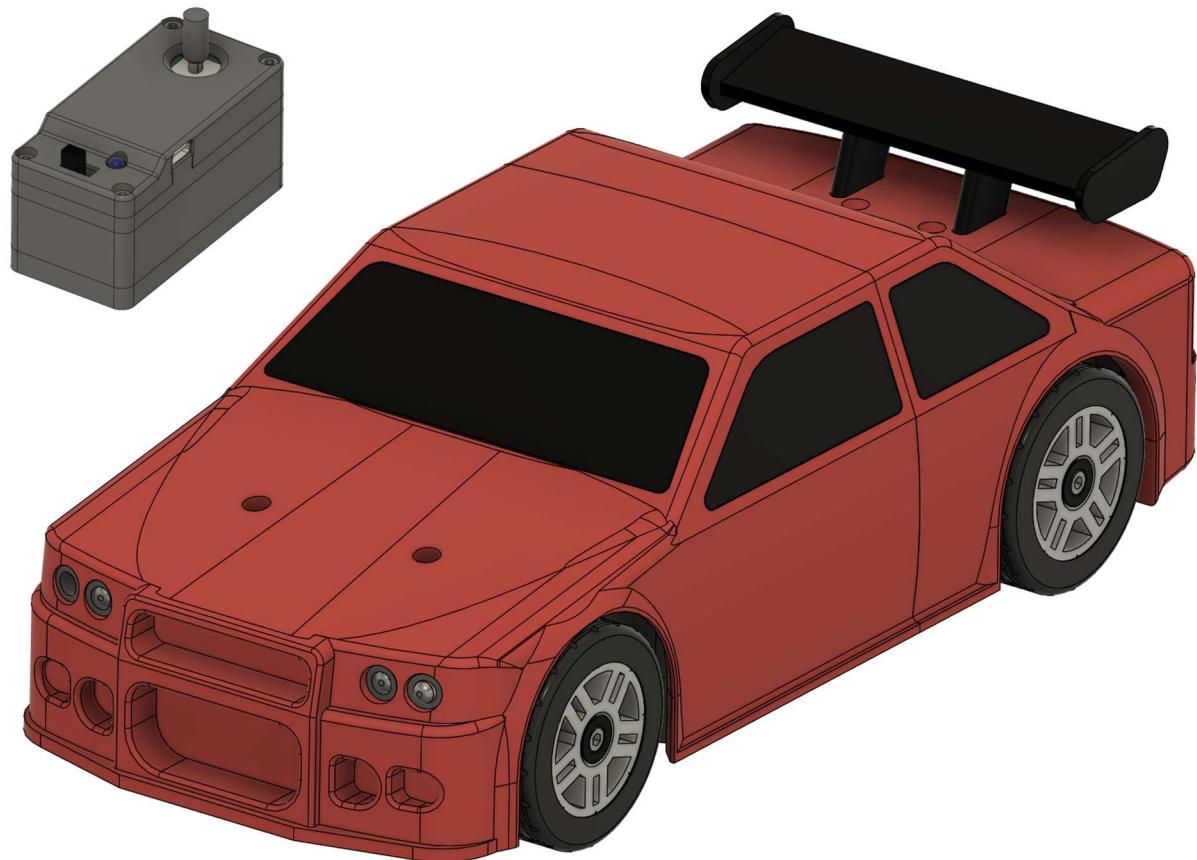


EasyRC – Open Source Modellbauplattform



Betriebsanleitung

Inhaltsverzeichnis

1.	Hinweise und Spezifikationen	3
1.1	Sicherheitshinweise	3
1.2	Spezifikationen der EasyRC-Plattform	4
2.	Teilelisten	5
2.1	Elektronikbauteile	5
2.2	Metallbauteile	7
2.3	Gedruckte Bauteile	8
3.	Bauanleitung	15
3.1	Fernbedienung	19
3.2	Antriebsstrang und Hinterräder	25
3.3	Vorderräder und Lenkung	35
3.4	Finalisierung Chassis	42
3.5	Karosserie	49
4.	Programmierung	57
4.1	Installation Arduino IDE	57
4.2	Mögliche und benötigte Code-Modifikationen	59
4.3	Programmieren der Mikrocontroller	61
5.	Erste Schritte und Bedienung	63
5.1	Batterien einsetzen und Geräte anschalten	63
5.2	Lenkung fertig montieren	64
5.3	Fahrzeug fertig montieren	66
5.4	Steuerungsprinzip und Funktionen des Fahrzeugs	67
5.5	Trimmung der Lenkung	68
5.6	Feintuning der Fahreigenschaften	69
6.	Mögliche Fehler und Lösungsansätze	70
7.	Anhang	71
7.1	Änderung der Geräte-Adresse der EasyRC-Plattform	71
7.2	Änderung des Funkkanals der EasyRC-Plattform	72
8.	Impressum	73

1. Hinweise und Spezifikationen

1.1 Sicherheitshinweise

EasyRC ist eine open-source-Plattform für drahtlos ferngesteuerte Modellfahrzeuge, die als Bausatz aus 3D-gedruckten Bauteilen montiert werden. Das Benutzerhandbuch muss vor dem Zusammenbau und der Inbetriebnahme vollständig gelesen und verstanden werden, um Gefahrenquellen zu erkennen und Verletzungen oder Schäden zu vermeiden.

Für ferngesteuerte Modellfahrzeuge gilt grundsätzlich, dass diese nicht im öffentlichen Straßenverkehr bewegt werden dürfen. Bei ferngesteuerten Modellen besteht die Gefahr, dass der Steuernde die Kontrolle verliert, wenn die Drahtlosverbindung durch Überschreiten der maximalen Übertragungsreichweite oder Interferenzen gestört wird. EasyRC beinhaltet dafür eine Sicherheitsroutine, die beim Ausbleiben von Funksignalen der Fernsteuerung für wenigstens eine Sekunde die Motoren deaktiviert (die Fernsteuerung überträgt pro Sekunde 20 Datenpakete an das Modellfahrzeug, sodass diese Sicherung nicht durch einzelne verlorene Datenpakete ausgelöst werden kann). Diese Sicherheitsroutine bleibt jedoch funktionslos bei Störsignalen, die vom Modellfahrzeug fälschlicherweise als Signal der Fernsteuerung interpretiert werden. Dafür prüft das Modellfahrzeug die individuelle Geräte-Adresse (MAC) der Fernsteuerung, die mit jedem Datenpaket gesendet wird. Nur wenn diese mit der intern gespeicherten Geräte-Adresse der im Bausatz mitgelieferten Fernsteuerung übereinstimmt, wird das Datenpaket weiterverarbeitet. Deshalb ist es beim gleichzeitigen Betrieb mehrerer EasyRC-Modelle wichtig, dass jedes mit einer anderen Geräte-Adresse programmiert wird, sodass jede Fernsteuerung spezifisch nur ein Fahrzeug steuert. Die Drahtlosverbindung basiert auf einer Funkfrequenz von 2,4 GHz, in der auch herkömmliches WLAN arbeitet. Daher besteht grundsätzlich in Häusern mit vielen verschiedenen Drahtlosnetzwerken ein erhöhtes Risiko für Verbindungsabbrüche. Sollte es zu häufigen Aussetzern kommen, kann der Sendekanal der EasyRC-Plattform geändert werden.

Die EasyRC-Plattform nutzt Batterien mit Systemspannungen von maximal ca. 12 V Gleichspannung. Diese Spannung ist ungefährlich für den Menschen; es kann zu keinen gefährlichen Stromschlägen kommen. Zudem ist die Platine des Fahrzeugs über eine Sicherung vor zu großem Stromfluss geschützt. Jedoch können Batterien bei unsachgemäßem Gebrauch (z.B. durch Überladen, Kurzschluss, mechanische Beschädigung) beschädigt werden, was zur Entzündung und zum Freisetzen gesundheitsschädlicher Inhaltsstoffe führen kann. Die im Modellsystem verwendeten Lithium-Ionen-Batterien dürfen deshalb nur mit dafür geeigneten Ladegeräten aufgeladen werden. Die maximale Spannung (Ladeschlussspannung) einer einzelnen Zelle beträgt 4.2 V. Diese Spannung darf nicht überschritten werden! Die Batterien dürfen nicht weiterverwendet werden, wenn sie mechanisch beschädigt sind oder überladen worden sind. Ein Kurzschluss muss unbedingt vermieden werden. Die Platinen der Fernsteuerung und des Fahrzeugs sind verpolungsgeschützt. Da das Fahrzeug mit drei in Serie geschalteten Batteriezellen betrieben wird, muss beim Einsetzen der Batterien dennoch unbedingt auf die richtige Ausrichtung (Polarität) geachtet werden, um Schäden an den Batterien zu vermeiden. Batterien und Elektronikkomponenten dürfen nicht im Restmüll entsorgt werden, sondern müssen entsprechend der örtlichen Möglichkeiten umweltgerecht entsorgt bzw. recycelt werden.

1.2 Spezifikationen der EasyRC-Plattform

1.2.1 Fernsteuerung

Abmessungen: 80x40x41 mm³ (LxBxH)

Gewicht: ca. 125 g

Mikrocontroller: ESP32-C3

Drahtloskommunikation: 2,4 GHz (ESP-NOW Protokoll)

Joystick: Alps 10k (Playstation 4 Modell)

Stromversorgung: 1x 18650 Li-Ion Batterie (3,7 V)

Stromverbrauch: ca. 0,5 W

Hinweise:

- Batterieanschluss verpolungsgeschützt
- Warnung bei niedriger Batteriespannung (<3,4 V): LED beginnt zu blinken
- Selbstkalibrierender Joystick

1.2.2 Fahrzeug

Abmessungen: 282x125x95 mm³ (LxBxH)

Maßstab: 1:16

Gewicht: ca. 1 kg (inklusive Batterien)

Mikrocontroller: ESP32-C3

Drahtloskommunikation: 2,4 GHz (ESP-NOW Protokoll)

Motortreiber: Pololu DRV8876 (QFN)

Antriebsmotor: JGA25-370 Bürstenmotor mit Getriebe (12 V, 1000 U/min)

Geschwindigkeit: max. 10 km/h

Lenkungsservo: MG90S Microservo mit Metallgetriebe

Stromversorgung: 3x 18650 Li-Ion Batterie (11.1 V)

