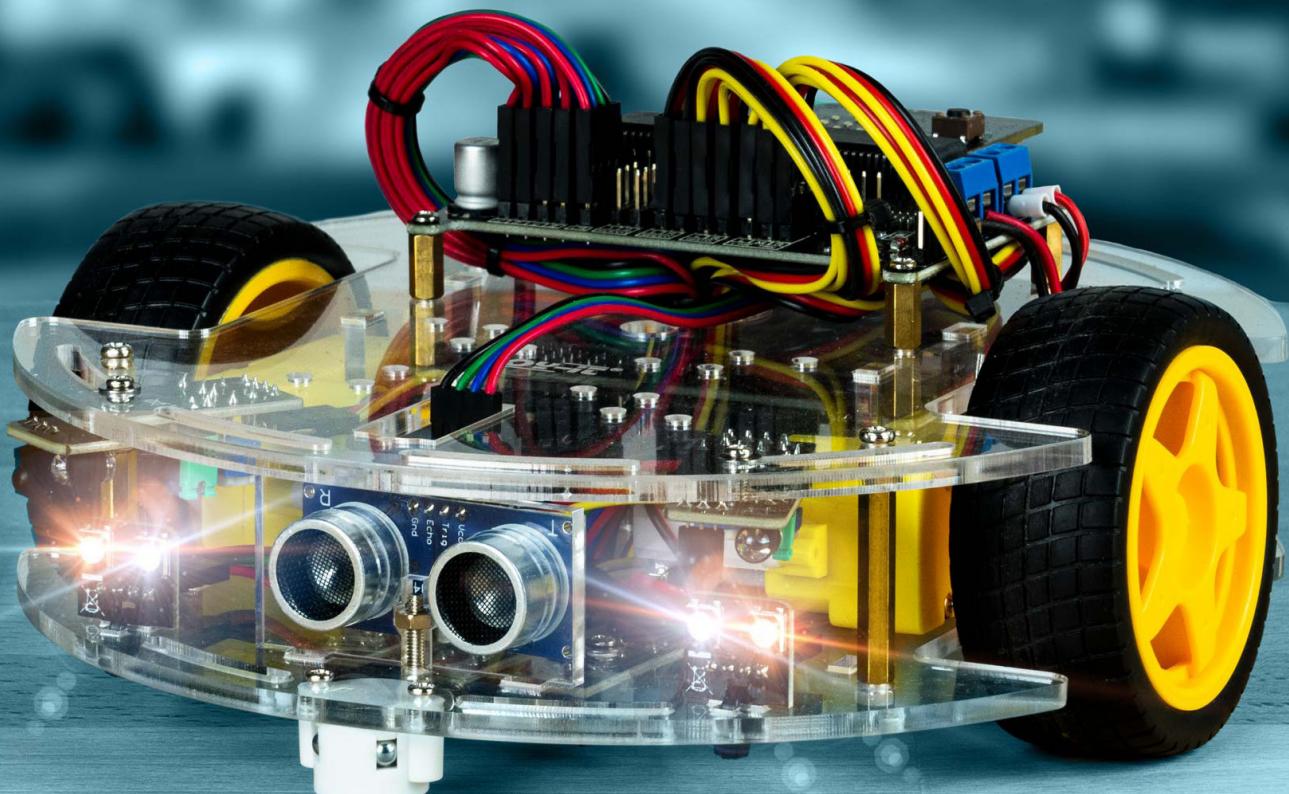


JOY-CAR

Educationroboter auf micro:bit Basis



INHALTSVERZEICHNIS

5	BAUTEILE	37
6	CHASSIS	38
7	ELEKTRONIK	41
8	MONTAGE-MATERIAL	45
9	MONTAGE	48
10	MONTAGE DER BODENGRUPPE	49
13	MONTAGE DES ANTRIEBSSTRANGS	49
15	MONTAGE DER ELEKTRONIK	52
19	MONTAGE DES CHASSIS-AUFSATZES	59
21	ULTRASCHALL- ALTERNATIVMONTAGE	65
24	ANSCHLUSSVERKABELUNG	
	TRAINING	
	SENSOREN	
	KOMMUNIKATION	
	DETAILS	
	QUAL DER WAHL	48
	MAKECODE	49
	EINFÜHRUNG	49
	DER ERSTE START	52
	JOY-CAR DIE ERWEITERUNG	59
	DAS VOLLE PROGRAMM	65



66

MICROPYTHON

66

EINFÜHRUNG

67

INTERFACE

69

DER ERSTE START

71

CODE

75

LOSLEGEN

76

SUPPORT

JOY-CAR

Das Joy-Car ist ein modular aufgebautes Robotik-Lernkit, welches zum Erlernen der Baugruppen und deren Funktionen als ganze Maschine dient. Die detaillierte Anleitung und die Programmierung machen das Lernen besonders einfach. Das Joy-Car verfügt über Sensoren wie z. B. Line Finder-Sensoren und einen Ultraschall-Sensor sowie über eine programmierbare RGB-Beleuchtung. Über einen zweiten micro:bit lässt sich das Joy-Car zusätzlich durch den Beschleunigungssensor über eine kabellose BT-Verbindung steuern.

BAUTEILE



Willkommen bei deinem eigenen Joy-Car-Bausatz! Es gibt vieles zu entdecken, doch auf diese Reise musst du nicht alleine gehen. In den folgenden Kapiteln zeigen wir dir Schritt für Schritt, wie du aus deinem Bausatz dein eigenes, fahrbares Projekt kreierst.

Mach dich doch schon mal mit den Bauteilen vertraut, danach legen wir los!



CHASSIS



BODENGRUPPE
1x



CHASSIS-AUFSATZ
1x



BATTERIE-HALTERUNG A
1x



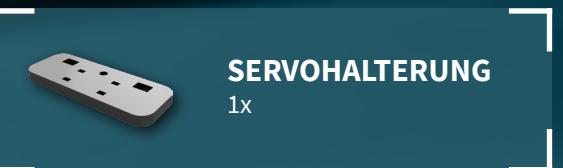
BATTERIE-HALTERUNG B
1x



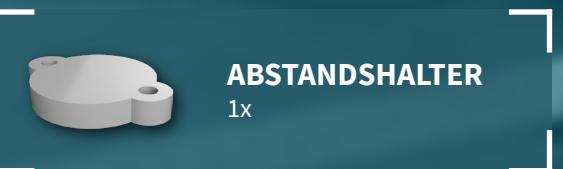
BATTERIE-HALTERUNG C
1x



MOTORHALTERUNG
2x



SERVOHALTERUNG
1x



ABSTANDSHALTER
1x



PARKHALTERUNG A
4x



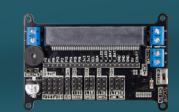
PARKHALTERUNG B
2x



ULTRASCHALL-HALTERUNG
1x



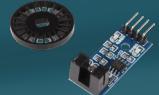
ELEKTRONIK



MAINBOARD
1x



LED BOARD
4x



SPEEDSENSOR
2x



**ULTRASCHALL-
SENSOR**
1x



**LINETRACKING-
SENSOR**
3x



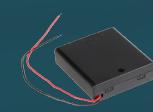
HINDERNISSENSOR
2x



SERVOMOTOR
2x



MOTOR
2x



BATTERIEFACH
1x



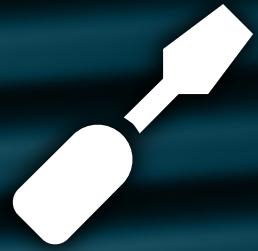
**3-PIN DUPONT-
KABELSATZ**
7x



**4-PIN DUPONT-
KABELSATZ**
5x



MICRO:BIT
1x
NICHT IN ALLEN JOY-CAR-SETS
ENTHALTEN!



MONTAGE-MATERIAL



SCHRAUBE
4x M2,5 x 5 mm



SCHRAUBE
12x M2,5 x 10 mm



SCHRAUBE
4x M2,5 x 22 mm



SCHRAUBE
6x M3 x 8 mm



SCHRAUBE
5x M3 x 14 mm



UNTERLEGSCHEIBE
30x M2,5



MUTTER
20x M2,5



MUTTER
15x M3



ABSTANDHALTER
4x M2,5 x 10 mm



ABSTANDHALTER
4x M3 x 30 mm

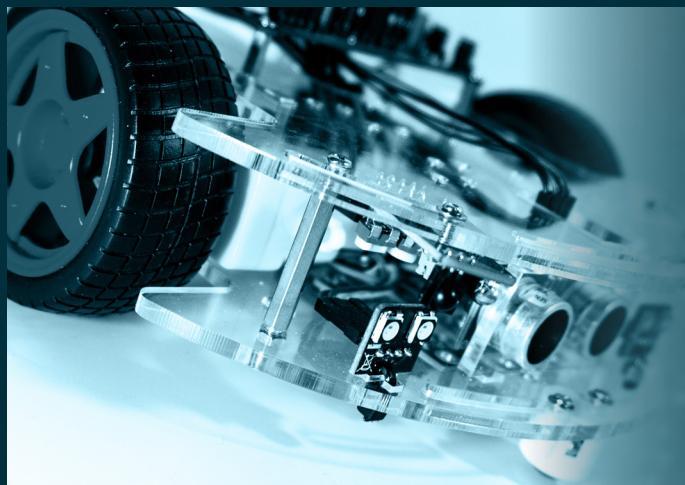


GLEITKUGEL
2x



KABELBINDER
15x

MONTAGE



Bevor du mit deinem Joy-Car loslegen kannst, muss dieses zunächst zusammengebaut werden. Aber keine Sorge! Auch wenn es viele Bauteile, Module und Kabel gibt, lassen wir dich nicht am Straßenrand stehen. In diesem Kapitel zeigen wir dir Schritt für Schritt wie du dein Joy-Car zusammensetzt.

Du wirst sehen, es ist leichter als gedacht und so lernst du alle Einzelheiten des Joy-Cars kennen!

MONTAGE DER BODENGRUPPE

Als Erstes erfolgt die Montage der Bodengruppe. Hier werden zunächst alle Halterungen und Abstandshalter angebracht, die im weiteren Verlauf des Aufbaus benötigt werden.

ACHTUNG! Alle Acrylteile sind mit einer Schutzfolie beklebt. Diese solltest du vor der Montage entfernen.

BODENGRUPPE

ANTRIEBS-
STRANG

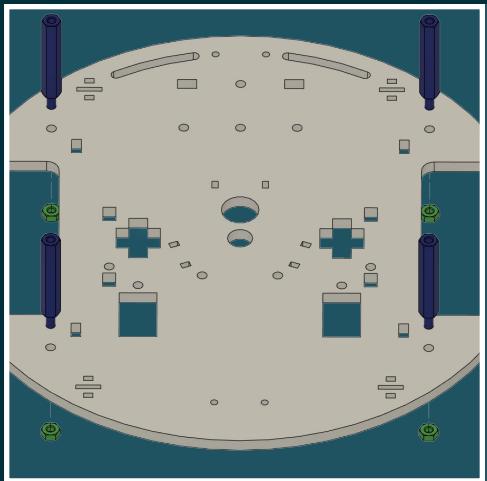
ELEKTRONIK

CHASSIS-
AUFSATZ

ULTRASCHALL
ALTERNATIVMONTAGE

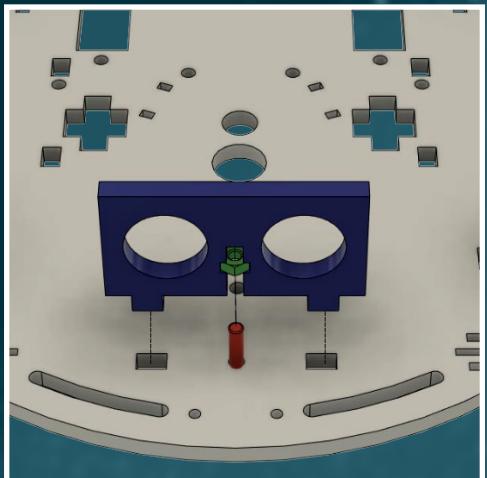
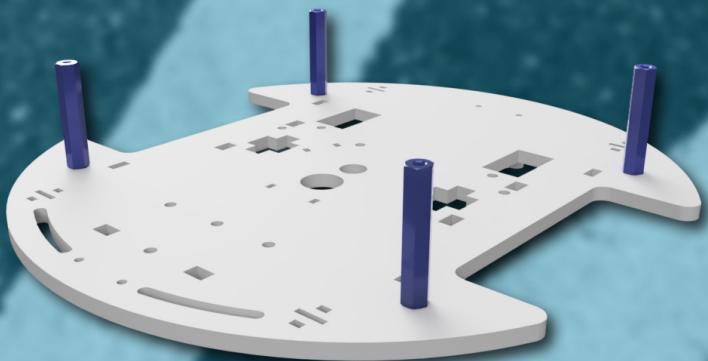
ANSCHLÜSSE

ABSCHLUSS



1. ABSTANDSHALTER ANBRINGEN

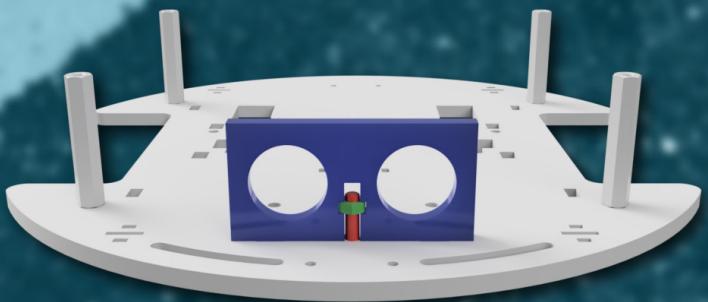
Bringe zunächst die 4 Abstandshalter ($M3 \times 30\text{ mm}$) an der Bodengruppe an und sichere diese von unten mit den passenden Muttern ($M3$).

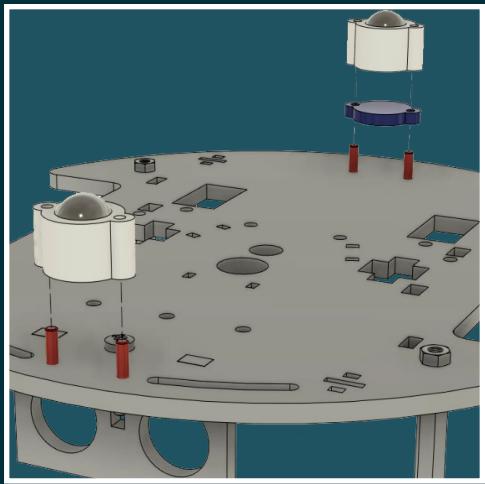


2. ULTRASCHALL-HALTERUNG ANBRINGEN

Bring nun die Ultraschall-Halterung an der Bodengruppe an. Sichere die Halterung von unten mit der passenden Schraube ($M3 \times 14\text{ mm}$) und der entsprechenden Mutter ($M3$).

ACHTUNG! Für fortgeschrittene Nutzer lässt sich der Ultraschall-Sensor auch beweglich, zusammen mit einem Servomotor, auf dem Chassis-Aufsatz montieren (siehe Ultraschall Alternativ-Montage). Für Einsteiger ist die folgende Montage jedoch am besten geeignet.

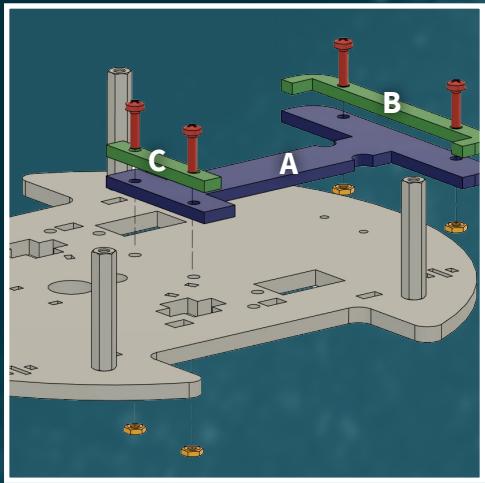




3. GLEITKUGELN MONTIEREN

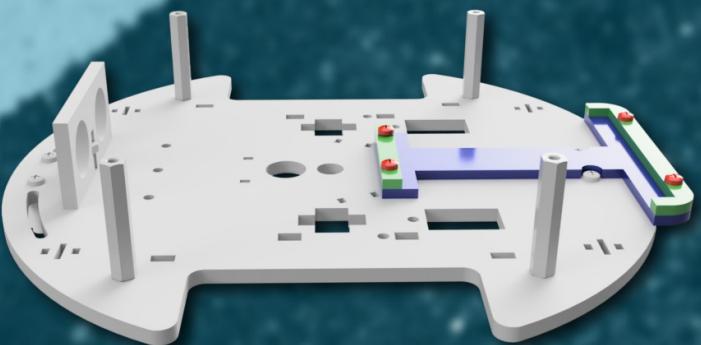
Bringe nun die Gleitkugeln an der Bodengruppe an. Verwende für die Vorderseite die Schrauben aus der Gleitkugel. Verwende für die Rückseite die Schrauben M2,5 x 10 mm.

ACHTUNG! Verwende zur Montage der Gleitkugel an der Hinterachse zusätzlich den Abstandshalter.



4. BATTERIEHALTERUNG ANBRINGEN

Nimm die Batteriehalterung A und lege diese auf die Oberseite der Bodengruppe. Das schmale Ende der Batteriehalterung sollte dabei zur Mitte der Bodengruppe zeigen. Platziere die Batteriehalterung B auf der breiten Seite der Batteriehalterung A und befestige diese mit den passenden Schrauben (M2,5 x 10 mm) und Muttern (M2,5). Die Batteriehalterung C wird auf der schmalen Seite der Batteriehalterung A platziert. Auch diese wird nun mit den passenden Schrauben (M3 x 14 mm) und Muttern (M3) fixiert.



MONTAGE DES ANTRIEBSSTRANGS

Im nächsten Schritt setzen wir den Antriebsstrang zusammen und montieren diesen auf dem Joy-Car. Der Antriebsstrang beinhaltet die Motoren und ist somit für den Vortrieb des Joy-Cars zuständig.



BODENGRUPPE

ANTRIEBS-
STRANG

ELEKTRONIK

CHASSIS-
AUFSATZ

ULTRASCHALL
ALTERNATIVMONTAGE

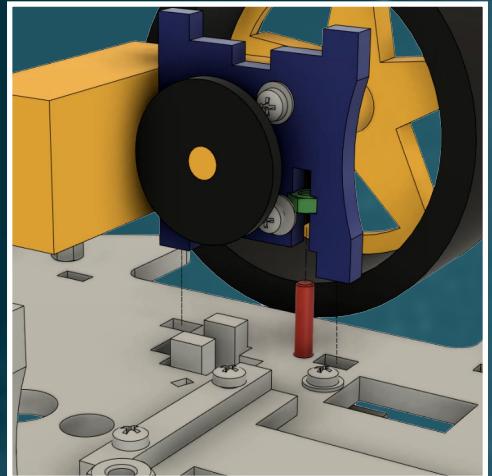
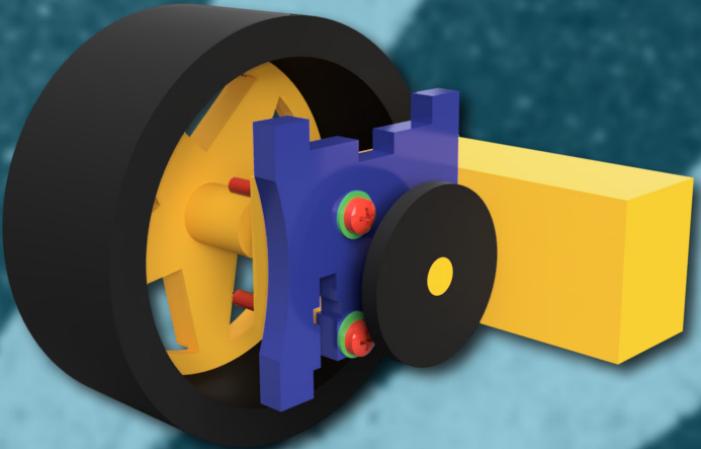
ANSCHLÜSSE

ABSCHLUSS

1. MOTOREN VORMONTIEREN

Setze die beiden Motoren in die Führungen der Motorhalterungen ein und befestige sie mit den passenden Schrauben (M2,5 x 22 mm) und Muttern (M2,5). Verwende zusätzlich pro Schraube jeweils zwei Unterlegscheiben (M2,5), eine am Schraubenkopf und eine am Schraubenende.

Stecke anschließend die schwarze Lochscheibe auf die Innenseite des Motors.

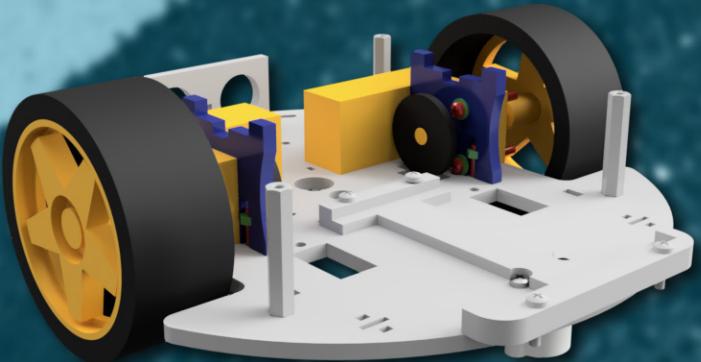
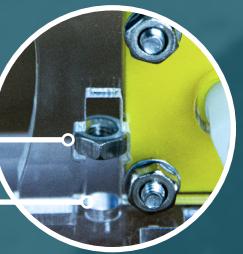


2. HALTERUNG AN CHASSIS ANBRINGEN

Bringe nun die Motorhalterungen an der Bodengruppe an. Sichere sie von unten mit der passenden Schraube (M3 x 14 mm) und Mutter (M3).

MUTTER ZUERST HIER EINSETZEN

SCHRAUBE DANN EINSETZEN UND
FESTZIEHEN



MONTAGE DER ELEKTRONIK

Als nächstes setzen wir alle elektronischen Bauteile in die Bodengruppe ein. Dazu zählen nicht nur die einzelnen Sensoren, sondern auch die LED-Module, welche als Scheinwerfer zum Einsatz kommen.



