm nach Abb. 53.

Alle mit [1)] gekennzeichneten Grafiken sind Screenshots der Seite: https://lab.open-roberta.org

	3. „Spielen“ Sie mit den Werten der Variablen, um eine homogene, flotte Fahrt zu erreichen (Umgebung nach Abb. 54). Veränderungen sind erst nach Neustart des Programms gültig.

Anhang: Vergrößerungen

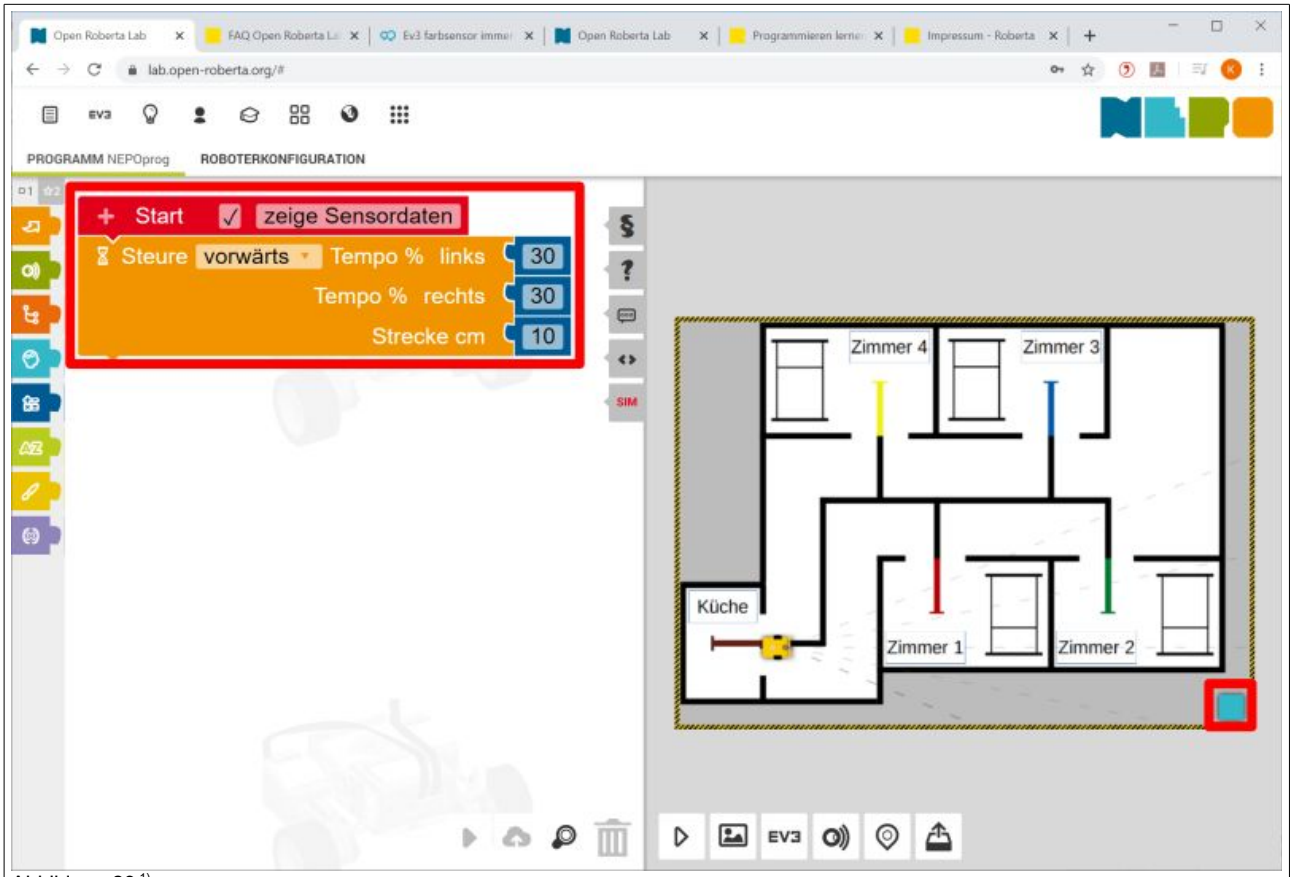


Abbildung 23 [1)]