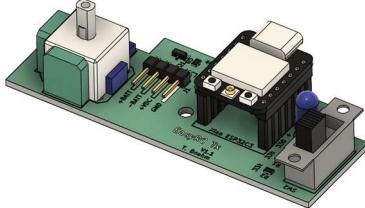
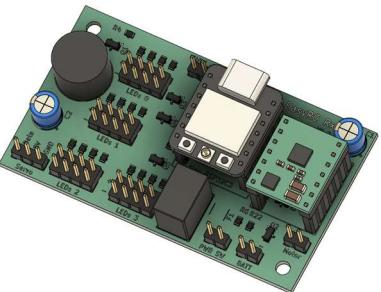
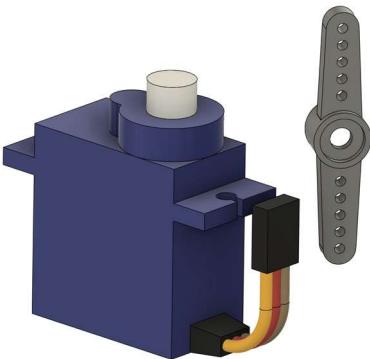
Stromverbrauch: ca. 0,8 W (Leerlauf); ca. 3-6 W (Fahrbetrieb)

Hinweise:

- Batterieanschluss verpolungsgeschützt
- Warnung bei niedriger Batteriespannung (<10,2 V): Motor deaktiviert, Buzzer piept
- Warnung bei fehlender Verbindung zur Fernsteuerung: Motor deaktiviert, Buzzer piept
- Warnung bei ausbleibenden Nutzereingaben (kein Input für >1 min): Buzzer piept
- Selbstrückstellende Sicherung bei zu großem Stromfluss (>1.5 A)
- Automatische Deaktivierung des Antriebsmotors bei Verbindungsverlust zur Fernsteuerung für >1 s

2. Teilelisten

2.1 Elektronikbauteile

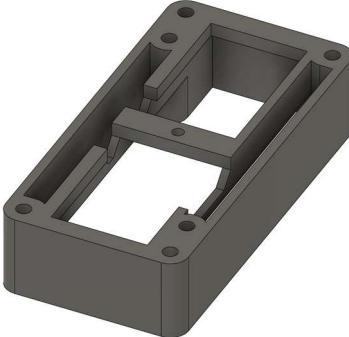
Abbildung	Bezeichnung	Beschreibung
	EasyRC Tx Platine	Leiterplatte mit Komponenten für die Fernsteuerung. Beinhaltet den Joystick sowie den Microcontroller, der die Drahtlosverbindung herstellt.
	EasyRC Rx Platine	Leiterplatte mit Komponenten für das Auto. Beinhaltet den Microcontroller für die Fahrzeugsteuerung und den Motortreiber für den Antriebsmotor.
	RC-Servomotor, Typ MG90S, mit Servohorn Farbkodierung des Anschlusskabels: Gelb: Datenkanal Rot: +5 V Braun: GND	Motor mit integriertem Getriebe und Elektronik, der den Winkel der Motorachse in einem Winkelumfang von ca. 180° entsprechend seinem Eingangssignal ausrichtet. Das Servohorn dient als Verbindung zum bewegten Element und wird mit der Motorachse verschraubt. Wichtig: Auf richtige Anschlussrichtung achten. Verpolung zerstört den Servomotor!
	DC-Getriebemotor JGA25-370 (1000 U/min) Farbkodierung des Anschlusskabels: Rot: +12 V Schwarz: GND	Motor mit integriertem Getriebe mit 1:10 Übersetzung. Die Drehzahl bei 12 V beträgt 1.000 U/min. Zur stabilen Kraftübertragung ist die Motorwelle in D-Form ausgeführt. Die Drehrichtung des Motors hängt von der Anschlussorientierung ab.

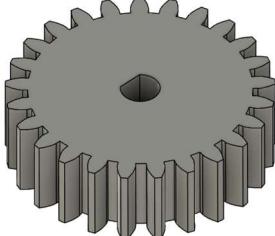
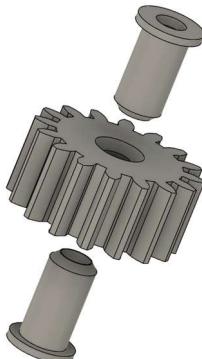
	<p>5-mm-LED</p> <p>Farbkodierung des Anschlusskabels: Rot: +5 V* Schwarz: GND</p> <p>*Gilt nur für EasyRC: Die benötigten Vorwiderstände für die LEDs sind auf der EasyRC-Platine integriert.</p>	<p>Leuchtdiode in verschiedenen Farben (Weiß, Gelb und Rot). Wichtig: Auf richtige Anschlussrichtung achten – die LED leuchtet nicht, wenn sie falsch angeschlossen ist und kann dadurch beschädigt werden!</p>
	<p>18650 Li Ion Batterie</p> <p>Spannungsfenster: 4.2 V (vollgeladen) 3.3 V (leer)</p>	<p>Lithium-Ionen-Batterie für den Betrieb der EasyRC-Plattform. Die Batterien müssen über ein separat erhältliches Ladegerät aufgeladen werden. EasyRC prüft die Batteriespannung und warnt den Nutzer, wenn die Batterieladung zur Neige geht.</p>
	<p>Batteriehalter für Einzelzellen</p> <p>Enthaltene Größen: 3x18650 1x18650</p>	<p>Wichtig: Vor dem Einsetzen der Batterien immer auf die richtige Polarität achten! Die Batterien und die angeschlossenen Komponenten können bei Verpolung beschädigt werden.</p>
	An/Aus-Schalter	Schaltet das Fahrzeugmodell an und ab.

2.2 Metallbauteile

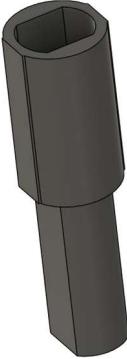
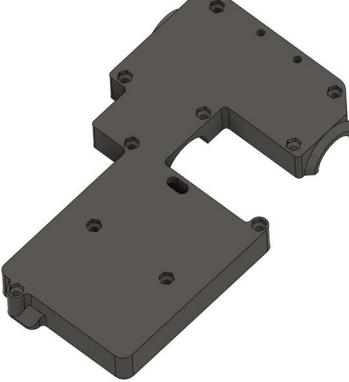
Abbildung	Bezeichnung	Beschreibung
	Zylinderkopfschraube M3 Enthaltene Größen: M3x8 mm M3x12 mm M3x16 mm M3x20 mm M3x22 mm M3x30 mm M3x35 mm M3x40 mm	Schraube mit metrischem Gewinde mit 3 mm Durchmesser. Der Schraubenkopf wird mit einem 2,5 mm Innensechskantschlüssel angetrieben.
	Sechskantmutter M3	Mutter für Schrauben mit metrischem 3-mm-Gewinde.
	Senkkopfschraube M3 Enthaltene Größe: M3x10 mm	Schraube mit metrischem Gewinde mit 3 mm Durchmesser. Der Schraubenkopf wird mit einem 2,0 mm Innensechskantschlüssel angetrieben.
	Linsenkopfschraube M2.5 Enthaltene Größe: M2.5x6 mm	Schraube mit metrischem Gewinde mit 2,5 mm Durchmesser. Der Schraubenkopf wird mit einem Kreuzschraubendreher (PH 1) angetrieben.
	Rillenkugellager Enthaltene Typen: 605 (5x14x5 mm) 608 (8x22x7 mm) 6802 (15x24x5 mm)	Die Größenbezeichnungen der Lager geben Innendurchmesser x Außendurchmesser x Höhe an.

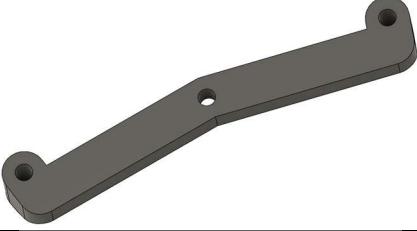
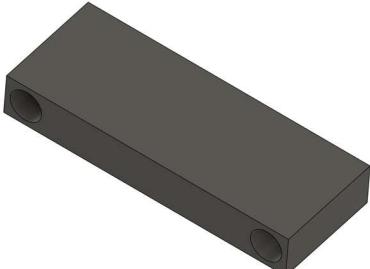
2.3 Gedruckte Bauteile

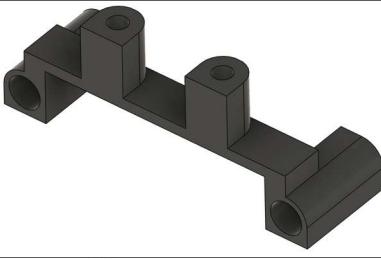
Abbildung	Bezeichnung	Benötigt in Schritt
	Fernbedienung, unterer Zwischenrahmen	1.1
	Fernbedienung, Deckel	1.2
	Fernbedienung, oberer Zwischenrahmen	1.2
	Fernbedienung, Boden	1.2

	Joystick-Verlängerung	1.3
	Fernbedienung, Batterieabdeckung	1.3
	Motorritzel (25 Zähne)	2.1
	Zwischenzahnrad (16 Zähne) mit Lageraufnahme (2x).	2.2
	Hohlzahnrad (24 Zähne) mit Abstandshalter	2.3

	Kronenzahnrad (15 Zähne)	2.3
	Kronenzahnräder (10 Zähne, 2x) mit Abstandshalter	2.3
	Differential-Körper	2.3
	Hinterradfelge mit Reifen und Spacer	2.4

	Hinterradachse (in zwei Ausführungen mit 2.5 mm Längenunterschied)	2.4
	Chassis-Boden	2.5
	Chassis-Abdeckung	2.5
	Vorderradfelge mit Reifen, Sicherungsring und Spacer	3.1
	Vorderradaufhängung mit Spacer und Lenkgestänge-Verbinder (zweimal (gespiegelt) vorhanden)	3.1, 3.2

	Lenkungsbügel	3.2
	Chassis-Sicherungsbügel	3.3
	Batteriefach-Abdeckung	4.2
	Chassis-Stoßfänger hinten	4.3
	Chassis-Stoßfänger vorn	4.3
	Servohorn-Aufsatz	4.5

	Karosserie, Heck	5.1
	Heckspoiler-Halterung	5.1
	Heckspoiler	5.1
	Rücklichtverklemmung und Auspuff (2x vorhanden)	5.4
	Karosserie, Front	5.2

	Scheiben (1x Front, 2x links, 2x rechts, 1x Heck)	5.2
	Scheibenhalterung	5.2
	Frontlicht-Verklemmung	5.3

3. Bauanleitung

Für den Zusammenbau des Modelfahrzeugs und der Fernsteuerung werden die Komponenten aus den Teilelisten benötigt. Als Werkzeuge werden Innensechskantschlüssel in den Größen 2 und 2,5 mm mit Kugelkopf benötigt (Teil des Lieferumfangs). Zudem werden ein Kreuzschraubendreher (PH1), eine kleine flache Zange und eine Pinzette benötigt bzw. sind hilfreich (nicht im Kit enthalten).

