

Bauanleitungen für Roboter

# Künstliche Intelligenz und Robotik

Version 1.2.0

© 2020-2021 Marco Lardelli.

Alle Urheber- und Leistungsschutzrechte sind vorbehalten. Dieses Buch ist nur für den privaten Gebrauch oder für den Gebrauch durch öffentliche Schulen (mind. zu 50% durch den Staat finanziert) bestimmt. Für alle Verwendungen, insbesondere Vorführung ausserhalb öffentlicher Schulen, Sendung, Bearbeitung und Vervielfältigung/Verbreitung bedarf es einer speziellen Bewilligung.

# Anhang A

## Bauanleitungen für den KI-Roboter

**Für die Korrektheit sämtlicher Inhalte wird keine Haftung übernommen.**

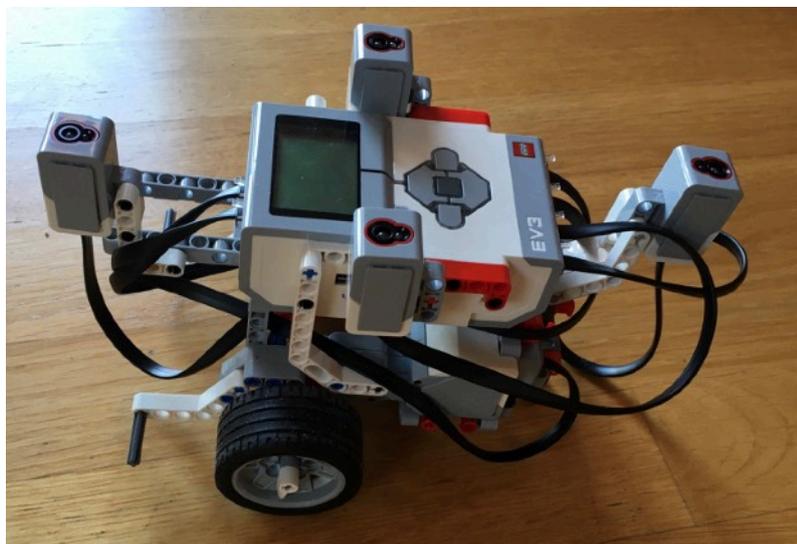
# 1. Überblick Roboter-Systeme

Der in diesem Buch zum Bau empfohlene Roboter kann mit mehreren verschiedenen technischen Plattformen aufgebaut werden. In der folgenden Tabelle findest du eine Übersicht:

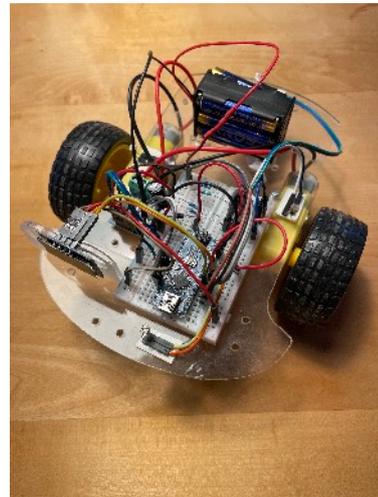
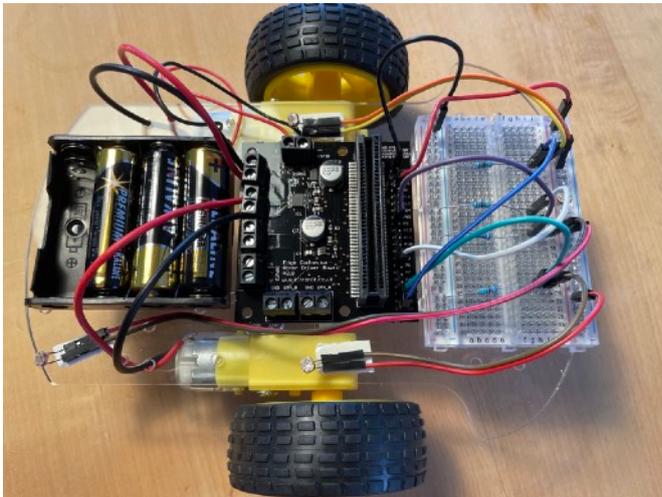
	Benötigtes Material	Ca. Kosten EUR	Programmiersprache	Bemerkungen
<b>LEGO Mindstorms EV3</b>	LEGO Mindstorms EV3 Set + 3 x Lichtsensor	420	Microsoft MakeBlock bzw. JavaScript	Aufbau und Programmierung sehr einfach. Sowohl die Education- wie auch die Consumer-Version können verwendet werden.
<b>LEGO Mindstorms Robot Inventor</b>	LEGO Mindstorms Robot Inventor Set + 3 x Lichtsensor	420	python	Aufbau sehr einfach. Zur Programmierung sind python-Kenntnisse hilfreich.
<b>BBC micro:bit</b>	micro:bit, Roboter-Kit, Motor-Driver-Board etc.	100	Microsoft MakeBlock bzw. JavaScript	Aufbau eher einfach, aber <b>einige Lötarbeiten erforderlich</b> . Programmierung einfach.
<b>Arduino</b>	Arduino Nano, Breadboard, Motor Driver, Roboter-Kit etc.	30-50	Arduino C	Etwas Erfahrung mit Elektronik sowie Programmierung ist erforderlich. <b>Wenige Lötarbeiten erforderlich</b> (Motorkabel).
<b>PGLU</b>	Bausatz PGLU Roboterfahrzeug <a href="https://pglu.ch">https://pglu.ch</a>	110	Arduino C	Etwas Erfahrung mit Elektronik sowie Programmierung ist erforderlich. <b>Wenige Lötarbeiten erforderlich</b> (Sensorkabel).

Hier ein paar Bilder der verschiedenen Roboter:

## 1. LEGO Mindstorms (hier die Education-Version)



## 2. BBC micro:bit (links) und Arduino (rechts)



Du kannst, bevor du dich für ein System entscheidest, auch zuerst einmal in die Bauanleitungen hineinschnuppern. Du siehst dann schnell, was dir am besten liegt.

Für den LEGO Mindstorms EV3 Roboter kannst du die **Education-Version** (graue Plastik-Box) **oder** auch die **Consumer-Version** verwenden (blaue Karton-Box), wobei die Education-Version mechanisch etwas besser funktioniert.

**ACHTUNG: DIE PROGRAMMIERUNG ERFOLGT Z. Z. VIA MAKECODE. DAZU MUSS DIE AKTUELLSTE FIRMWARE DER EDUCATION-VERSION AUF DEM BRICK INSTALLIERT SEIN. ES IST MÖGLICH, DASS DAS UPDATE EINES BRICKS DER HOME-VERSION AUF DIESE FIRMWARE IN ZUKUNFT NICHT MEHR MÖGLICH IST.**

Der Preis für die Bauteile der Arduino-Version hängt davon ab, wo man die Teile einkauft. Am billigsten sind asiatische Online-Shops. Allerdings dauert die Lieferung nach Europa oft einige Wochen.

Beim micro:bit-Roboter muss u. a. ein Stiftleistenstecker auf dem Motorcontroller-Board angelötet werden. Beim Arduino-Roboter müssen Kabel an die Motoren angelötet werden. Das ist nicht viel Arbeit, erfordert aber ein wenig Löt-Erfahrung. Wenn du das nicht selber machen kannst, kann dir vielleicht ein befreundeter Elektronik-Bastler damit helfen. Oder du lernst gleich vorher richtig löten (das kann auch Spass machen).

Für alle Versionen wird ein Computer (z. B. ein Notebook) mit USB-Anschluss, einem aktuellen Webbrowser und Internet-Anschluss benötigt. Für die Arduino-Version muss zusätzlich die Arduino-Entwicklungsumgebung („IDE“) installiert werden (eine Programmiersoftware).