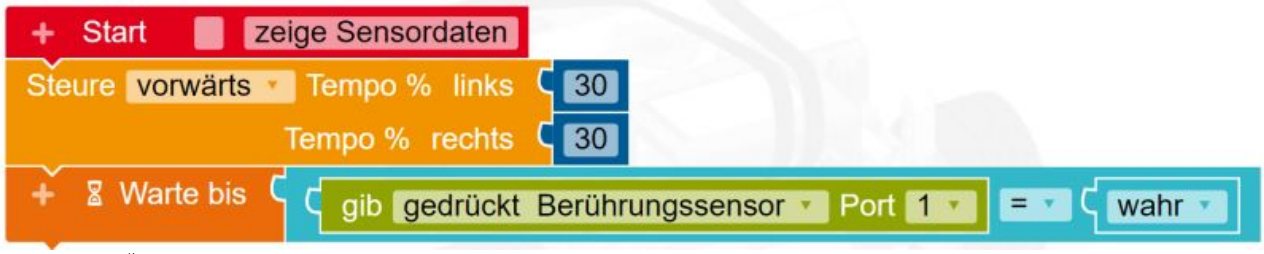


Abbildung 32 [1)]

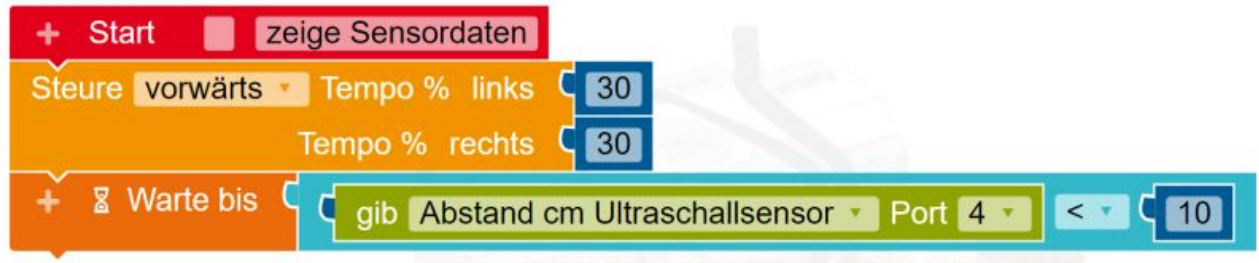


Abbildung 33 [1)]

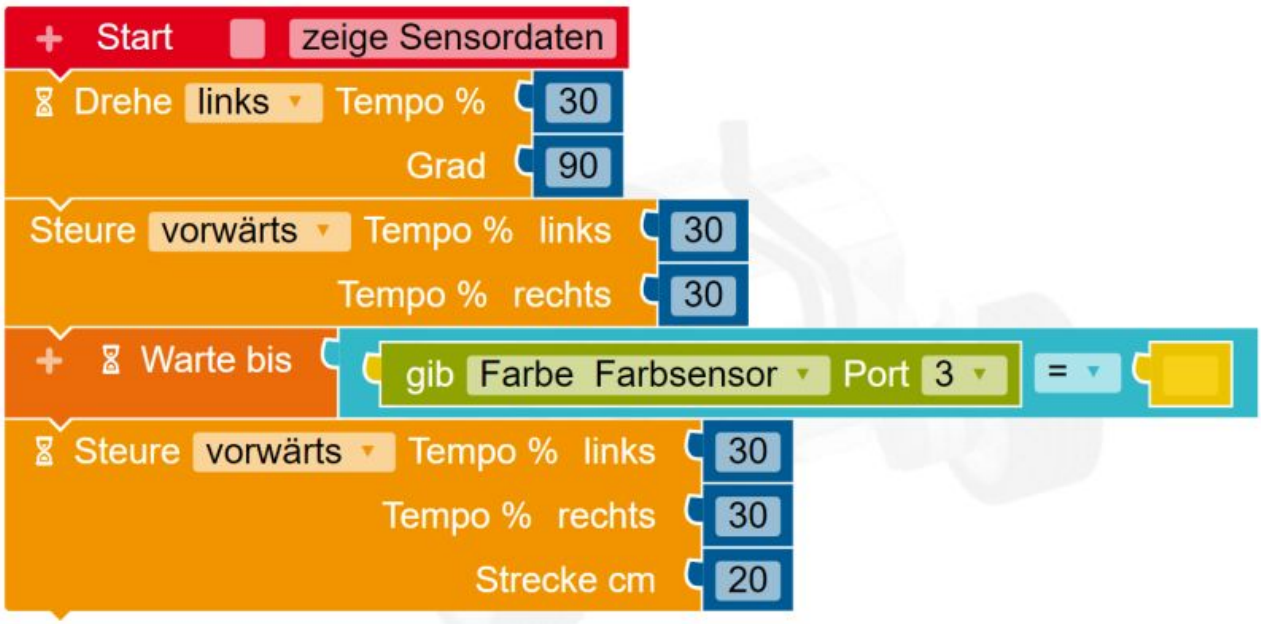


Abbildung 35 [1)]