Zu den gedruckten Bauteilen muss angemerkt werden, dass diese keine perfekte Passform aufweisen. Das ergibt sich aus der Fertigung: 3D-Druck erzeugt in bestimmten Ausrichtungen keine perfekten Toleranzen (z.B. werden in Z-Richtung beim Drucken Schritte von 0,2 mm ausgeführt, sodass sich die Form des Bauteils, sofern keine horizontale oder vertikale Kante vorliegt, der gewünschten Form nur annähern kann). Zudem benötigt 3D-Druck für bestimmte Geometrien Stützstrukturen, die die Form während des Drucks stabilisieren (man kann nicht in Luft drucken, sondern benötigt Kontakt zur Bodenplatte des Druckers, um eine Form stabil herstellen zu können). Diese Stützstrukturen, die auch Support genannt werden, verkleben teilweise mit den Bauteilen und sorgen deshalb für unsaubere Oberflächen. Die gedruckten Bauteile müssen vollständig von Support-Rückständen befreit werden, um eine korrekte Passung zu gewährleisten.

Teilweise kommt es auch zu Artefakten beim 3D-Druck, wie Stringing (es verbleiben dünne Kunststofffäden zwischen oder innerhalb von Bauteilen) oder Verzerrungen der Geometrie („Elefantfuß“ oder Warping: Das Bauteil verzieht sich unmittelbar über der Bodenplatte des Druckers durch die auftretenden Temperatur- und Luftbewegungsschwankungen während des Druckprozesses). Das EasyRC-Fahrzeug wurde so entwickelt, dass diese Effekte möglichst vollständig vermieden oder aber in ihrer Auswirkung auf die Funktion vernachlässigbar sind. Dennoch kann es vorkommen, dass Bauteile nicht perfekt in ihre Position passen, rotierende Elemente in ihrem Gehäuse schleifen, oder Aussparungen für Schrauben und Muttern zu eng ausfallen, um Verschraubungen einfach durchführen zu können.

Deshalb ist es wichtig, beim Zusammenbau auf diese möglichen Fehler zu achten: Bevor Bauteile verschraubt werden, sollte geprüft werden, ob sich Schrauben einfach in die vorgesehenen Kanäle schieben lassen. Falls das nur schwer geht, sollte der Kanal ggf. mit einer längeren Schraube vorbearbeitet werden, indem das Schraubengewinde als Feile genutzt wird und durch Auf- und Abbewegungen überschüssiger Kunststoff abgetragen wird. Bei rotierenden Elementen sollte vor dem Verschrauben immer die Leichtgängigkeit geprüft werden. Fällt ein Schleifgeräusch auf, sollte geprüft werden, ob irgendwo ein Kunststofffaden oder eine Oberflächenungenauigkeit vorliegt, der oder die die Reibung erzeugt. Überschüssiges Material sollte vorsichtig mit einem Schlitzschraubendreher oder Schleifpapier entfernt werden. Der Vorteil 3D-gedruckter Bauteile ist ihr geringer Preis und ihre schnelle Fertigung (bei kleinen Stückzahlen): Sollte ein Bauteil defekt sein, kann es innerhalb weniger Stunden neu gefertigt werden, sodass ein rascher Austausch beschädigter Komponenten möglich ist.

Das EasyRC-Modell wird mithilfe von metrischen Schrauben und Muttern zusammengesetzt. Es werden Muttern als Gewindestecksel verwendet, da kleine gedruckte Gewinde nur eine sehr geringe Stabilität aufweisen und deshalb nur wenige Male verwendet werden können, bevor sie abgerieben/abgetragen worden sind. Alternativ könnten gedruckte Bauteile auch mit Cyanacrylat-Kleber (Sekundenkleber) verbunden werden, was allerdings kein Lösen der Bauteile ohne ihre

Zerstörung ermöglicht. Sekundenkleber oder Heißkleber können aber als schnelle Reparaturmöglichkeit für gebrochene Bauteile genutzt werden.

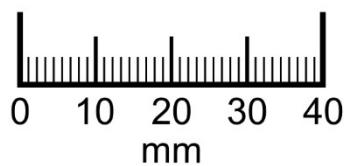
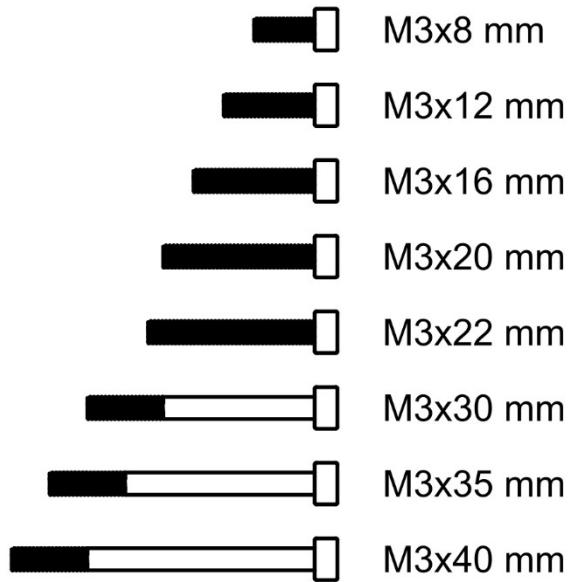
Bei den Schraubverbindungen ist es wichtig, die Muttern vor dem Verbinden der Komponenten korrekt in ihre jeweiligen Vertiefungen zu setzen. Das ist rein von Hand teilweise schwierig, was sowohl von der Position der Mutter in einem Bauteil als auch von den Toleranzen der Bauteile abhängt. Um das Einsetzen der Muttern zu vereinfachen, kann man diese auf eine Schraube drehen und mithilfe der Schraube im korrekten Winkel in die Vertiefung drücken. Alternativ setzt man die Mutter lose in die Vertiefung und greift sie von der gegenüberliegenden Seite mit einer Schraube, um sie in ihre Vertiefung zu ziehen.



Sechskant-Vertiefungen sind als Aufnahmepunkte für Muttern ausgelegt.
Die Mutter muss bis zum Boden der Tasche geschoben werden, um korrekt mit dem gegenüberliegenden Bauteil verschraubt zu werden.



Als Hilfestellung zum Finden der richtigen Schraubenlängen gibt es hier eine Größenliste. Wenn die Seite auf exaktem A4-Format ausgedruckt wird, entsprechen die gezeigten Größen genau den verschiedenen Schraubenlängen:

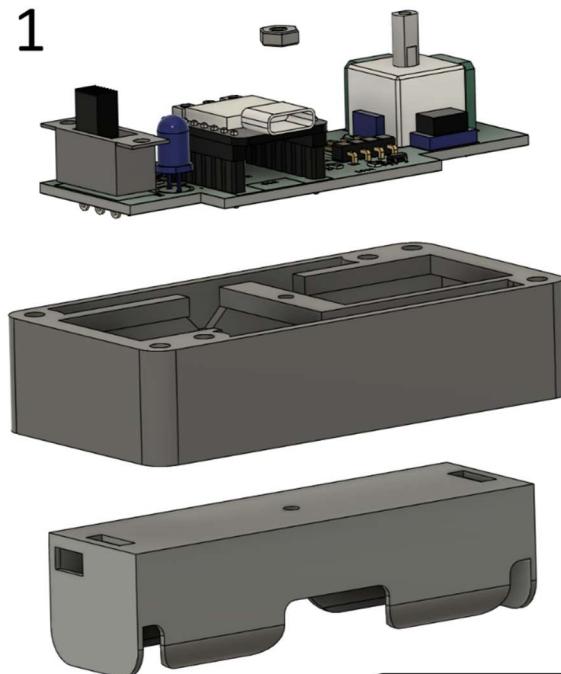


3.1 Fernbedienung

3.1.1 Einbau Leiterplatte und Batteriehalterung

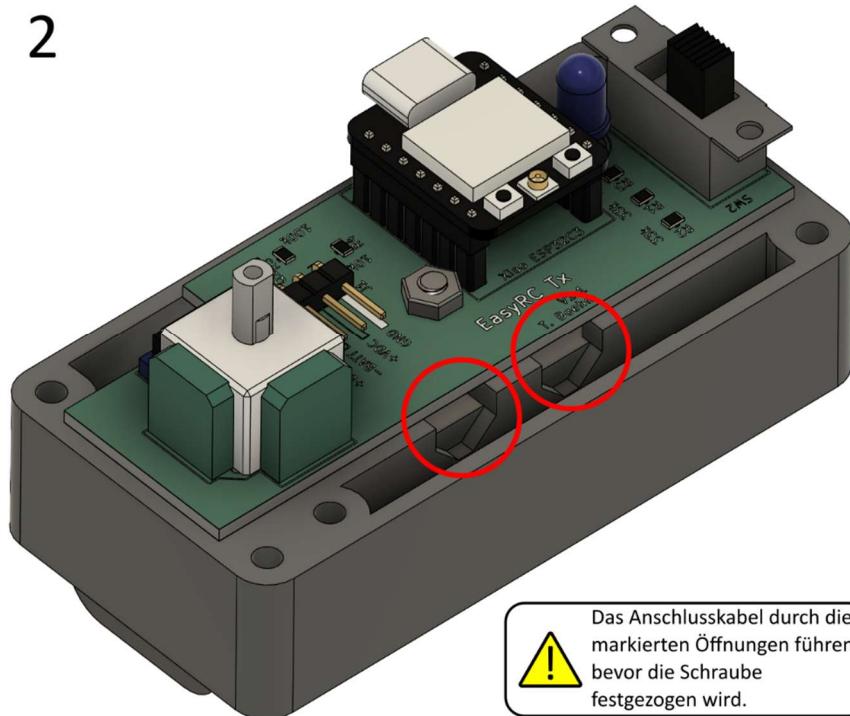
Benötigte Bauteile: Fernbedienung Zwischenrahmen unten, EasyRC Tx Leiterplatte, Batteriehalterung 1x 18650 Lilon Zelle, M3x10 mm Senkkopfschraube, M3 Mutter

1



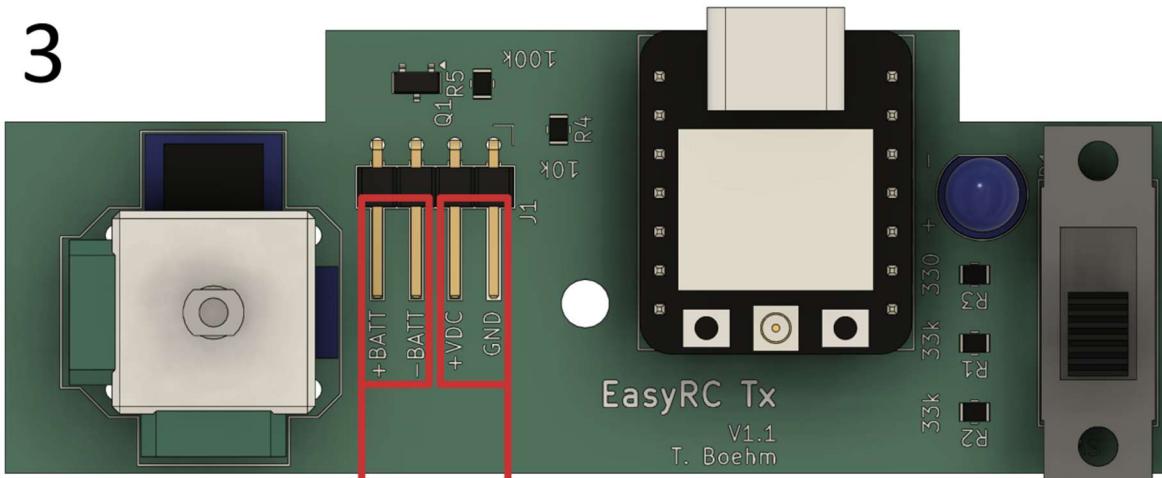
Auf die Anschlusskabel achten:
Die Kabel dürfen nicht zwischen
Rahmen und Batteriehalter
eingeklemmt werden.

