

ROBERTA
INITIATIVE

Open Roberta Neuronale Netze

Lernkarten mit vielen
Programmierideen

Mehr Informationen unter:
wiki.open-roberta.org

Open Roberta Lab

In 6 Schritten zum NEPO-Start. **So geht's:**

- Schritt 1** Open Roberta Programme können gleichermaßen auf einem Robotersystem oder in der Simulationsumgebung ausgeführt werden. Welches System wie vorzubereiten ist, wie die Verbindung mit dem Open Roberta Lab funktioniert und was außerdem benötigt wird, erfährst du im Open Roberta Wiki: wiki.open-roberta.org
- Schritt 2** Öffne das Open Roberta Lab und lege los!
- Schritt 3** Wähle  **Open Roberta xNN** als System aus
- Schritt 4** Erstelle das NEPO-Programm per »drag and drop«
- Schritt 5** Öffne mit einem Klick auf **SIM** die Simulationsumgebung.
- Schritt 6** Ein letzter Klick auf  im Simulationsfenster und dein Programm startet!

Open Roberta Lab

Für Anweisungen verwendest du für ein Robotersystem im »Open Roberta Lab« keine »normale« Sprache, sondern Blöcke der Programmiersprache »NEPO«.

Menü →

Programmansicht →

Modus
Wähle 1 für **Anfänger**-
oder 2 für **Experten**-
Blöcke.

In den **Block-Kategorien** findest du die Blöcke, die das System kennt.

Start **zeige Sensordaten**

Programmstart-Block
Platziere an den Sequenzverbindungen die Blöcke, die das System ausführen soll.

Soforthilfe

Öffne mit einem Klick die **Simulationsumgebung**

Wirf Blöcke zum Löschen in den **Mülleimer**.

Open Roberta Lab

Die Roboterkonfiguration umfasst die Aktoren und Sensoren sowie Angaben zum Raddurchmesser und zur Spurbreite passend zu deinem ausgewählten Robotersystem.

Konfigurationsansicht

Menü

The screenshot shows the 'Konfigurationsansicht' (Configuration View) in the Open Roberta Lab. At the top, there is a navigation bar with three tabs: 'PROGRAMM NEPOprog', 'ROBOTERKONFIGURATION' (which is active), and 'NEURONALES NETZ NEPOnn'. To the left of the main content area is a 'Menü' (Menu) with two categories: 'Aktion' (Action) in orange and 'Sensoren' (Sensors) in green. The main area displays the configuration for an 'XNN' (Neural Network) robot system. It includes fields for 'Raddurchmesser' (Wheel diameter) set to 5.6 cm and 'Spurbreite' (Track width) set to 18 cm. Below these are four sensors: Sensor 1 (Berührungssensor), Sensor 2 (Kreisel sensor), Sensor 3 (Farbsensor), and Sensor 4 (Ultraschallsensor). There are also four motors: Motor A, Motor B, Motor C, and Motor D. Motor A and Motor B are configured as 'Großer Motor' (Large Motor) with 'Regulierung' (Regulation) set to 'ja' (yes), 'Drehrichtung' (Rotation direction) set to 'vorwärts' (forward), and 'Seite' (Side) set to 'rechts' (right). Motor C is also a 'Großer Motor' with 'Regulierung' set to 'ja', 'Drehrichtung' set to 'vorwärts', and 'Seite' set to 'links' (left). Motor D is not configured. At the bottom right, there are icons for home, search, and trash.

In den **Block-Kategorien** findest du die Aktoren und Sensoren, die das System kennt.

Open Roberta Lab

Das Neuronale Netz kann frei konfiguriert und die Anzahl an **Eingabe-** und **Ausgabe-Neuron** sowie die Zahl der verborgenen Schichten und deren Neuronen bestimmt werden.

Ansicht **Neuronales Netz**

Menü →

Bestimme die **Aktivierungs-**funktion →

Bestimme wie Werte angezeigt und geändert werden →

Eine **Kante** verbindet zwei Neuronen verschiedener Schichten

Gewicht Bestimme die Stärke der Kante

Bias Bestimme den Schwellenwert des Neurons

SIM

Open Roberta Lab

Die Simulationsumgebung **SIM** ermöglicht es, Programme zu testen. Das ist insbesondere dann von Vorteil, wenn gerade kein Robotersystem zur Verfügung steht.

Programmansicht →



Bestimme mit diesen Steuerfunktionen Hintergrund und Elemente der Simulationsumgebung

Starte dein Programm mit einem Klick auf die **Start-Schaltfläche**.



Hallo Welt!!



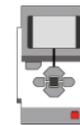
»Hallo! Ich stehe für die Simulation als 2D Roboter-Modell zur Verfügung. Übrigens: Meine Freund*innen nennen mich Roberta ;-). **Komm, wir fangen an!**«

Aufgabe 1

Im ersten Schritt zeigt dir Roberta, wie sie Text auf dem Bildschirm ausgibt. Nimm den Block »Zeige Text« aus der Kategorie »Aktion« und setze diesen an den Programmstart-Block.