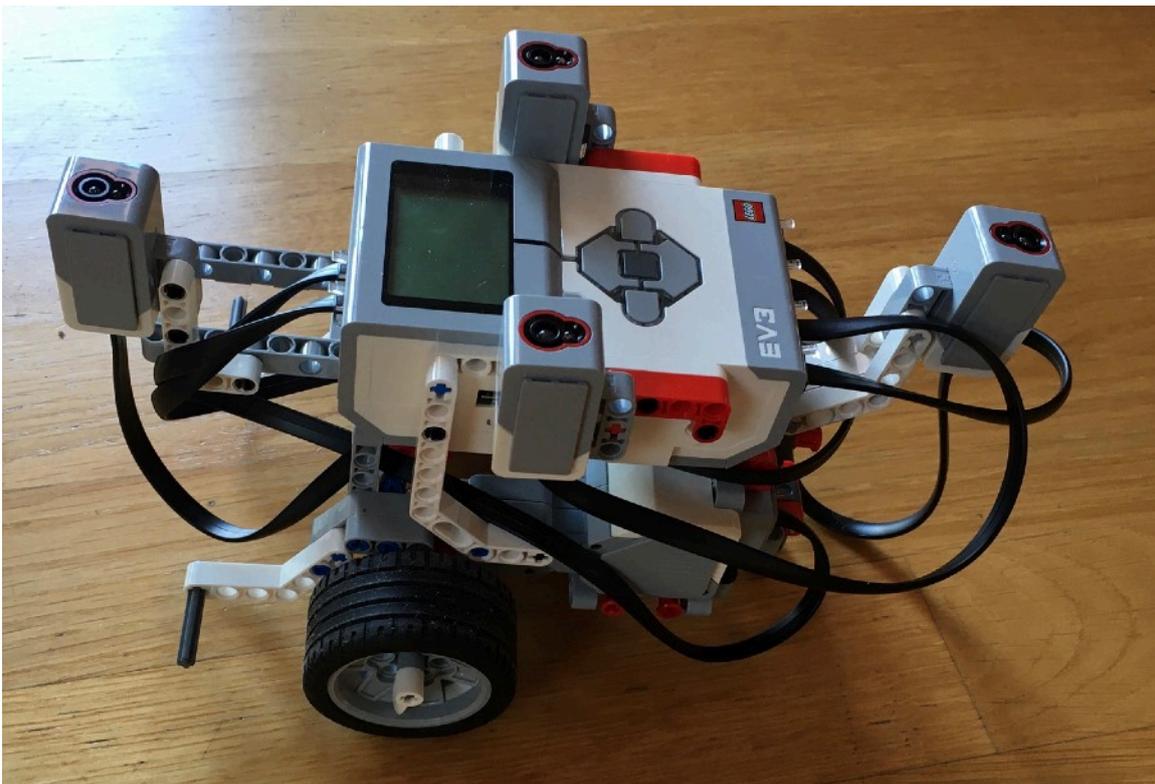
## 2. LEGO Mindstorms EV3

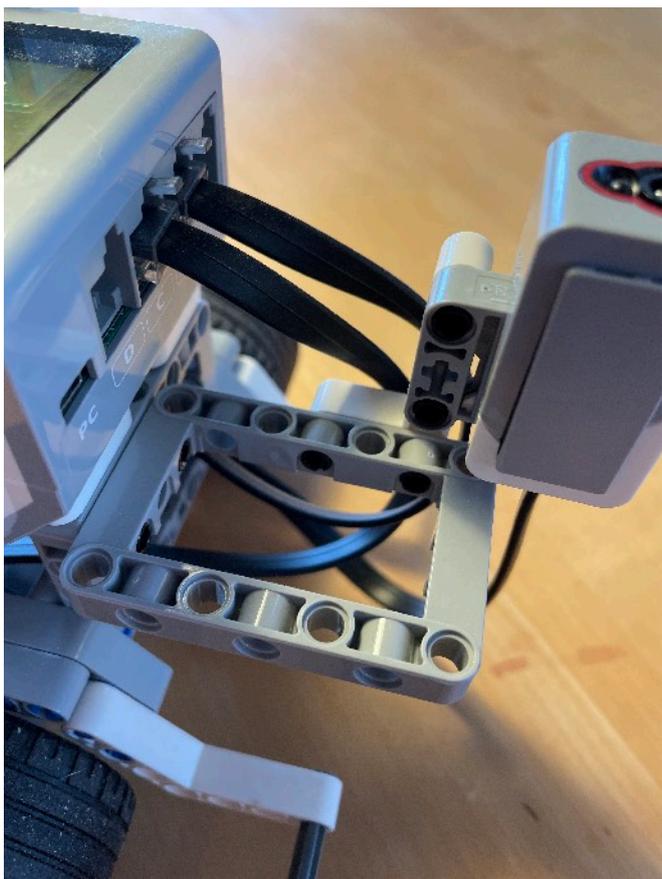
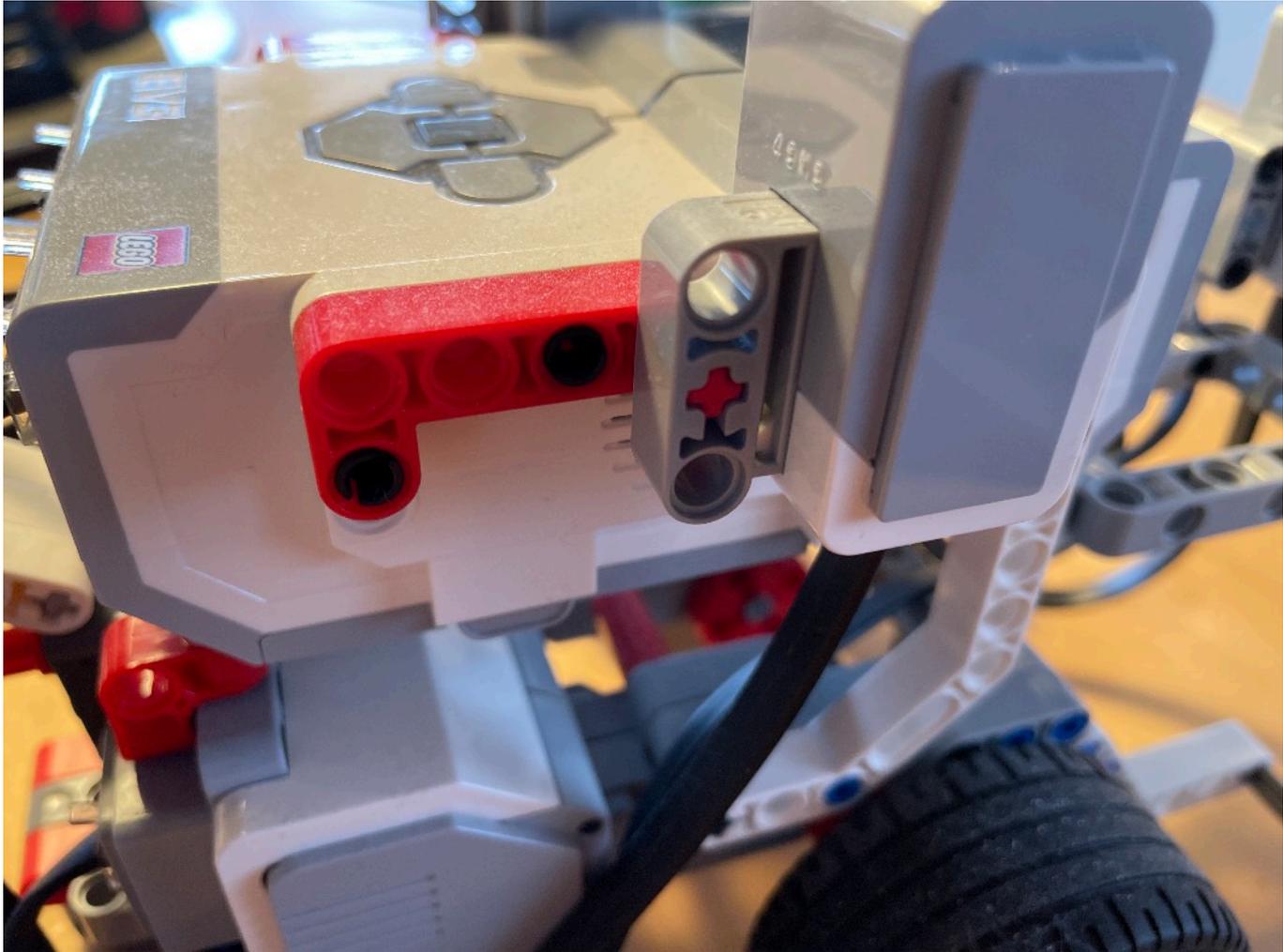
Der Aufbau des Roboters ist sowohl mit der Education-Version (graue Plastik-Box), wie auch mit der Consumer-Version (blaue Karton-Box) möglich. Benötigt werden das Kit sowie 3 zusätzliche Licht- und Farbsensoren (es braucht insgesamt vier Lichtsensoren, einer ist schon im Kit enthalten).