2



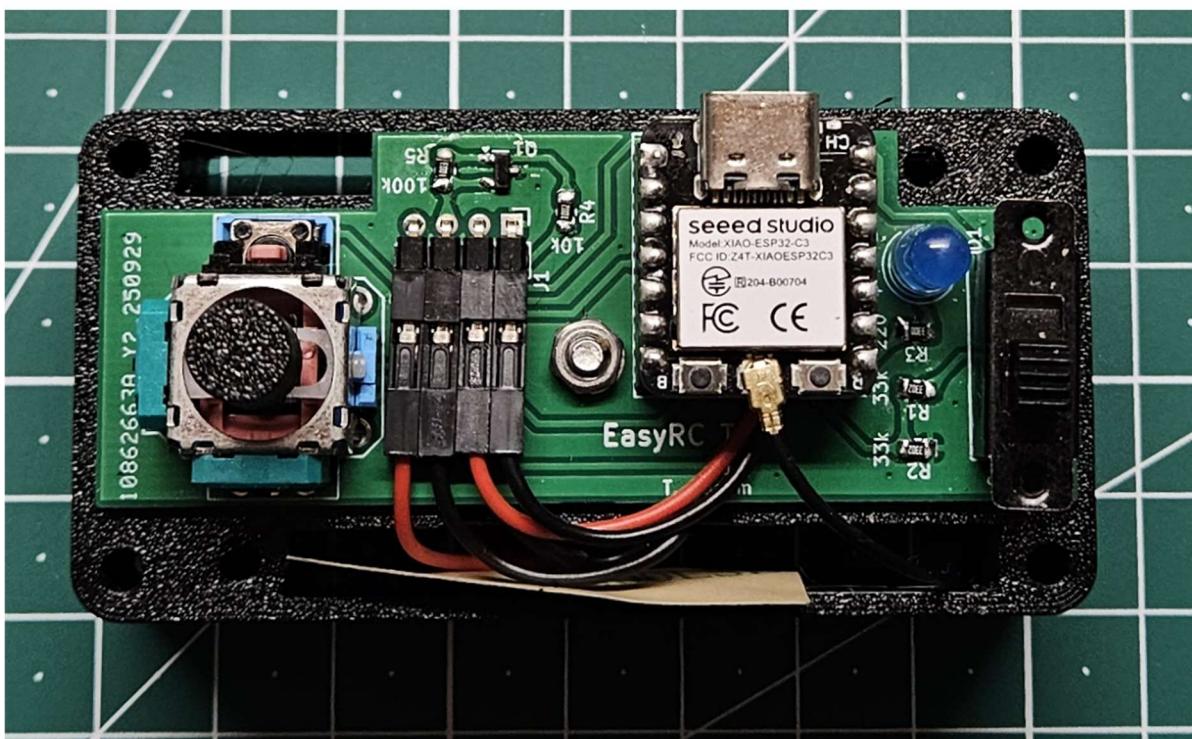
Das Anschlusskabel durch die
markierten Öffnungen führen,
bevor die Schraube
festgezogen wird.

3



1 2

Die Kabel müssen wie gezeigt angeschlossen werden und die Antenne seitlich verstaut werden.

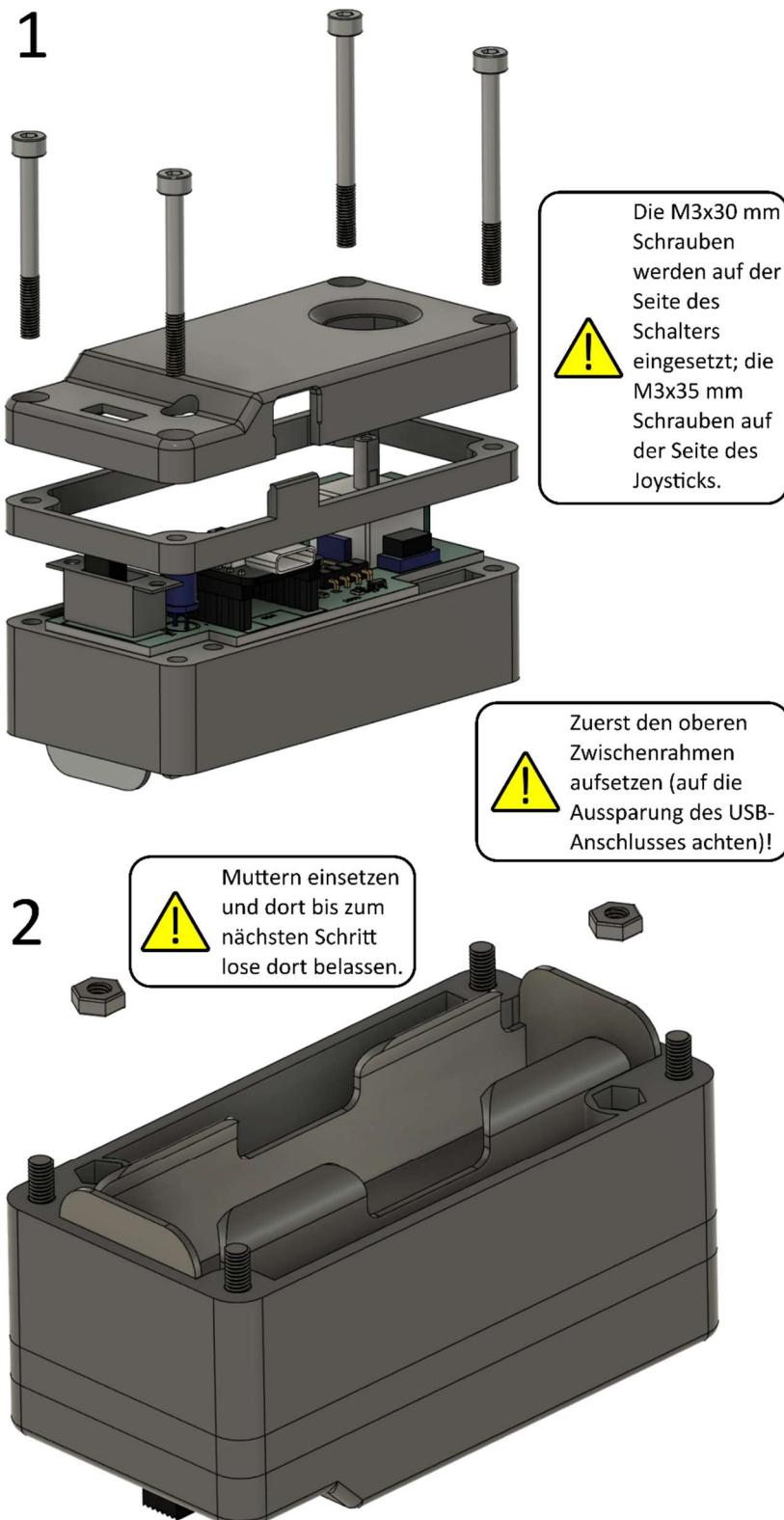


Anschlüsse: Die Batteriehalterung muss an (1) angeschlossen werden; das rote Kabel muss mit +BATT verbunden werden und das schwarze Kabel mit -BATT. Die Kabel vom Mikrocontroller müssen mit (2) verbunden werden; das rote Kabel wird an +VDC angeschlossen und das schwarze Kabel an GND.

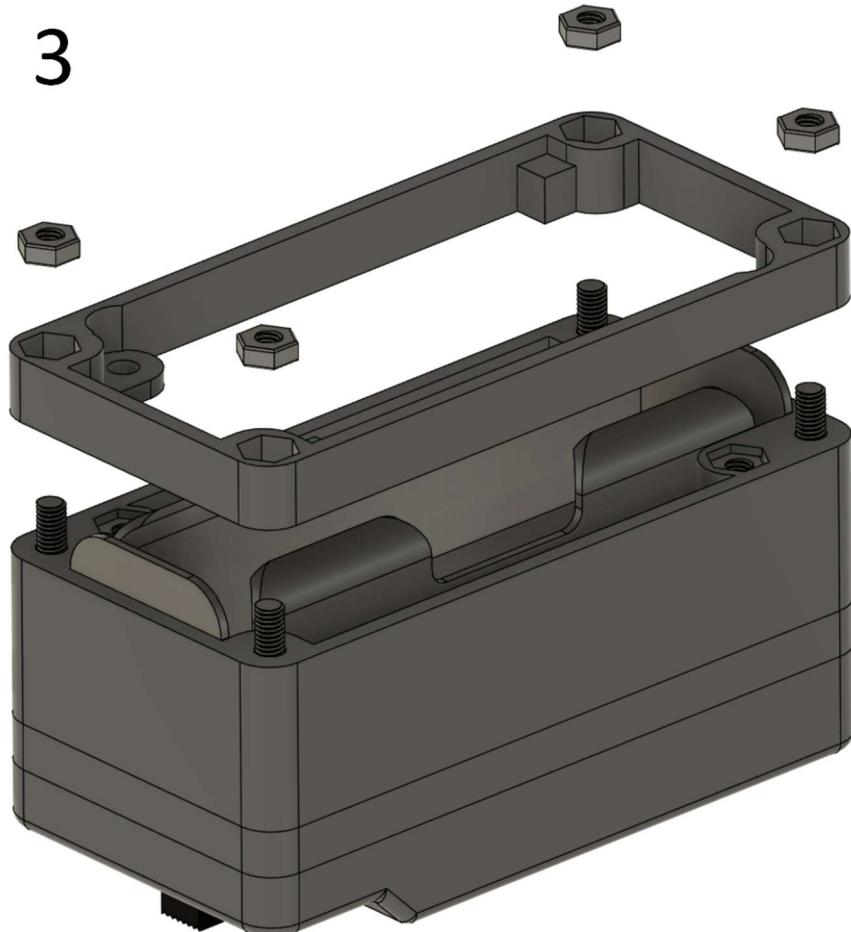
Achtung: Der Batterieanschluss (1) ist verpolungsgeschützt, der Mikrocontroller aber nicht! Sollte (2) falschherum angeschlossen werden, wird der Mikrocontroller unmittelbar zerstört, sobald die Fernsteuerung angeschaltet wird!