Bildschirm



- ▶ Starte dein erstes Programm in der Simulationsumgebung und öffne mit einem Klick auf  die Ansicht deines Systems

Aufgabe 2

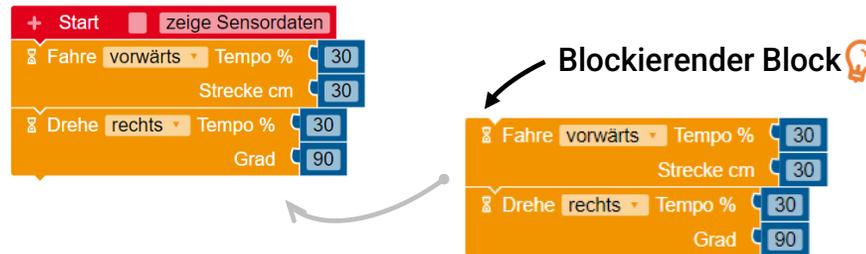
Ändere im Programm den Block »Zeige Text«, sodass Roberta nun „Hallo Welt!“ anzeigt.

- ▶ Prüfe!!



Aufgabe 1

Quadrat – eine bekannte geometrische Form mit vier gleich langen Seiten und vier gleich großen Winkeln. Versuche es mit Roberta!



▶ Probiere das Programm aus!

✍ Erkennst du im Programm ein Muster?

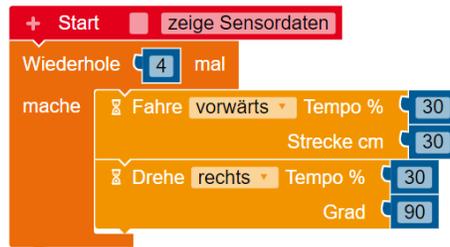


Ist dir das Symbol  aufgefallen?  markiert Blöcke, die beim Ausführen eine Weile benötigen. Beispielsweise der »Fahre«-Block, der solange andauert bis Roberta die vollständige Strecke gefahren ist.



Aufgabe 2

Optimiere dein Programm und verwenden den Block »Wiederhole ... mal« aus der Kategorie »Kontrolle«.



▶ Start das Programm!

📝 Was macht Roberta nun anders?



In der Programmierung müssen oft bestimmte Schritte wiederholt werden. Um dies einfacher zu machen, kann ein »Wiederhole« Block verwendet werden. Von dieser Kontrollstruktur namens **Schleife** gibt es verschiedene Typen. Eines haben sie alle gemeinsam: NEPO-Blöcke, die eine Schleife umklammert werden mehrfach ausgeführt.



Sicher ist clever

Es ist unangenehm, irgendwo anzustoßen. Auch Roberta will lieber auf Nummer sicher gehen. Deshalb benötigt sie die Wahrnehmung des Ultraschallsensors Gegenstände zu erkennen, die ihr im Weg stehen. Denn Sicherheit geht vor!

Aufgabe 1

Sie **wartet** geduldig und beobachtet aufmerksam. Falls ihr etwas zu nahe kommt, zeigt sie Entschlossenheit und **sagt** energisch "Stopp!".



- ▶ Starte das Programm und schiebe vorsichtig das blaue Hindernis mit der Maus in Robertas Nähe.



Aufgabe 2

So clever wie du, darf sich nun Roberta eigenständig dem Hindernis nähern. Füge in dein Programm den **nichtblockierenden** Block **Fahre vorwärts** Tempo % **30** vor »Warte bis« ein.

- ▶ Los, probier's aus!!!



Reize aus der Umwelt

Der italienischen Neurobiologen Valentino Braitenberg entwickelte wissenschaftliches Gedankenexperiment, um grundlegende Konzepte in der Robotik und Kognitionswissenschaft zu veranschaulichen. Es basiert auf einfachen robotischen Systemen, die mit Sensoren und Aktoren ausgestattet sind.

Verschiedene Verbindungsmuster zwischen Sensor und Aktor führen zu einem unterschiedlichen Verhalten.

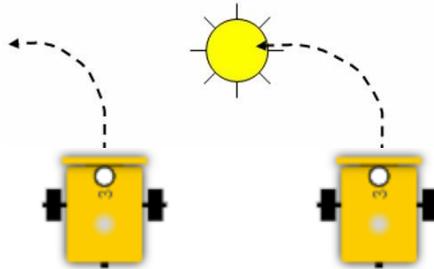
Braitenberg zog Parallelen zu simplen biologischen Lebensformen und verglich die Verhaltensweisen seiner künstlichen Wesen mit Emotionen wie Furcht, Aggression, Liebe oder Neugier



Reize aus der Umwelt

Der Neurobiologe Valentino Braitenberg entwickelte ein faszinierendes Gedankenexperiment, um grundlegende Konzepte in der Robotik und Kognitionswissenschaft zu veranschaulichen. Seine Experimente basieren auf simplen robotischen Systemen mit Sensoren und Aktoren. Durch unterschiedliche Verbindungsmuster zwischen Sensor und Aktor können Reize in der Umgebung wahrgenommen und als unterschiedliche Verhaltensweisen auswirken.