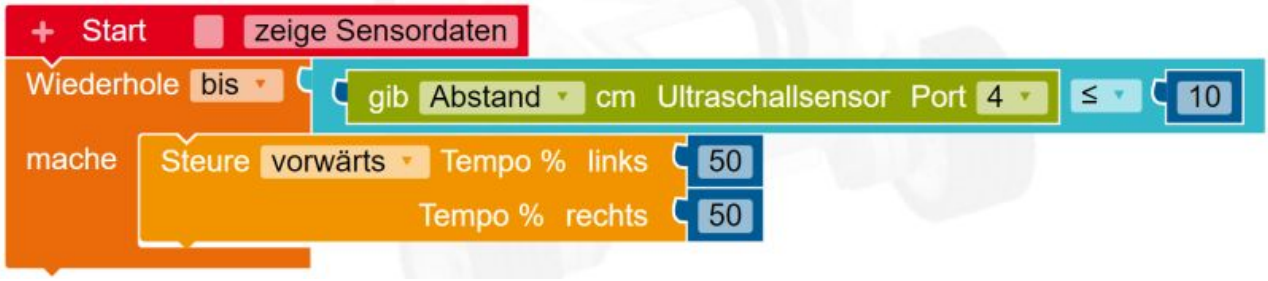


Abbildung 39 [1)]

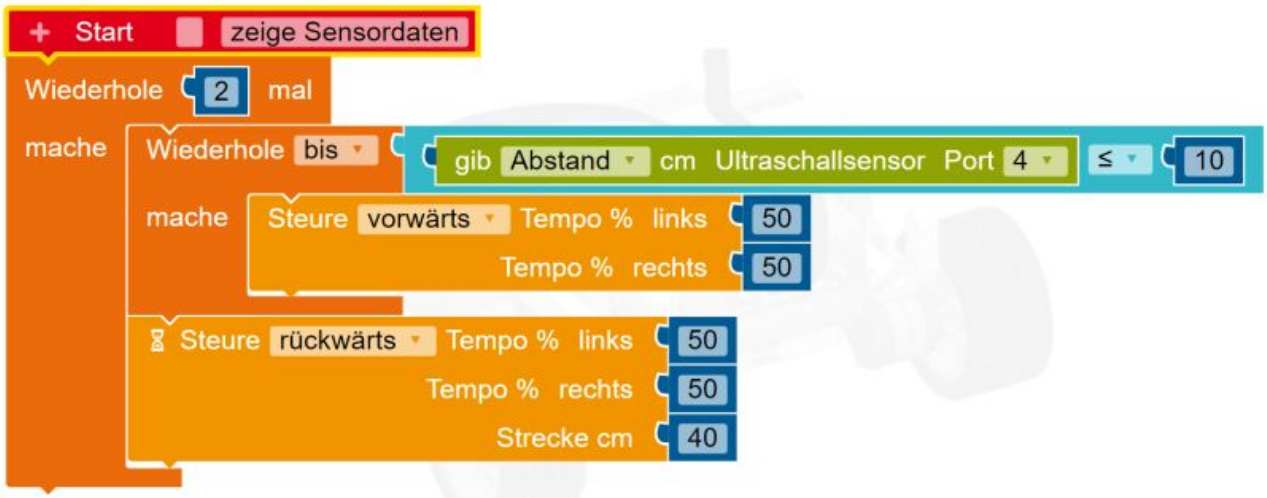


Abbildung 40 [1)]

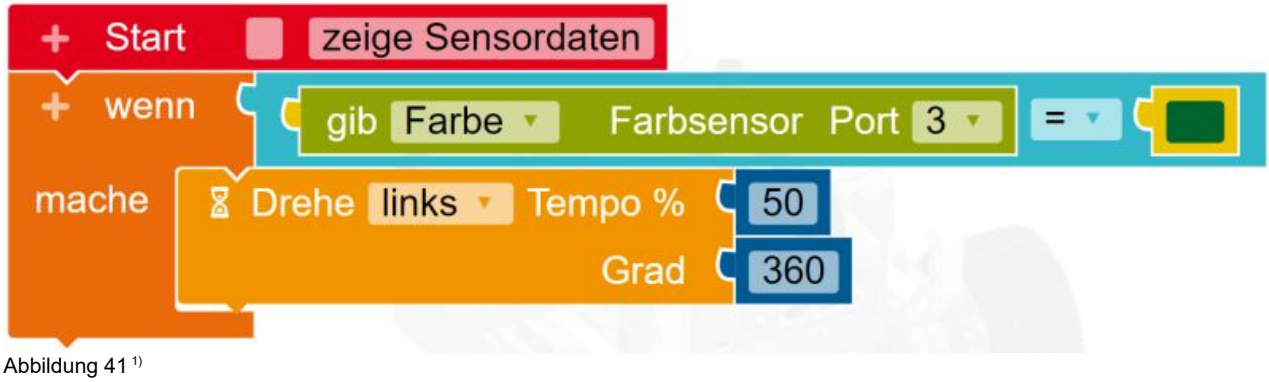


Abbildung 41 [1)]

