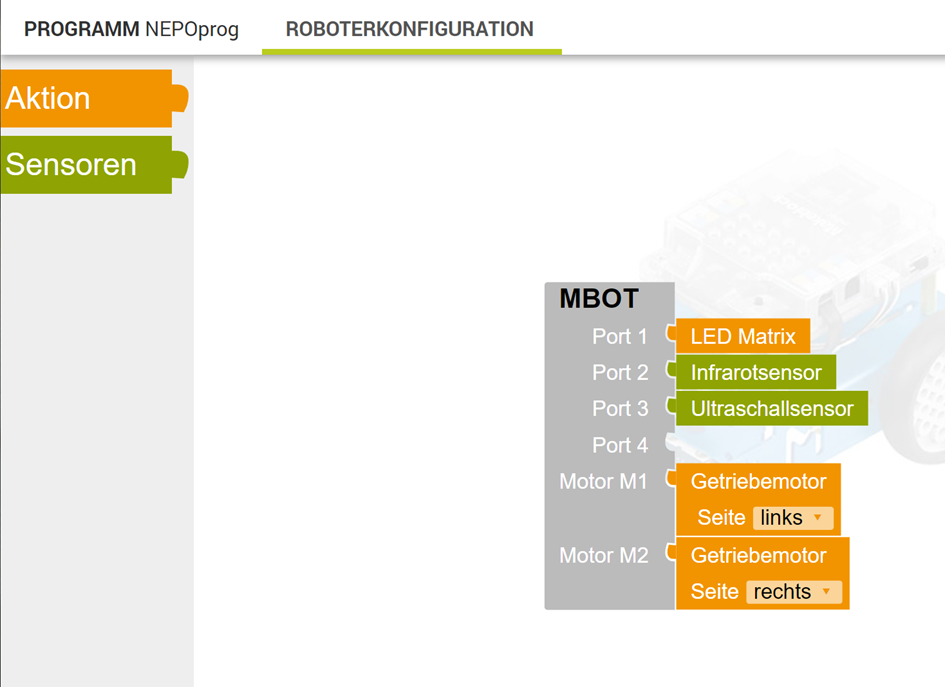
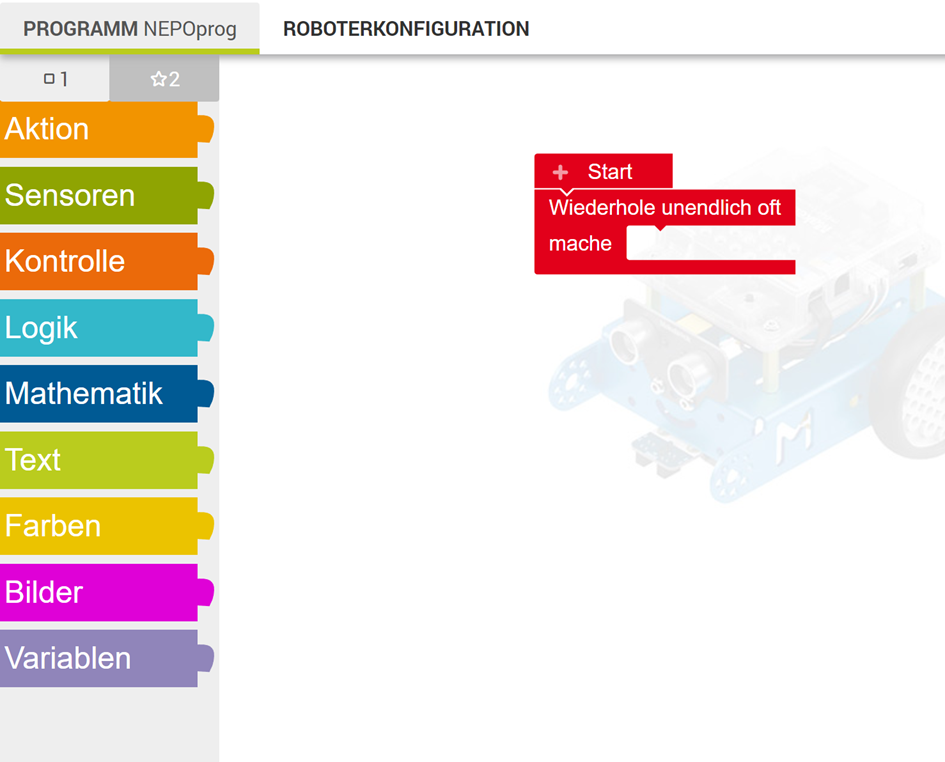
# Einführung in die Open Roberta Lab Umgebung:

