

Octopus

A1



Aufgabe:

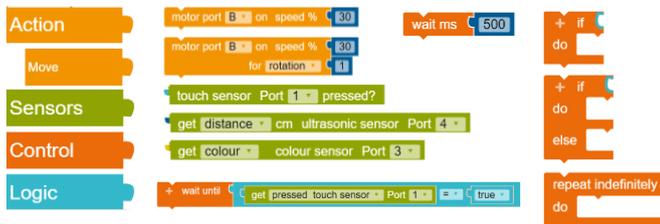
Verbinde den NXT-Baustein (CPU) mit allen Motoren und den Sensoren (Berührung, Ultraschall, Licht und Geräusch). Logge dich beim Open-Roberta Lab ein und verbinde den NXT-Baustein mit dem Computer.

Programmiere deinen Roboter so, dass er bei einem Sensorsignal mit einer Bewegung oder einem Ton reagiert. Probiere verschiedene Sensoren aus!

Zeit: 45' - 90'

Sozialform: PA

Blöcke:



Modell:



Roberta:



- Der Berührungssensor wird gedrückt, ein Motor dreht sich.
- Der Berührungssensor wird gedrückt, ein Motor bewegt sich, der Sensor wird erneut gedrückt, der Motor stoppt.
- Zwei Motoren (B+C) laufen mit 75% Leistung, sobald der Ultraschallsensor eine Distanz von weniger als 15cm misst, stoppen beide Motoren.
- Der Geräuschsensor erkennt ein lautes Geräusch, der Roboter wartet 1000 Millisekunden und antwortet mit einem akustischen Signal.
- Beide Motoren (B+C) laufen mit 100% Leistung, sobald der Lichtsensor eine Lichtquelle erkennt.



.....

Tipps:



-  Benutze die Funktionen „wait until...“, um Aktionen per Sensorsignal zu starten.
-  Baue Geräusche in dein Programm ein. So hörst du, wo genau im Programm Roberta gerade steckt.

Nachgedacht:



Besprich mit deinem Partner die Fragen:

- 1) Was können die einzelnen Sensoren? Was nicht?
- 2) Wie kannst du in den Sensorblöcken andere Sensorfunktionen abrufen?
- 3) Was genau tut die „wait-until-Funktion“?
- 4) Wie kannst du die Motoren ansteuern (auswählen)?
- 5) Wie kannst du die Motorenleistung und Drehrichtung verändern?
- 6) Wie speicherst du deine Programme, so dass du sie später wiederfindest?

Teste dich:



Wozu brauchst du diese Blöcke?

wait ms 500

motor port B on speed % 80

motor port B on speed % 30 for rotation 1

wait until get pressed touch sensor Port 1 true

touch sensor Port 1 pressed?

get distance cm ultrasonic sensor Port 4

Nachgefragt:

Was geschieht in diesem Programm?

start show sensor data

wait until get distance cm ultrasonic sensor Port 4 30

play file 5

drive forwards speed % 30 distance cm 20

Lernziele:



Du kannst:

alle Sensoren und Motoren benennen und deren Funktion erklären.

alle Sensoren und Motoren richtig an den NXT-Baustein anschliessen.

die passenden Blöcke in der Programmiersoftware dem richtigen Sensor oder Motor zuordnen.

mit dem "wait-until-Block" einfache Aktions-Reaktions-Programme schreiben.

deine Programme speichern und wieder abrufen.

den NXT-Baustein mit dem Open-Roberta Lab verbinden.

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

= kann ich sehr gut = kann ich gut = kann ich mit etwas Hilfe = übe ich noch

Fahrschule

A2



Aufgabe:

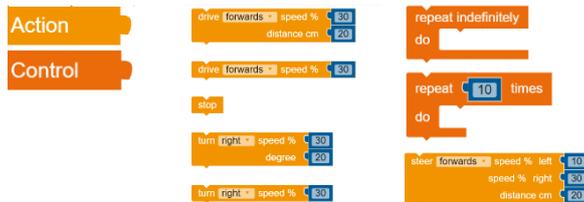
Baue den Domabot nach Anleitung zusammen. Überprüfe, ob alle Sensoren (Ultraschall, Licht, Geräusch) und die Motoren (B+C) vom NXT-Baustein erkannt werden. Die Anleitung findest du auf: <http://die-denkschule.ch/roberta/bauanleitungen/nxt-domabot/>
Logge dich beim Open-Roberta Lab ein und verbinde den NXT-Baustein mit dem Computer.