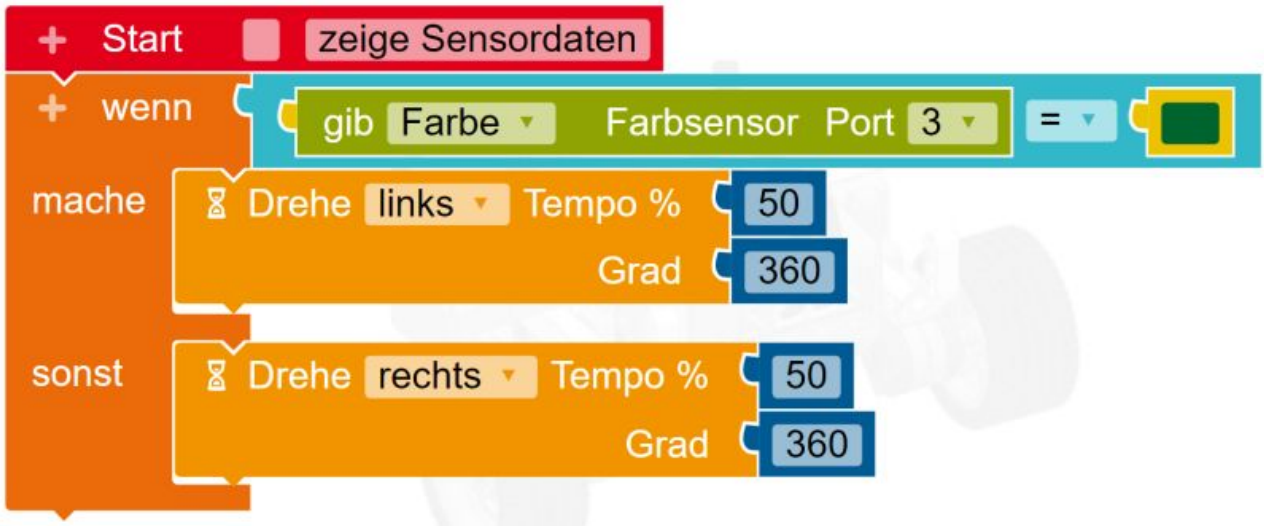


Abbildung 42 [1)]