Braitenberg zog dabei interessante Parallelen zu einfachen biologischen Lebensformen und verglich das Verhalten seiner künstlichen Wesen mit Emotionen wie Furcht, Aggression, Anziehung oder Neugier.





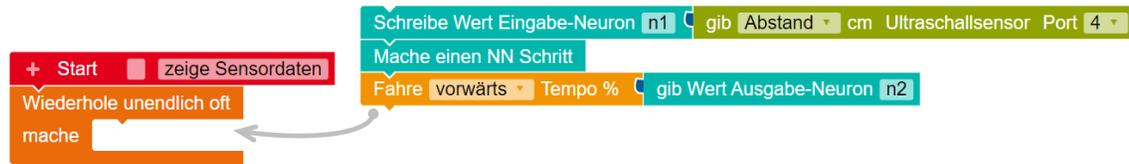
Aufgabe 1

Roberta, als künstliches Wesen, findet das Verhalten **Anziehung** faszinierend, denn es bewirkt, dass sie sich wie magisch von einem Reiz angezogen fühlt und mutig darauf zugeht. Zusätzlich zur Fähigkeit, Reize mit dem Ultraschallsensor zu erkennen, nutzt Roberta ihr **anfänglich vorkonfiguriertes Neuronales Netz**.

Erstelle ein neues Programme und verwende aus Kategorie »Neuronales Netz« die Blöcke »Schreibe Wert Eingabe-Neuron«, »Mache einen NN Schritt« und, um das Ergebnis direkt mit »gib Wert Ausgabe-Neuron« aus dem Neuronale Netz in den Block »Fahre« zu übergeben.



Denk dran: Wiederholung ist wichtig! Gib Robertas Neuronalem Netz ständig Reize, damit es immer weiter rechnen und neue Ausgabewerte generieren kann.



- ▶ Teste dein erstes Neuronales Netz in der Simulation.
Ändert sich Robertas Verhalten, wenn du das Gewicht anpasst? Versuch's mal!



Roberta, als künstliches Wesen, findet das Verhalten **Anziehung** faszinierend, denn es bewirkt, dass sie sich wie magisch von einem Reiz angezogen fühlt und mutig darauf zugeht. Zusätzlich zur Fähigkeit, Reize mit dem Ultraschallsensor zu erkennen, nutzt Roberta ihr **anfänglich vorkonfiguriertes Neuronales Netz**.

Aufgabe 1

Erstelle ein neues Programme und verwende aus Kategorie »Neuronales Netz« die Blöcke »Schreibe Wert Eingabe-Neuron«, »Mache einen NN Schritt« und, um das Ergebnis direkt mit »gibt Wert Ausgabe-Neuron« aus dem Neuronale Netz in den Block »Fahre« zu übergeben.



▶ Teste dein erstes Neuronales Netz in der Simulation und schiebe vorsichtig das blaue Hindernis mit der Maus in Robertas Nähe.

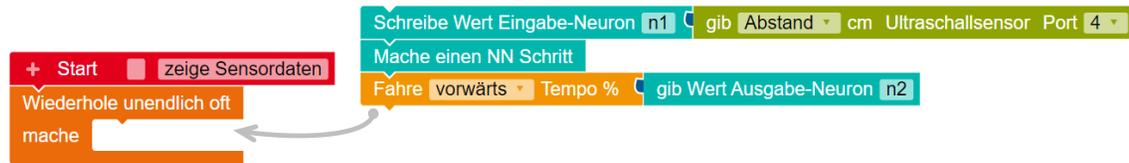
✎ Wie verhält sich Roberta?





Aufgabe 2

Überrascht? Vermutlich hat sich Roberta kaum oder gar nicht dem Hindernis genähert. Denk dran: Wiederholung ist wichtig! Versorge Robertas Neuronales Netz ständig mit Reizen, damit es immer weiter rechnen und neue Ausgabewerte generieren kann.



▶ Prüfe erneut!

✎ Ändert sich Robertas Verhalten, wenn du das Gewicht anpasst? Versuch's mal!

Neugierig geworden?

Details zum Geschäftsfeld Smart Coding and Learning



Weitere Lernkarte zu Künstliche Neuronale Netze mit Open Roberta



Neugierig geworden?

Entdecke Sie weitere Details!

Smart Coding and Learning



Roberta-Initiative



Künstliche Neuronale Netze mit Open Roberta



Lernkarte zu Künstliche Neuronale Netze mit Open Roberta



Neugierig geworden?

Tauche Sie tiefer ein! Scannen Sie jetzt hier für mehr spannende Details!

Smart Coding and Learning



Roberta-Initiative



Künstliche Neuronale Netze mit Open Roberta



Lernkarte zu Künstliche Neuronale Netze mit Open Roberta