3.1.2 Befestigung Deckel und Boden

Benötigte Bauteile: Fernbedienung (vorbereitet), Fernbedienung Deckel, Fernbedienung Zwischenrahmen oben, M3x30 mm Schraube (2x), M3x35 mm Schraube (2x), M3 Mutter (6x)



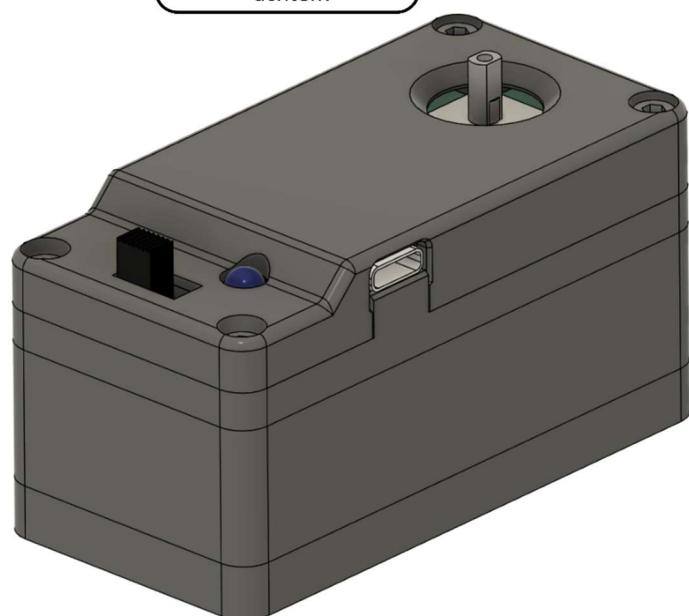
3



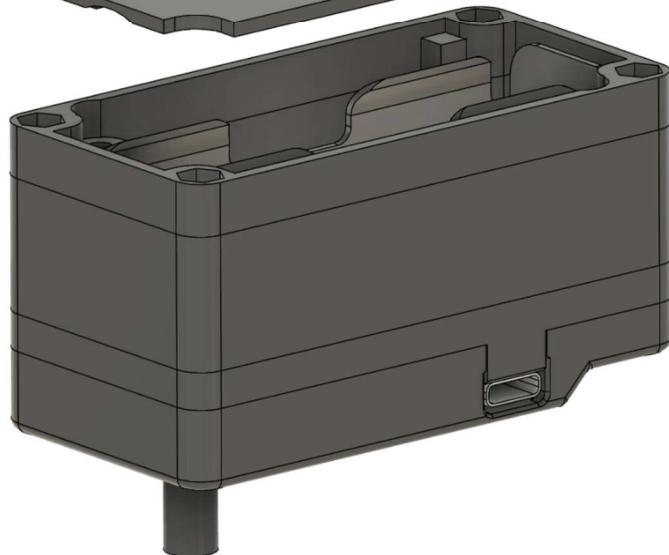
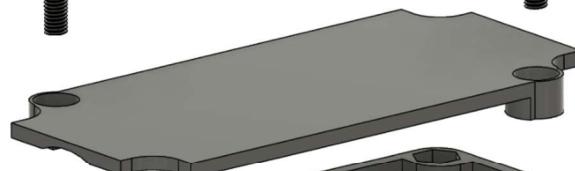
3.1.3 Abschluss Fernbedienung

Benötigte Bauteile: Fernbedienung (vorbereitet), Joystick-Verlängerung, Batterieabdeckung, M3x8 mm Schraube (2x)

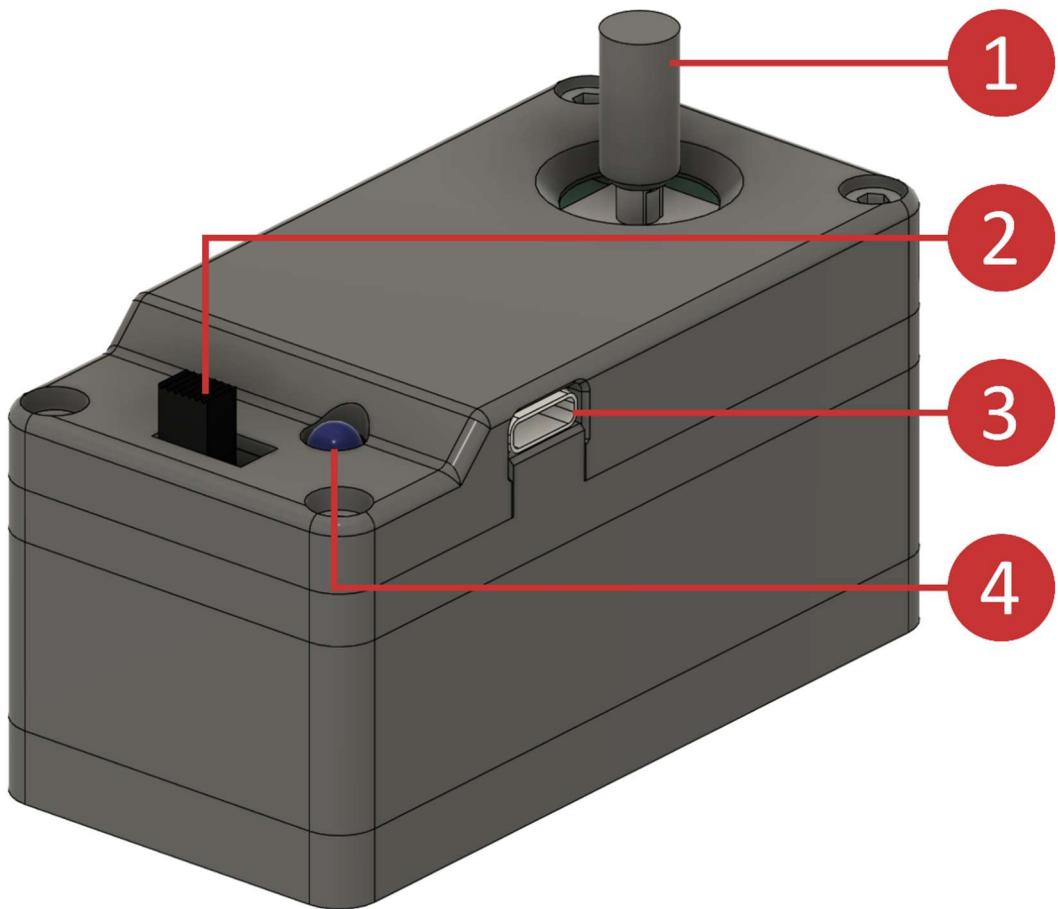
1



2



3.1.4 Übersicht Fernbedienung



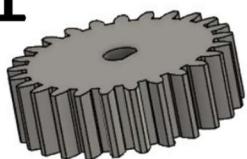
Zahl	Bezeichnung	Erklärung
1	Joystick	Joystick zur Steuerung des Fahrzeugs (vorwärts/rückwärts-links/rechts). Drücken des Joysticks löst die Hupe des Fahrzeugs aus.
2	An/Aus-Schieber	Position links: aus; Position rechts: an
3	USB-Anschluss	USB-Anschluss Typ C
4	Status-LED	Leuchtet durchgehend, wenn die Fernbedienung angeschaltet ist. Blinkt 2x pro Sekunde, wenn die Batterie schwach ist.

3.2 Antriebsstrang und Hinterräder

3.2.1 Vorbereitung Motor

Benötigte Bauteile: 1x DC-Getriebemotor, 1x Motorritzel

1



! Das Motorritzel muss entsprechend der D-Form der Motorwelle gedreht sein!

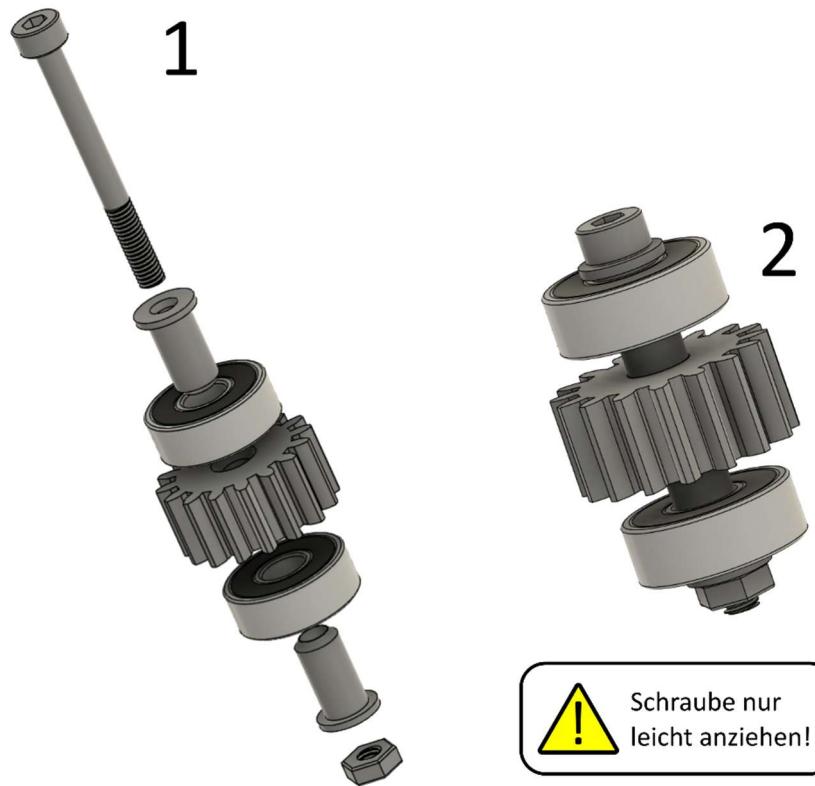
2



! Das Motorritzel muss bündig mit der Motorwelle abschließen!