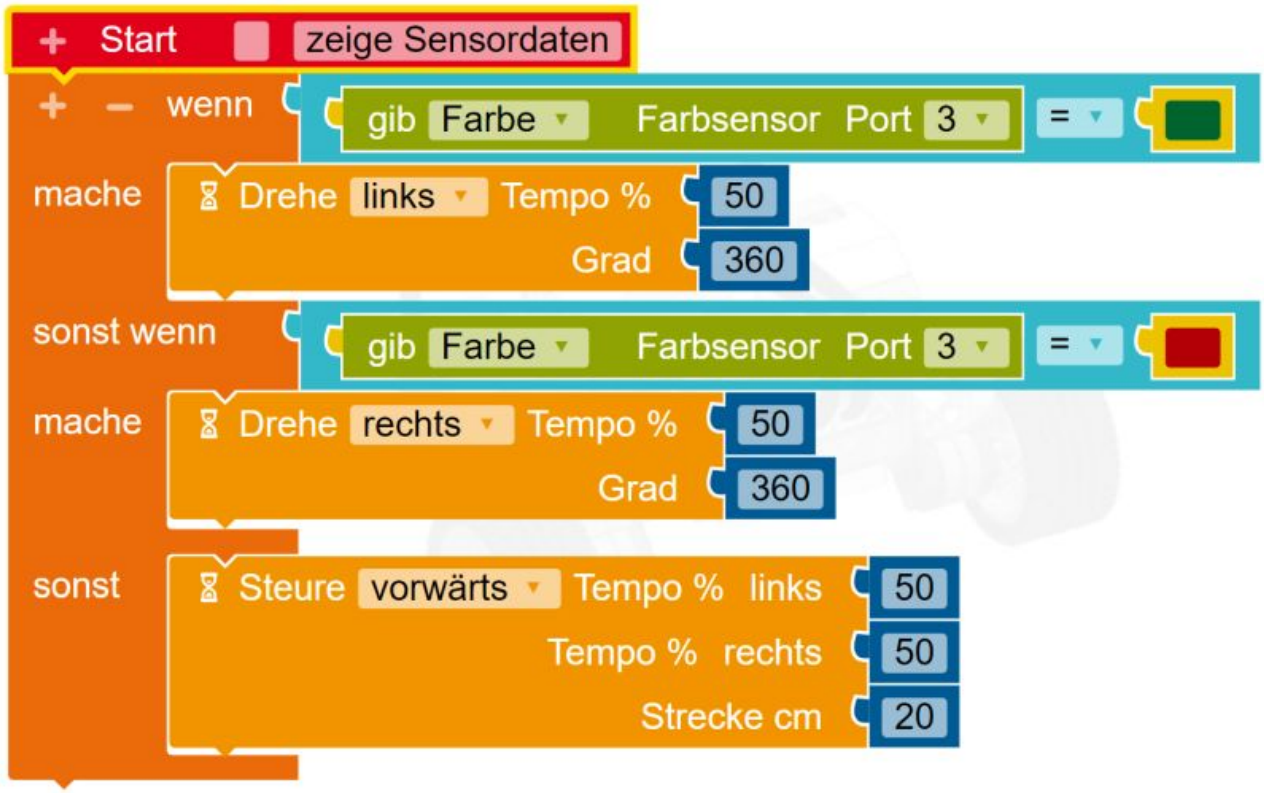


Abbildung 43 [1)]

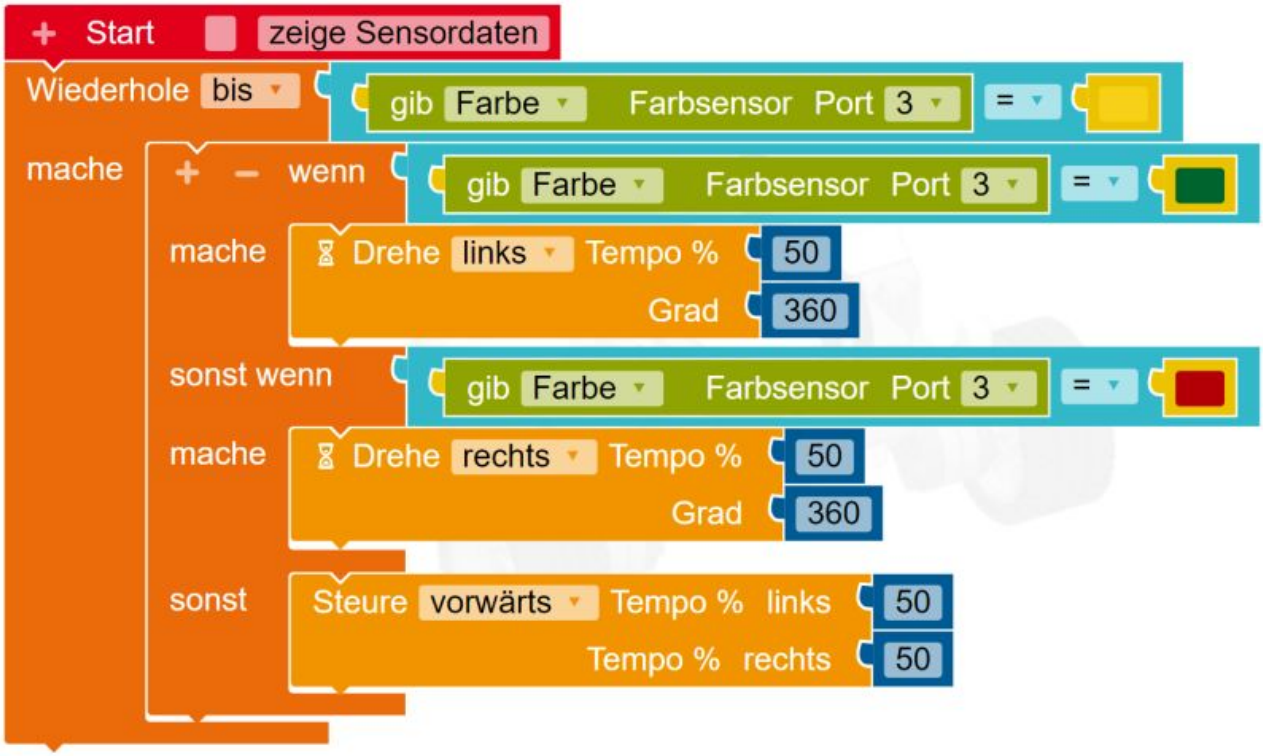


Abbildung 44 [1)]