BODENGRUPPE

ANTRIEBS-
STRANG

ELEKTRONIK

CHASSIS-
AUFSATZ

ULTRASCHALL
ALTERNATIVMONTAGE

ANSCHLÜSSE

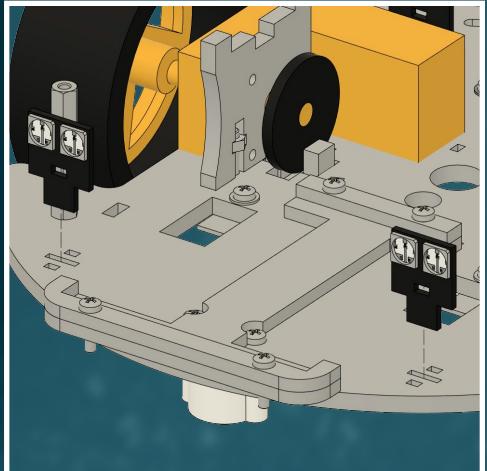
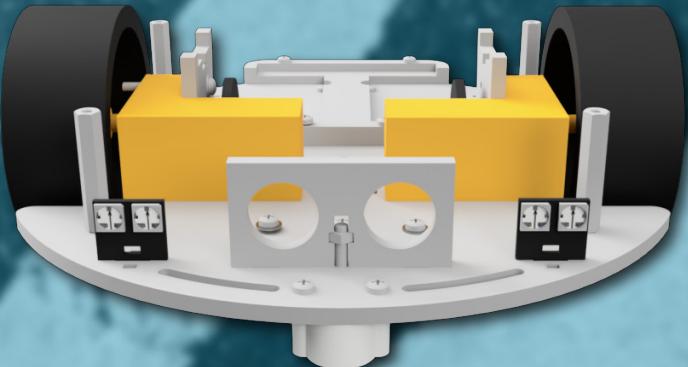
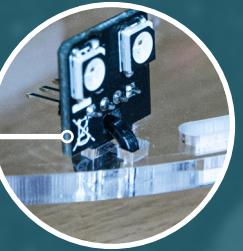
ABSCHLUSS



1. SCHEINWERFER VORNE

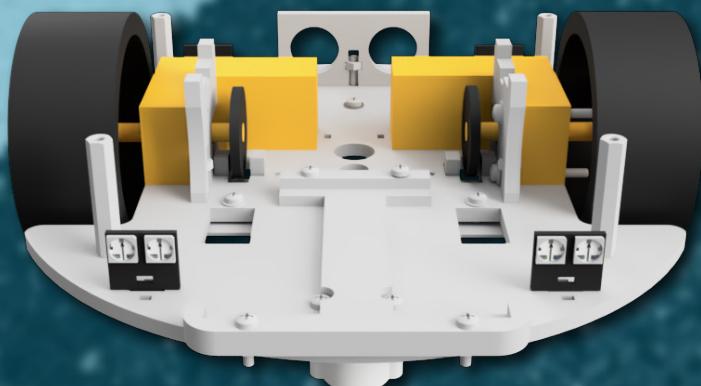
Setze nun die WS2812B LED-Scheinwerfer-Module in die Front der Bodengruppe ein. Befestige diese mit den beiliegenden Kabelbindern. Du kannst einfach jeweils einen Kabelbinder durch die Öffnung in den Scheinwerfern und durch die Öffnungen in der Bodengruppe führen und diese dann festziehen.

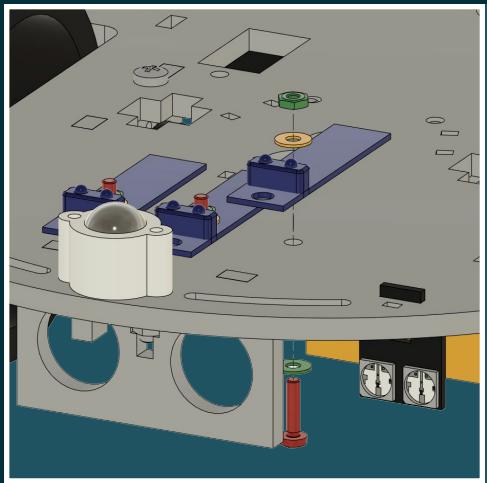
KABELBINDER DURCH ÖFFNUNGEN DER BODENGRUPPE UND DER SCHEINWERFER FESTZIEHEN.



2. SCHEINWERFER HINTEN

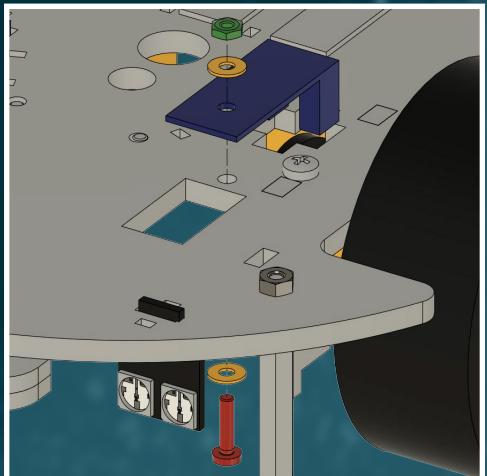
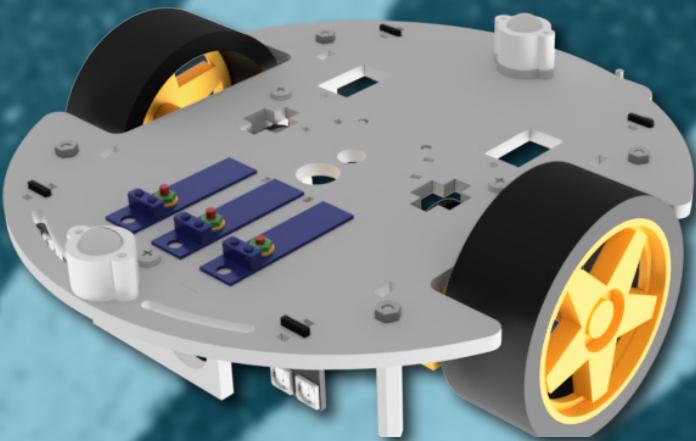
Montiere nun auch die Scheinwerfer-Module am Heck der Bodengruppe mit den beiliegenden Kabelbindern.





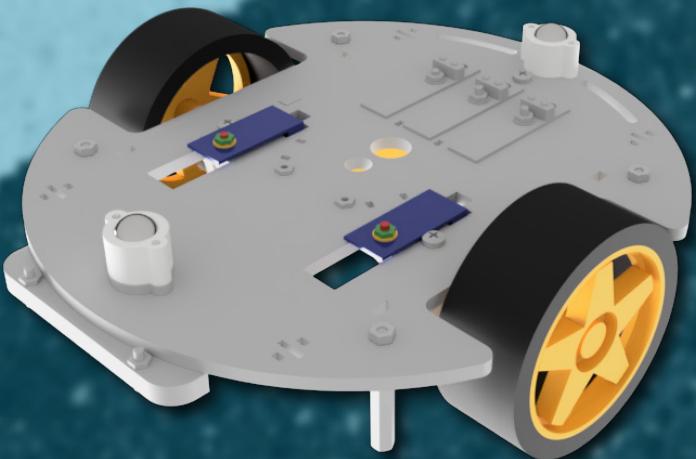
3. LINETRACKING-SENSOREN

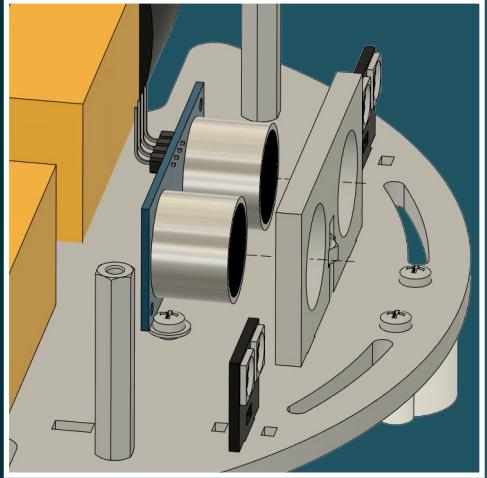
Bring die Linetracking-Sensoren auf der Unterseite der Bodengruppe mit den passenden Schrauben (M2,5 x 10 mm) und Muttern (M2,5) an. Verwende zusätzlich pro Schraube jeweils 2 Unterlegscheiben (M2,5).



4. SPEED-SENSOREN

Setze die Speed-Sensoren auf der Unterseite der Bodengruppe ein. Befestige diese mit den passenden Schrauben (M2,5 x 10 mm) und Muttern (M2,5). Verwende pro Schraube wieder jeweils 2 Unterlegscheiben (M2,5).

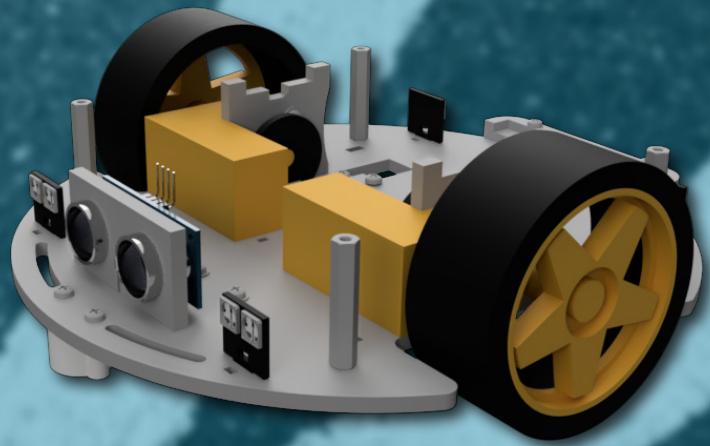




5. ULTRASCHALL-SENSOR

Setze den Ultraschall-Sensor in die Halterung ein. Bei Bedarf kann dieser mit etwas Klebstoff zusätzlich befestigt werden. Die vier Anschlüsse des Sensors sollten dabei nach oben zeigen.

ACHTUNG! Solltest du dich im 2. Arbeitsschritt der Montage der Bodengruppe dazu entschieden haben, den Ultraschall-Sensor auf einem beweglichen Servo zu montieren, so musst du diesen Schritt überspringen.



BODENGRUPPE

ANTRIEBS-
STRANG

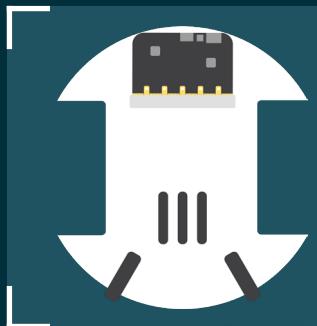
ELEKTRONIK

CHASSIS-
AUFSATZ

ULTRASCHALL
ALTERNATIVMONTAGE

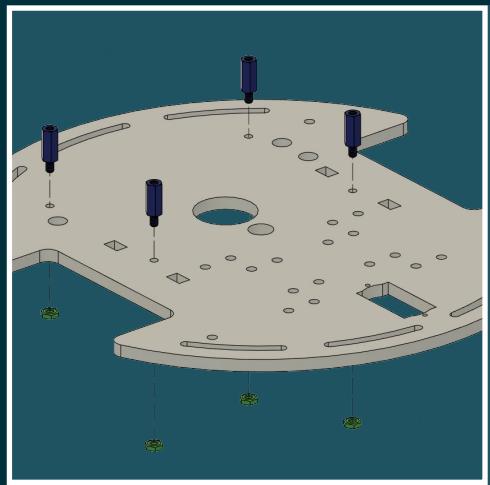
VERKABELUNG

ABSCHLUSS



MONTAGE DES CHASSIS-AUFSATZES

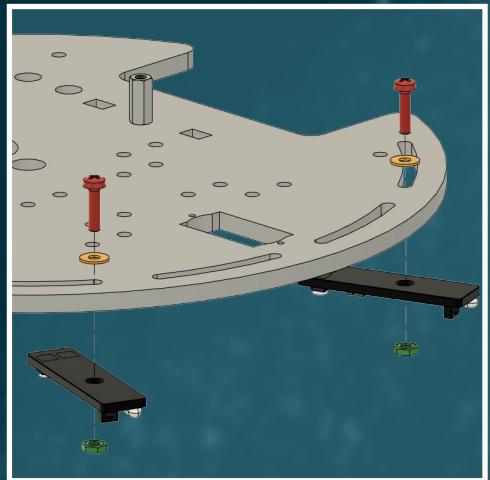
Die Bodengruppe ist soweit fertig zusammengesetzt, daher kümmern wir uns nun um den Chassis-Aufsatz. Dieser beinhaltet, neben der Mainboard-Halterung, auch die Hindernis-Sensoren.



1. ABSTANDHALTER ANBRINGEN

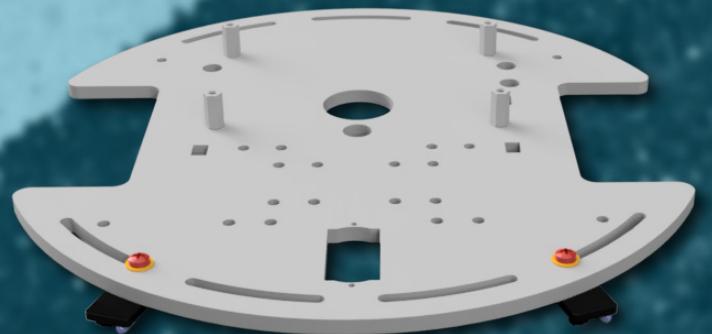
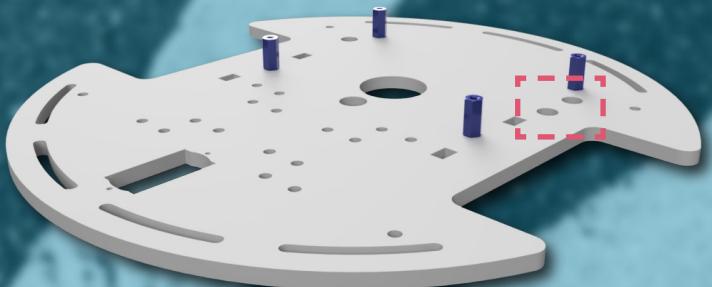
Bringe die 4 Abstandhalter ($M2,5 \times 10$ mm) am Chassis-Aufsatz an und sichere diese von unten mit den passenden Muttern ($M2,5$).

ACHTUNG! Achte darauf, dass sich die beiden markierten Löcher auf der in Fahrtrichtung linken Seite befinden.



2. HINDERNIS-SENSOREN

Montiere nun die Hindernis-Sensoren auf der Unterseite des Chassis-Aufsatzes mit den passenden Schrauben ($M2,5 \times 10$ mm), Unterlegscheiben und Muttern ($M2,5$).



BODENGRUPPE

ANTRIEBS-
STRANG

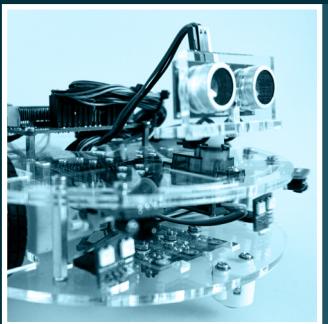
ELEKTRONIK

CHASSIS-
AUFSATZ

ULTRASCHALL
ALTERNATIVMONTAGE

ANSCHLÜSSE

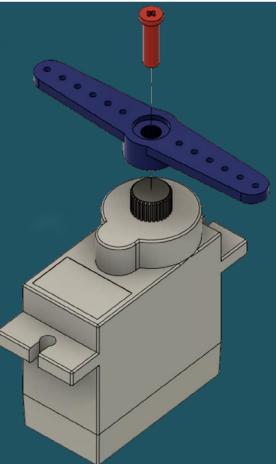
ABSCHLUSS



⚠ ULTRASCHALL-ALTERNATIVMONTAGE ⚠

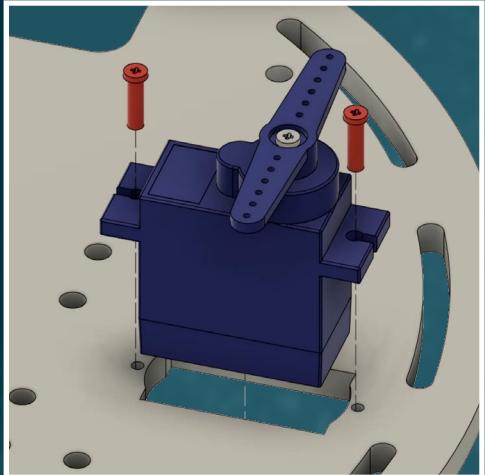
Der Ultraschall-Sensor lässt sich alternativ auch auf dem Chassis-Aufsatz montieren. Hier wird er zusätzlich mit einem Servomotor montiert und bietet damit einen größeren Messbereich.

Solltest du diese Variante bevorzugen und die Montage auf der Bodengruppe übersprungen haben, kannst du mit diesem Kapitel fortfahren. Ansonsten fahre einfach mit dem nächsten Kapitel fort.



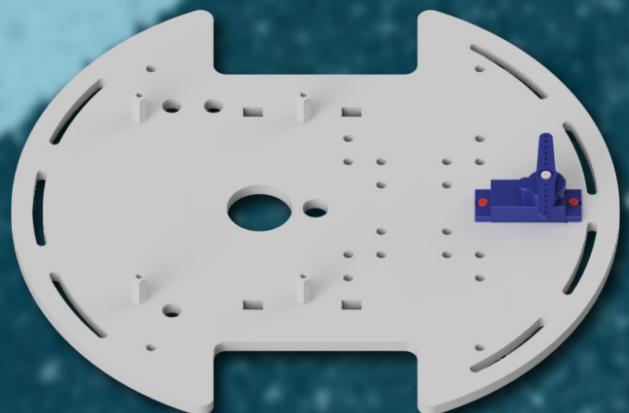
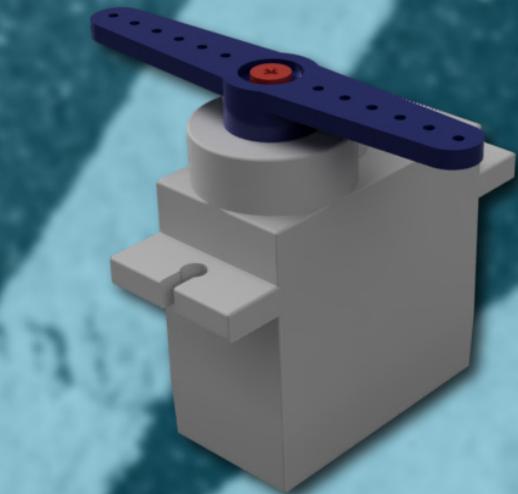
1. SERVOARM ANBRINGEN

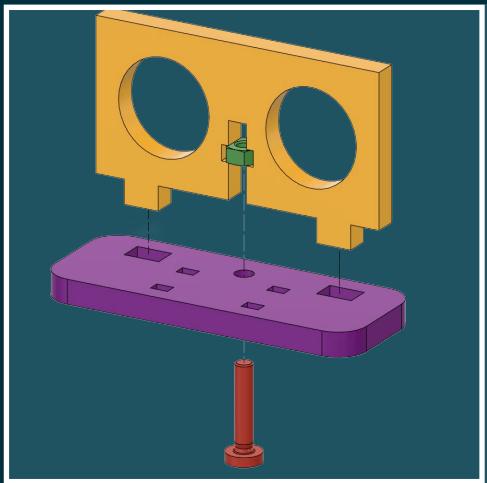
Setze den länglichen Servoarm auf den Getriebekopf des Motors. Fixiere diesen nun mit der dem Motor beiliegenden Schraube.



2. SERVOMOTOR EINSETZEN

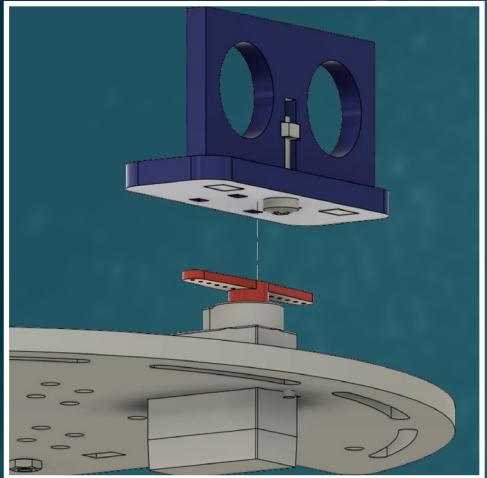
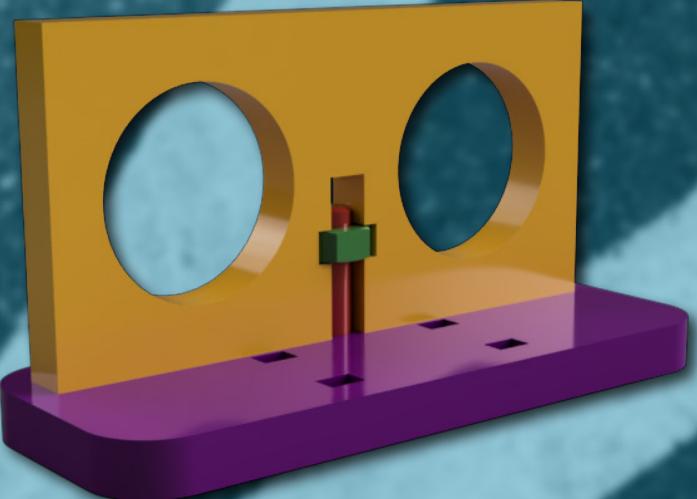
Setze nun den Servomotor mit dem Kabel voran in die Aussparung des Chassis-Aufsatzes ein. Befestige ihn am Chassis mit den dem Motor beiliegenden Schrauben.





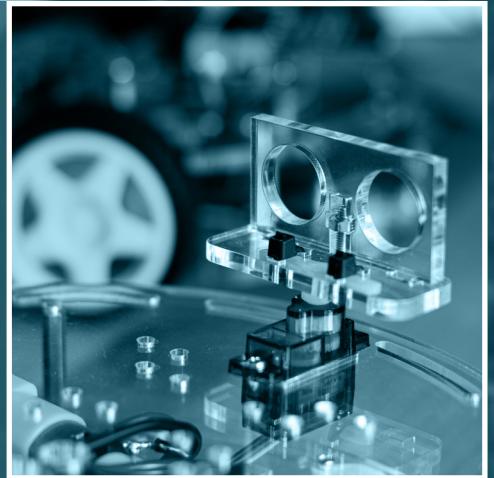
3. ULTRASCHALL-HALTERUNG ZUSAMMENSETZEN

Setze nun die Ultraschall-Halterung auf die Servohalterung und verschraube diese mit der passenden Schraube (M3 x 14 mm) und der entsprechenden Mutter (M3).



4. ULTRASCHALL-HALTERUNG MONTIEREN

Platziere nun die Ultraschall-Halterung auf dem Servoarm und befestige diese mit den beiliegenden Kabelbindern.



BODENGRUPPE

ANTRIEBS-
STRANG

ELEKTRONIK

CHASSIS-
AUFSATZ

ULTRASCHALL
ALTERNATIVMONTAGE

ANSCHLÜSSE

ABSCHLUSS



ANSCHLUSSVERKABELUNG

Nun ist es an der Zeit die Elektronik mit dem Mainboard des Joy-Cars zu verkabeln.