Für die Education-Version findest du eine brauchbare Bauanleitung unter der folgenden URL:

<https://education.lego.com/en-us/support/mindstorms-ev3/building-instructions#robot>

Baue das „Gyro-Sensor“-Modell auf. Montiere aber anstatt des Gyro-Sensors einen Lichtsensor. Die übrigen 3 Lichtsensoren kannst du gemäss den folgenden Bildern montieren (du kannst auch eine andere Lösung wählen, die Anordnung der Sensoren sollte aber ungefähr dem Bild entsprechen).





Verbinde Motoren und Sensoren gemäss der folgenden Tabelle mit dem Brick:

Aktor bzw. Sensor	Eingang / Ausgang des Bricks
Motor links	B
Motor rechts	C
Lichtsensoren vorne	1
Lichtsensoren hinten (wo die Kugel ist!)	4
Lichtsensoren links	2
Lichtsensoren rechts	3

Für die **Consumer-Version** kannst du dir einen einfachen 2-rädrigen Roboter („Trike“) gemäss der folgenden Anleitung bauen:

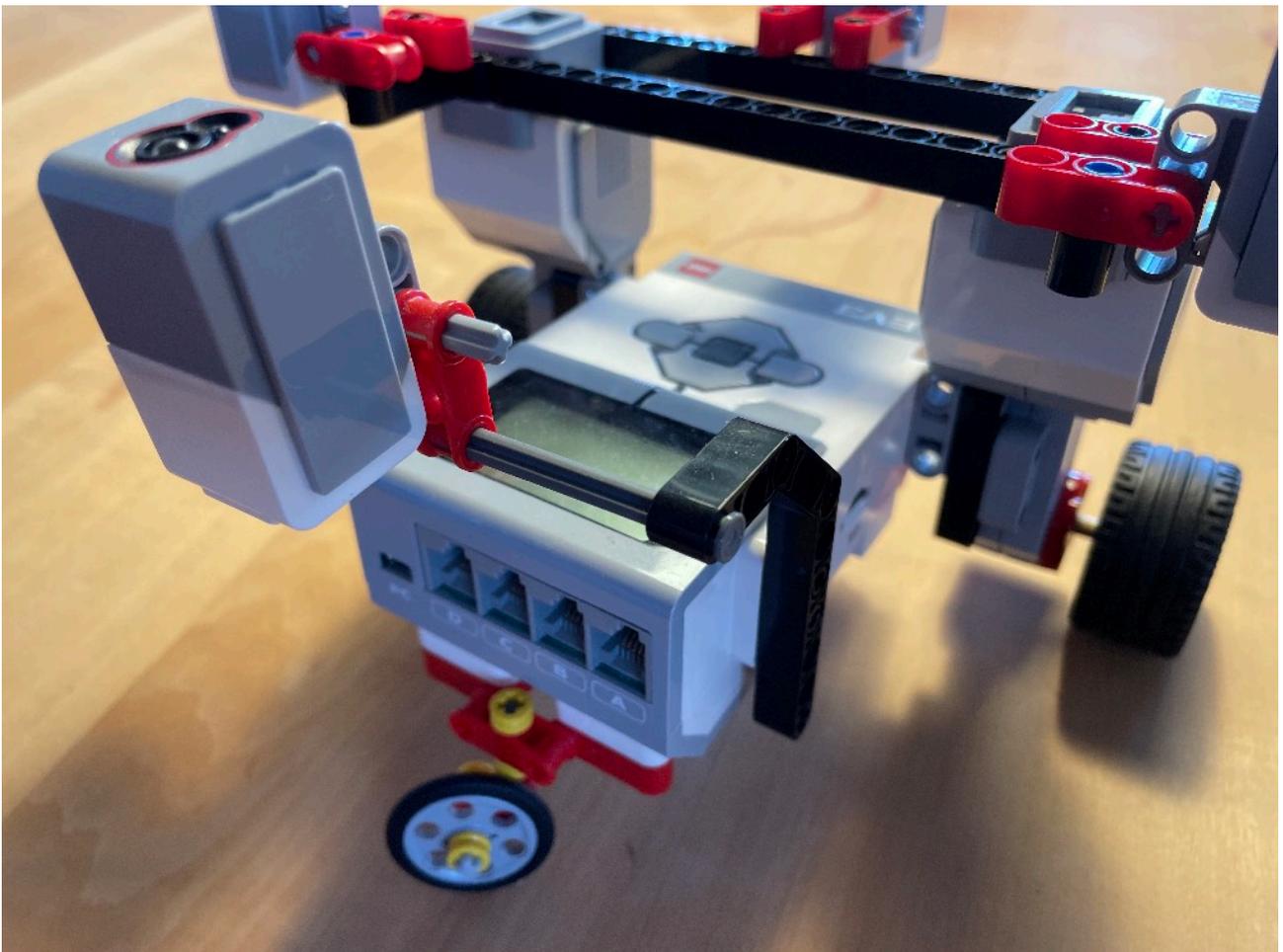
<https://www.youtube.com/watch?v=3bvn8AQ8m7o>  
(Englisch, aber man sieht alles ziemlich genau im Bild)

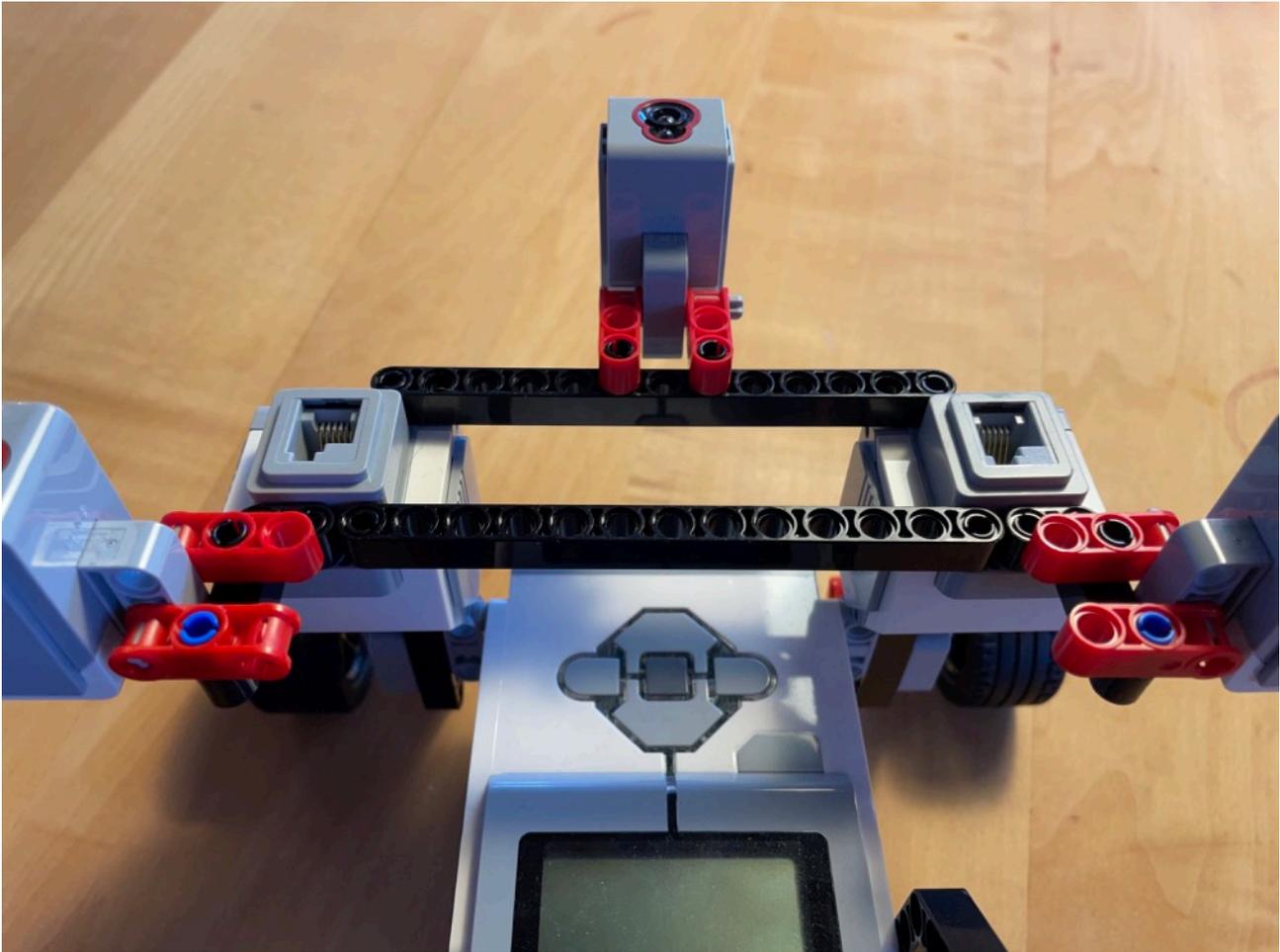
Fertig (inkl. der Sensoren) sieht der Roboter so aus:





Montiere nun mit ein paar weiteren Bauteilen die vier Lichtsensoren:





Die Verbindung von Motoren und Sensoren mit dem Brick ist gleich wie bei der Education-Version (siehe Tabelle oben). **Hinten ist dort, wo sich das kleine Rad befindet!**

Die Programmierung erfolgt über Microsoft MakeCode im Webbrowser:

<https://makecode.mindstorms.com>

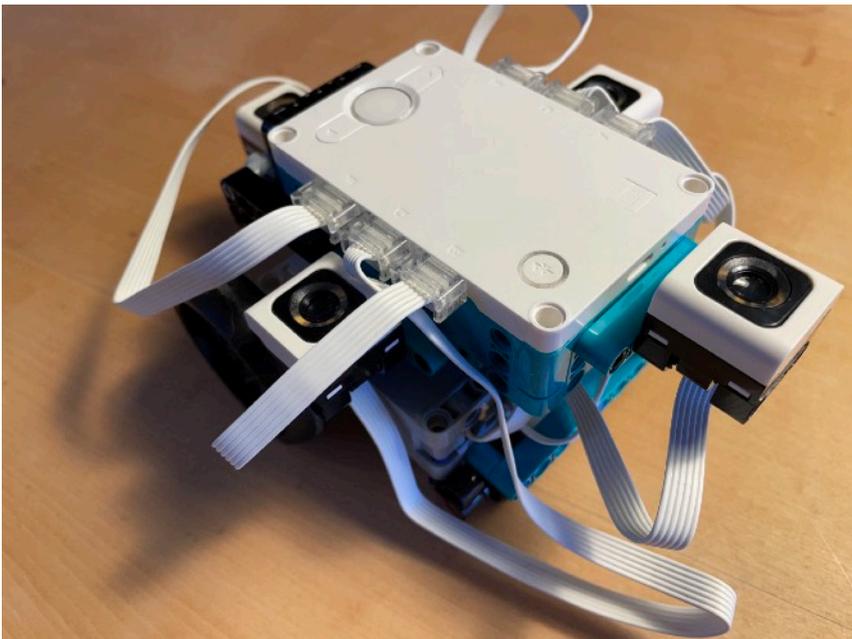
**BITTE BEACHTE, DASS U. U. (FALLS DU EINE ÄLTERE VERSION DES LEGO-BRICKS HAST), NOCH EIN SOFTWARE-UPDATE NÖTIG IST, DAMIT DER BRICK MITTELS MAKECODE PROGRAMMIERT WERDEN KANN.**

Über den Button „Importieren“ kannst du das Programm in die Programmierumgebung laden und bearbeiten. Dann kannst du es mit dem „Herunterladen“-Button auf deinen Computer herunterladen (z. B. auf den Desktop). Nun verbindest du den Brick mit dem beiliegenden USB-Kabel mit dem Computer. Der Brick wird sich wie ein USB-Stick am Computer anmelden. Du kannst nun das heruntergeladene Programm auf den Brick kopieren.

### 3. LEGO Mindstorms Robot Inventor

Der Lichtsuchende Roboter baut auf dem Roboterfahrzeug „Tricky“ auf. Du brauchst den Distanz-Sensor nicht zu montieren (wir brauchen ihn nicht und montieren an seiner Stelle einen Lichtsensor).

Montiere nun die vier Licht/Farbsensoren an allen Seiten des Bricks. Dazu kannst du z.B. grüne 6-er Stäbe verwenden (und natürlich schwarze Verbinder).

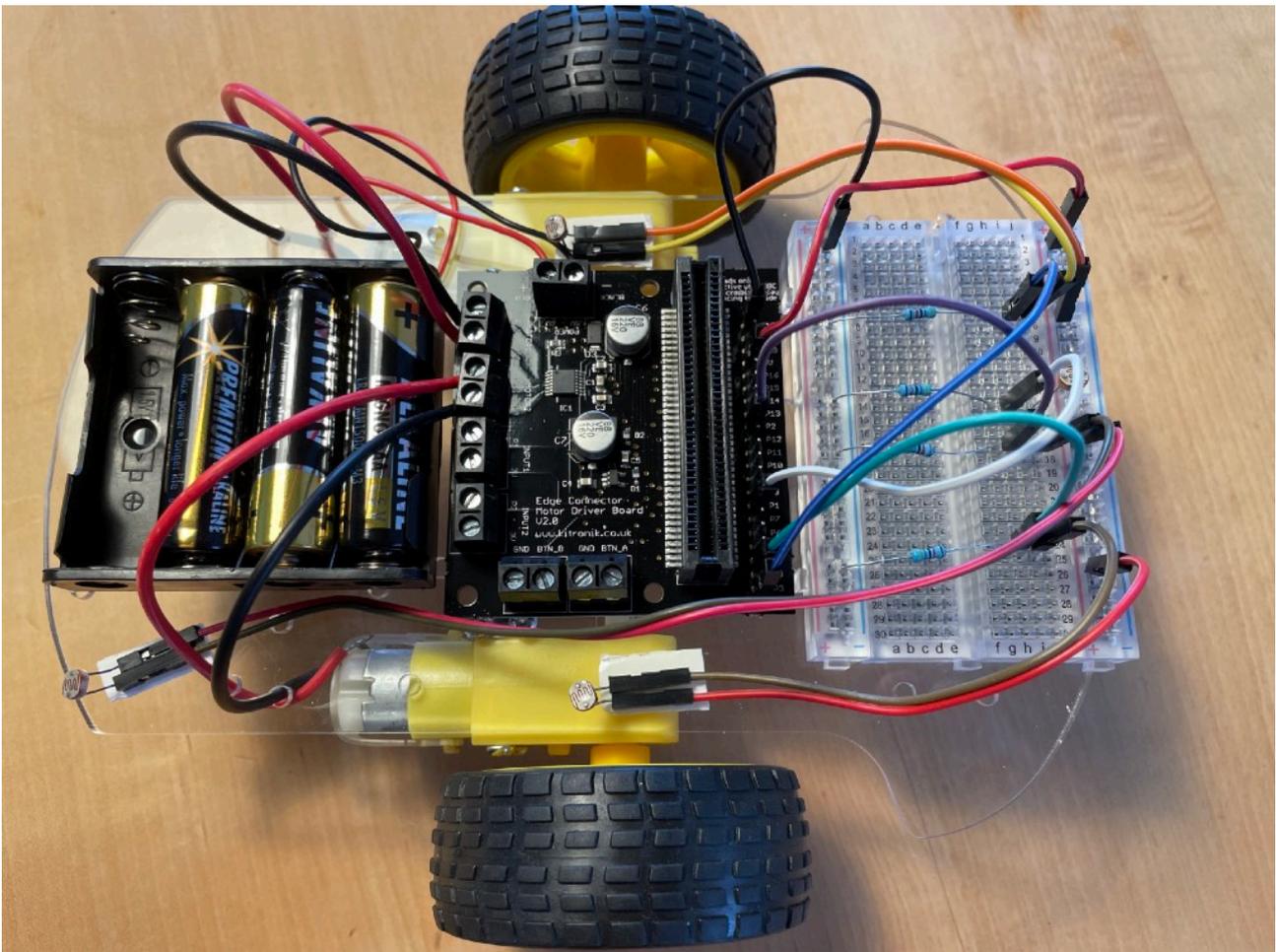


Die Motoren und Sensoren verbindest du wie folgt mit dem Brick:

Motor links	E
Motor rechts	A
Lichtsensor links	D
Lichtsensor rechts	C
Lichtsensor vorne	F
Lichtsensor hinten	B