Abbildung 45 [1)]

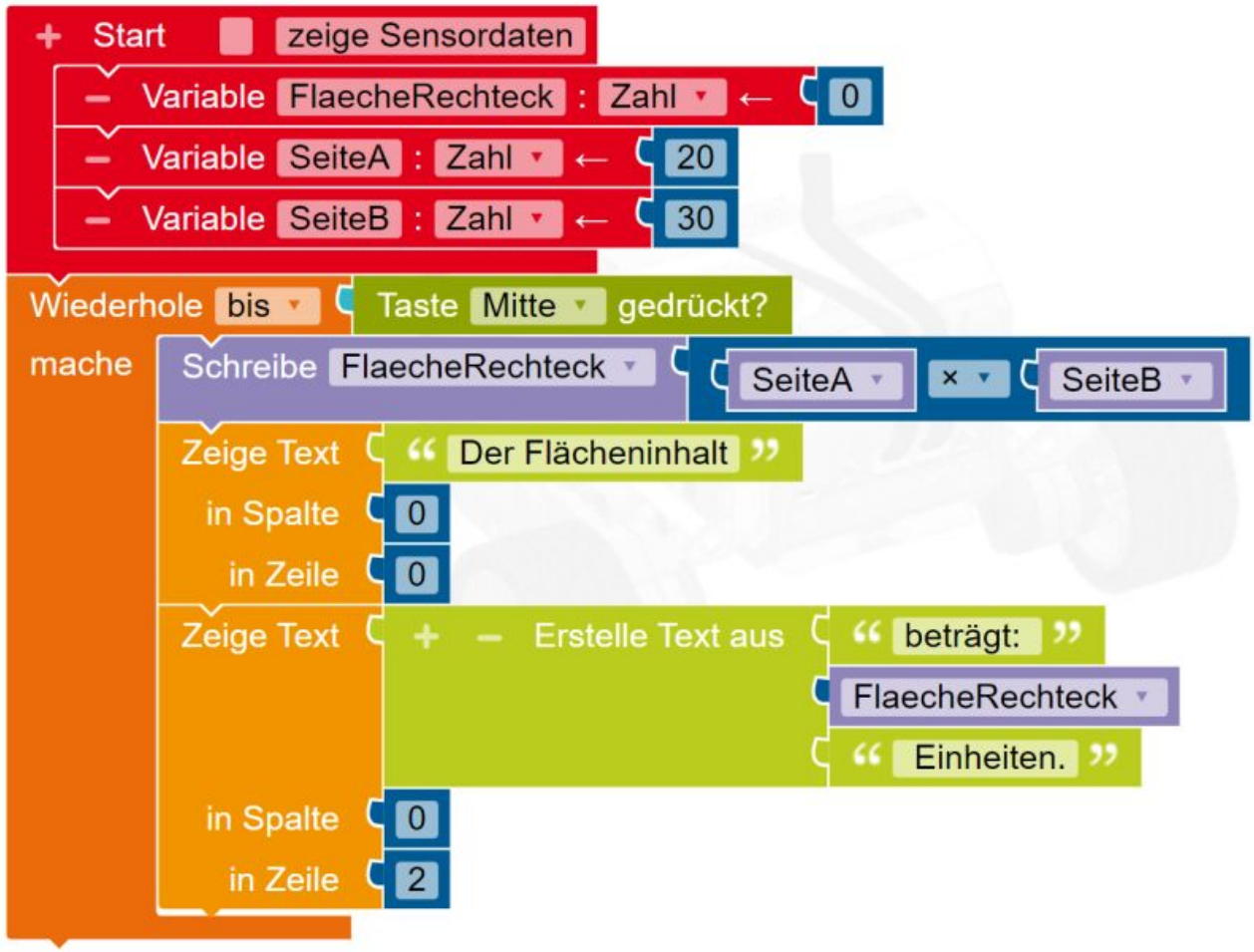


Abbildung 46 [1)]