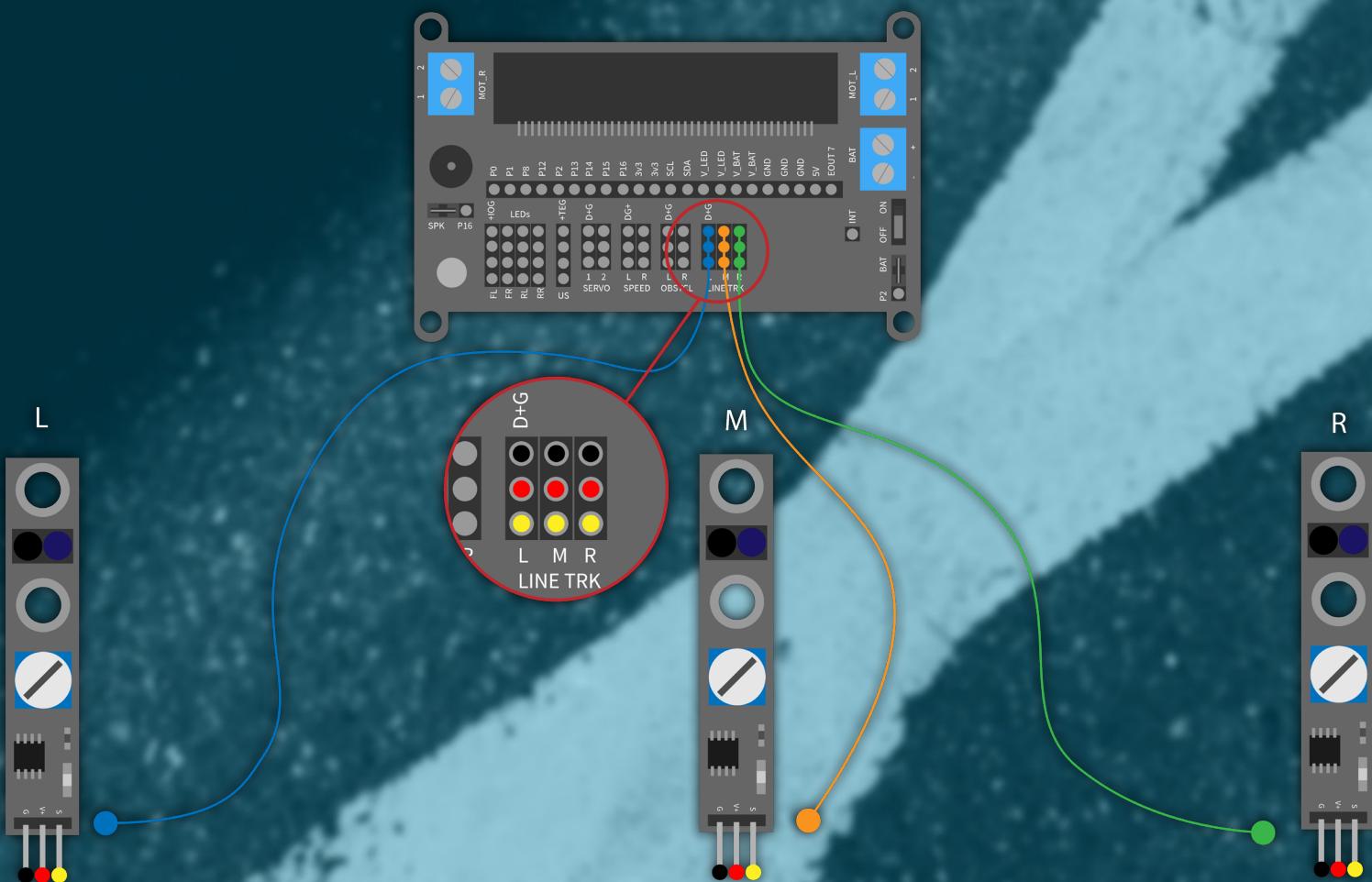
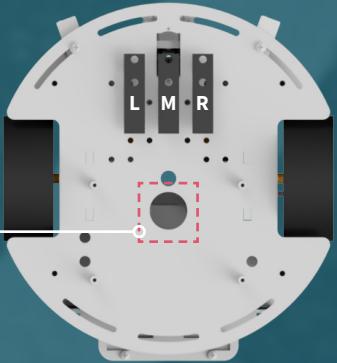
ACHTUNG! Mainboard, Chassis-Aufsatz und Bodengruppe sind noch nicht miteinander verschraubt. Dennoch ist es jetzt am einfachsten die Kabel anzuschließen und zu verlegen und erst danach die drei Einheiten fest miteinander zu verbinden.

1. LINETRACKING-SENSOREN VERKABELN

Die drei Linetracking-Sensoren werden nun mit jeweils einem Drei-Pin-Kabel verbunden. Das andere Ende der Kabel kann dann durch die Öffnungen in der Bodengruppe und des Chassis-Aufsatzes geführt und, wie unten zu sehen, am Board des Joy-Cars angeschlossen werden.

ACHTUNG! Die Empfindlichkeit der Sensoren kann zusätzlich justiert werden. Die hierfür nötigen Details findest du [hier](#).

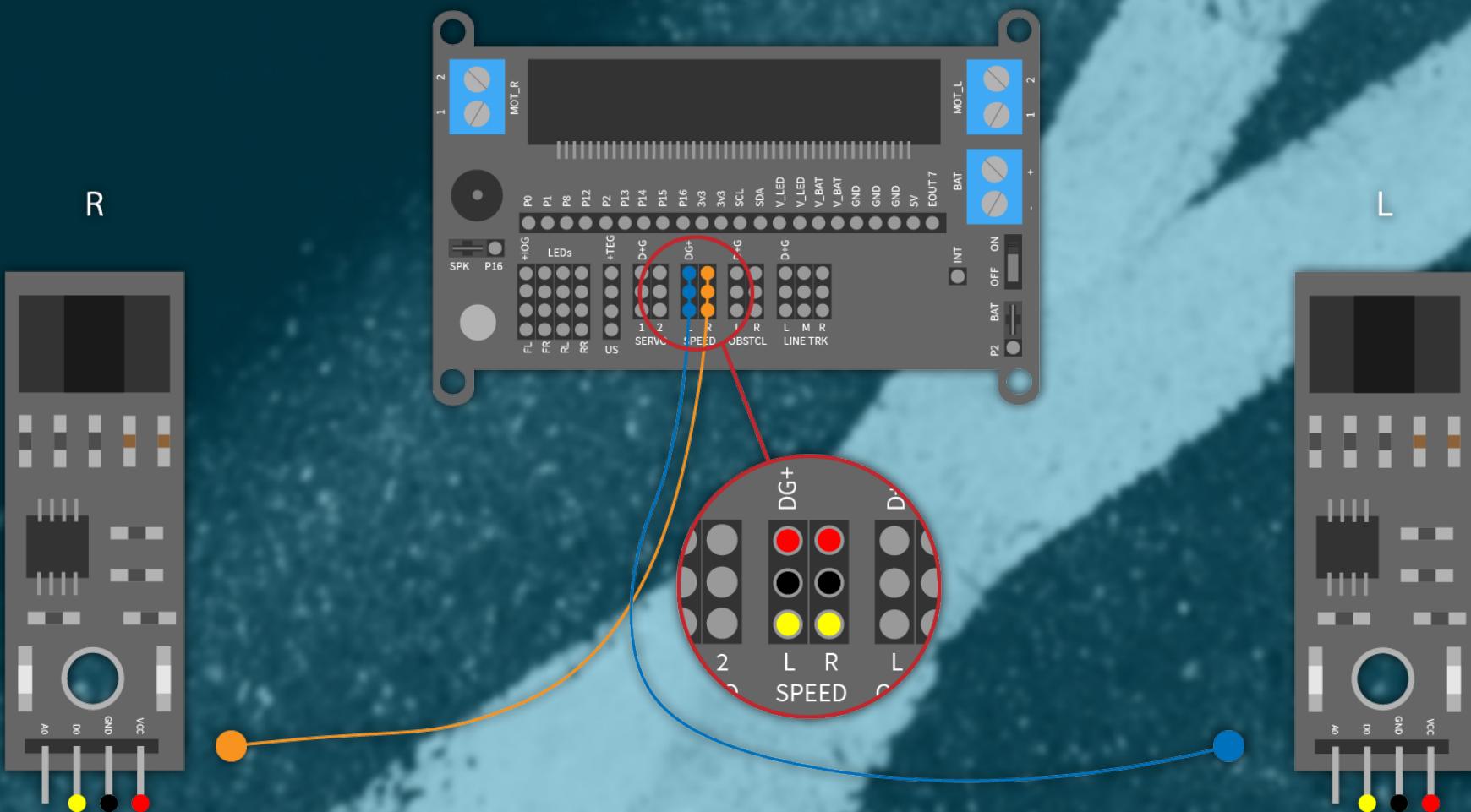
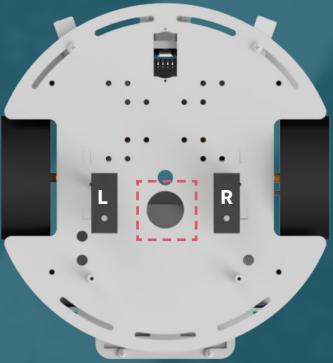
EMPFOHLENE
KABELEDURCHFÜHRUNG



2. GESCHWINDIGKEITS-SENSOREN VERKABELN

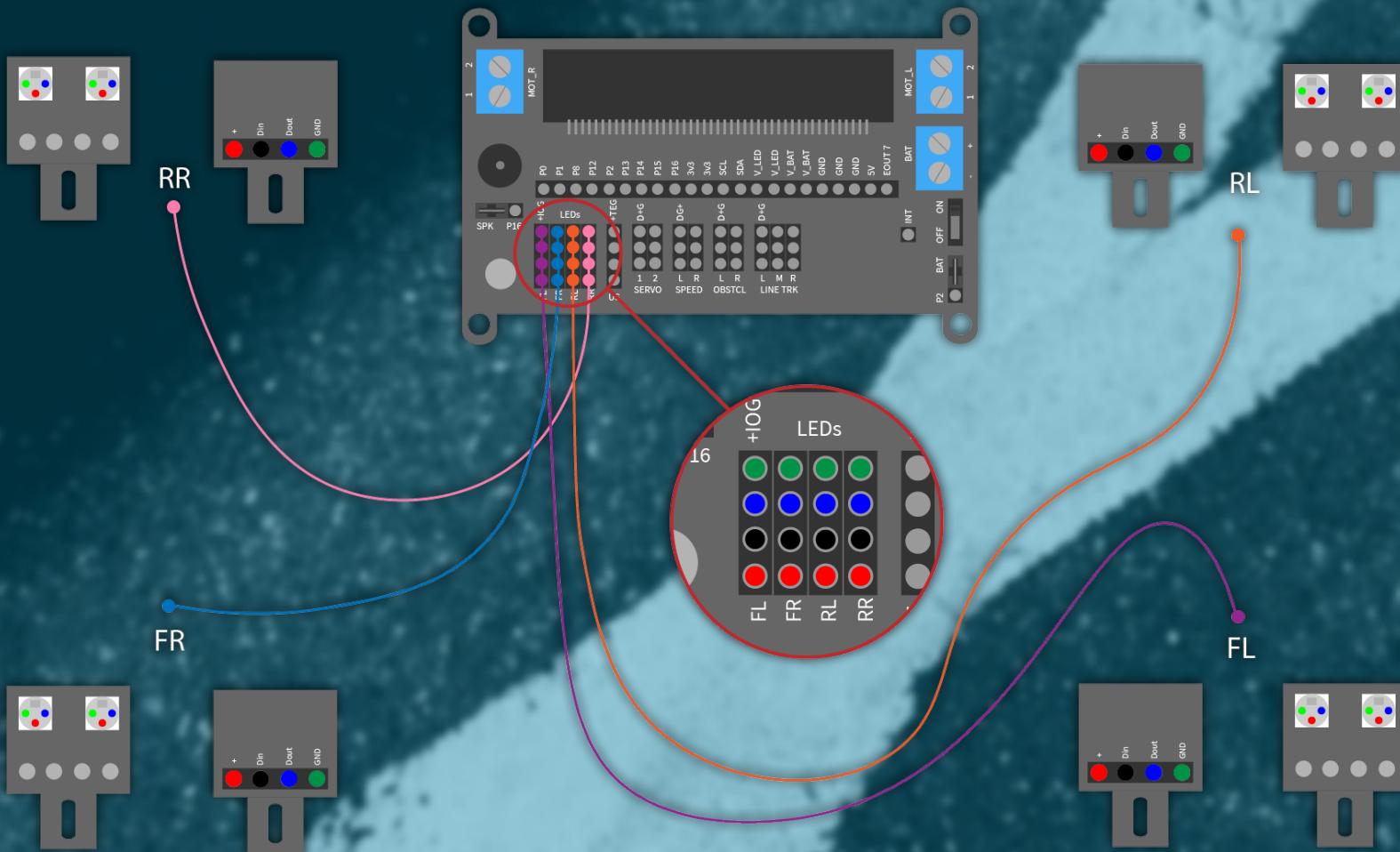
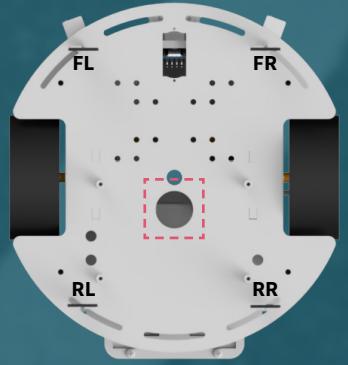
Auch die beiden Geschwindigkeits-Sensoren werden mit jeweils einem Drei-Pin-Kabel verbunden und mit dem Board des Joy-Cars verbunden.

ACHTUNG! Die Geschwindigkeits-Sensoren verfügen zwar über 4 Pins, werden jedoch nur mit einem 3-Pin Kabel ange- schlossen. Der übrige Pin wird nicht verbunden.



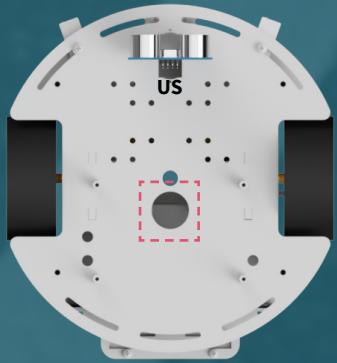
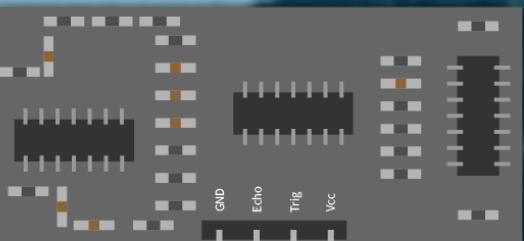
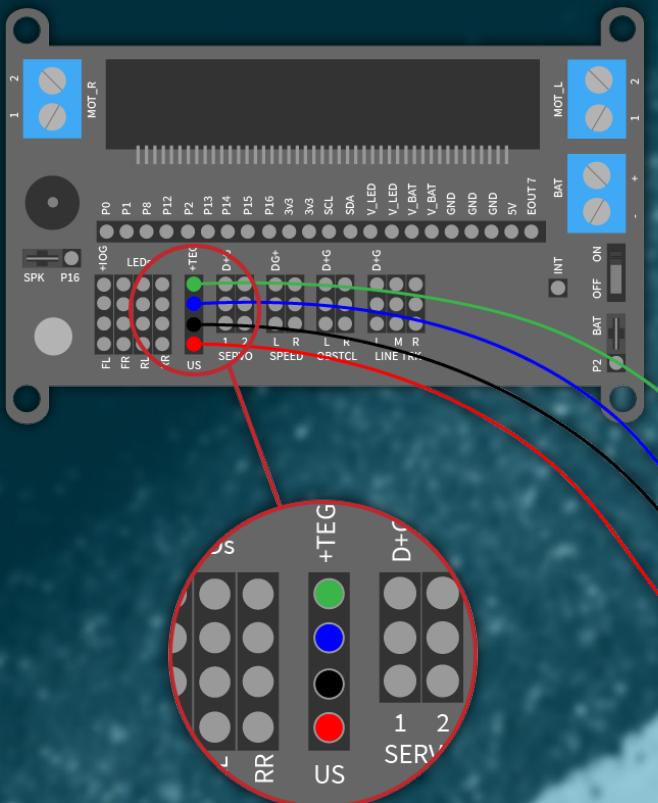
3. SCHEINWERFER VERKABELN

Die vier Scheinwerfer-Module werden mit jeweils einem Vier-Pin-Kabel verbunden und auf das Mainboard des Joy-Cars geführt.



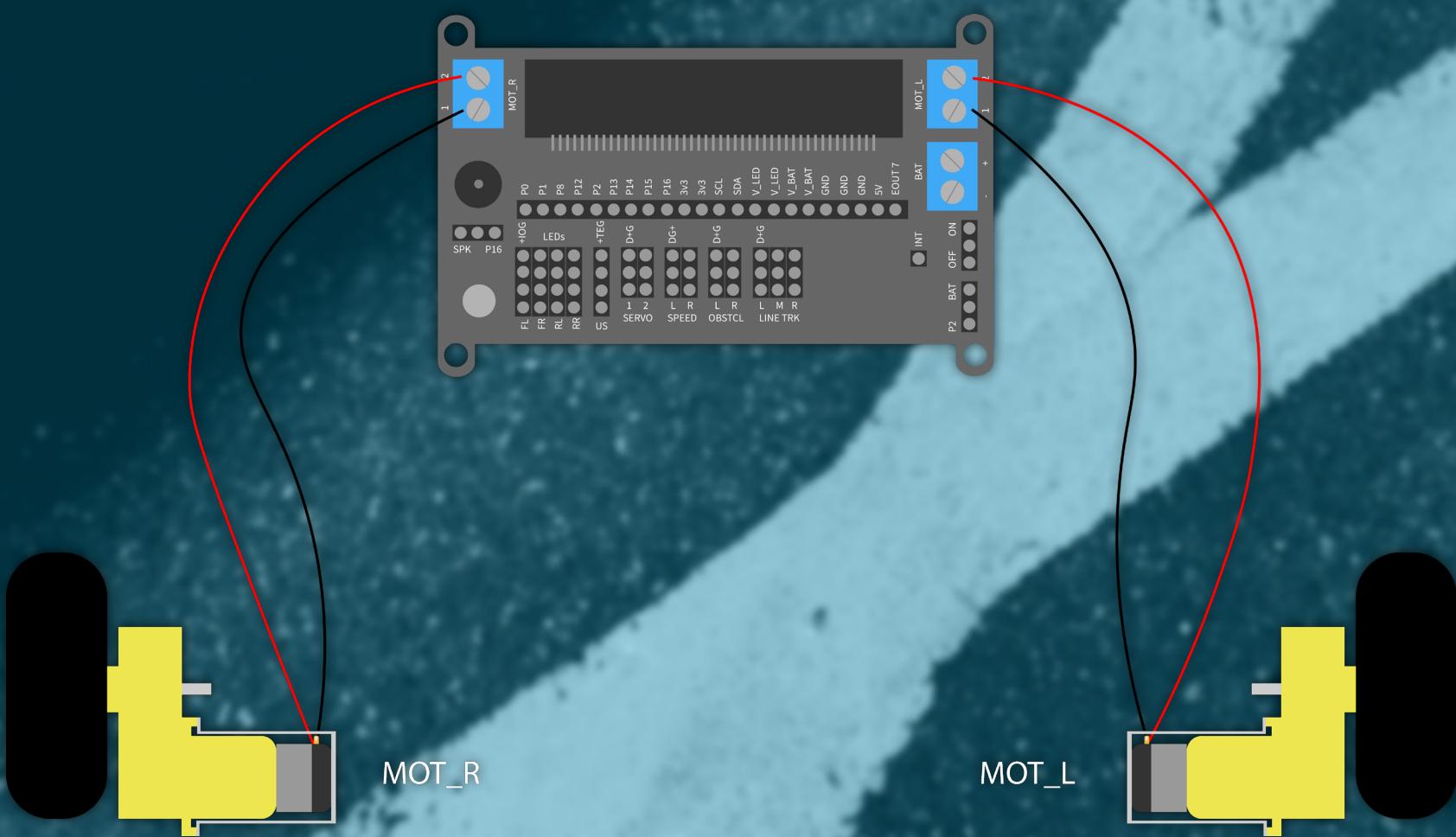
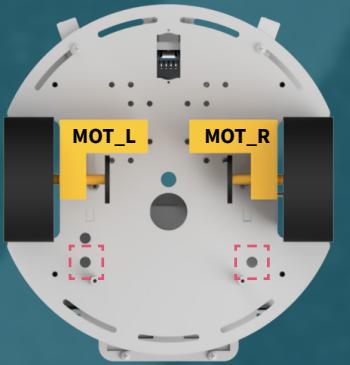
4. ULTRASCHALL-SENSOR VERKABELN

Der Ultraschall-Sensor wird ebenfalls mit einem Vier-Pin-Kabel mit dem Mainboard verbunden.



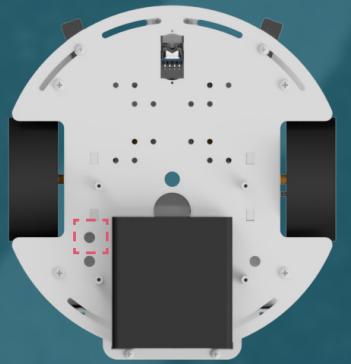
5. MOTOREN VERKABELN

Die beiden Motoren sind bereits mit jeweils zwei Kabeln vorverkabelt. Diese werden zu den seitlichen Verbindern auf dem Board des Joy-Cars geführt. Du benötigst hier einen Schraubendreher, um die Klemmen zu lösen und wieder zu befestigen, nachdem du die Kabel eingesteckt hast.



6. BATTERIEFACH VERKABELN

Das Batteriefach ist, genau wie die Motoren zuvor, ebenfalls mit zwei Kabeln vorverkabelt. Auch diese werden in den entsprechenden Klemmen auf dem Board des Joy-Cars befestigt. Hierbei ist das rote Kabel für die „+“-Klemme und das schwarze Kabel für die „-“-Klemme vorgesehen. Nach dem Anschluss kannst du das Batteriefach einfach in die bereits montierte Batteriehalterung legen. Dort liegt es sicher und kann nicht verrutschen.