Öffne den Roboter Konfigurator und definiere die Anschlüsse. Trage Raddurchmesser und Radabstand korrekt ein. Verbinde den NXT-Baustein mit dem Computer und lade deine Programme auf den Roboter. Starte das Programm über die Bedientasten.

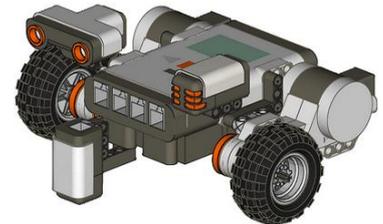
Zeit: 45' - 90'

Sozialform: PA

Blöcke:



Modell:



Roberta:



- Roberta fährt gerade eine bestimmte Strecke vorwärts, stoppt, fährt genau dieselbe Strecke rückwärts und stoppt erneut.
- Roberta fährt einen vollen Kreis (360°) auf der Stelle.
- Dein Roboter fährt einen vollen Kreis (360°) mit nur einem Motor. Das eine Rad bleibt auf der Stelle stehen.
- Dein Roboter fährt im ZickZack durch den Raum.
-  Roberta wiederholt eine der obigen Aufgaben 10 Mal (20 Mal, unendlich, bis der Berührungssensor gedrückt wird, etc.)

Tipps:



-  Dein Roboter soll genaue Kurven fahren. Bestimme die Drehung des Roboters über die Anzahl Radumdrehungen (Dezimalbrüche funktionieren auch!)
-  Benutze „Schleifen“, um einen Programmablauf zu wiederholen.
-  Baue Geräusche in dein Programm ein. So hörst du, wo genau im Programm Roberta gerade steckt.

Fahren mit Sensoren

A3



Aufgabe:

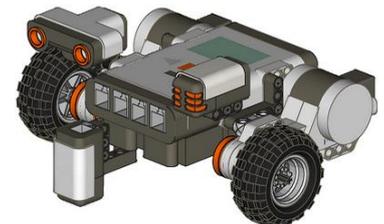
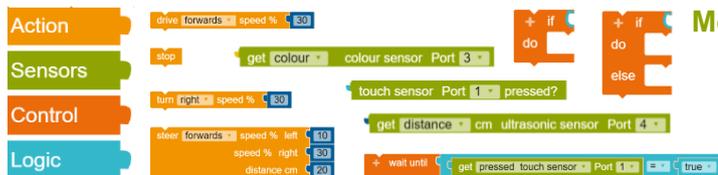
Baue den Domabot nach Anleitung zusammen. Überprüfe, ob alle Sensoren (Ultraschall, Licht, Geräusch) und die Motoren (B+C) vom NXT-Baustein erkannt werden. Die Anleitung findest du auf: <http://die-denkschule.ch/roberta/bauanleitungen/nxt-domabot/>
Logge dich beim Open-Roberta Lab ein und verbinde den NXT-Baustein mit dem Computer.

Öffne den Roboter Konfigurator und definiere die Anschlüsse. Trage Raddurchmesser und Radabstand korrekt ein. Verbinde den NXT-Baustein mit dem Computer und lade deine Programme auf den Roboter. Starte das Programm über die Bedientasten.

Zeit: 45' - 90'

Sozialform: PA

Blöcke: Modell:



Roberta:



- Dein Roboter fährt vorwärts und bleibt genau 25cm vor einem Hindernis (z.B. einer Wand) stehen.
- Dein Roboter fährt geradeaus. Sobald ein lautes Geräusch ertönt, wechselt der Roboter die Richtung.
- Dein Roboter fährt im Kreis. Sobald das Licht ausgeht, bleibt der Roboter stehen. Sobald das Licht wieder angeht, fährt der Roboter weiter im Kreis.
-  Entwickle deinen Roboter weiter. Kombiniere Teile aus den Aufgaben 1-3 zu einem neuen Programm weiter. Deine Kreativität ist gefragt!

Tipps:



-  Benutze „Verzweigungen“ `if...; else...`, um die Sensoren „entscheiden“ zu lassen.
-  Miss mit dem Roboter auf „Port-View“ die Sensorwerte, damit du geeignete Schwellenwerte bestimmen kannst.
-  Baue Geräusche in dein Programm ein. So hörst du, wo genau im Programm Roberta gerade steckt.