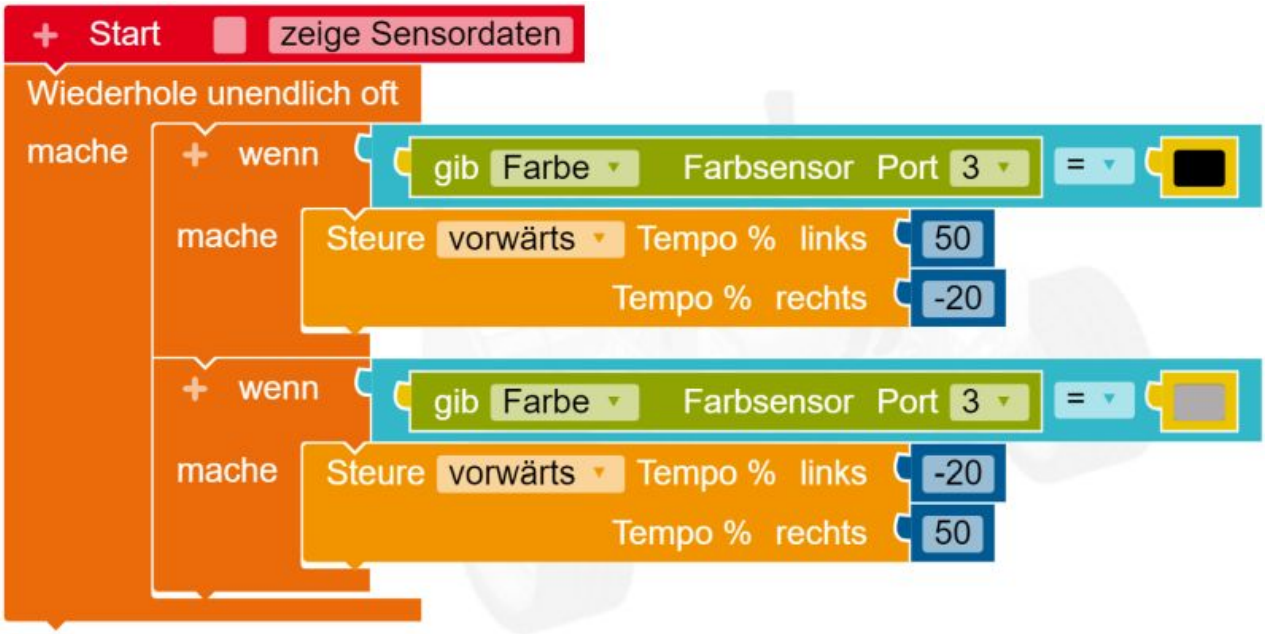


Abbildung 49 [1)]

Alle mit [1)] gekennzeichneten Grafiken sind Screenshots der Seite: https://lab.open-roberta.org

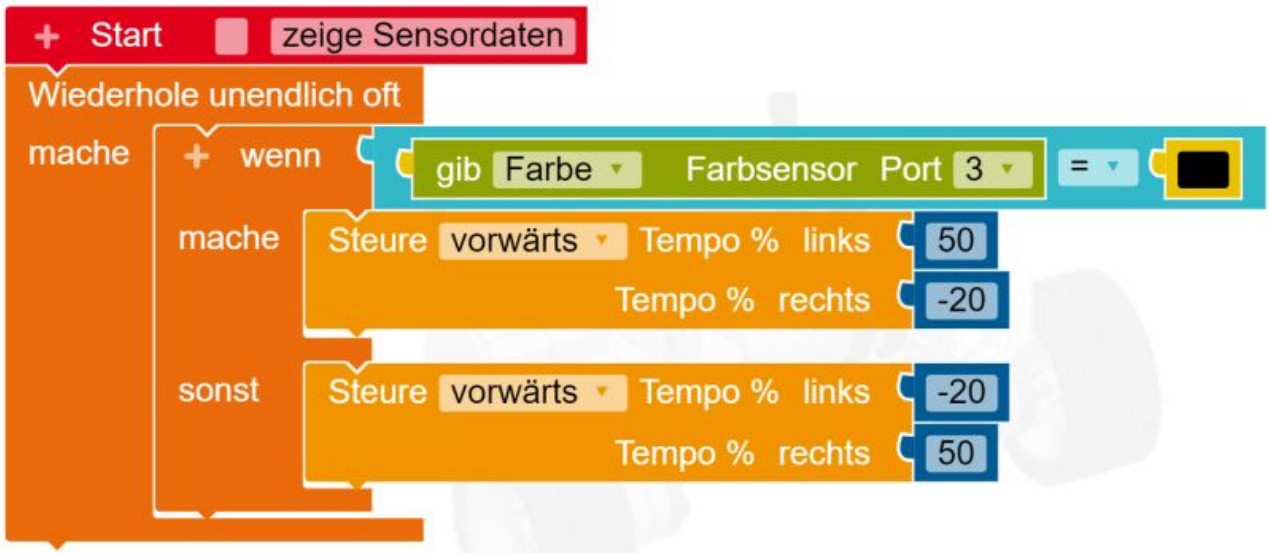


Abbildung 51 [1)]