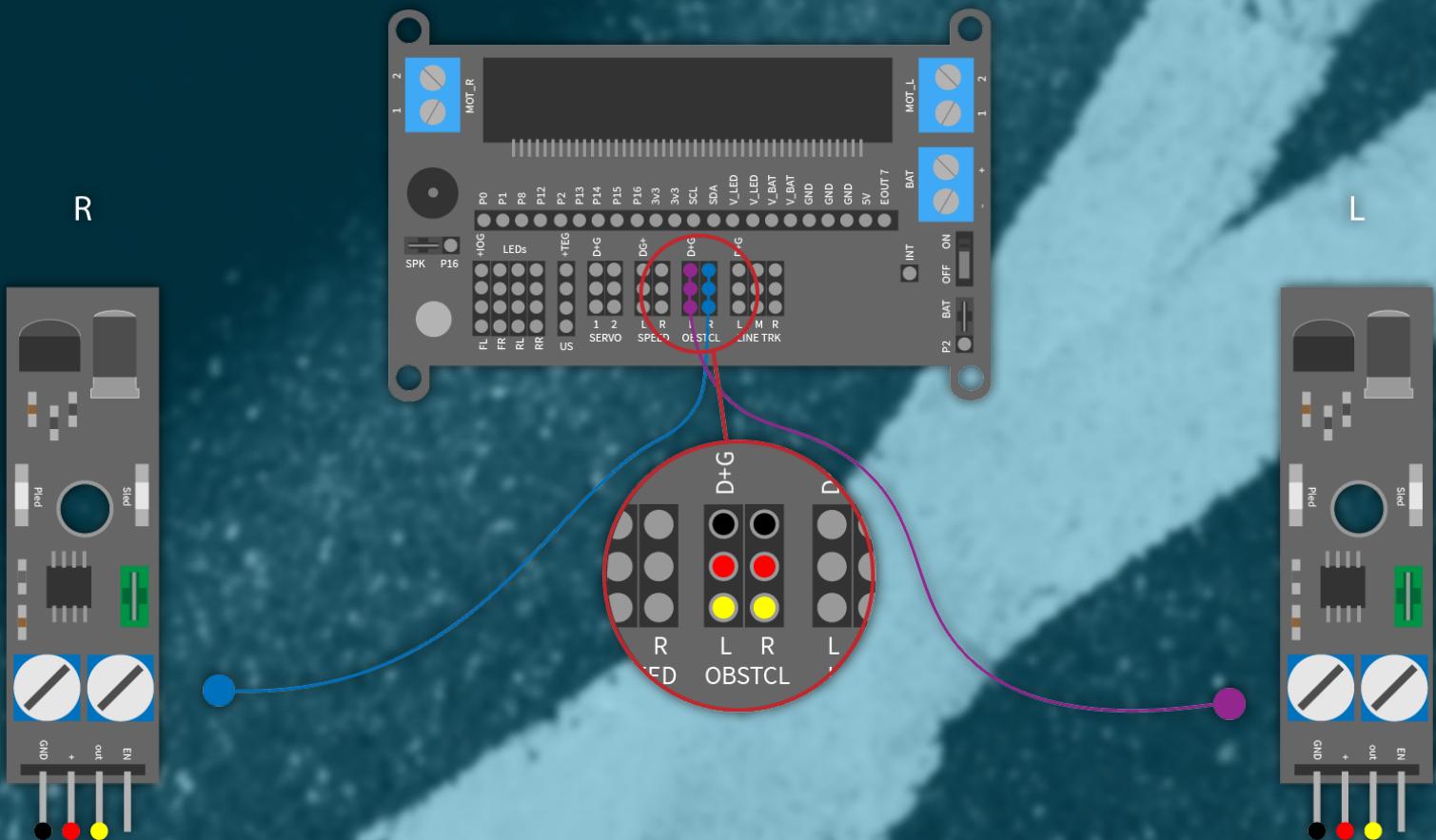
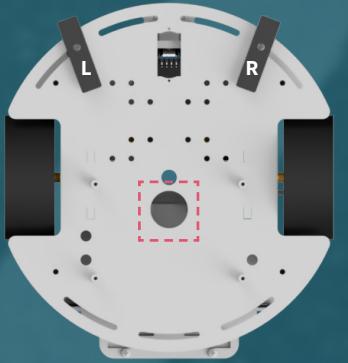


7. HINDERNIS-SENSOREN VERKABELN

Die beiden Hindernis-Sensoren haben zwar vier Pin-Leisten ausgeführt, benötigen jedoch nur drei Kabel. Sie werden daher auch nur mit jeweils einem Drei-Pin-Kabel angeschlossen und zum Board geführt.

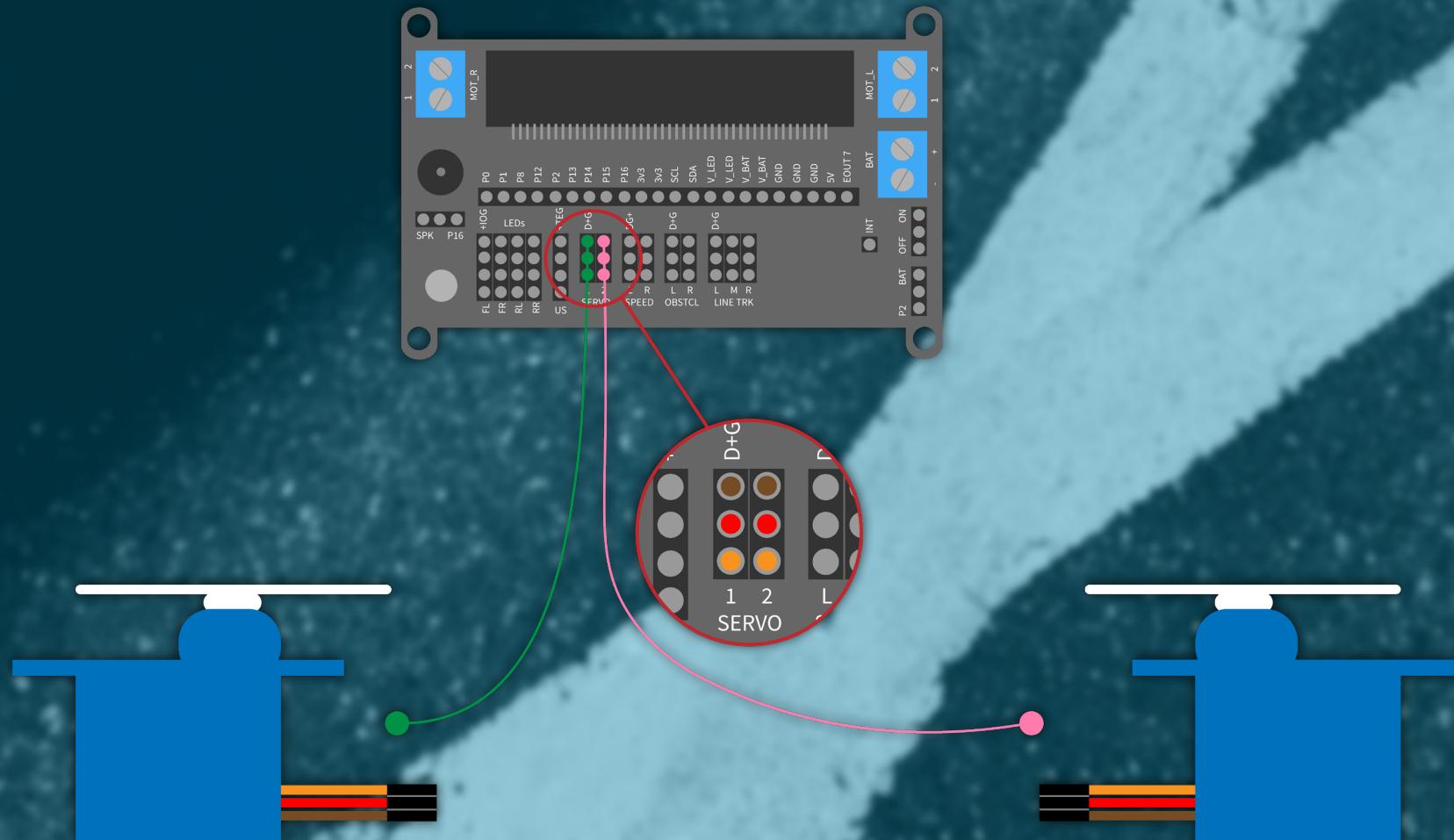
ACHTUNG! Die Hindernis-Sensoren verfügen zwar über 4 Pins, werden jedoch nur mit einem 3-Pin Kabel angeschlossen.
Der übrige Pin wird nicht verbunden.

ACHTUNG! Die Empfindlichkeit der Sensoren kann zusätzlich justiert werden. Die hierfür nötigen Details findest du [hier](#).



8. OPTIONAL: SERVO-MOTOREN VERKABELN

Solltest du den Ultraschall-Sensor in der **Alternativmontage** mit einem Servo-Motor montiert haben, so wird dieser nun mit einem Drei-Pin-Kabel am ersten Servo-Anschluss angeschlossen. Auch ein zweiter Servo-Motor, der für eine individuelle Programmierung genutzt werden kann, kann optional nun mit angeschlossen werden. **Solltest du weder die Alternativmontage verwendet haben, noch einen optionalen Servo-Motor anschließen wollen, dann kannst du diesen Schritt einfach überspringen.**



ABSCHLUSS

Du hast es fast geschafft! Da nun alles montiert und verkabelt ist, setzen wir nun nur noch den Chassis-Aufsatz auf die Bodengruppe, verschrauben das Mainboard und setzen den micro:bit ein.

BODENGRUPPE

ANTRIEBS-
STRANG

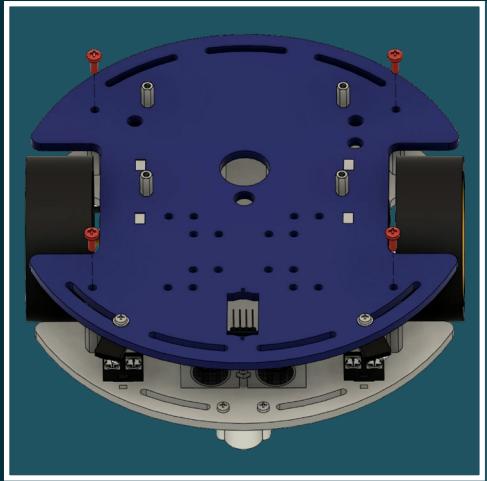
ELEKTRONIK

CHASSIS-
AUFSATZ

ULTRASCHALL
ALTERNATIVMONTAGE

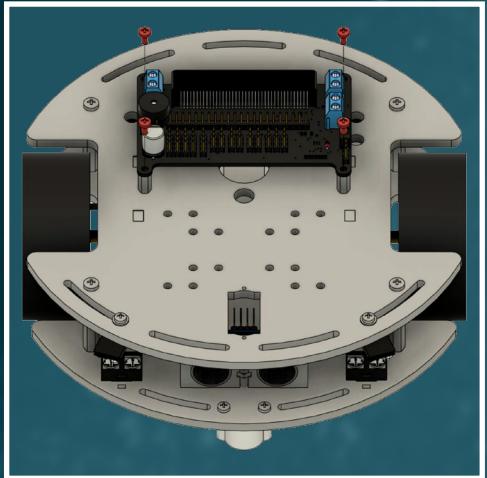
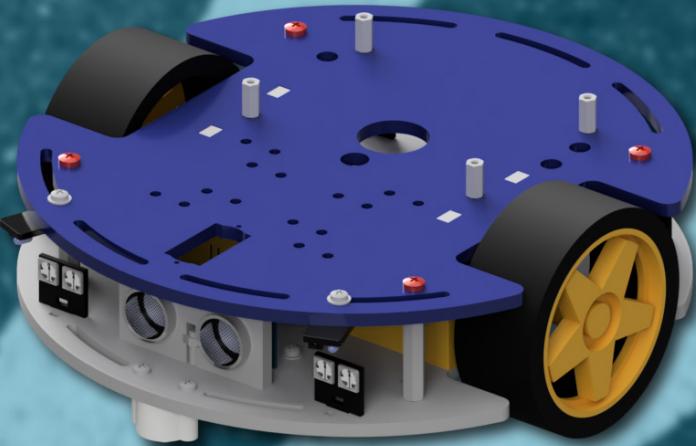
ANSCHLÜSSE

ABSCHLUSS



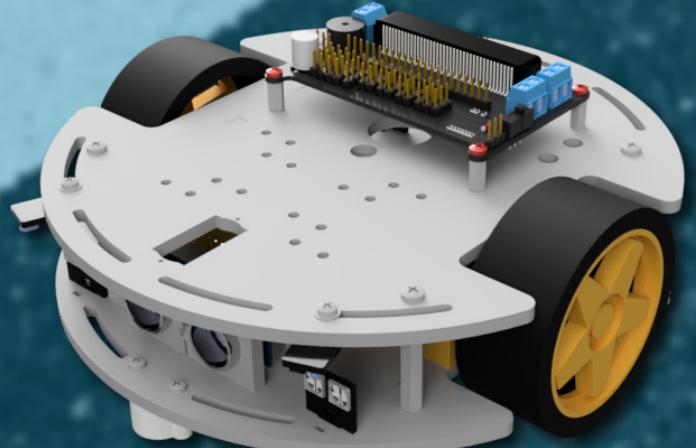
1. HOCHZEIT

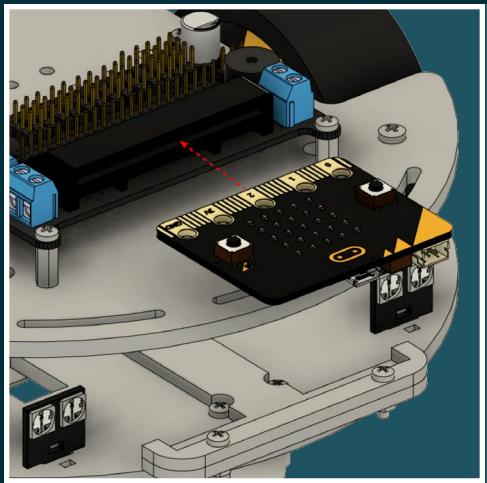
Da nun alle Sensoren montiert und alle Kabel angeschlossen sind, kann der Chassis-Aufsatz mit der Bodengruppe verheiratet werden. Setze dazu den Chassis-Aufsatz auf die Bodengruppe auf und verschraube ihn an den vier Abstandshaltern mit den entsprechenden Schrauben (M3 x 8 mm)



2. MAINBOARD

Setze nun das Mainboard des Joy-Cars auf die bereits angebrachten Abstandshalter des Chassis-Aufsatzes auf und verschraube dies mit den vier Schrauben (M2,5 x 5 mm).

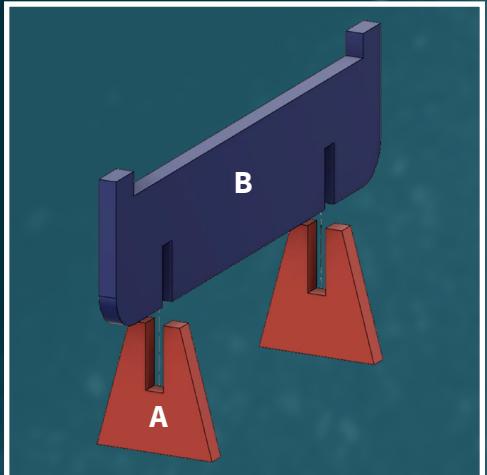
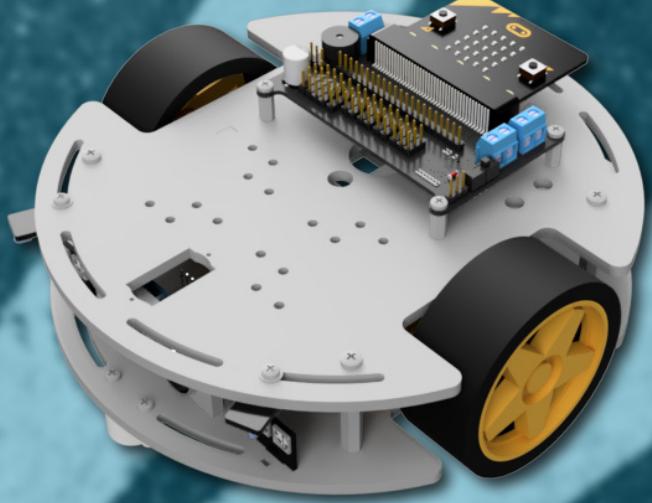




3. MICRO:BIT EINSETZEN

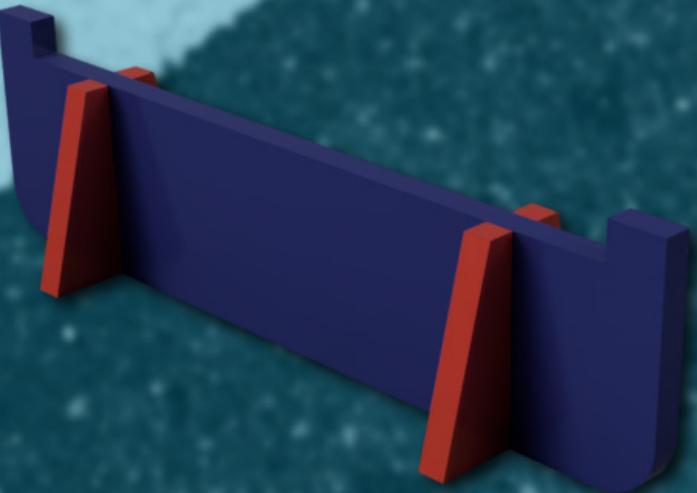
Schiebe nun deinen micro:bit in die Halterung des Mainboards ein. Achte darauf, dass dabei die beiden Knöpfe des micro:bits nach oben zeigen.

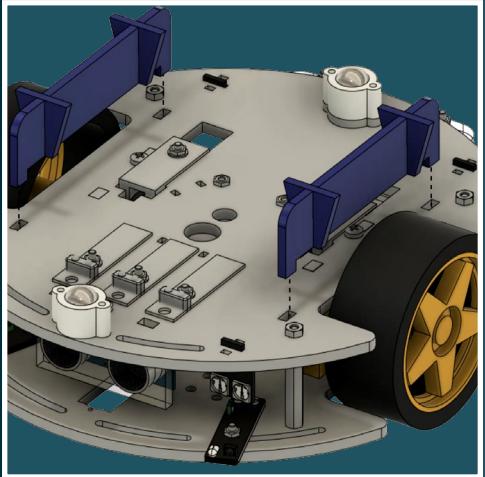
ACHTUNG! Je nach Ausführung ist der micro:bit im Set nicht enthalten und muss zusätzlich erworben werden.



4. PARKHALTERUNG

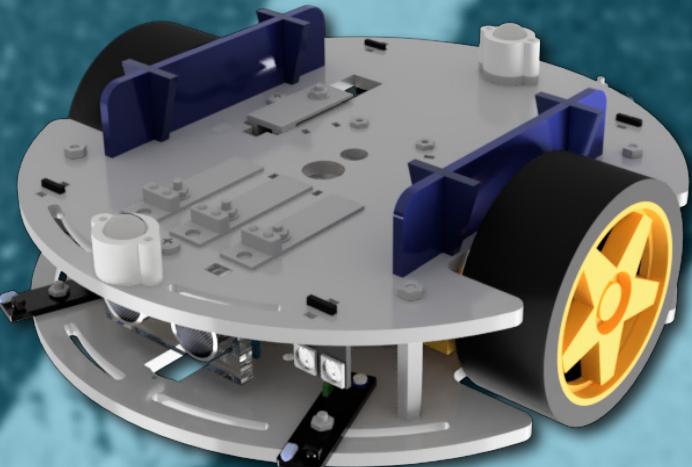
Nimm die beiden großen Parkhalterungen B und setze in beide jeweils zwei der Parkhalterungen A ein. Du kannst das ganze zusätzlich mit etwas Klebstoff stabilisieren.





4. JOY-CAR AUFSETZEN

Du kannst das Joy-Car auf die Parkhalterungen aufsetzen, wenn du z.B. die Motoren programmierst. So fährt das Joy-Car nicht sofort davon, wenn du deinen Code ausprobierst.

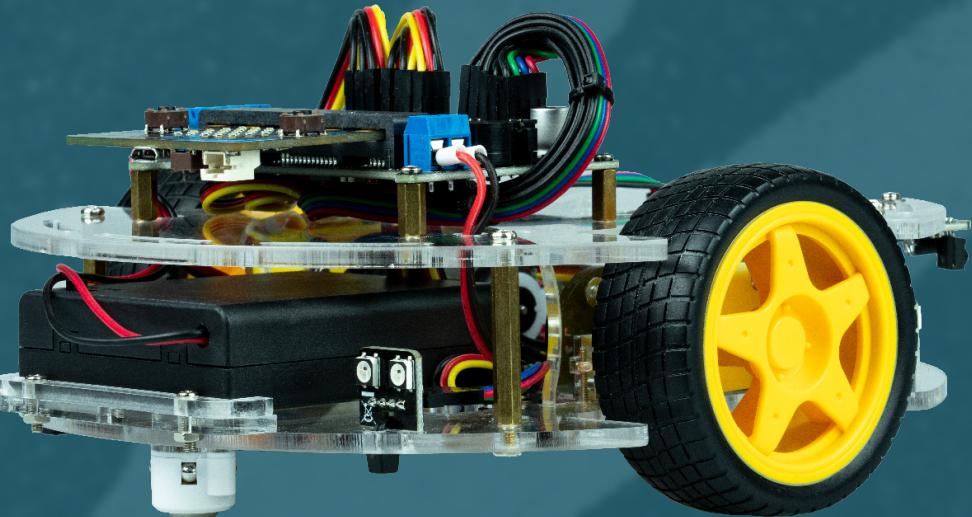
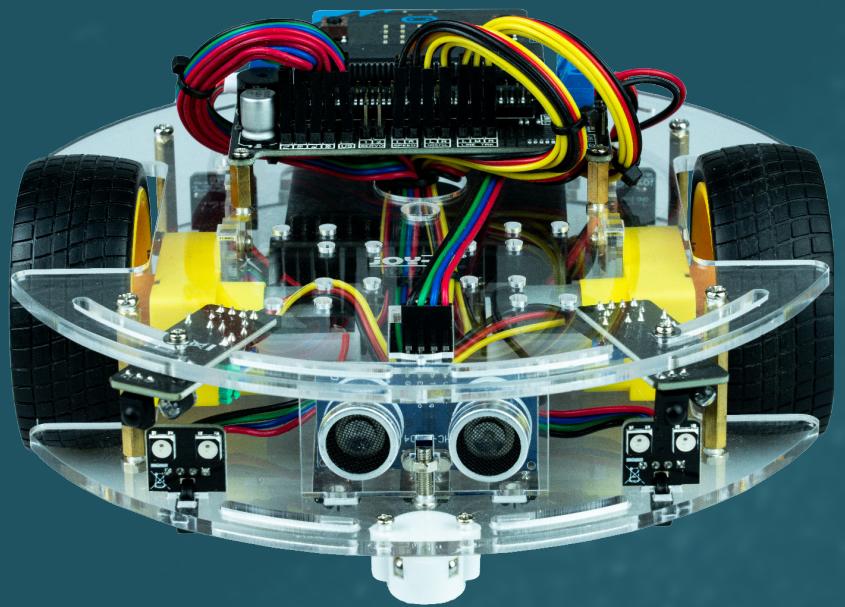


Die Montage deines Joy-Cars ist nun abgeschlossen. Du kannst nun entweder mit dem nächsten Kapitel fortfahren, hier erklären wir dir Schritt für Schritt wie die einzelnen Sensoren funktionieren und wie diese eingesetzt werden können, oder du stürzt dich direkt in die Programmierung. Auch hier haben wir in den nächsten Kapiteln die passenden Einführungen und Erklärungen für dich vorbereitet.