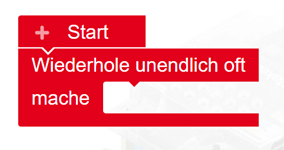
1. Entwicklungsumgebung auswählen? => Mbot (Mbot 2 noch in Beta und ohne Simulation)
2. Reiter Roboter Konfiguration führt uns zum Aufbau des Roboters und seiner Ausrüstung
   1. Aktion -> Motoren und LED-Matrix?
   2. Sensoren -> Alle Sensoren die ausgerüstet werden können



1. Programm Reiter dient der Planung unseres Programms:
   1. Seite 1 ist der Anfänger Modus (weniger Bausteine und Kategorien)
   2. Seite 2 ist der Expertenmodus (mehr Bausteine und neue Kategorien Listen / Funktionen und Nachrichten



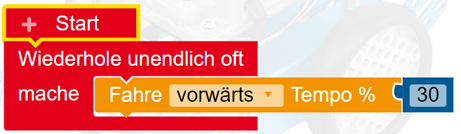
1. Roter Startblock gibt den Beginn des einzigen Programms an (Alles was NICHT mit ihm verbunden ist wird nicht ausgeführt)



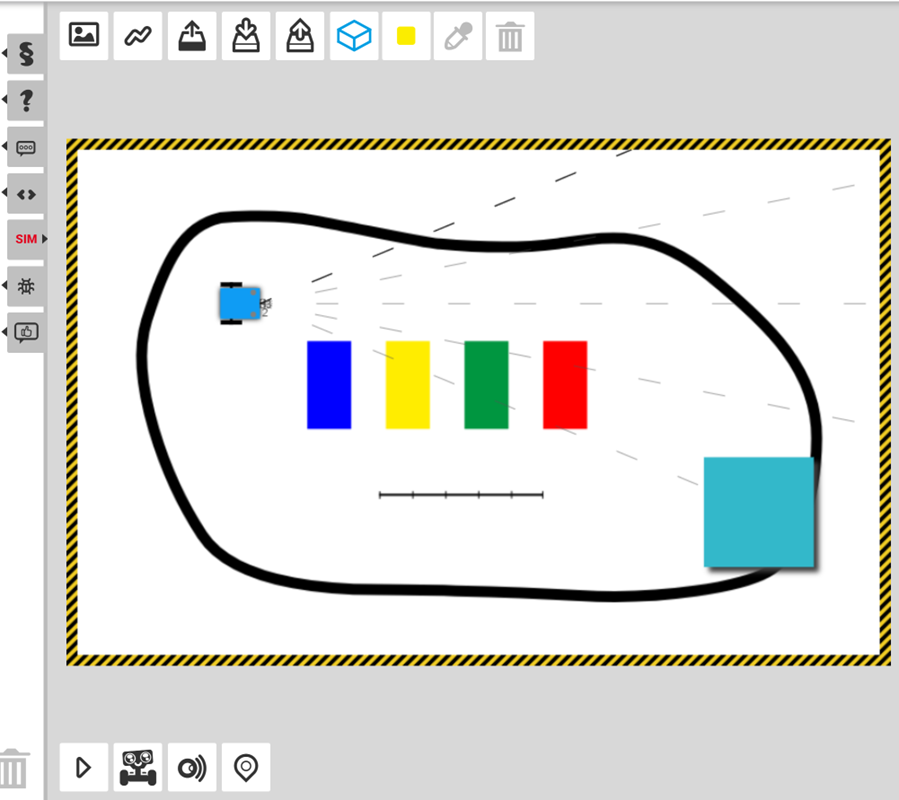
1. Sequenzielle Programmierung durch aneinanderreihen verschiedener „Bausteine“ aus den Kategorien durch das Drag-and-Drop-Prinzip / Entfernen von Blöcken durch Auswahl und del / Entf (Bausteine = Legosteine mit dem Steckprinzip)
2. Wie einzelne Blöcke miteinander verbunden werden dürfen zeigen die Ausstülpungen der Blöcke an