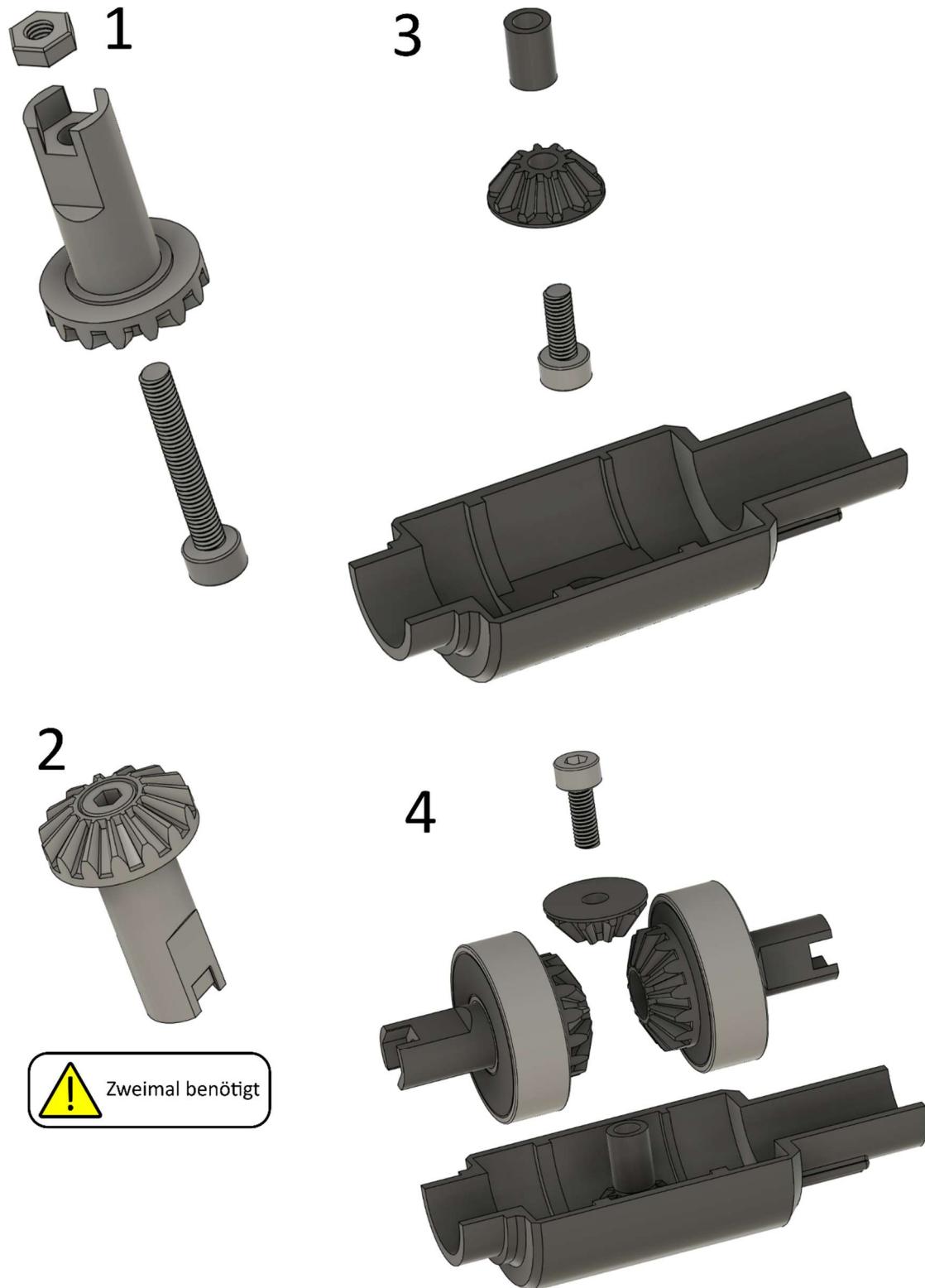
3.2.2 Zwischenzahnrad

Benötigte Bauteile: Zwischenzahnrad mit Lageraufnahme (2x), Kugellager 605 (5x14x5 mm, 2x), M3x30 mm Schraube, M3 Mutter



3.2.3 Differential

Benötigte Bauteile: Differential-Körper (2x), Kronenzahnräder (10 Zähne) mit Abstandshalter (2x), Kronenzahnrad (15 Zähne, 2x), Hohlzahnrad (24 Zähne) mit Abstandshalter (2x), Kugellager 608 (8x22x7 mm, 2x), Kugellager 6802 (15x24x5 mm, 2x), M3x8 mm Schraube (2x), M3x20 mm Schraube (2x), M3 Mutter (2x)



5



! Vor dem Verschließen des
Gehäuses etwas Schmierfett
auf die Zahnräder auftragen!

6



! Das Differential auf Leichtgängigkeit prüfen:
Die innere Achse muss sich ohne
Widerstand drehen lassen. Falls Widerstand
auftritt, sind die Zahnräder nicht korrekt
zueinander ausgerichtet, sodass deren
Anordnung korrigiert werden muss.

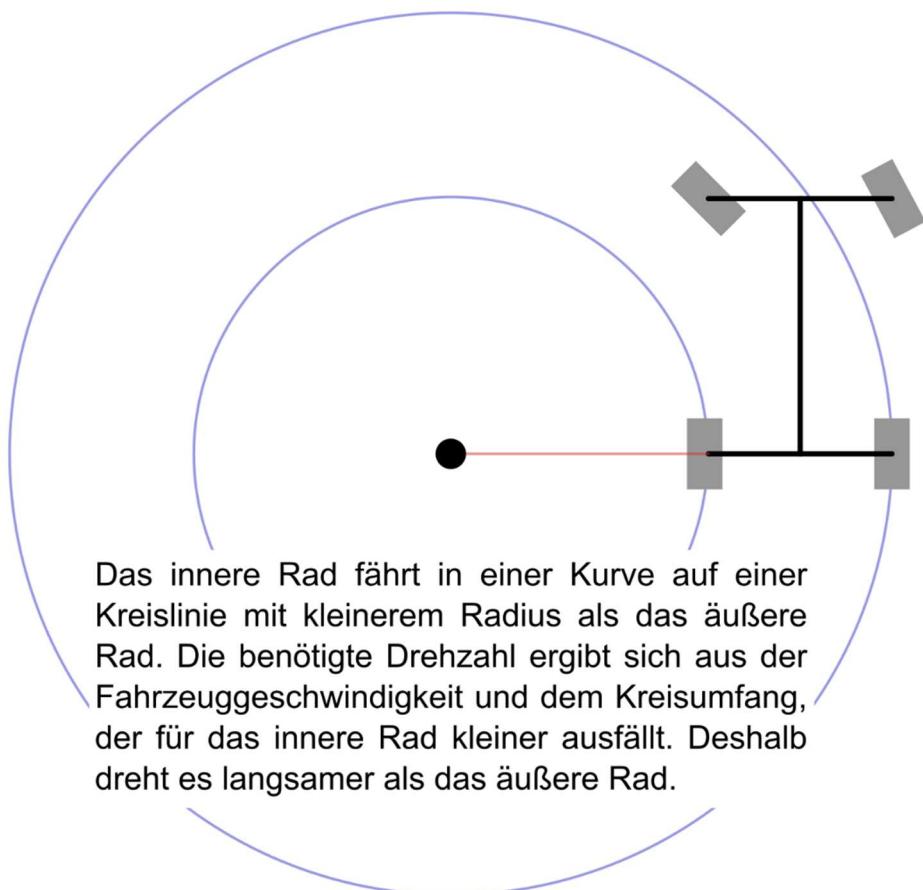
7



Schon gewusst?

Ein Differential ist in jedem Auto verbaut. Ein Differential wird benötigt, um bei einer angetriebenen Achse unterschiedlich schnelle Drehzahlen der Räder zu ermöglichen. Das ist für Kurvenfahrten wichtig: In einer Kurve muss das innere Rad langsamer als das äußere Rad drehen können, damit die Haftung nicht verloren geht. Fahrzeuge mit Hinter- oder Vorderradantrieb benötigen je nur ein Differential zwischen dem linken und rechten angetriebenen Rad. Autos mit Allradantrieb benötigen sogar drei Differentiale (vorn und hinten zwischen den linken und rechten Rädern, sowie ein Mitteldifferential zwischen der vorderen und hinteren Achse).

Der Nachteil eines Differentials wird deutlich, wenn ein Rad die Bodenhaftung verliert. Dann dreht dieses Rad frei, während das andere Rad stillsteht – das Auto steckt fest. Für Straßenfahrzeuge spielt das keine Rolle, da dieser Fall nie auftritt. Für Geländefahrzeuge ist das hingegen ein häufiges Problem. Dafür gibt es Spezialdifferentiale, die selbstsperrend wirken, oder manuell zu betätigende Differentialsperren: Die Differentialfunktion wird mechanisch ausgehebelt, sodass beide angetriebenen Räder gleich schnell drehen müssen. Die Differentialsperre darf aber nur im Gelände aktiviert werden, da im normalen Straßenverkehr zusätzlicher Reifenabrieb, schlechtere Kurvenhaftung und eine unnötig starke Beanspruchung des Antriebsstrangs auftreten.



3.2.4 Hinterräder

Benötigte Bauteile: Hinterradfelge (2x), Reifen (2x), Spacer Hinterrad (2x), Hinterradachse (2x), Kugellager 608 (8x22x7 mm, 2x), M3x30 mm Schraube (2x), M3 Mutter (2x)



3.2.5 Verbindung Antriebsstrang und Chassis

Benötigte Bauteile: Hinterrad (2x), Differential, Zwischenzahnrad, Motor mit Ritzel (alle bereits vorbereitet), Chassis-Boden, M3x30 mm Schraube (8x), M3 Mutter (14x)