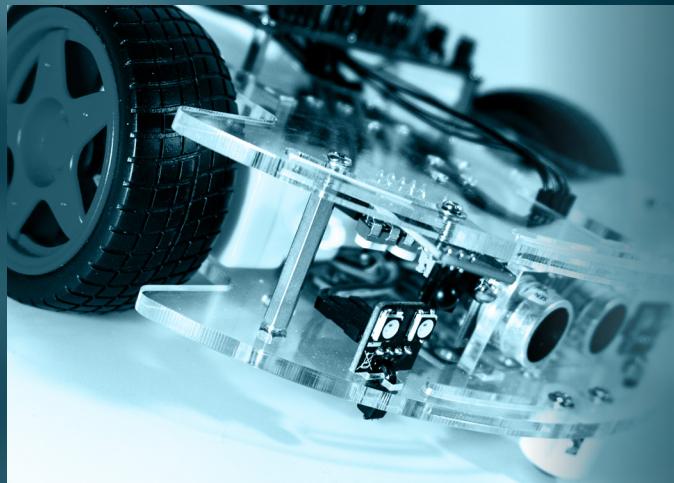
???

Du bist dir noch immer unsicher was die Verkabelung deines Joy-Cars angeht? Du weißt noch immer nicht so recht, wo lang du die Kabel führen sollst und irgendwie sieht alles nicht richtig aus?

Schau dir doch an unserer Beispielverkabelung ab, wie du die Kabel am besten führen kannst und an welchen Stellen du sie mit Kabelbindern befestigen kannst, damit auch die Verkabelung deines Joy-Cars einen guten Eindruck macht.



TRAINING



Dein Joy-Car ist zusammengebaut und frisch poliert? Super! Doch losfahren kannst du noch nicht. In diesem Kapitel gehen wir ins Detail und erklären dir die Module, wie diese funktionieren und wie sie mit deinem Joy-Car kommunizieren.

Für eigene Projekte wird dir dieses Wissen weiterhelfen!

Du kannst nicht warten und möchtest lieber direkt loslegen? Überspringe dieses Kapitel einfach und mach mit dem nächsten weiter, dort geht es dann richtig los!



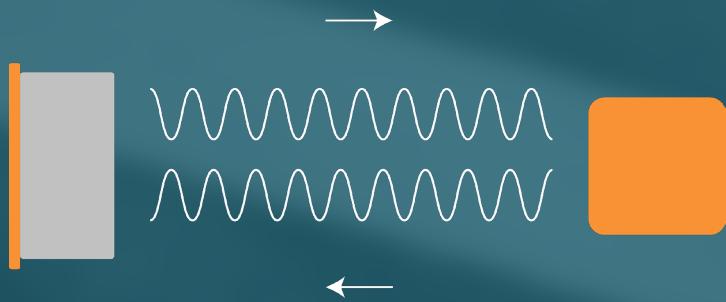
SENSOREN



ULTRASCHALL-SENSOR

Der Ultraschall-Sensor kann beim Joy-Car zur Erkennung von Objekten und Hindernissen in einem Entfernungsbereich zwischen 2 und 300 cm genutzt werden. So kann auch in größerer Entfernung einem Hindernis ausgewichen oder sogar auf dieses zugesteuert werden.

Der Ultraschall-Sensor ist dazu in der Lage, Entferungen präzise zu messen, indem er hochfrequente Schallimpulse aussendet. Trifft dieser Impuls auf ein Objekt so wird der Schallimpuls reflektiert. Das reflektierte Echo wird vom Sensor erkannt. Aus der Zeitspanne, welche sich zwischen dem Aussenden und dem Empfangen des Ultraschallimpulses ergibt, lässt sich dann die Entfernung zum Objekt berechnen.





INFRAROT-SENSOREN

Der Hindernissensor, der Linetracking-Sensor und der Speedsensor arbeiten alle nach dem gleichen Prinzip: Infrarot. Die Sensoren nutzen eine Infrarot-LED und einen Infrarotempfänger um das LED Licht zu erkennen.

HINDERNIS-SENSOR

Der Hindernis-Sensor kann Gegenstände in der Nähe des Joy-Cars erkennen. Dazu wird mit der LED Infrarotlicht nach vorne gestrahlten. Kommt ein Gegenstand / Hindernis in diesen Lichtstrahl wird das Licht reflektiert und kann von dem Infrarotempfänger erkannt werden. Dieser Reichweite dieses Sensors kann mit Hilfe der Potentiometer eingestellt werden. Dieser Sensor kann allerdings nur die, durch die Potentiometer, voreingestellte Distanz erkennen und nicht so wie der Ultraschallsensor aktiv die Entfernung zum nächsten Gegenstand messen.

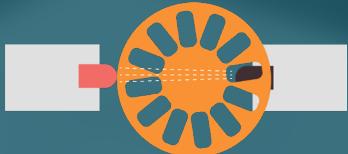
LINETRACKING-SENSOR

Der Linetracking-Sensor strahlt das Infrarotlicht nach unten ab. Befindet sich eine helle Oberfläche unter dem Linetracking-Sensor, wird das Licht reflektiert und von dem Infrarotempfänger erkannt. Wird das Infrarotlicht aber auf eine schwarze und nicht reflektierende Oberfläche gestrahlten, wird keine Infrarotlicht zurückgeworfen, dass der Infrarotempfänger erkennen kann. Um einer Linie zu folgen, werden mindestens 2 Sensoren, aber besser 3 Sensoren, benötigt. Mit Hilfe von 3 Sensoren ist man dann in der Lage festzustellen wo sich die schwarze Linie gerade befindet und in welche Richtung man steuern muss um dieser zu folgen.

SPEED-SENSOR

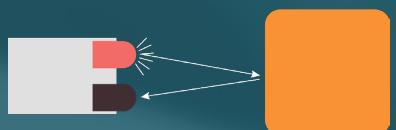
Beim Speed-Sensor sind sich Infrarotempfänger und Infrarot-LED direkt gegenübergestellt. Zwischen Empfänger und LED befindet sich die Lochscheibe, welche auf die Welle des Motors montiert ist. Dreht sich die Motorwelle jetzt, dreht sich auch die Lochscheibe. Diese unterbricht immer wieder das Infrarotlicht der LED. Wenn man die Anzahl der Löcher in der Lochscheibe kennt (in diesem Fall 20), weiß man, dass sich das Rad von Unterbrechung zu Unterbrechung 1/20 gedreht hat. Mit Hilfe dieser Information kann man die gefahrene Strecke ermitteln und nimmt man die vergangene Zeit von Unterbrechung zu Unterbrechung dazu, kann man die Geschwindigkeit bestimmen.

SPEED-SENSOR



IR-LICHT ERREICHEN SENSOR DURCH ÖFFNUNGEN IN DER LOCHSCHEIBE

HINDERNIS-SENSOR



IR-LICHT WIRD VON NAHE GELEGENEM OBJEKT REFLEKTIERT

LINETRACKING-SENSOR



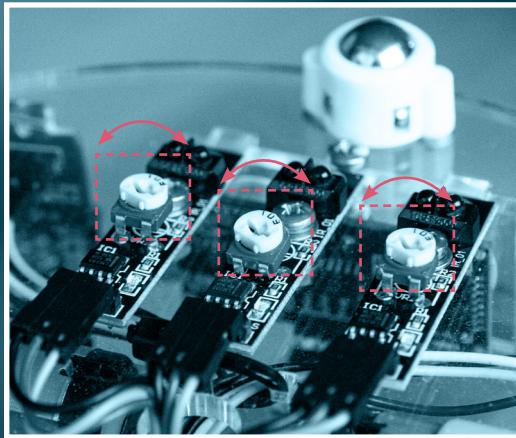
IR-LICHT WIRD VON SCHWARZEM UNTERGRUND NICHT REFLEKTIERT, JOY-CAR KANN GERADE AUS FAHREN UM ZU FOLGEN

IR-LICHT WIRD VOM WEISSEM UNTERGRUND REFLEKTIERT UND VOM SENSOR ERKANNT. RICHTUNG MUSS KORRIGIERT WERDEN UM ZU FOLGEN



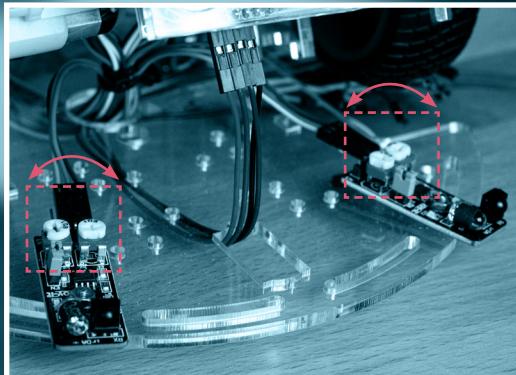
EMPFINDLICHKEIT (LINETRACKER- & HINDERNIS-SENSOREN)

Die Empfindlichkeit der Sensoren kann angepasst werden, falls diese nicht mehr zuverlässig funktionieren. Dabei sind die Linetracking-Sensoren und die Hindernis-Sensoren mit Potentiometern ausgestattet, die mit einem Schraubendreher justiert werden können, um die Empfindlichkeit einzustellen. Auf jedem Sensor ist zusätzlich eine LED angebracht, die nur aufleuchtet, wenn der Sensor gerade etwas erkennt. Sie hilft dir dabei die Funktion deines Sensors zu überprüfen und diesen so gut wie möglich einzustellen.



LINETRACKING-SENSOREN

Nimm dir hierfür am besten ein weißes Blatt Papier und klebe darauf einen Streifen schwarzes Klebeband. Stelle nun dein Joy-Car abwechselnd auf das Blatt Papier und auf den schwarzen Klebestreifen. Deine Sensoren sind richtig eingestellt, wenn die LED auf jedem Sensor aufleuchtet, wenn das Joy-Car auf dem Blatt Papier steht und wenn die LEDs wieder erlischen, sobald du das Joy-Car auf den Klebestreifen stellst.



HINDERNIS-SENSOREN

Löse zunächst den Chassis-Aufsatz von der Bodenplatte, damit du die Potentiometer der Hindernis-Sensoren leichter erreichen kannst. Die Hindernis-Sensoren haben, im Gegensatz zu den Linetracking-Sensoren, jeweils zwei Potentiometer. Hier kannst du sowohl die Stärke der Infrarot-LED, als auch die Empfindlichkeit des Sensors justieren. Halte hier einen Gegenstand vor die Sensoren. Die LED sollte dabei aufleuchten und wieder erlischen, wenn du den Gegenstand entfernst.

TIPP: Du erhöhest die Empfindlichkeit, indem du den Potentiometer im Uhrzeigersinn drehst. Drehst du gegen den Uhrzeigersinn verringst du die Empfindlichkeit.



KOMMUNIKATION

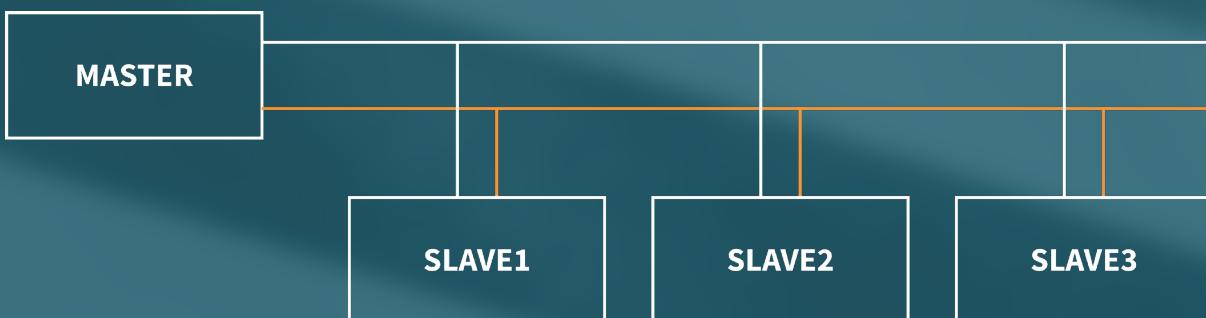


I2C

Der Begriff I2C steht für Inter-Integrated Circuit, stellt einen seriellen Datenbus dar und beschreibt, wie Geräte miteinander kommunizieren und ihre Daten austauschen. Die Daten werden dabei über zwei Leitungen übertragen, der SDA (Serial Data) und der SCL (Serial Clock) Leitung. Über die SDA-Leitung werden die eigentlichen Daten übertragen. Die SCL-Leitung gibt dabei lediglich die Taktfrequenz vor und signalisiert, wann ein Bit auf der Datenleitung anliegt. Beim I2C-Bus kommunizieren alle Geräte über das sogenannte Master/Slave-Prinzip. Hierbei wird die gesamte Kommunikation von einem einzelnen Gerät, dem micro:bit (Master) gesteuert und alle anderen Geräte warten dabei nur auf ihre Sendeerlaubnis und werden daher Slave genannt.

I2C kommt beim Joy-Car für die Kommunikation und Ansteuerung der Infrarot-Sensoren (IO Expander) und der Motor-Steuerungseinheit (PWM-Controller) zum Einsatz.

FÜR EXPERTEN: Für die Ansteuerung der Systeme über I2C werden die Adressen 0x70 (PWM-Controller) und 0x38 (IO-Expander) verwendet.

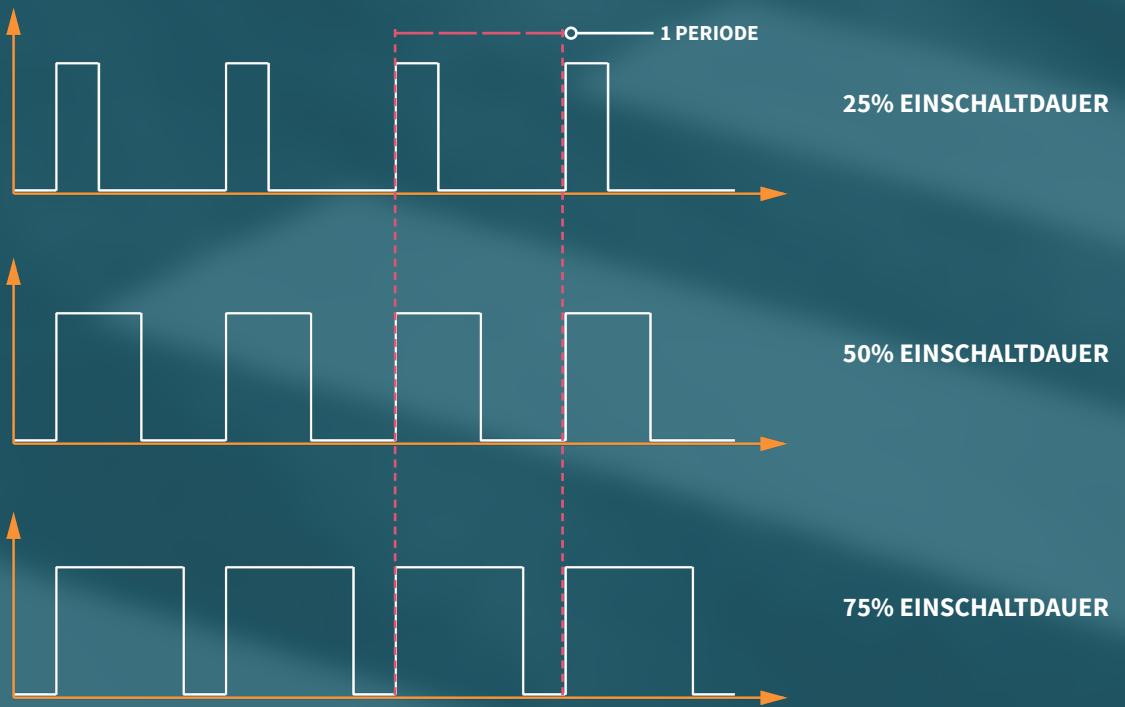




PWM

PWM steht für „Pulse Width Modulation“ bzw. Pulsweitenmodulation. Bei diesem Verfahren wird das Verhältnis von der Einschaltzeit zur festgelegten Periodendauer variiert. Die Pulsweitenmodulation wird genutzt um Verbraucher, wie z. B. Motoren oder LEDs in der Geschwindigkeit oder in der Helligkeit zu steuern.

Die Dauer einer Periode liegt in der Regel bei einigen wenigen Millisekunden oder weniger. Praktisch heißt das also, dass der entsprechende Verbraucher mehrere hundertmal pro Sekunde ein und ausgeschaltet wird. Je länger die Einschaltzeit in einer Periode ist, desto mehr Energie kann zum Verbraucher übertragen werden. Mit anderen Worten: Je länger die Einschaltzeit ist, desto schneller dreht der Motor oder desto heller leuchtet die LED. In den folgenden 3 Beispielen siehst du wie sich PWM-Signale mit 25%, 50% und 75% Einschaltzeit von einander unterscheiden. Bei 0% Einschaltzeit ist der Verbraucher aus. Wohingegen bei 100% Einschaltzeit läuft der Motor mit Vollgas und die LED leuchtet so hell es geht.





MOTOREN

Die Motoren werden über den verbauten PWM-Controller angesteuert. Hier kann sowohl die Geschwindigkeit über das PWM-Signal (0-255) reguliert werden, als auch die Drehrichtung (vorwärts & rückwärts) und der Bremsmodus (starkes & sanftes Bremsen) eingestellt werden.

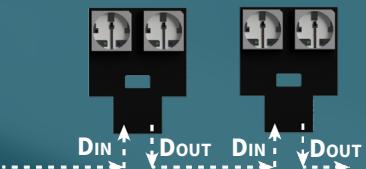
FÜR EXPERTEN: Der PWM-Controller kann per I2C über die Adresse 0x70 angesteuert werden. Hier stehen für die beiden Motoren insgesamt 4 Kanäle (2, 3, 4 & 5) zur Verfügung, die wie folgt verwendet werden können:

MOTOR RECHTS		MOTOR LINKS		FUNKTION
KANAL 2	KANAL 3	KANAL 4	KANAL 5	
0	PWM	0	PWM	Vorwärts
0	0	0	PWM	Links
0	PWM	0	0	Rechts
PWM	0	PWM	0	Rückwärts
255	255	255	255	Sanftes Bremsen
0	0	0	0	Starkes Bremsen



SCHEINWERFER

Für die Scheinwerfer werden adressierbare WS2812B RGB LEDs verwendet. In jeder dieser LEDs ist ein Controller verbaut, der es ermöglicht, für jede LED einzeln die Farbe und die Helligkeit festzulegen. Die Daten werden über eine Busleitung, die an einen Pin vom micro:bit angeschlossen ist, zur ersten LED übertragen. Diese Busleitung wird dann von der ersten LED zur zweiten LED, von der zweiten LED zur dritten LED und so weiter fortgeführt. Die Daten werden dann über diese Busleitung von LED zu LED übertragen. Mit anderen Worten bilden die LEDs eine Art Lichterkette, bei der jedes Licht gesondert gesteuert werden kann.



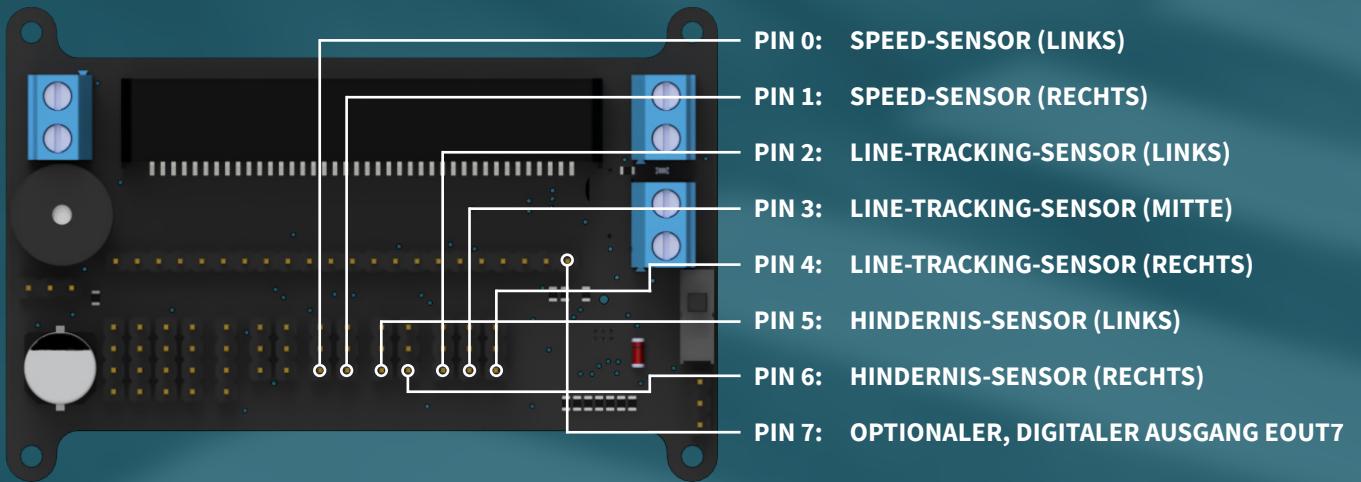
Beim Joy-Car ist diese LED Kette so gelöst worden, dass 2 WS2812B LEDs auf einem Scheinwerfermodul montiert sind. Diese beiden LEDs sind schon auf der Platine mit der Busleitung verbunden. Bei den Anschlusspins der Platine gibt es einen „Din“ (data in) und einen „Dout“ (data out) Pin. Mit Hilfe dieser Pins werden die Platinen miteinander verbunden. Um die Verkabelung übersichtlich zu halten, werden die LED Platinen nicht direkt mit der Busleitung miteinander verbunden. Die Busleitung wird zum Mainboard des Joy-Cars zurück geführt und dort von dem „Dout“ Pin auf den „Din“ Pin des folgenden LED Moduls geroutet.

ACHTUNG! Ist ein LED Modul nicht verbunden, funktionieren die in der Kette weiter hinten liegenden Module nicht mehr, da die Datenverbindung unterbrochen ist.



IO-EXPANDER

Der IO-Expander ist an deinem Joy-Car eine zentrale Einheit, an dem die meisten Sensoren angeschlossen sind. Da der micro:bit nicht über genügend Eingänge für alle Sensoren verfügt werden diese an den IO-Expander angeschlossen. Dieser kommuniziert dann mit dem micro:bit über die I2C Schnittstelle. Auf diese Weise werden lediglich zwei Ports des micro:bits belegt. Es ist sogar ein unbenutzter, digitaler Ausgang am IO-Expander frei, den du für eigene Projekte und Sensoren verwenden kannst. Der IO-Expander ist wie folgt aufgebaut:



FÜR EXPERTEN: Der IO-Expander wird über die I2C-Adresse 0x38 angesprochen. Er prüft die an ihm angeschlossenen Sensoren und gibt die Ergebnisse zusammengefasst als Byte zurück. Dabei steht jedes Bit für einen Pin des Expanders. Bei einer entsprechenden Erkennung wird das Bit des Sensors auf 1 (True) gesetzt.