**Start der Programmierung:**

1. Aktion: Fahre Vorwärts Tempo (Aktion)



1. Um unser kleines Programm zu testen, müssen wir den SIM Knopf auf der rechten Seite des Bildschirms drücken:



1. Einstellungen und Infos zum Roboter:
2. Zum Starten des Programms, den Play Knopf unten links in der Ecke drücken:



1. Daten des Roboters anzeigen lassen:



1. Den Roboter an den Anfang der Simulation zurücksetzen:



1. Zum Verändern der Umgebung des Roboters:
2. Der Roboter hinterlässt eine Spur:



1. Simulationseinstellung hochladen:



1. Füge ein Hindernis der Umgebung hinzu



1. Erzeugung eines Farbfelds:



1. Änderung der Farbe eines Objekts:

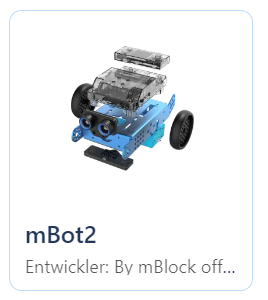


1. Löschen eines Objekts in der Umgebung:



# Einführung in die Makeblock Umgebung:

1. Mblock herunterladen: ([Download](https://mblock.makeblock.com/en-us/download/))
2. Gerät Mbot 2 hinzufügen:

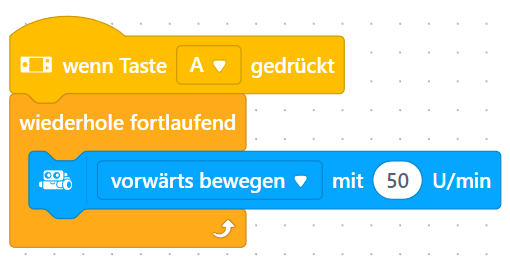
1. Die geladenen Mbot 2 Codeblöcke ermöglichen euch jetzt das Programmieren mit den speziellen Funktionen unseres Roboters.
2. Unter der Kategorie Ereignisse befinden sich die Blöcke, um das Programm starten zu lassen:



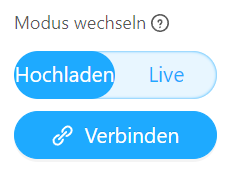
1. Sechseckige Blöcke sind Bedingungen, die entweder wahr oder falsch sein können. Runde Blöcke stellen Zahlenwerte dar:

1. Unter den Kategorien Quad-RGB-Sensor und Ultraschall 2 findet ihr die Codeblöcke der Sensoren und unter den Kategorien mBot 2 Chassis und mBot 2-Erweiterung befinden sich Bewegungsbefehle.
2. So könnt ihr wie in Open Roberta Lab Programme erstellen, welche der Roboter bei gewissen Ereignissen ausführt:



1. Um das Programm jetzt auf den Roboter zu übertragen, müsst ihr zuerst den Roboter mit dem USB-Kabel mit eurem Laptop verbinden.
2. Geht nun unten links in der auf den Knopf “Hochladen” und anschließend auf “Verbinden”:



1. Verbindet den Roboter nun mit eurem Laptop:



1. Nun könnt ihr euer Programm durch den Knopf “Hochladen” auf den Roboter laden, welches auch sofort gestartet wird.
2. Nachdem das Programm hochgeladen wurde, könnt ihr den Roboter vom Kabel lösen. Falls ihr euer Programm verändert und es ausprobieren wollt, führt einfach nochmal Schritt 7 aus.

**Hinweis:**

Nur Bewegungsbefehle mit einem klar definiertem Ende verwenden, da diese sonst unbegrenzt lange ausgeführt werden

# Aufgaben: (Hinweise)

Link zum GitHub-Repository: <https://github.com/codecentric/girls-day-bielefeld-2023>

**Open Roberta Lab: (**[**open-roberta-lab**](https://lab.open-roberta.org/) **/** [**Einführung in die Open Roberta Lab Umgebung:**](#_qaprb4arkmqy)**)**

1. *Roboter-Rennen*

* Lade die Simulationseinstellung Roboter-Rennen in deiner Simulation hoch

In unserer ersten Aufgabe geht es darum, dem Roboter eine Runde auf unserer Rennstrecke fahren zu lassen. Platziere den Roboter dafür auf der markierten Stelle und versuche mit Hilfe der Code-Blöcke, ihn eine Runde absolvieren zu lassen. Da wir uns dabei an die offiziellen Roboter-Renn-Regeln halten, gilt: Abkürzungen verboten!