Nachgedacht:



Besprich mit deinem Partner die Fragen:

- 1) Was genau tun „Verzweigungen“ `if...; else...`?
- 2) Wie kannst du mehrere „Schleifen“ und „Verzweigungen“ in einem Programm kombinieren?
- 3) Wie kannst du mit dem Roboter Sensorwerte messen?
- 4) Auf welche Arten kannst du eine Schleifen-Funktion beenden und das Programm trotzdem fortsetzen?

Teste dich:



Wozu brauchst du diese Blöcke?



.....

.....



.....

.....

.....

Nachgefragt:

Wie heissen diese NXT-Bauteile?



.....



.....



.....



.....

Lernziele:



Du kannst:

mit der "Verzweigungen" `if...; else...` Wenn-Dann-Programme schreiben.

|||||
□ □ □ □

mit „wait-Blöcken“ Programmabläufe verzögern oder von Ereignissen (Sensoren) abhängig machen.

□ □ □ □

mit verschiedenen "Schleifen" Programmabläufe beliebig oft wiederholen.

□ □ □ □

Sensorwerte mit dem NXT-Baustein messen und ablesen.

□ □ □ □

||||| = kann ich sehr gut

|||| = kann ich gut

||| = kann ich mit etwas Hilfe

|| = übe ich noch

Linienfolger

A4a



Aufgabe:

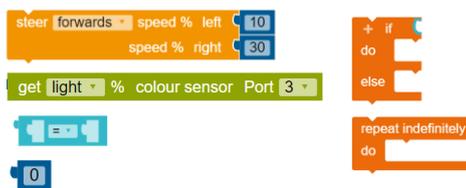
Baue den Domabot nach Anleitung zusammen. Überprüfe, ob alle Sensoren (Ultraschall, Licht, Geräusch) und die Motoren (B+C) vom NXT-Baustein erkannt werden. Die Anleitung findest du auf: <http://die-denkschule.ch/roberta/bauanleitungen/nxt-domabot/>
Logge dich beim Open-Roberta Lab ein und verbinde den NXT-Baustein mit dem Computer.

Öffne den Roboter Konfigurator und definiere die Anschlüsse. Trage Raddurchmesser und Radabstand korrekt ein. Verbinde den NXT-Baustein mit dem Computer und lade deine Programme auf den Roboter. Starte das Programm über die Bedientasten.

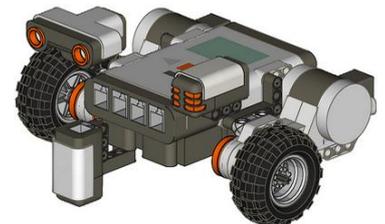
Zeit: 20' - 45'

Sozialform: PA

Blöcke:



Modell:



Roberta:



- Dein Roboter fährt vorwärts einer dunklen/hellen Linie entlang.
- Dein Roboter folgt der Linie auch um scharfe Kurven nach links ODER rechts.
- Dein Roboter folgt der Linie auch um scharfe Kurven nach links UND rechts.
-  Der Roboter folgt der Linie in alle Richtungen sehr genau und präzise.
-  Der Roboter folgt der Linie in alle Richtungen mit möglichst hohem Tempo.

Tipps:



-  Benutze „Verzweigungen“ `if...; else...`, um die Sensoren „entscheiden“ zu lassen.
-  Miss mit dem Roboter auf „Port-View“ die Sensorwerte, damit du geeignete Schwellenwerte bestimmen kannst.
-  Verändere gezielt einzelne Konfigurationen wie Motorenleistung, Lenkung oder den Schwellenwert der Sensoren. Beobachte, was dann passiert.
-  Du benötigst nur 7 Blöcke für den Linienfolger

Nachgedacht:



Besprich mit deinem Partner die Fragen:

- 1) Was genau tun „Verzweigungen“ `if...; else...`?
- 2) Was kann der Lichtsensor eigentlich erkennen, resp. messen?
- 3) Was tun „Logik-Blöcke“?
- 4) Wozu brauchst du „Relationszeichen“ beim Programmieren?

Teste dich:



Was bedeuten diese Relationszeichen?



.....



.....



.....



.....



.....

Debuggen:

Ein Robotiker hat dieses Programm für einen Linienfolger erstellt. Markiere den/die Fehler und beschreibe kurz, was verbessert werden muss.

```

+ start [show sensor data]
repeat indefinitely
do
+ if [get light % colour sensor Port 3 < 30]
do
steer forwards speed % left 10
speed % right 30
distance cm 20
else
steer forwards speed % left 30
speed % right 10
distance cm 20
    
```

.....