BEISPIELBYTE: 1 0 1 1 0 1 1 1

○	○	○	○	○	○	○	○
BIT 0 - Speed-Sensor L - True	BIT 1 - Speed-Sensor R - True	BIT 2 - Line-Tracking-Sensor L - True	BIT 3 - Line-Tracking-Sensor M - False	BIT 4 - Line-Tracking-Sensor R - True	BIT 5 - Hindernis-Sensor L - True	BIT 6 - Hindernis-Sensor R - False	BIT 7 - EOUT7 - True



DETAILS



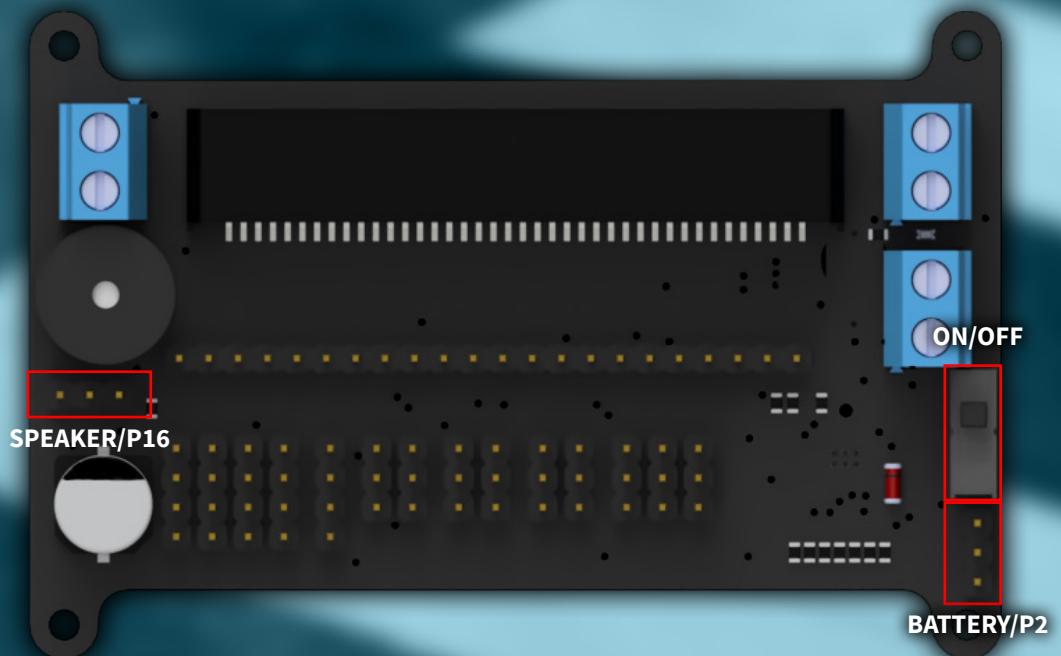
JUMPER

Auf dem Mainboard deines Joy-Cars kannst du über die Jumper weitere Konfigurationen vornehmen. Hier kannst du den **Speaker [SPK]** und die **Batteriespannungsmessung [BAT]** deaktivieren und so die Pins P16 bzw. P2 auf der Stifteleiste des Mainboards aktivieren. So kannst du dir zwei weitere Pins für deine eigene Entwicklung aktivieren, falls du sie benötigst.



ON/OFF

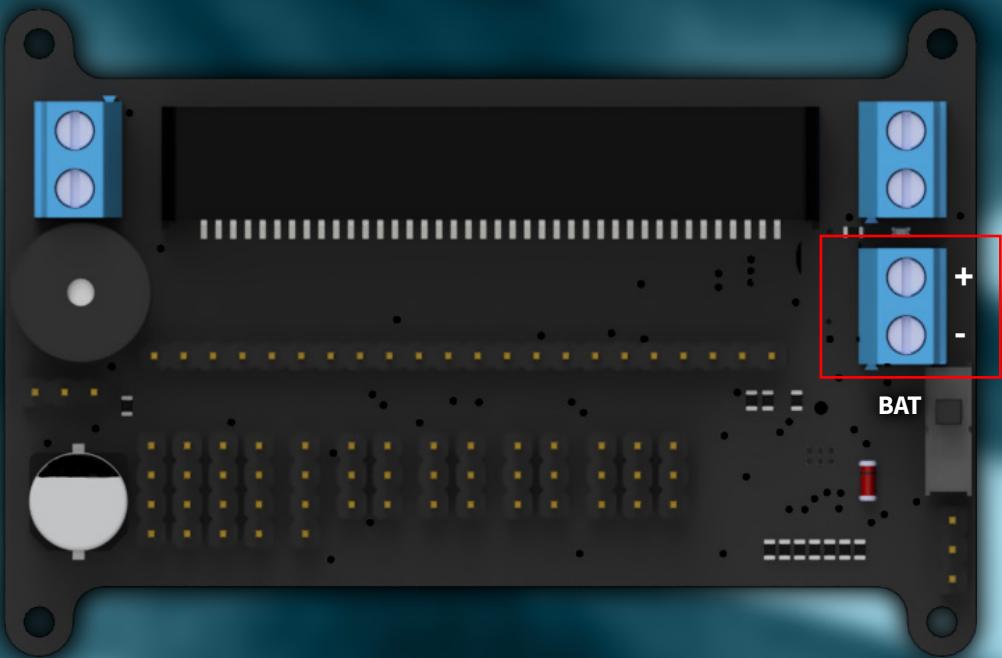
Um dein Joy-Car ein- bzw. auszuschalten musst du nicht immer wieder die Batterien einlegen bzw. herausnehmen. Du kannst die Spannungsversorgung bequem über den **Ein/Aus-Schalter** trennen.





ENERGIEVERSORGUNG

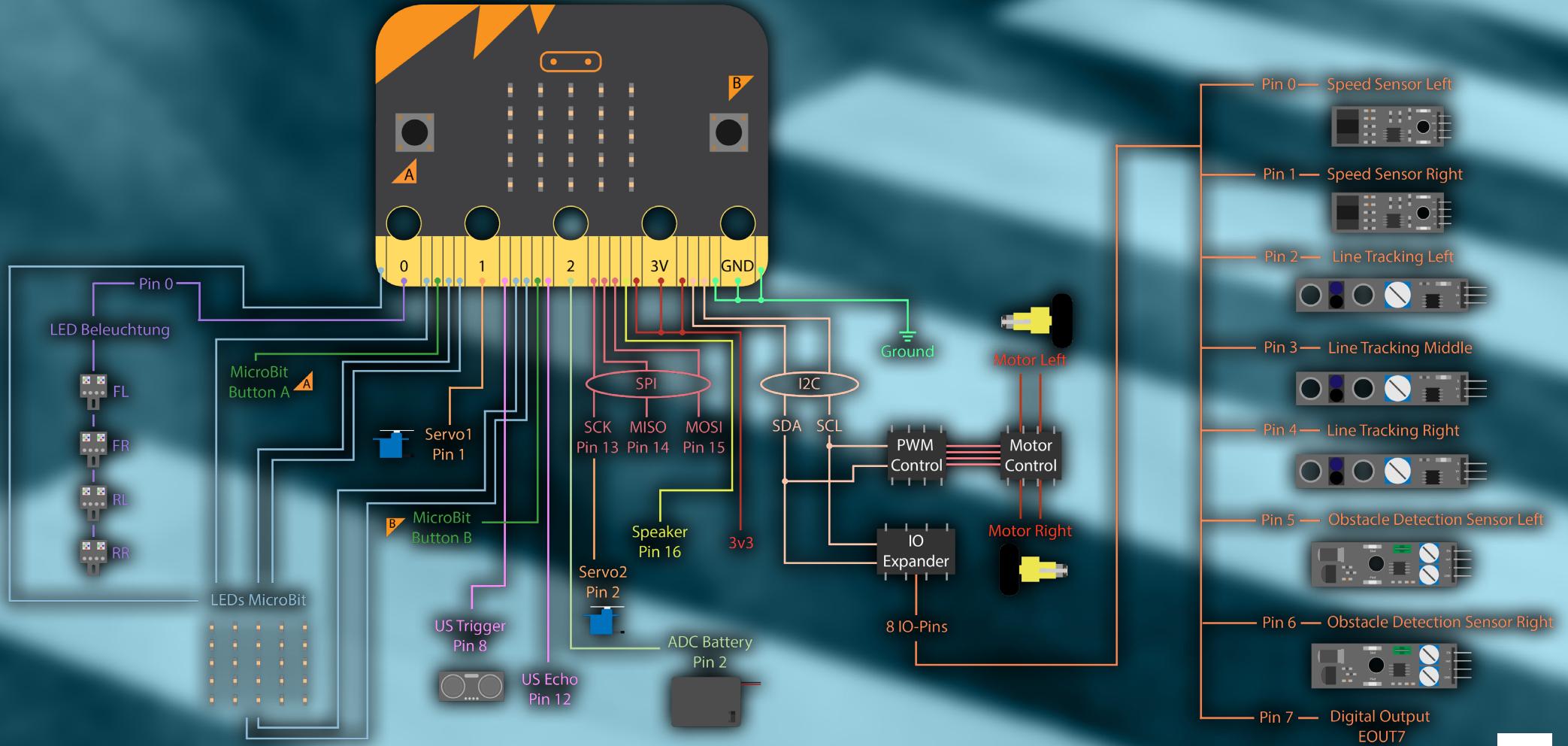
In der Montageanleitung hast du bereits erfahren, dass du die Batteriehalterung an dem **BAT-Terminal** anschließen kannst. Solltest du dennoch einmal eigene Modifikationen vornehmen, bist du nicht an die Batteriehalterung gebunden. Hier ist es gut zu wissen: An dem BAT-Terminal kannst du eine beliebige Spannungsquelle zwischen 4,5-9 V anschließen.





ALLES AM RECHTEN FLECK

Das Mainboard des Joy-Cars bildet natürlich nur die Verbindungseinheit zwischen den einzelnen Sensoren und Modulen und dem micro:bit. Du möchtest wissen, wo und wie die einzelnen Einheiten mit dem micro:bit in Verbindung stehen? Oder vielleicht möchtest du sogar selbst Änderungen vornehmen? Auf unserer schematischen Abbildung haben wir alle Einheiten zusammengefasst und zeigen dir, wie sie vom Micro:Bit angesteuert werden.



QUAL DER WAHL

MAKECODE ODER MICROPYTHON?



MAKECODE? MICROPYTHON? NA WAS DENN NUN...?

Du kannst dein Joy-Car sowohl mit MakeCode als auch mit MicroPython programmieren. Du fragst dich, was davon denn nun richtig ist? Wir helfen dir weiter!

MakeCode und „Mu für MicroPython“ sind beides Entwicklungsumgebungen. Prinzipiell gibt es hier kein richtig oder falsch. Mit beiden Varianten kannst du alle Funktionen deines Joy-Cars verwenden und auch eigene Anwendungen auf dem Joy-Car programmieren. Die beiden Varianten unterscheiden sich eher in ihrer Zielgruppe:

MakeCode richtet sich vor allem an Anfänger. Du hast kaum oder noch keine Programmiererfahrung? Dann bist du bei MakeCode richtig. Hier lassen sich alle Funktionen über grafische Blöcke zusammenschieben, die dann von der Entwicklungsumgebung für den micro:bit übersetzt werden. Gleichzeitig können etwas fortgeschrittenere Nutzer die Umgebung auf JavaScript oder Python umstellen und von dort aus weiterprogrammieren. Grundsätzlich gilt aber: **Du musst bei MakeCode keine einzige Zeile Code schreiben, wenn du dies nicht möchtest.**

MicroPython richtet sich eher an Fortgeschrittene. Du kannst bereits programmieren und weißt, wie du eigene Projekte realisierst? Dann bist du mit MicroPython am besten bedient. MicroPython ist eine Implementierung der Python-Programmiersprache und optimiert für Mikrocontroller. Auch wenn sich die Mu-Entwicklungsumgebung ebenfalls an Anfänger richtet, ist es dennoch empfehlenswert, wenn du hier bereits ein paar Grundkenntnisse besitzt.



UND DAS FÄHRT AUCH!?

Nun kommt der Teil in dem deinem Joy-Car Leben eingehaucht wird, der Programmierung. Wir haben das Joy-Car so entworfen, das einige Funktionen eines echten Autos nachempfunden werden können. Mal abgesehen vom Fahren, kannst du verschiedene Lichtelemente eines Autos nachbilden (z. B. Abblendlicht, Fernlicht, Bremslicht, Blinker, etc.). Des Weiteren hat das Joy-Car mit dem Buzzer eine (Mehrton-) Hupe an Board. Aber auch Sensoren, wie z. B. den Ultraschallsensor, findet man in heutigen Autos. Du möchtest kein Auto nachbauen sondern dir ein bunt blinkendes Spaßmobil programmieren? Keine Sorge, auch das ist möglich. Auf den nächsten Seiten findest du Erklärungen und Beispiele wie du dein Joy-Car programmieren kannst.





EINFÜHRUNG

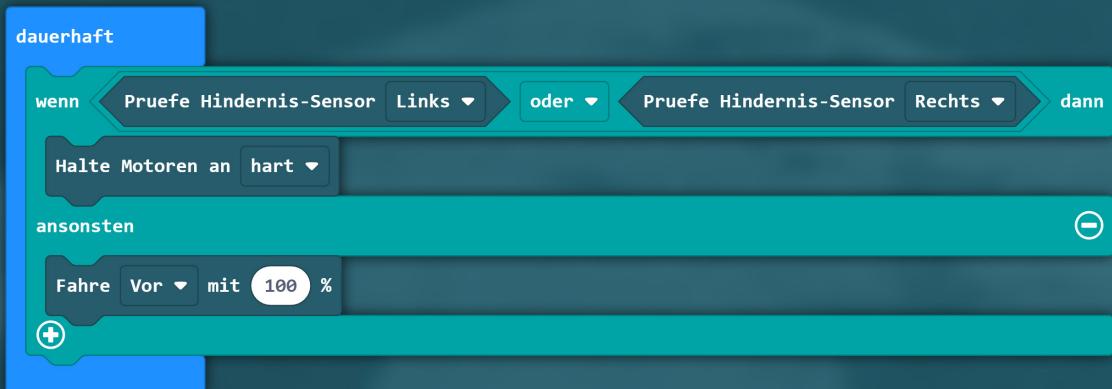
MAKECODE



MAKECODE?

Du hast bisher nur wenig Programmiererfahrung? Dann ist MakeCode der perfekte Einstieg, um dich mit dem Joy-Car vertraut zu machen. Mit MakeCode kannst du dir farbige Blöcke zusammenstellen, die die Funktionalitäten des Joy-Cars abbilden, ohne auch nur eine Zeile Code schreiben zu müssen. Und dennoch bringt dir dieses System das Programmieren näher und bereitet dich auf eine praxisorientierte Programmierung vor.

Mit MakeCode kann es beispielsweise spielend einfach sein, das Joy-Car zum fahren zu bringen und zu stoppen, sobald ein Hindernis erkannt wird:



Du kennst dich bereits aus? Du hast keine Lust auf Erklärungen? Du willst lieber direkt losfahren, anstatt dir dein Joy-Car selbst zu entwickeln? **HIER** erfährst du direkt, wie du mit unserer für dich entwickelten Programmierung sofort starten kannst.

EIN NEUER START

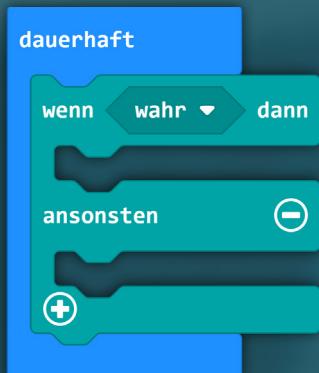
Jedes Projekt besteht zu Beginn aus zwei Grundblöcken. Dem „**beim Start**“-Block und dem „**dauerhaft**“-Block. Alle Anweisungen, die sich innerhalb dieser beiden Blöcke befinden, werden vom Programm ausgeführt. Dabei wird jedoch der „**beim Start**“-Block lediglich einmalig ausgeführt, sobald du dein Programm startest. Der „**dauerhaft**“-Block hingegen wird immer und immer wieder ausgeführt. Sobald alle Befehle und Anweisungen, von oben nach unten, in diesem Block abgearbeitet wurden, beginnt die Ausführung wieder von Anfang an. Hier kannst du beispielsweise dauerhaft die Sensoren überprüfen und auf neue Ereignisse reagieren.



MIT WENN UND ABER...

Prinzipiell wird die Programmiersprache JavaScript durch die Blöcke bildlich abgebildet. Dadurch lassen sich auch Abfragen und Abhängigkeiten, die du vielleicht bereits kennengelernt hast, verwenden. So kannst du beispielsweise prüfen, ob eine bestimmte Bedingung erfüllt ist. Dort kannst du auch zusätzliche Anweisungen unterbringen, die nur ausgeführt werden, wenn die vorherige Bedingung erfüllt wurde: **wenn <Bedingung> dann...**

Optional kannst du diesem Block auch eine Alternative hinzufügen, die ausgeführt wird, falls die vorherige Bedingung **nicht** erfüllt wurde. Dies ist die „**ansonsten**“-Anweisung.





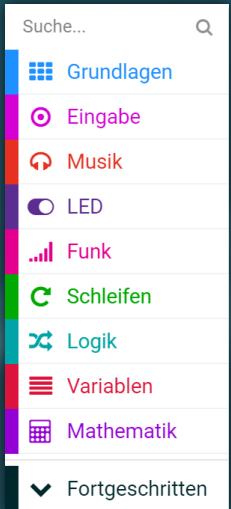
IMMER UND IMMER WIEDER...

Auch Schleifen lassen sich mit den Blöcken problemlos abbilden. Hier kannst du beispielsweise etwas so lange ausführen lassen, bis die Bedingung nicht mehr erfüllt ist („**während**-Block) oder etwas auf Basis einer bestimmten Anzahl wiederholen („**x-mal wiederholen**-Block).



AUF DER SUCHE!

Im linken Bereich deines Projektfensters findest du auch die **Blockübersicht**, mit allen Blöcken, die dir zur Verfügung stehen. Sie sind nach Kategorien geordnet und lassen sich aufklappen. Sie bieten dir einen tollen Überblick über all deine Möglichkeiten. Klick dich doch einmal durch die Kategorien und finde selbst heraus, welche Wege du gehen kannst.



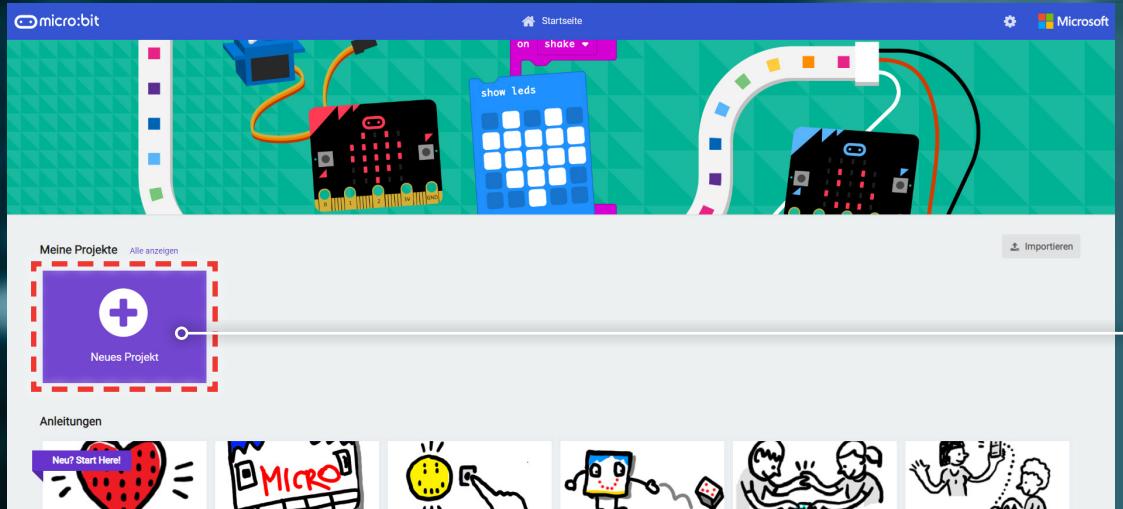


DER ERSTE START



EIN NEUES KAPITEL

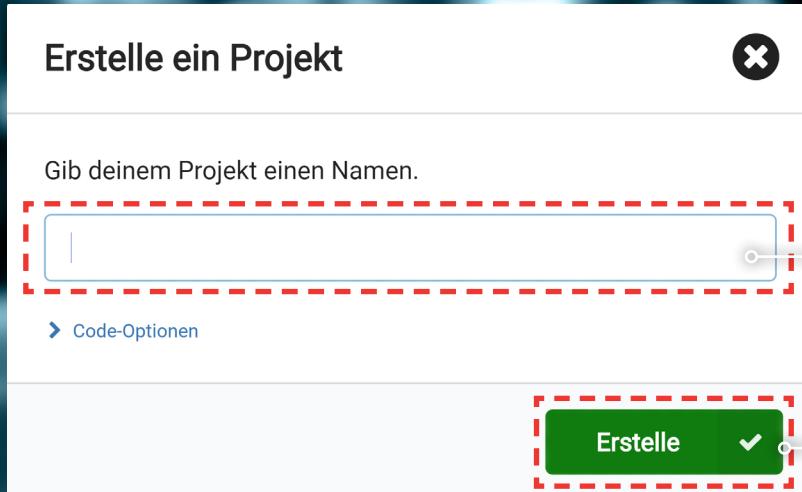
Du hast noch keine Erfahrungen mit der MakeCode-Entwicklungsumgebung gesammelt, möchtest aber gerne mit einem kleinen Beispiel starten? Wir bringen dir die Umgebung näher! Auf [HTTPS://MAKECODE.MICROBIT.ORG/](https://makecode.microbit.org/) gelangst du zur Entwicklungsumgebung.



Hier gehts los, sobald du die Seite betrittst. Im mittleren Bereich der Seite werden dir all deine Projekte aufgelistet. Außerdem kannst du hier ein neues Projekt erstellen.

Probiere es doch einmal aus und erstelle dein erstes Projekt!

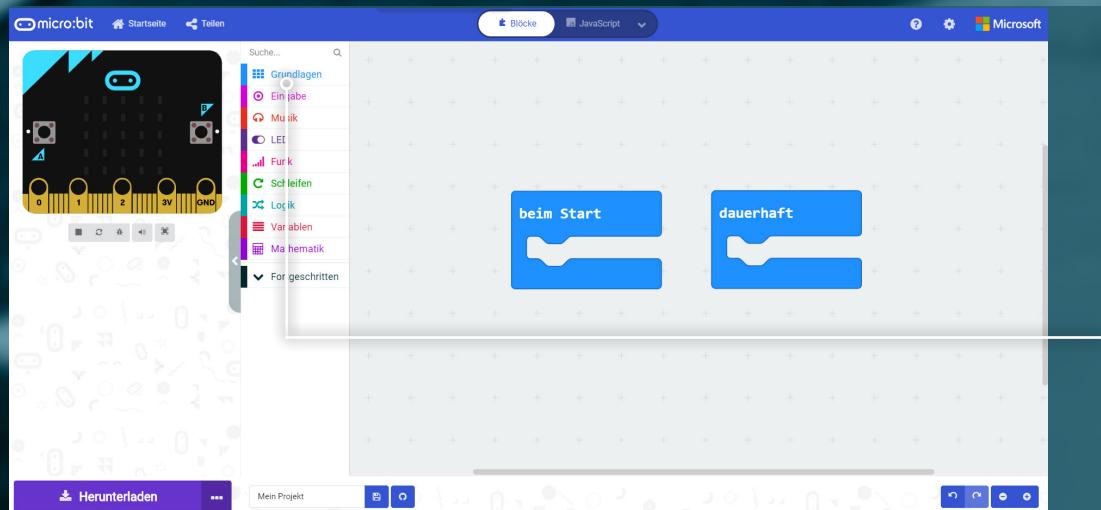
ERSTELLE HIER DEIN ERSTES PROJEKT



Du musst deinem neuen Projekt nur einen Namen geben und schon kann es los gehen.