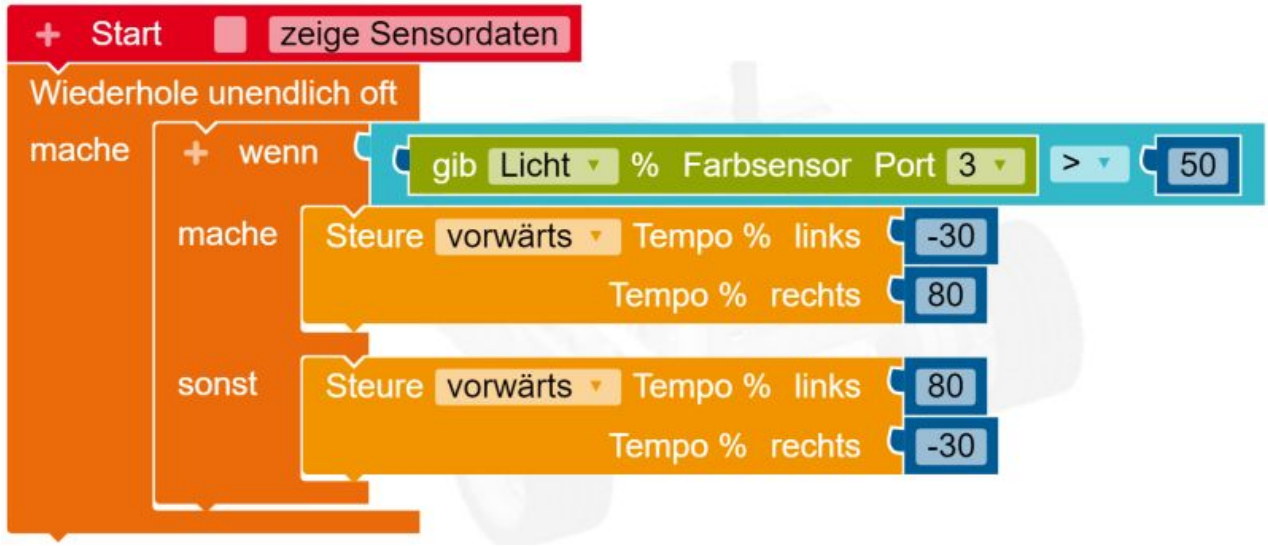


Abbildung 52 [1)]

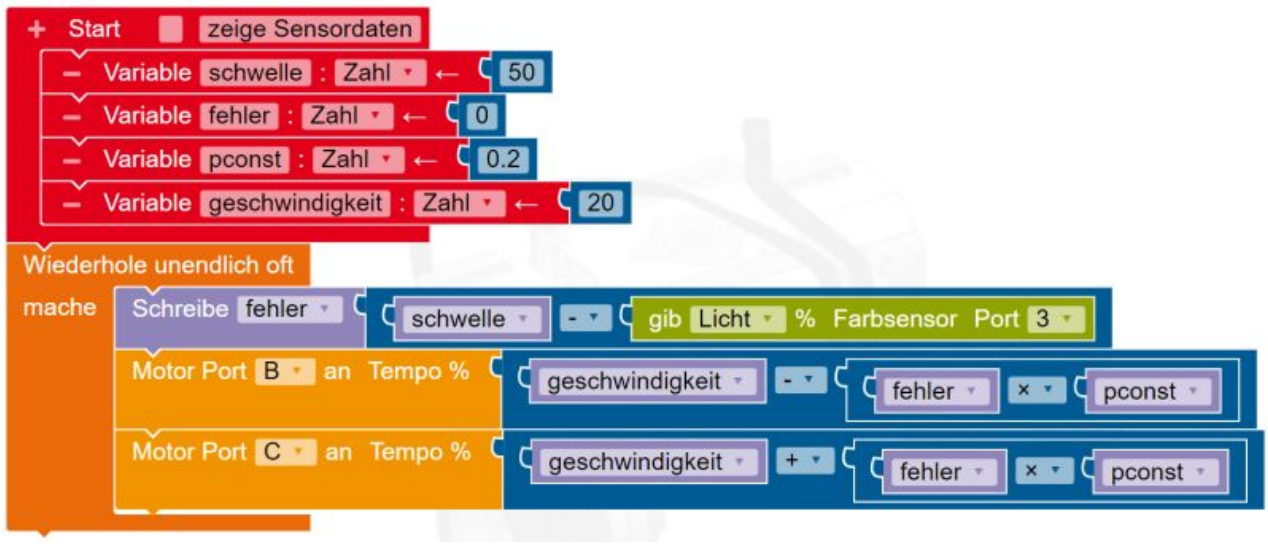


Abbildung 53 [1)]