.....

.....

.....

.....

Erstelle dieses Programm ebenfalls im Open Roberta Lab und teste deine Verbesserung im Simulator.

Lernziele:



Du kannst:

mit der "Verzweigungen" `if...; else...` Wenn-Dann-Programme schreiben.

Fehler in deiner Programmierung durch gezieltes Testen erkennen, beschreiben und beheben (debuggen).

einfache Programme mit dem Simulator im Open Roberta Lab testen und „debuggen“.

Relationszeichen benennen und in Logik-Blöcken richtig einsetzen.

= kann ich sehr gut
 = kann ich gut
 = kann ich mit etwas Hilfe
 = übe ich noch

Staubsauger

A4b



Aufgabe:

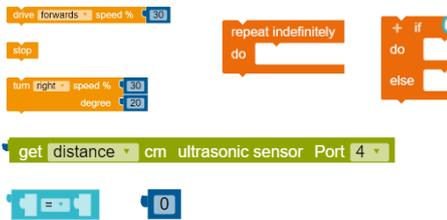
Baue den Domabot nach Anleitung zusammen. Überprüfe, ob alle Sensoren (Ultraschall, Licht, Geräusch) und die Motoren (B+C) vom NXT-Baustein erkannt werden. Die Anleitung findest du auf: <http://die-denkschule.ch/roberta/bauanleitungen/nxt-domabot/>
Logge dich beim Open-Roberta Lab ein und verbinde den NXT-Baustein mit dem Computer.

Öffne den Roboter Konfigurator und definiere die Anschlüsse. Trage Raddurchmesser und Radabstand korrekt ein. Verbinde den NXT-Baustein mit dem Computer und lade deine Programme auf den Roboter. Starte das Programm über die Bedientasten.

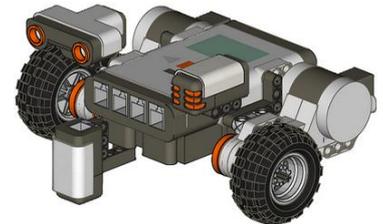
Zeit: 20' - 45'

Sozialform: PA

Blöcke:



Modell:



Roberta:



- Dein Roboter fährt vorwärts und bleibt genau 20cm vor einem Hindernis (z.B. einer Wand) stehen.
- Dein Roboter fährt vorwärts, erkennt ein Hindernis und fährt wieder eine bestimmte Strecke zurück.
- Dein Roboter fährt beliebig im Raum umher und weicht Hindernissen nach einem **bestimmten** Muster aus.
-  Dein Roboter fährt beliebig im Raum umher und weicht Hindernissen nach einem **zufälligen** Muster aus.

Tipps:



-  Benutze „Verzweigungen“ `if...; else...`, um die Sensoren „entscheiden“ zu lassen.
-  Miss mit dem Roboter auf „Port-View“ die Sensorwerte, damit du geeignete Schwellenwerte bestimmen kannst.
-  Verändere gezielt einzelne Konfigurationen wie Motorenleistung, Lenkung oder den Schwellenwert der Sensoren. Beobachte, was dann passiert.
-  Du benötigst nur 7 Blöcke für den Staubsauger.

Nachgedacht:



Besprich mit deinem Partner die Fragen:

- 1) Was genau tun „Verzweigungen“ `if...; else...`?
- 2) Was kann der Ultraschallsensor eigentlich erkennen, resp. messen?
- 3) Was tun „Logik-Blöcke“?
- 4) Wozu brauchst du „Relationszeichen“ beim Programmieren?

Teste dich:



Was bedeuten diese Relationszeichen?



.....



.....



.....



.....



.....

Debuggen:

Ein Robotiker hat dieses Programm für einen Staubsauger erstellt. Markiere den/die Fehler und beschreibe kurz, was verbessert werden muss.

```

+ start
repeat indefinitely
do
+ if
do
drive forwards speed % 50
else
stop
turn right speed % 30
degree 90
    
```

.....

.....

.....

.....

Erstelle dieses Programm ebenfalls im Open Roberta Lab und teste deine Verbesserung im Simulator.

Lernziele:



Du kannst:

mit der "Verzweigungen" `if...; else...` Wenn-Dann-Programme schreiben.

Fehler in deiner Programmierung durch gezieltes Testen erkennen, beschreiben und beheben (debuggen).

einfache Programme mit dem Simulator im Open Roberta Lab testen und „debuggen“.