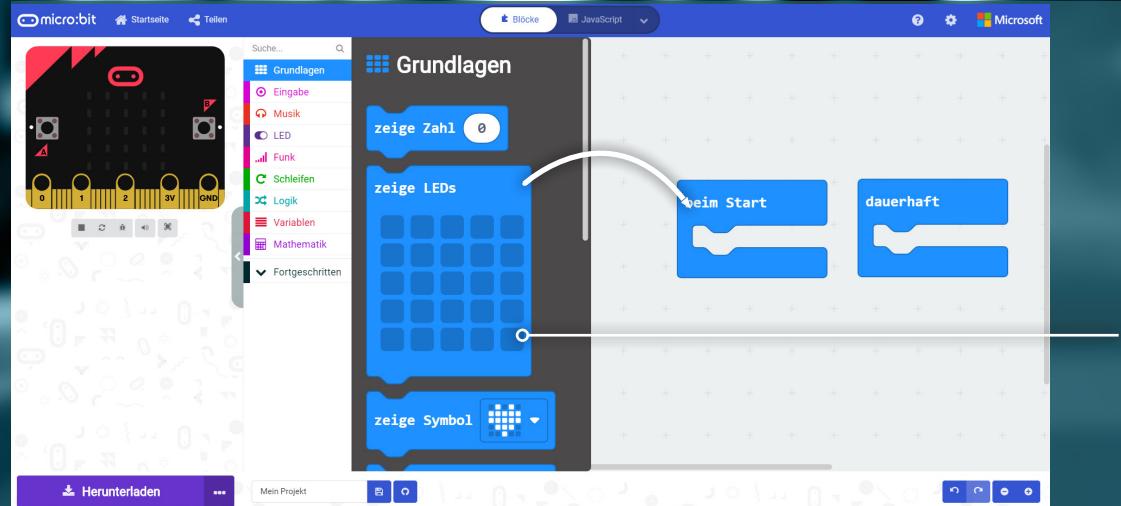
GIB HIER DEINEM PROJEKT EINEN NAMEN

BESTÄTIGE DANACH DEINE EINGABE



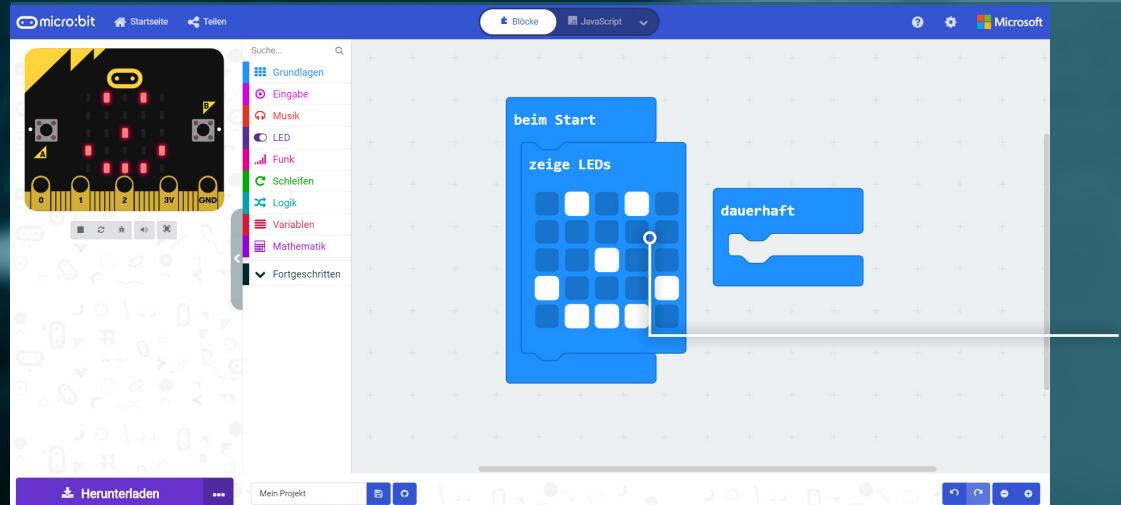
Als nächstes startest du direkt in der Entwicklungsumgebung und kannst dir deine ersten Blöcke zusammensetzen. Für dein erstes Projekt erstellen wir nun gemeinsam ein kleines Beispielprojekt.

ÖFFNE IN DER BLOCKÜBERSICHT DEN GRUNDLAGEN-BEREICH



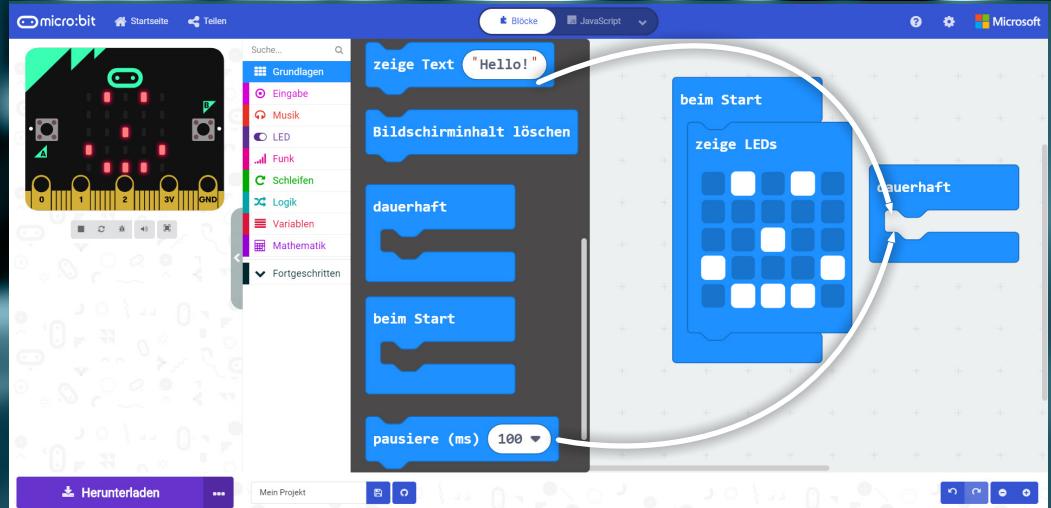
Du wirst sehen, dass sich die Blockübersicht vergrößert hat und sich die Kategorie, die du gerade angeklickt hast, geöffnet hat.

NIMM DEN „ZEIGE LEDS“-BLOCK UND ZIEHE IHN MIT DEINER MAUS EINFACH IN DEN „BEIM START“-AUSFÜHRUNGS-BLOCK



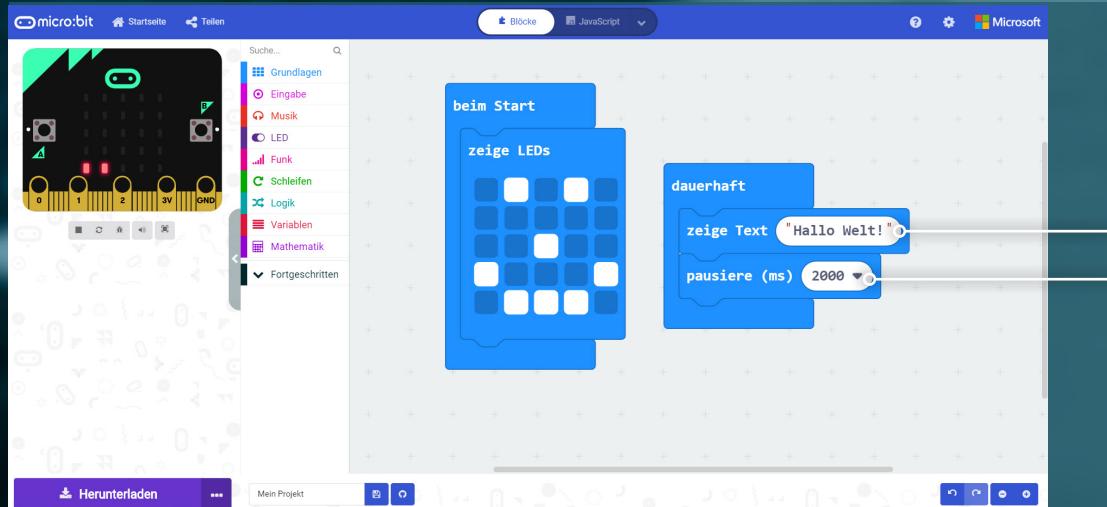
Du kannst die einzelnen Kästchen anklicken und somit einstellen, welche der LEDs aktiviert werden sollen.

KLICKE DIE LEDS AN, DIE DU AKTIVIEREN MÖCHTEST



Doch das war nur der Teil, der einmalig beim Start ausgeführt wird. Ziehe jetzt aus der Grundlagen-Kategorie den „zeige Text“-Block und einen „pausierte“-Block in deinen „dauerhaft“-Ausführungs-Block.

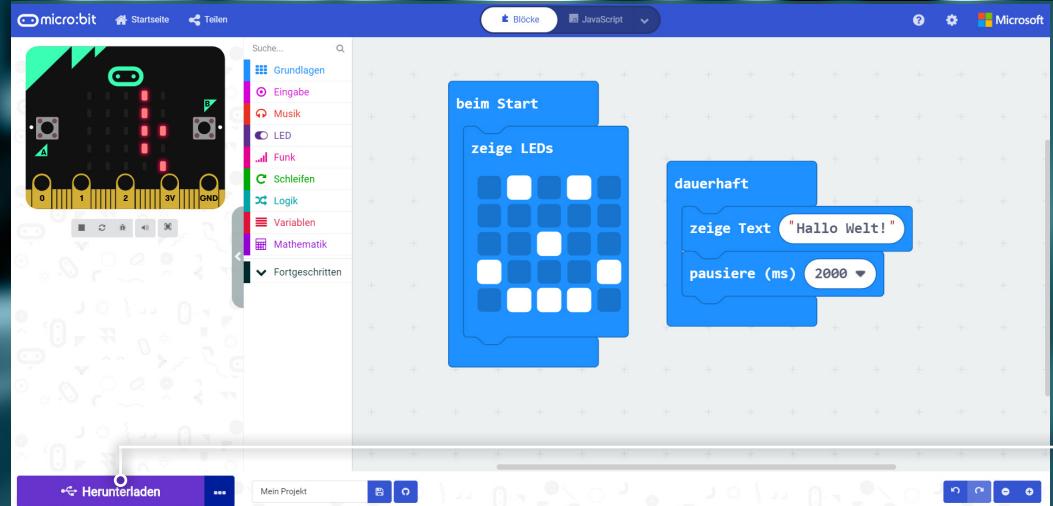
NIMM DEN „ZEIGE LEDS“-BLOCK UND ZIEHE IHN MIT DEINER MAUS EINFACH IN DEN „BEIM START“-AUSFÜHRUNGS-BLOCK



Du kannst den Text und auch die Dauer der Pause zusätzlich ändern, indem du auf die weißen Felder klickst.

GIB HIER DEINEN INDIVIDUELLEN TEXT EIN

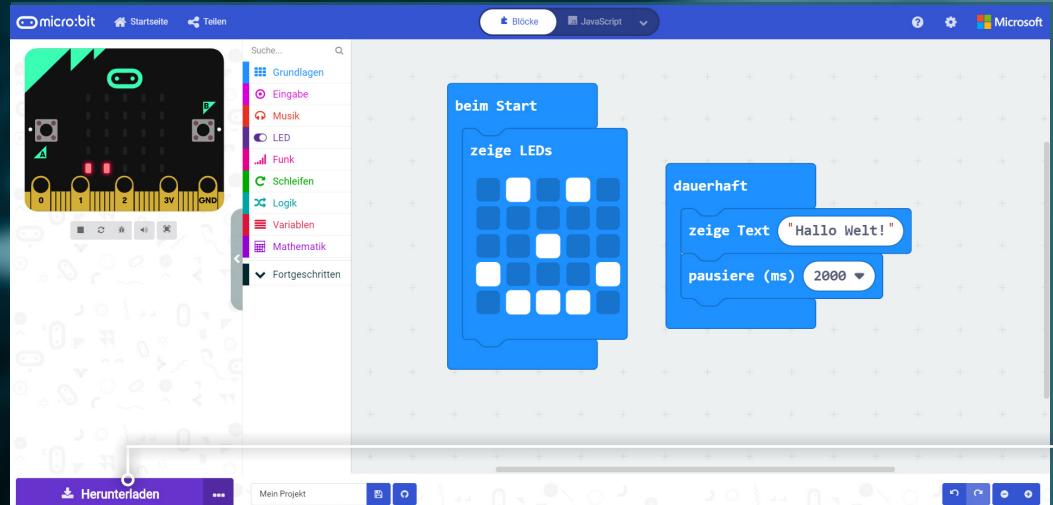
STELLE HIER DIE DAUER DER PAUSE EIN



Dein erster Beispielcode ist fertig und kann jetzt auf deinen micro:bit übertragen werden.

Schließe dazu zuerst deinen micro:bit an deinen Computer an. In den meisten Fällen sollte dein micro:bit automatisch erkannt und gekoppelt werden, sodass dein Code bequem direkt auf dein Gerät geladen wird.

KLICKE HIER FÜR DIE AUTOMATISCHE ÜBERTRAGUNG. DAS USB-SYMBOL NEBEN DEM „HERUNTERLADEN“ BEDEUTET, DASS DEIN MICRO:BIT ERKANNNT UND AUTOMATISCH GEKOPPELT WURDE



In manchen Fällen erkennt der Browser den micro:bit leider nicht, sodass die automatische Kopplung fehlschlägt. In diesem Fall kannst du deine Programmierung als Datei herunterladen.

KLICKE HIER FÜR DEN MANUELLEN DOWNLOAD. DAS DOWNLOAD-SYMBOL NEBEN DEM „HERUNTERLADEN“ BEDEUTET, DASS MIT DIESEM BUTTON DER DOWNLOAD AKTIVIERT WIRD



Auch wenn möglicherweise der Browser deinen micro:bit nicht erkannt hat, so sollte er in deinem Windows-Explorer dennoch als Laufwerk erkannt worden sein.

HIER KANNST DU DEINEN MICRO:BIT ALS DATENTRÄGER ÖFFNEN



Kopiere dann deine Programmierung, die du zuvor heruntergeladen hast, in den Datenträger-Ordner deines micro:bits. Dein Code installiert sich dann automatisch.

KOPIERE DIE DATEI AUF DEINEN MICRO:BIT

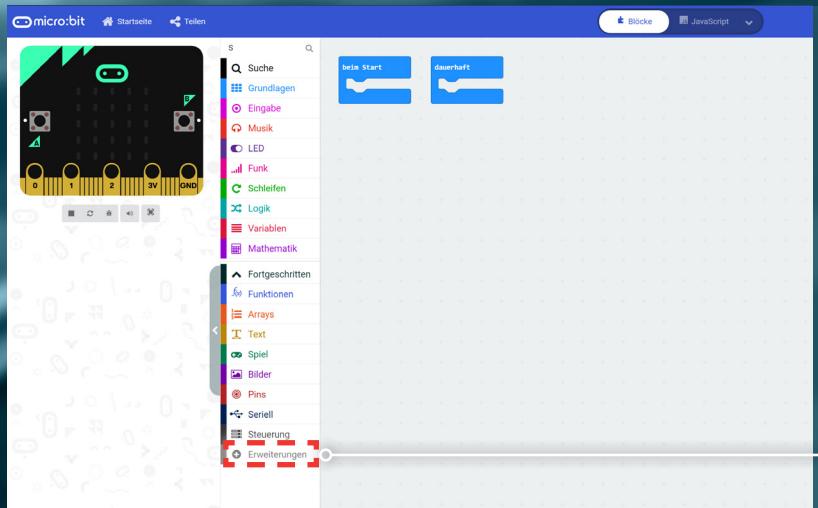


Die automatische Installation dauert nur ein paar Sekunden. Danach wird der Code immer automatisch ausgeführt, wenn du deinen micro:bit mit Strom versorgst.

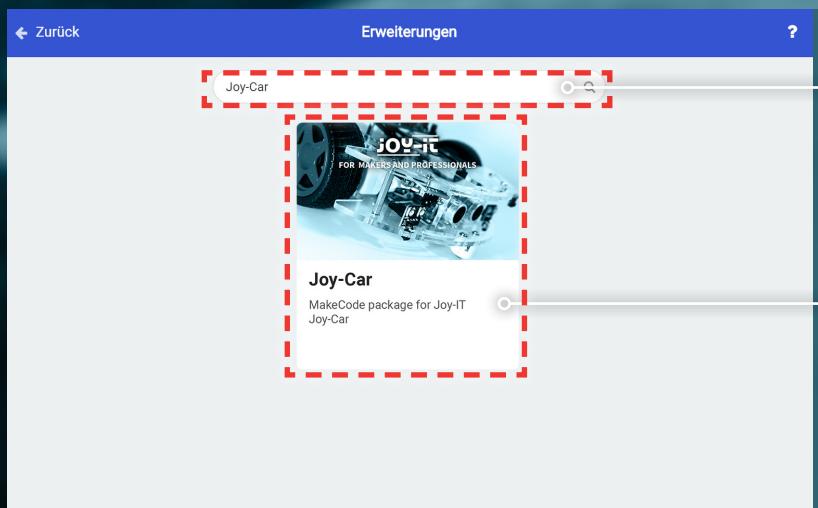


DIE JOY-CAR ERWEITERUNG

Alle Funktionalitäten des Joy-Cars haben wir für dich in einer eigenen Erweiterung zusammengestellt. Um diese für dein Projekt zu aktivieren, öffne in deiner **Blockübersicht** zunächst den Reiter **Fortgeschritten** und klicke auf **Erweiterungen**. Suche nun in der Suche nach dem Joy-Car und klicke auf unsere Erweiterung. Sie wird dann automatisch zu deinem Projekt hinzugefügt.



ERWEITERUNGSMENÜ ÖFFNEN



„JOY-CAR“ IN DER SUCHE EINGEBEN, MIT ENTER BESTÄTIGEN

ERWEITERUNG ANKlicken, DIE INSTALLATION ERFOLGT AUTOMATISCH



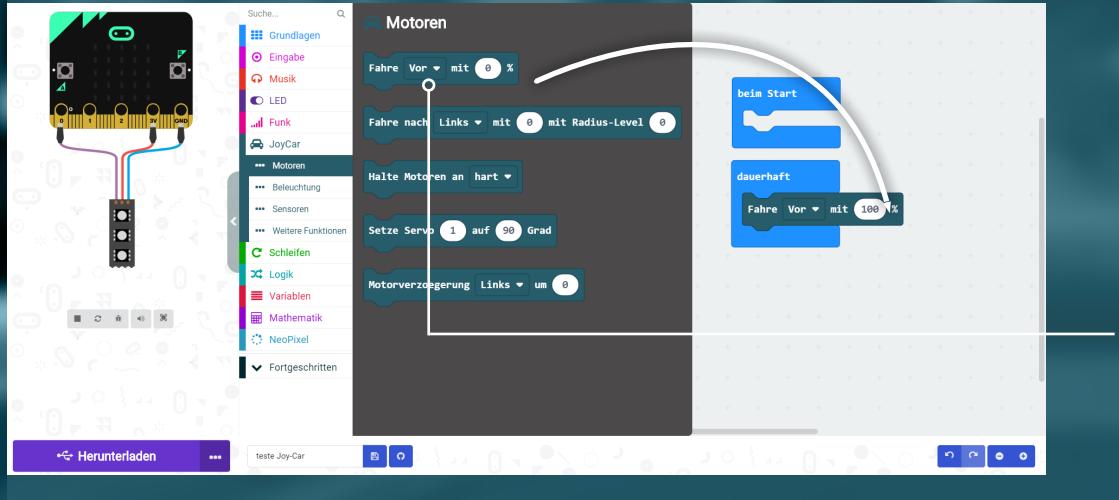
JOY-CAR

DIE ERWEITERUNG



ALLES AN EINEM ORT

Nachdem du über das Erweiterungsmenü die Joy-Car Erweiterung zu deinem Projekt hinzugefügt hast, findest du in deiner Blockübersicht den Reiter JoyCar. Hier sind alle Funktionalitäten des Joy-Cars vereint, sodass du direkt loslegen kannst. Die Funktionen sind hierbei nach **Motoren, Beleuchtung, Sensoren und weiteren Funktionen** kategorisiert. Die einzelnen Funktionen sollten leicht verständlich sein. Trotzdem gehen wir auf den nächsten Seiten für dich noch einmal genauer darauf ein.



Nach der Installation der Erweiterung kannst du auch hier wieder jeden beliebigen Block aus der Übersicht einfach in dein Programm ziehen.

So kannst du beispielsweise dein Joy-Car mit nur einer einzigen Blockanweisung vorwärts fahren lassen.

ZIEH DIR DEINEN GEWÜNSCHTEN BLOCK EINFACH IN DEINE ANWENDUNG



MEIN BLOCK

Über die Blöcke der Joy-Car Erweiterung erhältst du also Zugriff auf alle Funktionen deines Joy-Cars. So kannst du dir deine Blöcke und deine gesamte Programmierung individuell zurecht legen. Auf den folgenden Seiten haben wir dir jeden einzelnen Block noch einmal zusammengefasst und seine Funktion erklärt.



MOTOREN

Die Motoren sind der Antrieb des Joy-Cars. Mit ihnen kannst du das Joy-Car vor und zurück bewegen, verschiedene Geschwindigkeiten fahren, abbiegen und bremsen. Auch die zwei Servo-Motoren können in dieser Kategorie angesteuert werden.

Fahre Vor ▾ mit 0 %

FAHREN

Fahre vorwärts oder rückwärts. Die Geschwindigkeit kannst du zusätzlich in Prozent zwischen 0 (kein Antrieb) und 100 (maximaler Antrieb) wählen.

Motorverzögerung Links ▾ um 0 %

MOTORVERZÖGERUNG

Durch Fertigungstoleranzen kann es vorkommen, dass sich die beiden Motoren nicht mit der exakt gleichen Geschwindigkeit drehen. Hiermit kannst du eine dauerhafte Verlangsamung eines Motors in Prozent einstellen.

Halte Motoren an hart ▾

BREMSEN

Bremse das Joy-Car bis zum Stillstand. Du kannst hier zusätzlich wählen zwischen einer harten Vollbremsung oder einem sanften Bremsen, bei dem das Joy-Car langsam ausrollt.