**ACHTUNG: FÜR UNS IST DIE SEITE MIT DEN BEIDEN KLEINEN KUGELRÄDERN HINTEN!**

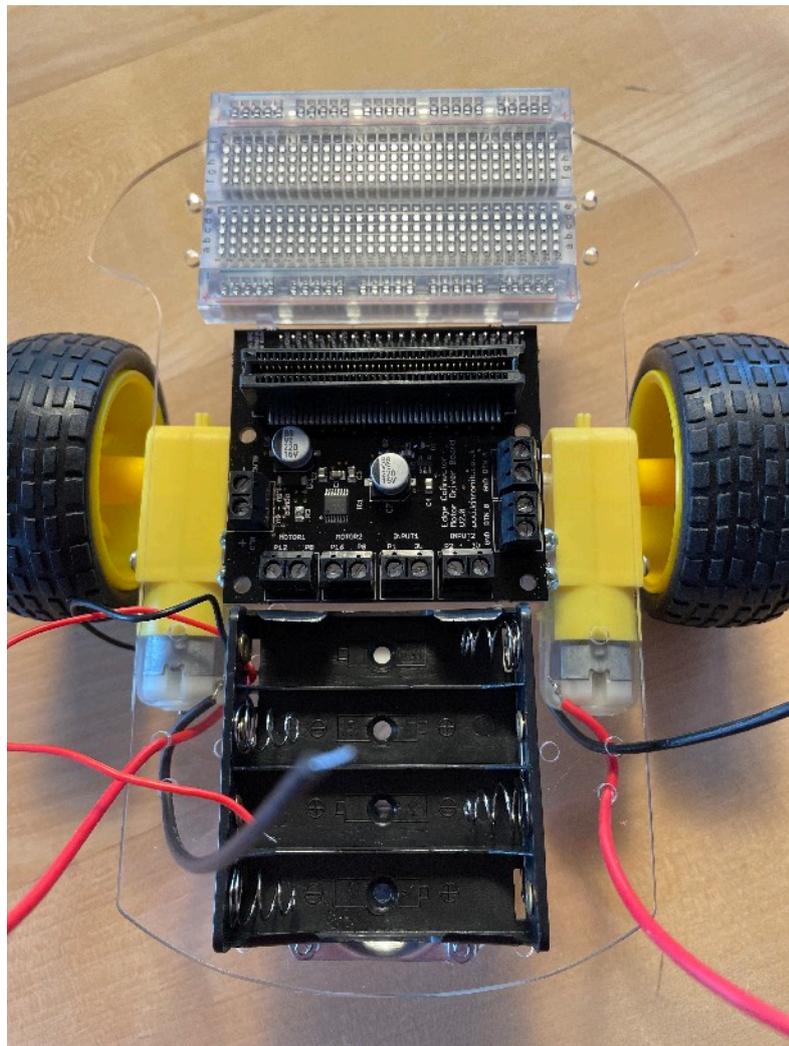
## 4. BBC micro:bit



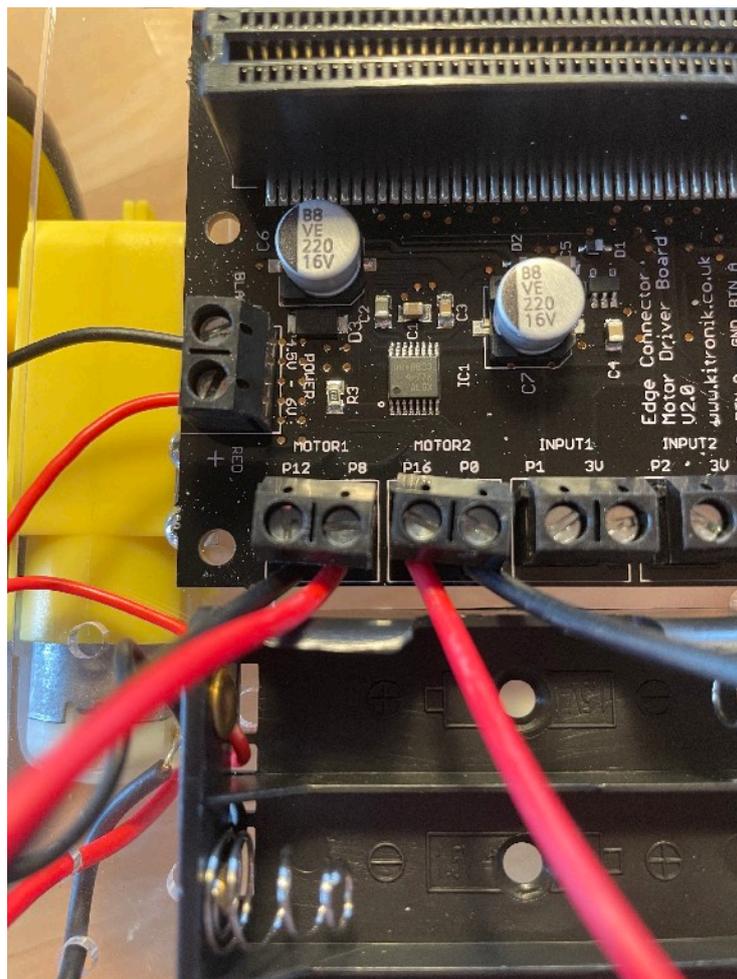
Zum Aufbau deslichtsuchenden KI-Roboters mit einer micro:bit-Steuerung werden folgende Bauteile benötigt:

- BBC micro:bit Einplatinencomputer
- „Kitronik Motor Driver Board for the BBC micro:bit“  
(URL: <https://kitronik.co.uk/collections/microbit-accessories/products/5620-motor-driver-board-for-the-bbc-microbit-v2> )  
**Es muss eine Steckerleiste eingelötet werden!**
- Car Robot Kit (siehe Bild). Es gibt diverse brauchbare Versionen. Wichtig ist, dass du eine Version mit nur 2 angetriebenen Rädern wählst (nicht vier). Bei Versionen mit den gelben „TT-Motoren“ kann nicht viel schief gehen.
- Batteriehalterung für 4x AA Batterien  
(idealerweise mit einem integrierten ON/OFF-Schalter). Oft beim Car Robot Kit schon dabei.

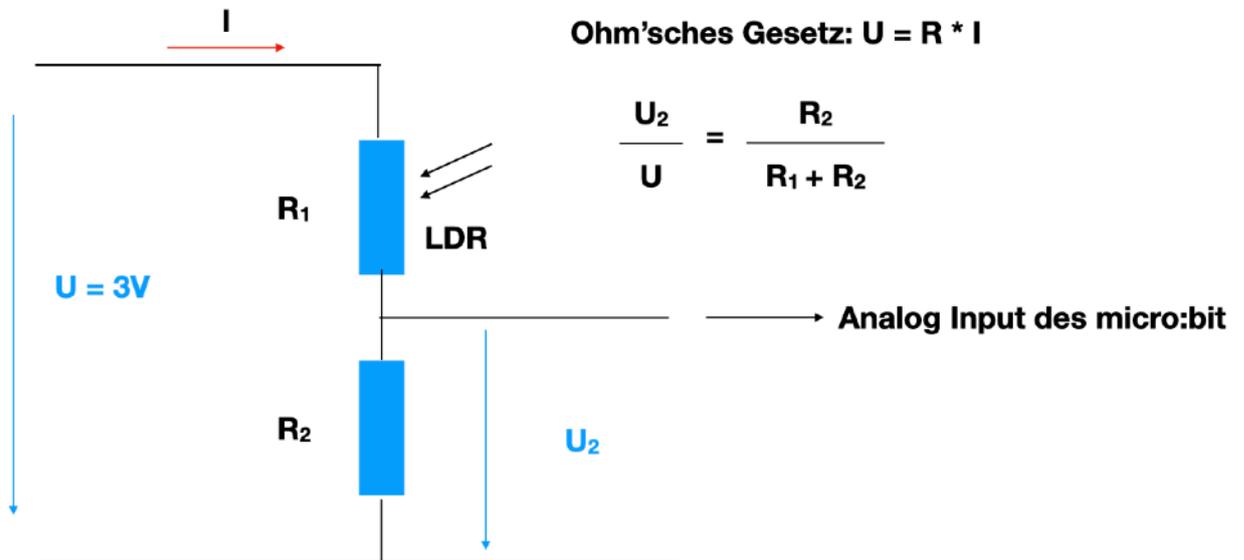
- Mini-Breadboard (halbe Grösse, d. h. 400 Kontakte)
  - 4 Widerstände 47kOhm 1/4W
  - 4 Stück LDR (lichtempfindlicher Widerstand) vom Typ 5539
  - Jumper-Kabel gemäss folgender Liste:
    - 2x 20cm männlich-weiblich
    - ca. 15x10cm männlich-weiblich
  - Doppelseitiges Montage-Klebeband (stark)
  - 4 AA Batterien (Alkali-Mangan)
1. Löte zuerst die Motorkabel an den Motoren fest. Wenn du keinen LötKolben hast, oder noch nicht löten kannst, kann dir sicher ein befreundeter Elektronik-Bastler dabei helfen (das ist in 2-3 Minuten erledigt). Baue dann das Car-Robot Kit zusammen (gemäss beiliegender Anleitung).
  2. Klebe das Breadboard, das Controller-Board und den Batteriehalter mit dem starken Montageband auf dem Chassis fest:



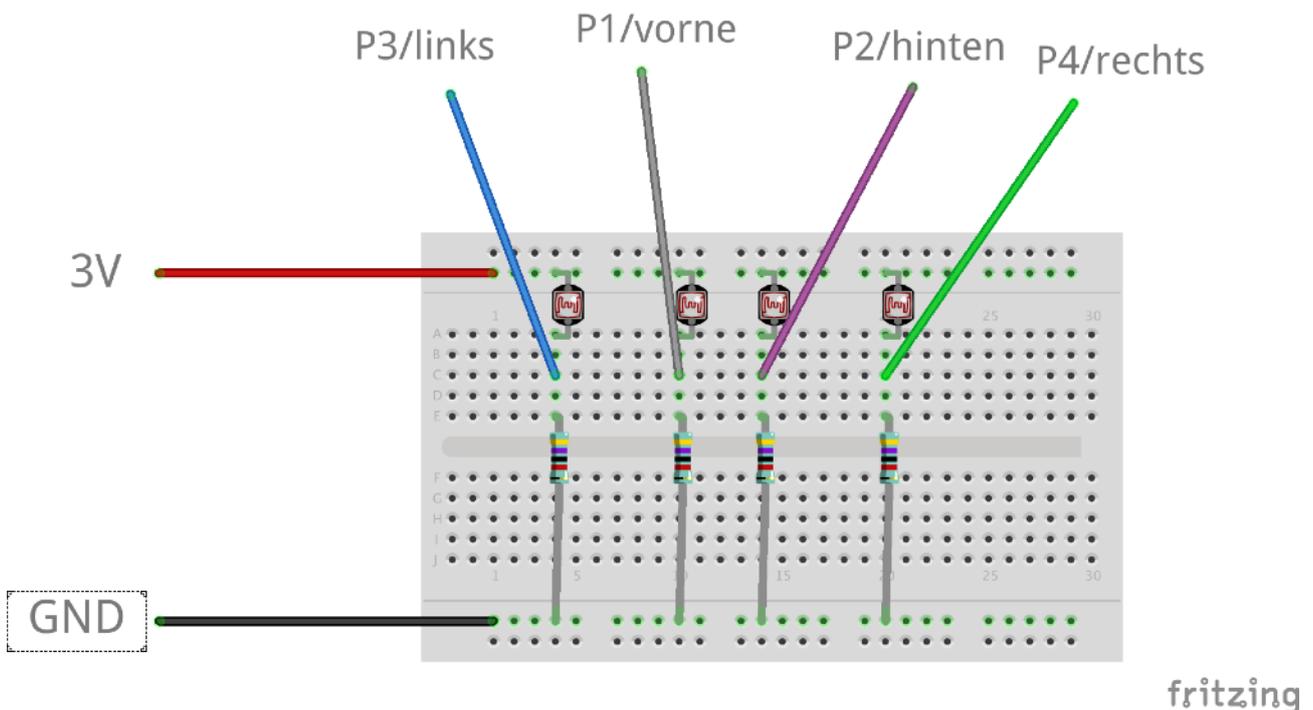
3. Klemme die Batteriekabel und die Motorkabel auf dem Controller-Board fest. Bitte **beachte unbedingt die Polarität der Stromversorgung** (plus=rot, minus=schwarz). Der Anschluss MOTOR1 des Controller-Boards sollte mit dem (von hinten gesehen) linken Motor belegt werden. Wenn du die Motoren mit falscher Polarität anschliesst (d. h. rot und schwarz verwechselst), dann dreht der entsprechende Motor in der falschen Richtung (d. h. zum Bsp. rückwärts statt vorwärts). Vertausche dann einfach die Kabel.



4. Stecke nun die Schaltung für die Sensoren auf dem Breadboard. Für jeden Sensor musst du einen Spannungsteiler aus einem Widerstand und einem LDR (lichtempfindlicher Widerstand) bauen:



Verbinde dazu zuerst die Pins GND und 3V mit dem Breadboard. GND zur nächsten blauen Minus-Leiste und 3V zur gegenüberliegenden roten Plus-Leiste. Nun kannst du die Spannungsteiler zwischen diesen Leisten aufbauen. Zwischen den beiden Widerständen kann man eine zur Lichtstärke proportionale Spannung  $U_2$  abgreifen. Verbinde diese Punkte wie folgt mit dem Controller-Board:



**AUF DEM BREADBOARD-SCHEMA SIND DIE LDR DIREKT EINGEBAUT EINGEZEICHNET, ABER DU MUSST SIE NATÜRLICH ÜBER JUMPER-KABELN AN DER RICHTIGEN STELLE PLATZIEREN.**

Breadboard	Motor-Controller Board (analoge Eingänge)
Sensor vorne	Pin P1
Sensor hinten	Pin P2
Sensor links	Pin P3
Sensor rechts	Pin P4

Verbinde nun die Sensoren (LDR) mit dem Breadboard. Den VORNE-LDR kannst du direkt auf dem Breadboard einbauen. Für die anderen brauchst du Jumper-Kabel (männlich auf der Breadboard-Seite und weiblich auf der LDR-Seite). Falls der LDR in einem Kabel nicht gut hält, kannst du den Stecker mit einer kleinen Flachzange andrücken. Du kannst die Sensoren mit etwas Montageband befestigen (die Fläche des LDR sollte nach oben zeigen)

Das sieht dann insgesamt etwa so aus, wie auf dem Bild am Anfang der Anleitung.

Nun kannst du das Programm zum Testen der Motoren auf das micro:bit laden und dann das micro:bit in das Controller-Board stecken (**das Display des micro:bit muss unbedingt zur Steckerleiste hinzeigen!**). Wenn du nun die Batterien einlegst, sollte zuerst das linke Rad vorwärts (d. h. so, dass sich der Roboter vorwärts bewegen würde) und dann rückwärts drehen. Danach sollte das rechte Rad das Gleiche tun.

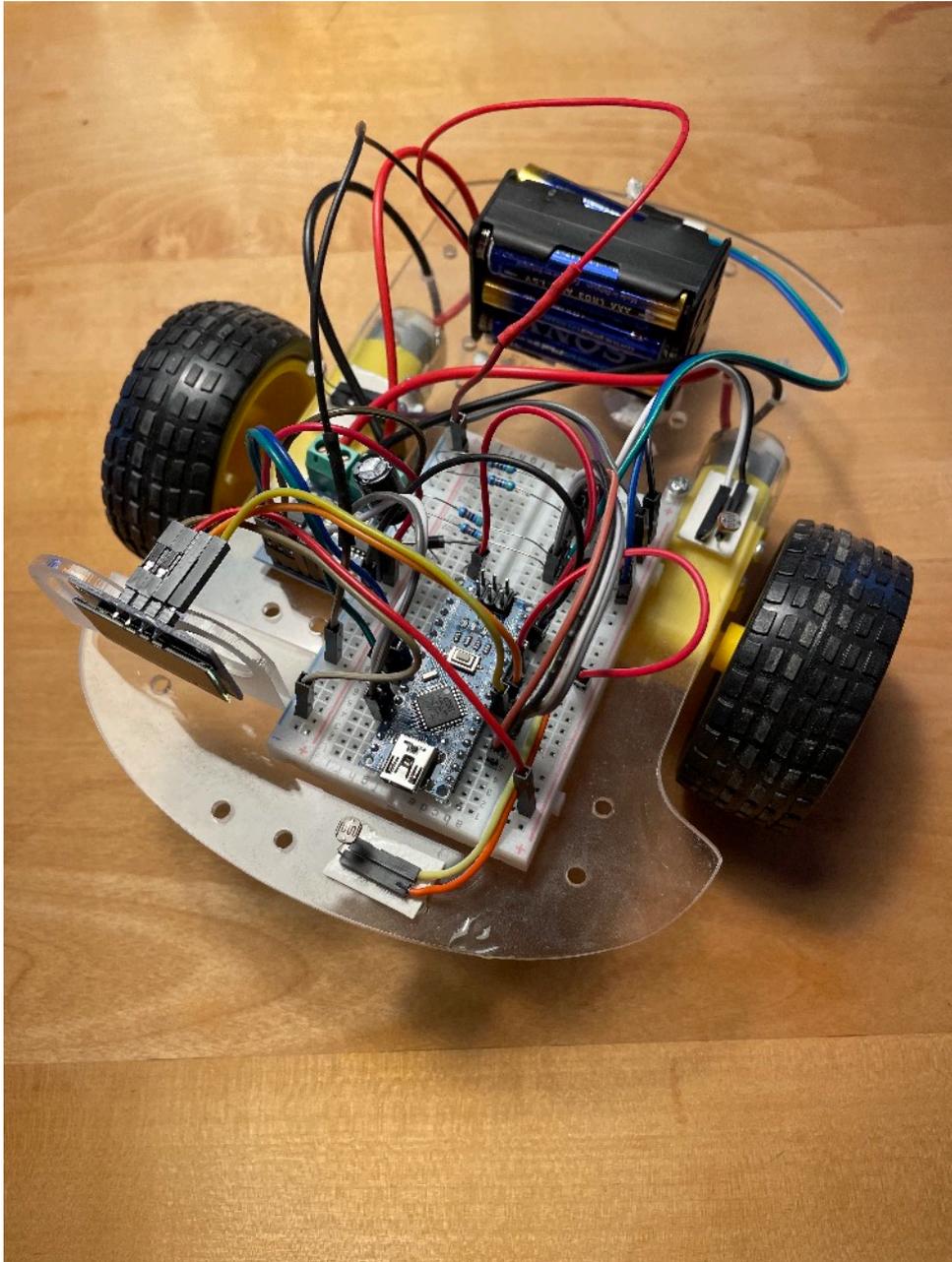
Regeln zur Fehlerbehebung:

- Falls das rechte Rad zuerst dreht: Anschlüsse MOTOR1 und MOTOR2 vertauscht.
- Falls ein Rad zuerst rückwärts dreht und dann vorwärts: Polarität des entsprechenden Motors vertauscht.

Nun kannst du das Programm zum Testen der Sensoren auf das micro:bit laden. Wenn du die Batterie einlegst, werden auf dem Display die Lichtstärken der Sensoren dargestellt (Anzahl LEDs/Lämpchen von oben nach unten). Leider ist die Darstellung des Displays leicht gestört, da die analogen Eingänge auch für das Display benutzt werden (und somit das Display von den Sensoren teilweise gestört wird). Nimm eine Taschenlampe und halte sie einzeln auf die verschiedenen Sensoren. Es sollten jeweils in einer Spalte mehrere Lämpchen aufleuchten. Falls das bei einem Sensor nicht klappt, hast du noch einen Fehler in der Schaltung oder einen Wackelkontakt. Überprüfe in diesem Fall die Schaltung noch einmal ganz genau.

Wenn beide Testprogramme korrekt laufen, kannst du das KI-Programm (inkl. dem default-NN) auf dem micro:bit installieren und den Roboter mit einer starken Taschenlampe testen!

## 5. Arduino



## 5.1 Benötigte Bauteile

Zum Aufbau deslichtsuchenden KI-Roboters mit einer Arduino-Steuerung werden folgende Bauteile benötigt:

- Arduino-Nano (Original oder Kopie)
- USB-Programmierkabel passend zum Arduino-Nano
- „Two Channel Motor Driver Board L9110S“  
(z. B. <https://kitronik.co.uk/products/46134-two-channel-motor-driver-board-l9110s> )
- Breadboard (halbe Grösse, d.h. 400 Kontakte)
- Car-Robot-Kit. Es gibt diverse brauchbare Versionen. Wichtig ist, dass du eine Version mit nur 2 angetriebenen Rädern wählst (nicht vier). Die gelben „TT-Motoren“ sind brauchbar.
- Batteriehalterung für 6x AAA Batterien  
(du kannst auch eine 9V Blockbatterie verwenden, aber der Roboter wird damit deutlich weniger lange funktionieren)
- 4 Widerstände 47kOhm 1/4W
- 4 Stück LDR (lichtempfindlicher Widerstand) vom Typ 5539
- 1x Schottky-Diode vom Typ 1N5817 (oder ähnlich)
- 1x Elektrolytkondensator 500uF, 16V
- Genügend Jumper-Kabel aller Sorten (10cm und 20cm).
- Doppelseitiges Montage-Klebeband (stark)
- 6 AAA Alkali-Mangan Batterien (empfohlen) bzw. 9V Blockbatterie (hält nicht sehr lange!)

Zudem benötigst du zur Programmierung einen Computer mit USB-Anschluss und die sog. integrierte Entwicklungsumgebung (IDE) für den Arduino. Du kannst diese unter folgender URL herunterladen:

<https://www.arduino.cc/en/main/software>

**BITTE BEACHTE, DASS ZUR PROGRAMMIERUNG MANCHER ARDUINO NANO VARIANTEN (Z.B. KOPIEN AUS ASIATISCHEN ONLINE-SHOPS) EIN ZUSÄTZLICHER USB-TREIBER INSTALLIERT WERDEN MUSS. EIN SOLCHER IST BEIM ORIGINAL VON ARDUINO NICHT NÖTIG.**

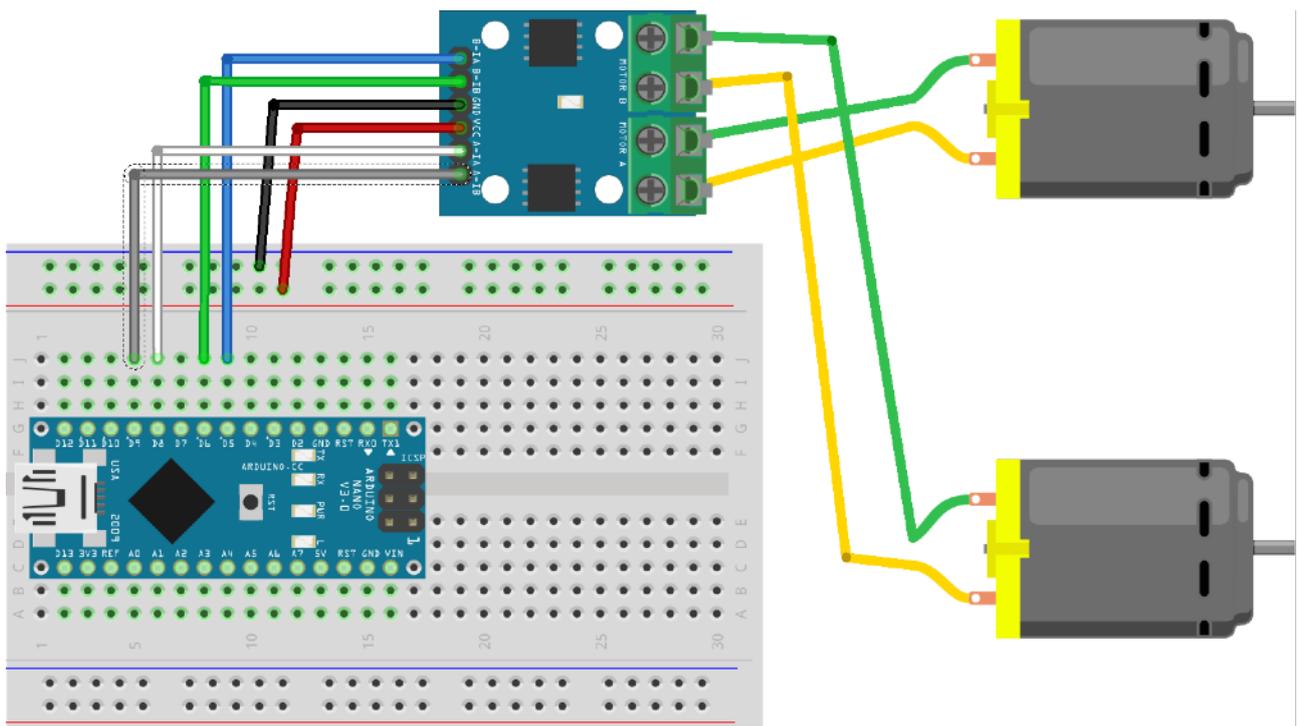


## 5.3 Motorsteuerung

Nun verbinden wir das Motor-Controller Board mit den Motoren und dem Arduino:

Beachte, dass wir das Motorcontroller-Board mit den vollen 9V ansteuern (nicht den stabilisierten 5V). Das gibt den Motoren mehr Kraft und verhindert, dass der Arduino überlastet wird (die Motoren brauchen ziemlich viel Strom). Die Stromversorgung geschieht über die mittleren beiden Pins (rot/schwarz) des Boards.