Relationszeichen benennen und in Logik-Blöcken richtig einsetzen.

= kann ich sehr gut
 = kann ich gut
 = kann ich mit etwas Hilfe
 = übe ich noch

Achtung! Kakerlaken!

A5



Aufgabe:

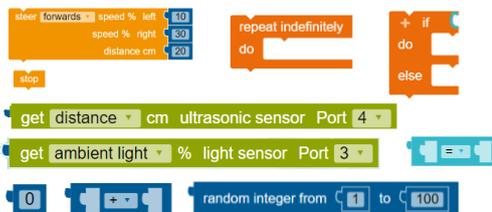
Baue den Domabot nach Anleitung zusammen. Überprüfe, ob alle Sensoren (Ultraschall, Licht, Geräusch) und die Motoren (B+C) vom NXT-Baustein erkannt werden. Die Anleitung findest du auf: <http://die-denkschule.ch/roberta/bauanleitungen/nxt-domabot/>
Logge dich beim Open-Roberta Lab ein und verbinde den NXT-Baustein mit dem Computer.

Öffne den Roboter Konfigurator und definiere die Anschlüsse. Trage Raddurchmesser und Radabstand korrekt ein. Verbinde den NXT-Baustein mit dem Computer und lade deine Programme auf den Roboter. Starte das Programm über die Bedientasten.

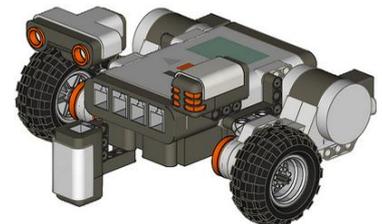
Zeit: 20' - 45'

Sozialform: PA

Blöcke:



Modell:



Roberta:



- Solange es dunkel ist, bleibt dein Roboter stehen. Wenn das Licht angeht, beginnt dein Roboter sofort herumzufahren und weicht dabei Hindernissen nach einem **bestimmten Muster** aus.
- Dein Roboter fährt bei hellem Licht im Raum umher und weicht Hindernissen nach einem **zufälligen Muster** aus.
- Dein Roboter fährt bei Licht beliebig im Raum umher und bleibt in einem dunkeln „Versteck“ (z.B. unter einem Tisch) stehen.
-  Dein Roboter kombiniert alle obigen Aufgaben gleichzeitig.

Tipps:



-  Benutze mehrere „Verzweigungen“ `if...; else..` in der logisch richtigen Reihenfolge, um entsprechenden.
-  Miss mit dem Roboter auf „Port-View“ die Sensorwerte, damit du geeignete Schwellenwerte bestimmen kannst.
-  Zeichne ein Flussdiagramm mit den passenden Fragen, um die Logik des Programmes aufzuzeigen.

Erklärt:



Unten siehst du ein Beispiel, wie „Mathe-Blöcke“ eingesetzt werden können. Besprich mit deinem Partner, was genau in diesem Beispiel mit den Zahlenwerten geschieht und wozu das Ergebnis dienen soll.

```

steer backwards speed % left random integer from 1 to 4 x 25
speed % right random integer from 1 to 4 x 25
distance cm 20
    
```

Programmiere diese Vorlage an der richtigen Stelle in deinem Kakerlaken-Programm exakt nach.

Teste dein neues Programm im Simulator des Open Roberta Lab.

Nachgedacht:



Besprich mit deinem Partner die Fragen:

- 1) Wie kannst du mehrere „Verzweigungen“ `if...; else...` und Schleifen in einem Programm verschachtelt einsetzen?
- 2) Was bewirkt „random integer“ im obigen Programm?
- 3) Was bedeutet „integer“ in der Mathematik?
- 4) Welche Ergebnisse sind im Beispiel mit den Wertes 1, 4 und 25 möglich? Warum?

Lernziele:



Du kannst:

Fehler in deiner Programmierung durch gezieltes Testen erkennen, beschreiben und beheben (debuggen).

einfache Programme mit dem Simulator im Open Roberta Lab testen und „debuggen“.

mehrere "Verzweigungen" `if...; else...` in einem Algorithmus der Logik einer Aufgabe entsprechend einsetzen.

mit „Mathe-Blöcken“ und „Zufallswerten“ die Motorenleistung oder Motorensteuerung direkt beeinflussen.



= kann ich sehr gut
 = kann ich gut
 = kann ich mit etwas Hilfe
 = übe ich noch