Setze Servo 1 auf 90 Grad

SERVOMOTOREN

Die beiden optionalen Servomotoren können hiermit angesteuert und in einem Winkel zwischen 0 und 180 Grad eingestellt werden.

Fahre Vor ▾

nach Links ▾

mit 0 %

mit Radius-Level 0

KURVEN

Fahre eine Links- oder Rechtskurve. Auch hier kannst du die Geschwindigkeit in Prozent einstellen. Zusätzlich kannst du das Radius-Level der Kurve von 0 - 5 angeben (0 = Enge Kurve, 5 = Weite Kurve).

Setze PWM-Signale auf Kanal 2 0

Kanal 3 0 Kanal 4 0

Kanal 5 0

PWM-SIGNAL

Die Motoren können auch direkt über PWM-Signale angesteuert werden. Dazu kann an jeden Kanal ein PWM-Wert zwischen 0 und 255 gesendet werden.



SCHEINWERFER

Die vier LED-Module, also die Scheinwerfer des Joy-Cars, können in dieser Kategorie angesteuert werden. So kannst du hier die Frontscheinwerfer ansteuern, Blinker und Bremslichter aktivieren und noch weitere Funktionen finden.

Licht An ▾

SCHEINWERFER

Hiermit kontrollierst du die Scheinwerfer. Vorne wird hierbei ein weißes Licht aktiviert und hinten leuchtet ein rotes Licht.

Bremslicht An ▾

BREMSLICHT

Hiermit kontrollierst du ein helleres, rotes Licht auf den hinteren Scheinwerfern deines Joy-Cars.

Blinker An ▾ auf Seite Links ▾

BLINKER

Kontrolliere den Blinker für eine Seite (Links/Rechts) des Joy-Cars. Dabei fangen jeweils die vorderen und hinteren Scheinwerfer auf der ausgewählten Seite an gelb zu blinken.

Warnblinker An ▾

WARNBLINKER

Hiermit kontrollierst du das Warnlicht. Dabei fangen alle Scheinwerfer an gelb zu blinken.

Rückfahrlicht An ▾

RÜCKFAHRLICHT

Kontrolliere hiermit das Rückfahrlicht. Dies lässt ein weißes Licht auf den hinteren Scheinwerfern aufleuchten.



SENSOREN

Die Sensoren an deinem Joy-Car ermöglichen es dir erst auf bestimmte Ereignisse zu reagieren. Hindernisse, Linien, Markierungen und Geschwindigkeit? All das können die Sensoren an deinem Joy-Car erkennen.

Prüfe Linefinder-Sensor Links ▾

LINEFINDER-SENSOREN

Prüft den Linken/Mittleren/Rechten Linefinder-Sensor, ob eine Linie auf dem Boden erkannt werden konnte. Die Funktion gibt als Antwort **Wahr** oder **Falsch** (True/False) zurück.

Prüfe Speed-Sensor Links ▾

SPEED-SENSOREN

Prüft den Linken/Rechten Speed-Sensor, ob das Signal durch die Lochscheibe unterbrochen wurde. Die Funktion gibt als Antwort **Wahr** oder **Falsch** (True/False) zurück.

Prüfe Hindernis-Sensor Links ▾

HINDERNIS-SENSOREN

Prüft den Linken/Rechten Hindernis-Sensor, ob ein Hindernis erkannt wurde. Die Funktion gibt als Antwort **Wahr** oder **Falsch** (True/False) zurück.

Prüfe Ultraschall-Sensor

ULTRASCHALL-SENSOR

Prüft den Ultraschallsensor auf die Entfernung zum nächstgelegenen erkannten Objekt. Die Funktion gibt die Entfernung als Antwort zurück.



WEITERE FUNKTIONEN

Hier findest du noch weitere Funktionen des Joy-Cars, die über die bisherigen Motorfunktionalitäten, Sensorabfragen und Lichteinstellungen hinaus gehen.

Spiele Musik Dadadum ▾ wiederhole einmal ▾

BUZZER

Spiele eine vordefinierte Melodie über den Buzzer ab. Du kannst dabei zusätzlich auswählen, ob die Melodie nur einmal gespielt wird oder ob diese dauerhaft wiederholt werden soll.

Batteriespannung auslesen

BATTERIESPANNUNG

Die Batteriespannung kann über den Analog-Digital-Converter-Pin des micro:bit abgefragt werden. So kann beispielsweise die noch verfügbare Batteriekapazität ermittelt werden.



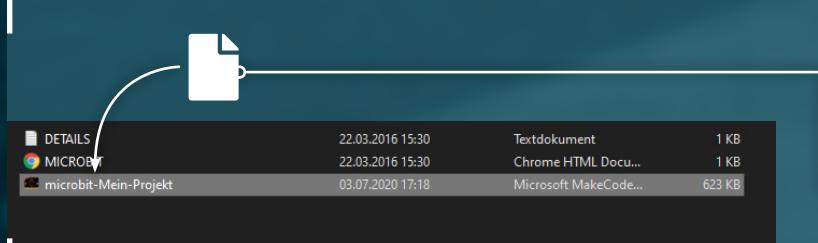
DAS VOLLE PROGRAMM



MAXIMALER SPASS, MINIMALER AUFWAND?

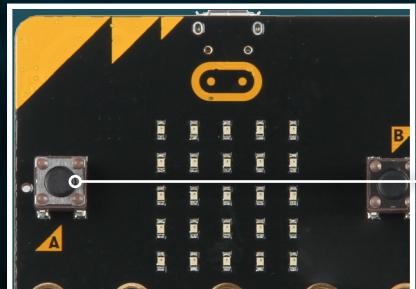
Du möchtest lieber sofort losfahren? Du kannst auch, anstatt dein Joy-Car selbst zu entwickeln, unser vorbereitetes Skript auf deinem micro:bit installieren. Hier sind die wichtigsten Funktionen bereits in einer Anwendung mit drei verschiedenen Modi zusammengestellt.

Lade dir einfach die MakeCode-Beispieldatei auf unserer [Joy-Car-Webseite](#) herunter.



Kopiere dann deine Programmierung, die du zuvor heruntergeladen hast, in den Datenträger-Ordner deines micro:bits. Dein Code installiert sich dann automatisch.

KOPIERE DIE DATEI AUF DEINEN MICRO:BIT



Die unterschiedlichen Modi kannst du dann einfach mit dem **Button A** auf deinem micro:bit wechseln.

BUTTON A DRÜCKEN, UM DEN MODUS ZU WECHSELN



EINFÜHRUNG

MICROPYTHON



MICROPYTHON?

MicroPython ist eine Implementierung, die auf der Python 3 Sprache basiert. Sie wurde in der Programmiersprache C geschrieben und ist für die Anwendung auf Mikrocontrollern, wie dem micro:bit, optimiert. Anwender, die bereits mit den Grundlagen der Softwareprogrammierung vertraut sind, können direkt mit dieser Variante beginnen.

OHNE VORERFAHRUNG IN DER PROGRAMMIERUNG EMPFIEHLT ES SICH JEDOCH, ZUNÄCHST MIT DER MAKE:CODE VARIANTE AUS DEM VORHERIGEN KAPITEL EINZUSTEIGEN.



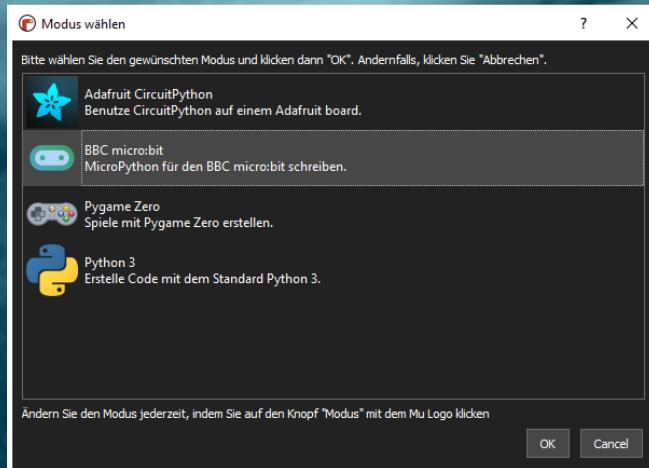
ENTWICKLUNGSUMGEBUNG

MicroPython kann mit dem Mu Editor kompiliert werden. Dieser Editor richtet sich vor allem an Anfänger und ist in seiner Handhabung daher besonders einfach zu bedienen.

Der Editor kann hier heruntergeladen werden: <https://codewith.mu/en/downloads>



INTERFACE



EINRICHTUNG

Beim ersten Start des Mu-Editors ist es zunächst notwendig, den gewünschten Modus zu wählen. Wähle hier **BBC MICRO:BIT** aus und bestätige die Auswahl mit **OK**.

Die detaillierte englische MicroPython Dokumentation kann zusätzliche Hilfestellung bieten und ist [hier](#) zu finden.

Die wichtigsten Optionen und Möglichkeiten des Editor-Interfaces werden in den nächsten Schritten erklärt.



REPL

REPL steht für **READ-EVAL-PRINT-LOOP** und stellt im Mu-Editor die Konsole dar. Hier können Ausgaben während der Codeausführung angezeigt werden und eigene Befehle an den micro:bit gesendet werden.

Zusätzlich gibt es Tastenkombinationen die zur Steuerung innerhalb des REPL verwendet werden können:

STRG + D: Soft reboot
STRG + C: Codeausführung stoppen



PRÜFEN

Hiermit kann der verfasste Quellcode überprüft werden. Fehler werden automatisch erkannt und entsprechend angezeigt.



AUFSPIELEN

Der verfasste Quellcode wird final überprüft, für den micro:bit kompiliert und anschließend auf das Gerät übertragen.

```
from microbit import *
uref = 0.00322265625
uratio = 2.7857142
while True:
    adcvolts = pin2.read_analog()
    voltaged = uref * adcvolts
    voltagep = voltaged * uratio
    print("Eingangsspannung = " + str(voltagep) + " V")
    sleep(2000)
```

BBC micro:bit REPL

```
MicroPython v1.9.2-34-gd64154c73 on 2017-09-01; micro:bit v1.0.1 with nRF51822
Type "help()" for more information.
>>>
>>> i2c.scan()
[25, 30, 56, 96, 112]
>>>
soft reboot
Eingangsspannung = 5.98792 V
Eingangsspannung = 5.9969 V
```

Micribit

STEUERUNGSELEMENTE

CODE-BEREICH

REPL EIN-/AUSGABE-BEREICH

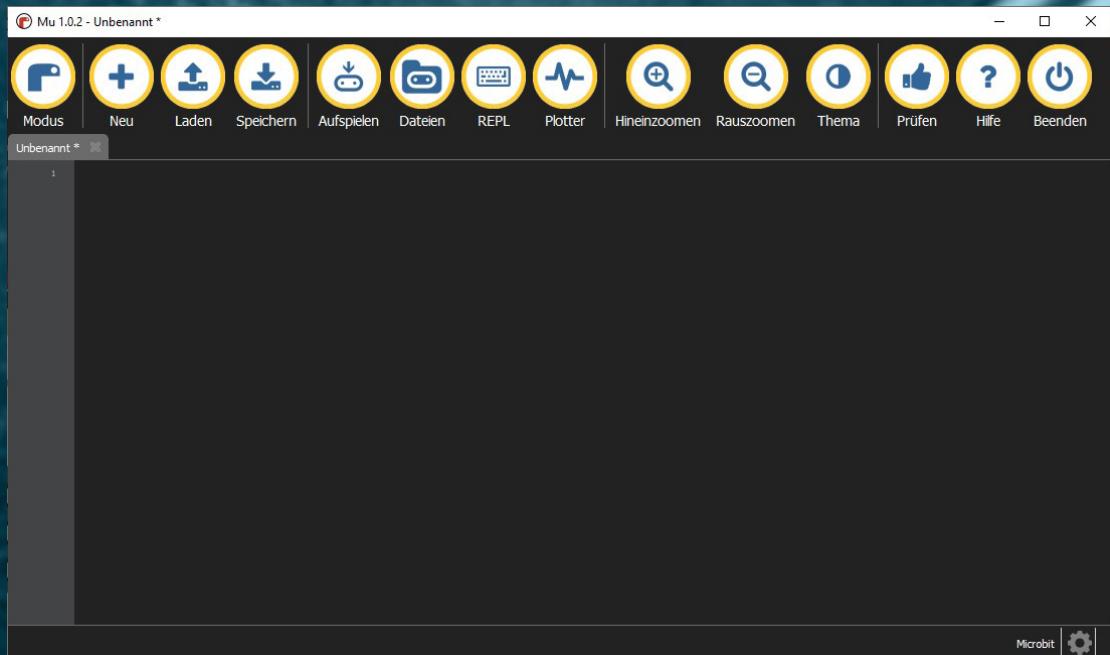


DER ERSTE START



EIN NEUES KAPITEL

Du hast noch keine Erfahrungen mit der Mu-Entwicklungsumgebung gesammelt, möchtest aber gerne mit einem kleinen Beispiel starten? Wir bringen dir die Umgebung näher!



Nach dem Start der Entwicklungsumgebung beginnst du direkt mit einem leeren Projekt. Von hier aus kann es direkt los gehen!

```
from microbit import *
text = "Hallo Welt!"
while True:
    display.scroll(text)
    print(text)
    sleep(3000)
```

Starte am besten mit einem simplen Test. Übernimm ruhig das linke Beispiel. Hierbei wird der Text „Hallo Welt!“ auf der LED-Matrix deines micro:bits und auf der Konsole angezeigt.

FÜGE DEINEN CODE HIER EIN

KLICKE, WENN DU FERTIG BIST, AUF AUFSPIELEN, UM DEN CODE AUF DEINEN ANGESCHLOSSENEN MICRO:BIT ZU ÜBERTRAGEN.

```
from microbit import *
text = "Hallo Welt!"
while True:
    display.scroll(text)
    print(text)
    sleep(3000)
```

BBC micro:bit REPL

```
MicroPython v1.9.2-34-gd64154c73 on 2017-09-01; micro:bit v1.0.1 with nRF51822
Type "help()" for more information.
>>>
>>>
soft reboot
Hallo Welt!
Hallo Welt!
```

Nachdem du auf Aufspielen geklickt hast, wird der Code sofort auf deinen micro:bit übertragen und automatisch gestartet. Die Ausgabe des Textes solltest du dann sowohl in der Konsole, als auch auf der LED-Matrix deines micro:bits sehen.

Von hier aus stehen dir alle Möglichkeiten offen! Sieh dir doch einmal unsere Beispieldateien an, die wir im nächsten Teilkapitel für dich zusammengestellt haben.

KONSOLEAUSGABE



CODE



MODIFIKATIONEN

Du möchtest lernen, wie das Joy-Car funktioniert? Optimierungen einpflegen? Eigene Entwicklungen und Ideen hinzufügen? Kein Problem!

In dem folgenden Teilkapitel haben wir unseren Code in seine Bestandteile zerlegt und ihn für dich in einzelne Dateien aufgeteilt. Wir erklären dir die relevanten Funktionen und was sie bewirken. So lernst du Schritt für Schritt alle Funktionen und kannst dir die für dich wichtigen Elemente zusammenstellen und deine eigene Joy-Car-Programmierung entwickeln.



DEN ÜBERBLICK BEHALTEN

Wir haben die Programmierung des Joy-Cars für dich in insgesamt 12 Dateien aufgeteilt. So lernst du jede Funktion kennen, verstehen und kannst dir das zusammenstellen, was du für deine eigene Entwicklung benötigst. Die folgenden Dateien haben wir für dich erstellt:

IO_EXPANDER_READ.PY
IO_EXPANDER_CONTROL.PY
MOTOR_CONTROL.PY
ADC_READ.PY
RC_JOYCAR.PY
RC_REMOTE.PY
RC_PLUS_JOYCAR.PY
RC_PLUS_REMOTE.PY
LIGHTS.PY
DEMO.PY
SONAR.PY
SERVO.PY

Auslesen der Sensordaten aus dem IO-Expander
Steuern des optionalen Sensors am 7. Ausgang des IO-Expanders
Steuerung der beiden Motoren durch den PWM-Controller
Auslesen der Versorgungsspannung durch den internen ADC
Simple Fernsteuerung des Joy-Cars (Joy-Car-Datei)
Simple Fernsteuerung des Joy-Cars (Fernsteuerungsdatei)
Erweiterte Fernsteuerung des Joy-Cars (Joy-Car Programmierung)
Erweiterte Fernsteuerung des Joy-Cars (Fernsteuerungsdatei)
Ansteuerung der Scheinwerfer-Module
Beispiel-Code
Ansteuerung des Ultraschall-Sensors
Ansteuerung der Servokanäle



IO_EXPANDER_READ.PY

Hier werden alle Sensorinformationen durch den IOExpander per I2C ausgelesen. Das Auslesen übernimmt dabei die Funktion **fetchSensorData()**. Es werden hierbei keine Übergabeparameter benötigt. Der Rückgabewert ist ein Dictionary welches die folgenden Informationen enthält:

- 0: Speed-Sensor Links
- 1: Speed-Sensor Rechts
- 2: LineTracking-Sensor Links
- 3: LineTracking-Sensor Mitte
- 4: LineTracking-Sensor Rechts
- 5: Hindernis-Sensor Links
- 6: Hindernis-Sensor Rechts
- 7: Optionaler Sensor

Die einzelnen Parameter geben den Sensorstatus als **boolsche Variable** an. **True** steht hierbei für eine HIGH-Erkennung und **False** für eine LOW-Erkennung.

ZUSATZINFORMATION: Die Funktion **zfill(s, width)** wird in dieser Datei verwendet. Sie ist eigentlich Bestandteil von Python, wurde bei der Portierung auf den micro:bit jedoch aus Speichergründen ausgelassen. Die Funktion ist jedoch für das Auslesen der Sensordaten relevant und wurde von uns daher hinzugefügt. Bei der Erstellung von eigenen Dateien, die das Auslesen der Sensordaten aufgreifen, ist diese Funktion notwendig und sollte daher ebenfalls übernommen werden.



IO_EXPANDER_CONTROL.PY

Ein am Ausgang 7 des IOExpanders angeschlossenes Gerät kann zusätzlich durch den IOExpander gesteuert bzw. auf HIGH/LOW gesetzt werden. Dies wird in diesem Beispiel durch die Funktionen **out7off()** bzw. **out7on()** demonstriert.



MOTOR_CONTROL.PY

Die Motoren können über den PWM-Controller gesteuert werden. Die **drive(PWM0, PWM1, PWM2, PWM3)** Funktion nimmt dabei vier Werte entgegen, die jeweils zwischen **0** und **255** liegen. Hierüber wird die Motorgeschwindigkeit definiert. **PWM0** und **PWM1** definieren dabei den linken Motor und steuern die Vorwärts- und Rückwärtsbewegung. **PWM2** und **PWM3** definieren dementsprechend den rechten Motor.

Die zusätzliche Funktion **stop()** setzt alle Motorbewegungen auf 0 und führt damit ein Bremsmanöver durch.



ADC_READ.PY

Die Versorgungsspannung kann über den internen Analog-Digital-Converter des micro:bit ausgelesen werden. Hierüber könnte beispielsweise eine Spannungsüberwachung oder eine Batterie-Warnung integriert werden. Die Funktion **supplyVoltage()** übernimmt dabei das Auslesen aus dem ADC-Wandler. Sie erwartet keine Übergabeparameter und liefert eine einzelne Variable als Rückgabewert. Diese enthält die gemessene Spannung.



RC_JOYCAR.PY & RC_REMOTE.PY

Mithilfe dieser beiden Dateien machst du aus deinem Joy-Car und einem weiteren micro:bit ein ferngesteuertes Auto. Lade die rc_joycar.py auf den micro:bit deines Joy-Cars und die rc_remote.py auf den micro:bit, den du als Fernsteuerung benutzt. Neige den micro:bit, den du als Fernsteuerung benutzt, in die Richtung in die du fahren möchtest und dein Joy-Car fährt in diese Richtung. Auf Button A nutzt du die Hupe. Der Button B schaltet das Licht ein.



RC_PLUS_JOYCAR.PY & RC_PLUS_REMOTE.PY

Diese Skripte funktionieren wie die normalen Fernsteuerungsskripte. Allerdings wurden die Fahrfunktionen grundlegend überarbeitet. Hier ist es nun möglich stufenlos die Geschwindigkeit in alle Richtungen anzupassen. Je weiter du den micro:bit in eine Richtung neigst umso schneller fährt bzw. umso enger biegt dein Joy-Car ab.



LIGHTS.PY

Die Ansteuerung der WS2812B-RGB-LED-Scheinwerfer-Module wird mit diesem Script übernommen. Dabei stehen hier die Funktionen **lights(x)** für das Abblendlicht, **highBeam(x)** für das Fernlicht, **breakLight(x)** für das Bremslicht, **reverseLight(x)** für das Rückfahrlicht und **indicator(y)** für den Blinker zur Verfügung. Dabei kann X die Werte 0 und 1 (Aus & An) übernehmen. Y, also die Funktion der Blinker, kann die Werte 0, 1, 2 und 3 übernehmen (Aus, Links, Rechts, Warnblinker).



DEMO.PY

Dieses Script dient als Beispiel-Anwendung und besteht aus drei Modi, die mit dem Button A durchgeschaltet werden können:

Modus 0: Standby, **Modus 1:** Linienfolgen, **Modus 2:** Hindernisse erkennen und Ausweichen



SONAR.PY

In diesem Script wird dir gezeigt, wie man das Sonar des Joy-Cars nutzen kann. Die Funktion **sonar()** erwartet keine Argumente und gibt den gemessenen Abstand zum nächsten Gegenstand in cm zurück.



SERVO.PY

Dieses Script zeigt dir, wie du die beiden Servokanäle nutzen kannst. Das Script **servo(x, y)** erwartet zwei Argumente. **x (1 - 2)** ist der Servokanal der angesteuert werden soll und **y (0 - 180)** ist die Position, die der Servo anfahren soll.



LOSLEGEN



AUFTANKEN UND LOS!

Dein Joy-Car ist nun einsatzbereit. Du kannst dir entweder nun deine eigene Programmierung entwickeln oder unsere Beispielcodes ausprobieren und die Funktionen deines Joy-Cars kennenlernen. Alle Dateien und Beispiele findest du zum Download auf unserer [Joy-Car-Webseite](#).

TIPP: WENN DU SOFORT LOSLEGEN WILLST, PROBIERE DOCH DIE DEMO.PY - DATEI AUS. HIERMIT KANNST DU SOFORT

SUPPORT



Wir sind auch nach dem Kauf für dich da. Sollten noch Fragen offen bleiben oder Probleme auftauchen stehen wir dir auch per E-Mail, Telefon und Ticket-Supportsystem zur Seite.

E-Mail: service@joy-it.net

Ticket-System: <http://support.joy-it.net>

Telefon: +49 (0)2845 9360 – 66 (10 - 17 Uhr)

Für weitere Informationen besuche unsere Website auf:

www.joy-it.net

Weitere Tipps, Anleitungen und Hilfestellung findest du außerdem auf:

JOYCAR.JOY-IT.NET