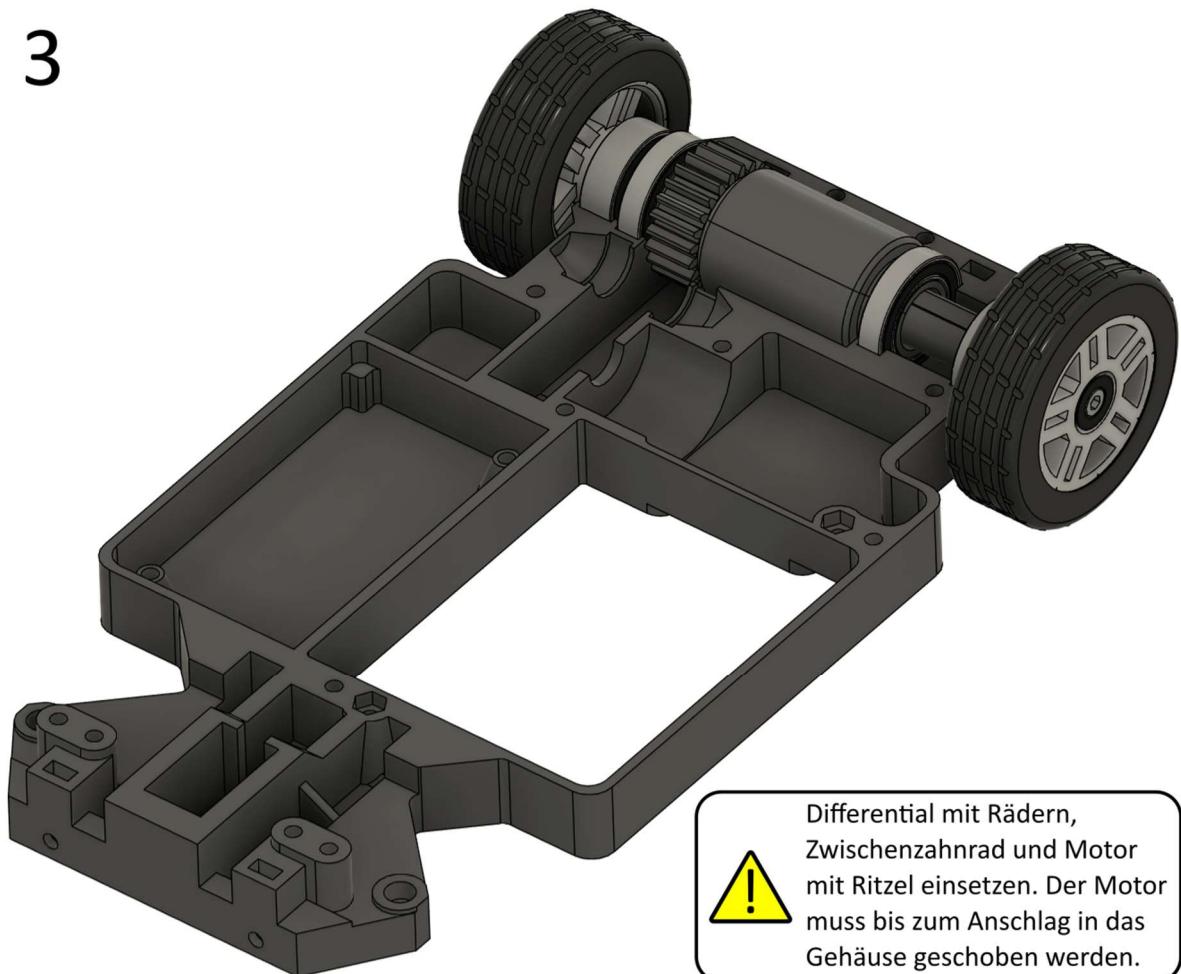
1



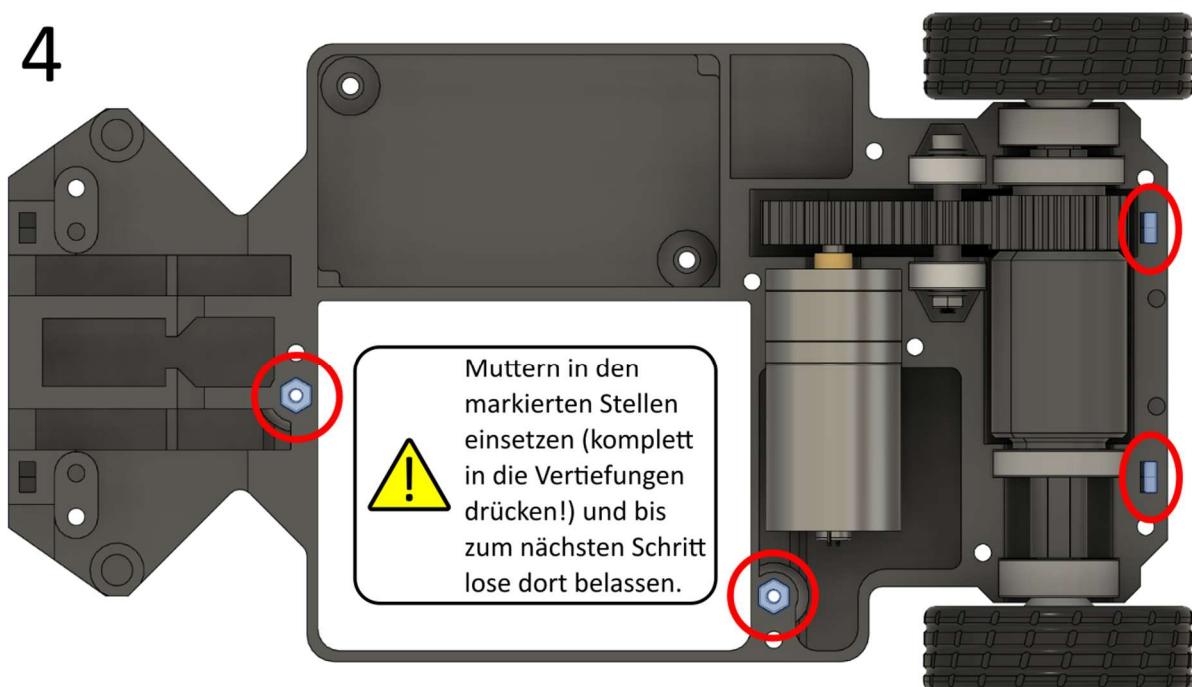
2



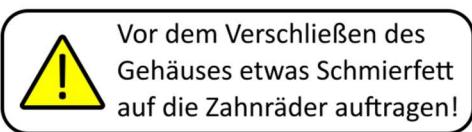
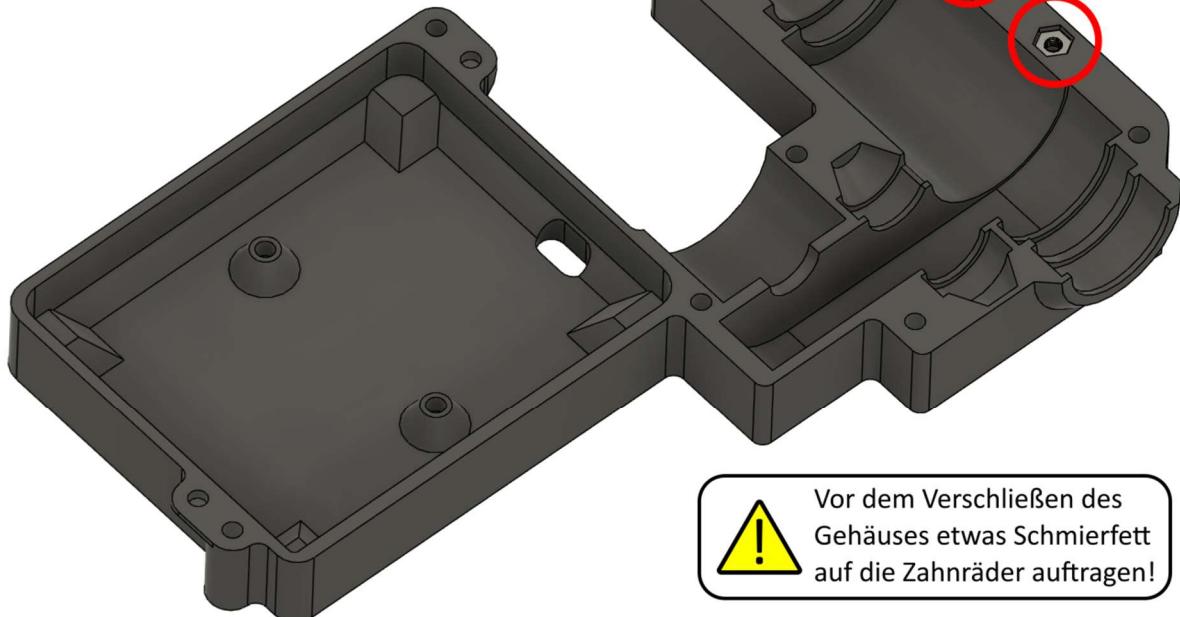
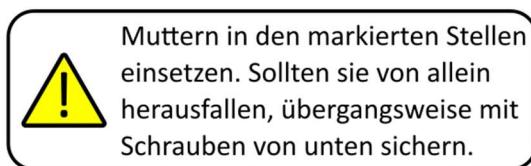
3



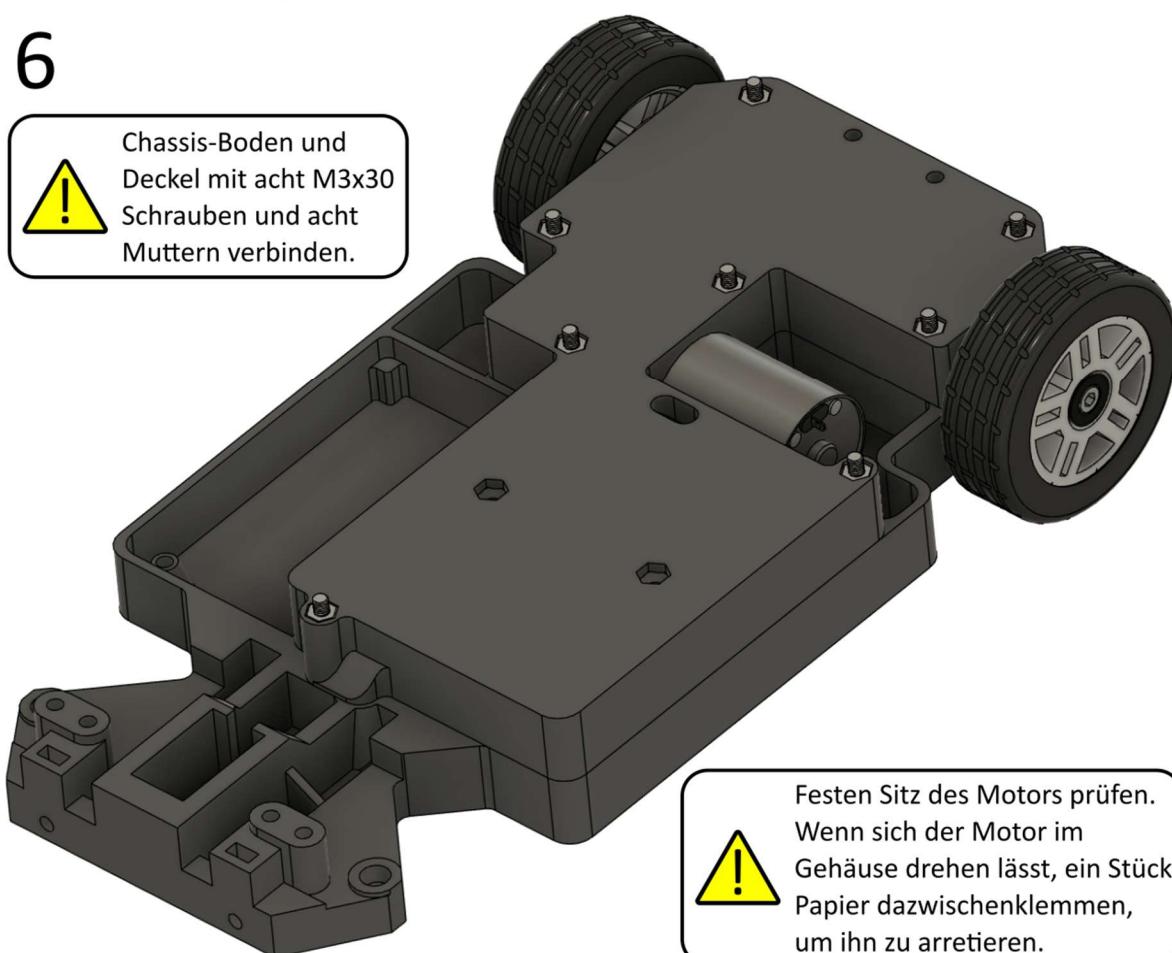
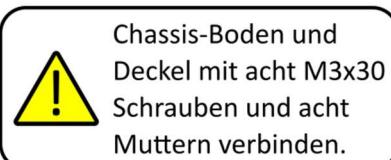
4



5



6



Schon gewusst?