1. Infrarotsensoren können Linien erkennen
2. Wenn man bei der Erkennung einer Linie von der Linie entgegen der Richtung der Linie steuert bleibt der Roboter in der Bahn

1. *Labyrinth (Leicht/Schwer)*

* Lade die Simulationseinstellung Labyrinth in deiner Simulation hoch

Rechts oder doch Links abbiegen? Du entscheidest, aber die Zeit läuft! Bei dieser Aufgabe musst du unserem MBot 2 helfen, möglichst schnell aus dem Labyrinth zu kommen in dem er sich verfahren hat. In der Simulation unter Ansicht der Sensordaten (Timer Values) kannst du die Zeit sehen.

1. Ultraschallsensoren können den kürzesten Abstand zu einem Objekt angeben
2. Eine 90 Grad Drehung entspricht ungefähr 17 Mal einer Drehung mit Tempo% 78
3. (Leicht) Mit einer stetigen Drehung bei jedem Hindernis kommt der Roboter irgendwann an seinem Ziel an
4. (Schwer) Schaut euch die Zufallszahlen des Roboters einmal genauer an :)
5. *Digitaler Stift*

* Lade die Simulationseinstellung Digitaler Stift in deiner Simulation hoch

1. Um unseren Roboter erstmal an seine neue Aufgabe als digitaler Stift zu gewöhnen, startest du am besten damit, dass du ihn den ersten Buchstaben deines Vornamens zeichnen lässt. Nutze dafür die Spur Funktion, die in der oberen Leiste der Simulation zu finden ist.
2. Jetzt, da ihr beide ein wenig Übung gesammelt habt, können wir einen Schritt weiter gehen und uns der ganzen Welt eine Nachricht zukommen lassen. Schreibe dazu zusammen mit dem Roboter die Nachricht “HALLO” in die Simulation
3. Bei b): Nutzt die fortgeschrittenen Programmierblöcke und Kategorie "Funktionen", um einzelne Buchstaben nicht doppelt programmieren zu müssen. Ihr könnt damit Vorgänge programmieren und komprimiert in einem Code-Block verwenden.

**Makeblock: (**[**mBlock**](https://ide.mblock.cc/) **/** [**Einführung in die Makeblock Umgebung:**](#_3gv62lmy2cfc)**)**

1. *Musik in meinen Sensoren*

Höre ich da Blau, Rot oder doch Grün? So muss es unserem Roboter gehen, wenn er über die Farbstrecke fährt. Doch damit wir auch Teil an der Musik haben können, müssen wir den Roboter dazu bringen, die Farben unter ihm in Töne umzuwandeln.

1. Arbeitet mit dem Block "spiele Note…” (Audio)
2. Der Quad-RGB-Sensor hat die Möglichkeit Farben zu erkennen
3. Für den optimalen Sound nutzt die Noten c(48) bis c(60)
4. *Süßer Transport(er)*

Der Traum ist wahr geworden: Eine Fabrik, die Süßigkeiten produziert. Das Transportsystem der Fabrik ist allerdings ausgefallen und du und der Mbot 2 müssen einspringen. Hilf dem Roboter dabei, die hergestellten Süßigkeiten zur Warenausgabe zu transportieren, indem du ihn die jeweiligen farbigen Linien folgen lässt:

1. Rot
2. Gelb

1. Cyan
2. Blau
3. Nutze die vier verschiedenen Quad-RGB-Sensoren des Roboters, um die Position der Linie zu erfassen
4. Mit vielen "wenn/sonst" Blöcken und den verschiedenen Farb Erkennungsmöglichkeiten lässt sich das Verhalten sehr gut programmieren
5. Um den Roboter in der Spur zu halten reichen meistens schon eine Drehung um ein paar Grad
6. *Gefahrenbremsung*

Da wir uns in einer Fabrik befinden, müssen wir unseren Roboter auch mit entsprechenden Sicherheitsvorkehrungen ausstatten. Wir möchten ja keine Süßigkeiten bei einem Unfall verlieren, oder?! Baue ihm dafür eine Notfallbremse ein, welche eine Kollision mit einem Objekt auf seiner Fahrbahn verhindert.

1. Nutze den Ultraschallsensor des Roboters
2. Stoppe solange bis der Ultraschallsensor kein Objekt, näher als eine bestimmte Distanz, erkennt