Verbinde das Board mittels folgender Tabelle mit dem Arduino:



fritzing

Arduino	Motor-Controller Board
Minus von Batterie	GND
Plus von Batterie	VCC
D5	B-1A
D6	B-1B
D8	A-1A
D9	A-1B

Nun kannst du die Motoren mit dem Motor-Controller-Board verbinden. Verbinde den „Motor-A“-Ausgang mit dem rechten Motor und den „Motor-B“-Ausgang mit dem linken Motor. Die Polarität spielt vorerst einmal keine Rolle (falls du sie falsch erwischst, läuft der entsprechende Motor in die falsche Richtung, was du aber leicht durch Tauschen der Motor-Kontakte korrigieren kannst).

Nun hast du eigentlich schon einen programmierbaren Roboter mit dem du erste Experimente machen kannst.

Installiere nun das Motor-Testprogramm auf dem Arduino:

1. Falls nötig, USB-Treiber für den Arduino Nano installieren (siehe oben)
2. Starte die Arduino-IDE
3. Öffne das Testprogramm in der Arduino-IDE
4. Stelle sicher, dass in der IDE folgende Einstellungen gemacht sind:  
„Tools“ > „Board“ > „Arduino AVR Boards“ > „Arduino Nano“
5. Verbinde den Arduino mittels des USB-Kabels mit dem Computer (die Batterie sollte dabei vom Breadboard getrennt sein)
6. Lade das Programm auf den Arduino durch „Sketch“ > „Upload“

Nun kannst du die USB-Verbindung trennen und die Stromversorgung anschliessen:

Der Roboter sollte nun folgendes machen:

- Zuerst dreht das linke Rad zuerst vorwärts, dann rückwärts (d. h. der Roboter würde zuerst vorwärts fahren, dann rückwärts)
- Dann dreht das rechte Rad zuerst vorwärts, dann rückwärts

Die Regeln zur Fehlerbehebung:

- Falls nun zuerst das rechte Rad dreht, hast du die Motoren am jeweils falschen Controller-Ausgang angeschlossen. Du musst also einfach die Motoren tauschen.
- Falls das linke oder das rechte Rad zuerst rückwärts dreht (statt vorwärts), musst du die Kabel am entsprechenden Motorcontroller-Ausgang tauschen.



Die Sensoren bestehen aus vier Spannungsteilern aus je einem Widerstand und einem LDR („Light **D**ependent **R**esistor“ = lichtempfindlicher Widerstand). In der Mitte des Spannungsteilers greifen wir eine der Lichtintensität proportionale Spannung ab und führen sie auf einen analogen Eingang (A0 bis A3) des Arduinos:

Sensor	Arduino Nano PIN
Sensor vorne	A2
Sensor hinten	A3
Sensor links	A0
Sensor rechts	A1

**ACHTUNG: IM BILD SIND DIE LDR DIREKT AUF DEM BREADBOARD EINGEZEICHNET. DU MUSST SIE ABER AN VERSCHIEDENEN STELLEN DES ROBOTERS MIT MONTAGEBAND MONTIEREN UND MITTELS JUMPER-KABELN MIT DEM BREADBOARD VERBINDEN.**

Es lohnt sich, als nächstes gleich das (optionale) OLED-Display zu montieren, weil dieses die Fehlersuche an den Sensoren sehr erleichtert.



Arduino bzw. Breadboard	Display
5V	VCC
GND	GND
A4	SDA
A5	SCK

## 5.7 Programmierung

Nun kann der Roboter mit der KI programmiert werden.

Der Vorgang ist genau gleich, wie beim Motortest-Programm. Wenn du das Display verwenden willst, musst du vorher die SSD1306-Library installieren (im Kit dabei):

„Sketch“ > „Include Library“ > „Add ZIP Library“

Die Software zeigt nach dem Starten Informationen über die Sensoren an. Das ist sehr nützlich, um Fehler in der Hardware (z. B. Wackelkontakte) zu finden.

## 6. PGLU-Roboterfahrzeug

### 6.1 Mechanik

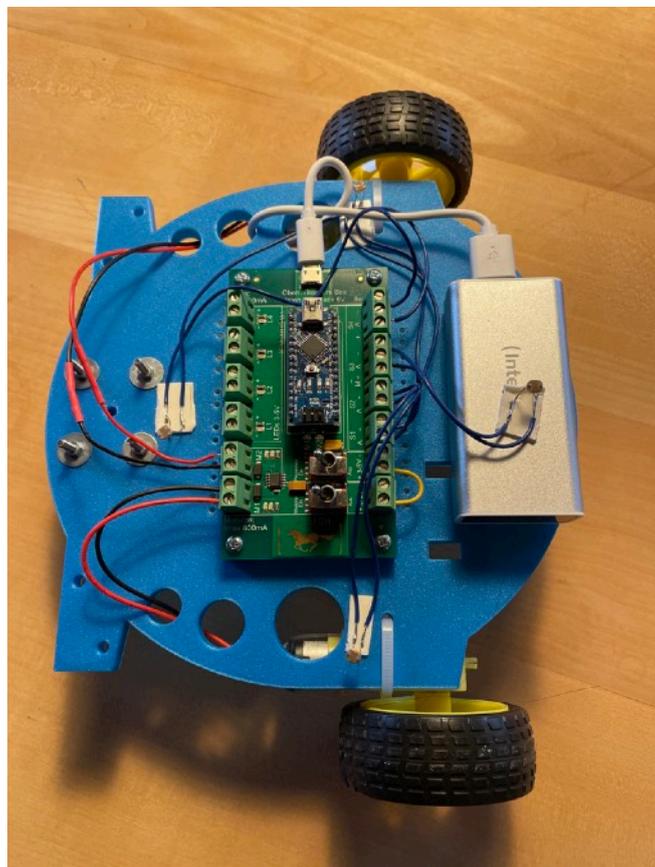
Baue das Roboterfahrzeug gemäss der Anleitung von PGLU zusammen (<https://pglu.ch>). Die Fühler und den Ultraschallsensor benötigen wir für die KI-Experimente nicht. Du kannst diese also weglassen, falls du den Roboter nur für den KI-Kurs verwenden möchtest.

Du benötigst 4 LDR Sensoren von PGLU (d.h. zwei Verpackungseinheiten mit je 2 Sensoren). Diese werden wie folgt mit den Sensor-Eingängen verbunden:

Sensor	PGLU Sensor-Eingang
Sensor vorne	S3
Sensor hinten	S4
Sensor links	S1
Sensor rechts	S2

Die Sensoren werden jeweils mit den Pins - und ^ verbunden (die Polarität spielt keine Rolle). Du kannst die LDR-Sensoren an kurzen Kabeln anlöten oder Jumperkabel verwenden. Zur Befestigung auf der Grundplatte eignet sich Montage-Klebeband.

Hier ein Bild des fertigen Roboters (Variante Chassis aus dem 3D-Drucker):



## 6.2 Programmierung

Die Programmierung erfolgt in Arduino C, da die grafische Programmierumgebung z.Z. noch keine Arrays unterstützt. Lies die Anleitung dazu (2020-PGLU-Anleitung-programmieren-ohne-Teachers-Box.pdf) genau durch und konfiguriere die Arduino-IDE entsprechend.

Es sind folgende Arduino-Sketches vorhanden:

- **Motor-Test**  
Mit diesem Sketch kannst du prüfen, ob die Motoren korrekt angeschlossen sind. Es sollte zuerst das linke Rad vorwärts und dann rückwärts drehen. Nachher sollte das Rechte Rad dasselbe tun. Bitte beachte dabei: hinten ist für uns immer dort, wo das einzelne Rad angebracht ist (beim PGLU-Roboterfahrzeug sind dort aber die Fühler angebracht, d.h. es ist eigentlich vorne). Wenn du unsicher bist, schau dir das Foto oben an.
- **Sensor-Test**  
Mit diesem Sketch kannst du testen, ob die Sensoren funktionieren. Die Sensoren S1-S4 steuern die LEDs L1-L4 an. Die Lichtstärke der LEDs ist proportional zur einfallenden Lichtstärke. Du kannst einzelne Sensoren mit einem Finger abdecken und prüfen, ob dabei die richtige LED dunkler wird.
- **KI-Code**  
Dies ist nun das Programm mit dem neuronalen Netz, welches den Roboter steuert (Lichtsuchfunktion). Wenn die ersten beiden Sketches korrekt funktionieren, sollte dieses Programm ebenfalls problemlos funktionieren.