Getriebe sind allgegenwärtig: Nicht nur im Auto, sondern auch in jeder Bohrmaschine, in mechanischen Uhren oder sogar in Salatschleudern sind Getriebe enthalten. Ein Getriebe sorgt dafür, dass die Drehzahl eines mechanischen Antriebs erhöht oder reduziert wird. Im Gegenzug wird das Drehmoment (die Kraft) kleiner oder größer, da sich die Leistung eines mechanischen Antriebs aus Drehmoment \times Drehzahl ergibt. Ein ideales Getriebe hat kaum Reibungsverluste, sodass der Wirkungsgrad meist bei >90% liegt. Das heißt, dass der Abtrieb eines Motors mit einer 1:10 Übersetzung zwar zehnmal langsamer dreht, aber dafür auch fast zehnmal so stark ist. Die Übersetzung ergibt sich aus dem Verhältnis der Zähnezahl des Zahnradpaars. Im EasyRC-Modell ist ein Getriebe bereits im Motor integriert, weil die gewünschte 1:10-Übersetzung für den Motor mit gedruckten Bauteilen nur über ein mehrstufiges Getriebe erreicht werden kann, was viel Platz kosten würde und die Fehleranfälligkeit erhöhen würde. Das gedruckte Getriebe im EasyRC-Modell (24:25-Übersetzung, also annähernd 1:1) dient nur dazu, den Motor weit genug von der Antriebsachse zu entfernen, um Platz für das Differential zu schaffen. Die Anzahl der Zähne des Zwischenzahnrads spielt dabei keine Rolle: Entscheidend für die Getriebeübersetzung sind lediglich die Zahnräder der Antriebs- und Abtriebswelle.

3.3 Vorderräder und Lenkung

3.3.1 Vorbereitung Vorderräder

Benötigte Bauteile: Vorderradfelge (2x), Reifen (2x), Vorderrad-Sicherungsring (2x), Vorderrad-Spacer (2x), Vorderradaufhängung (2x), Vorderradaufhängung-Spacer (2x), Kugellager 605 (5x14x5 mm, 2x), M3x16 mm Schraube (2x), M3 Mutter (2x)



3.3.2 Lenkgestänge

Benötigte Bauteile: Vorbereitetes Vorderrad (2x), Lenkungsbügel, Lenkgestänge-Verbinder (2x), M3x16 mm Schraube (2x), M3x20 mm Schraube (4x), M3 Mutter (4x)

1



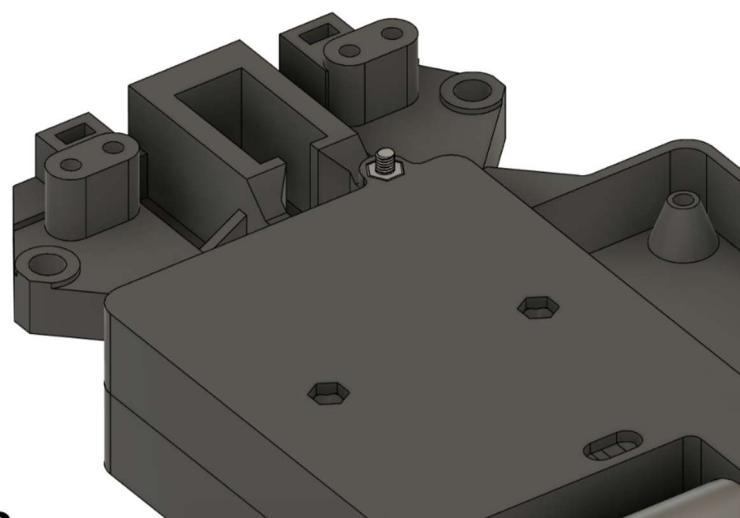
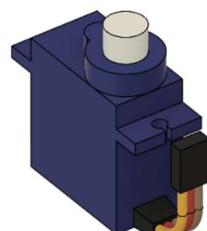
2



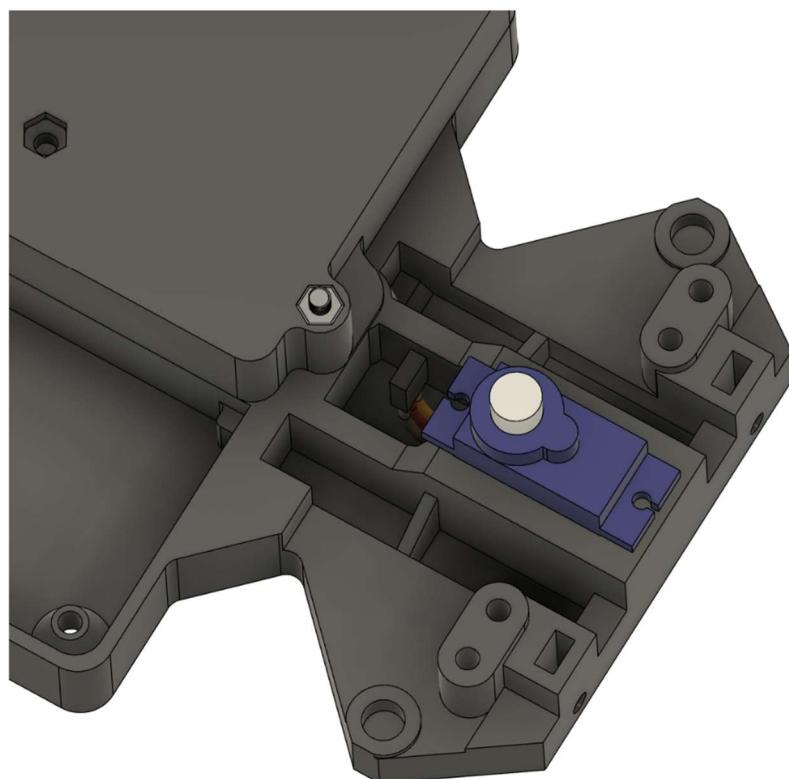
3.3.3 Verbindung von Lenkgestänge und Chassis

Benötigte Bauteile: Vorbereitetes Chassis, Vorbereitete Vorderräder mit Lenkgestänge, Chassis-Sicherungsbügel, MG90S Servomotor, M3x22 mm Schraube (2x), M3x30 mm Schraube (2x), M3 Mutter (4x)

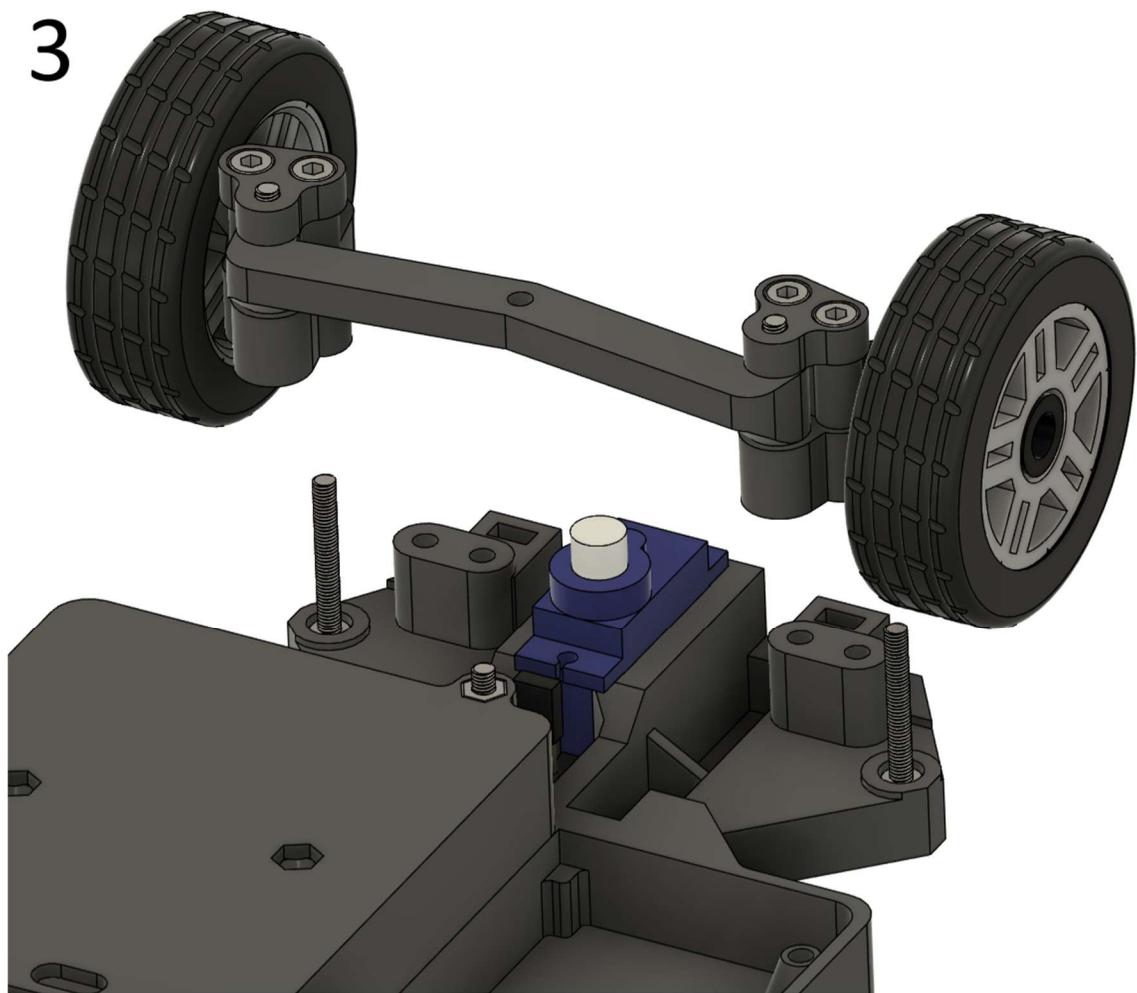
1



2



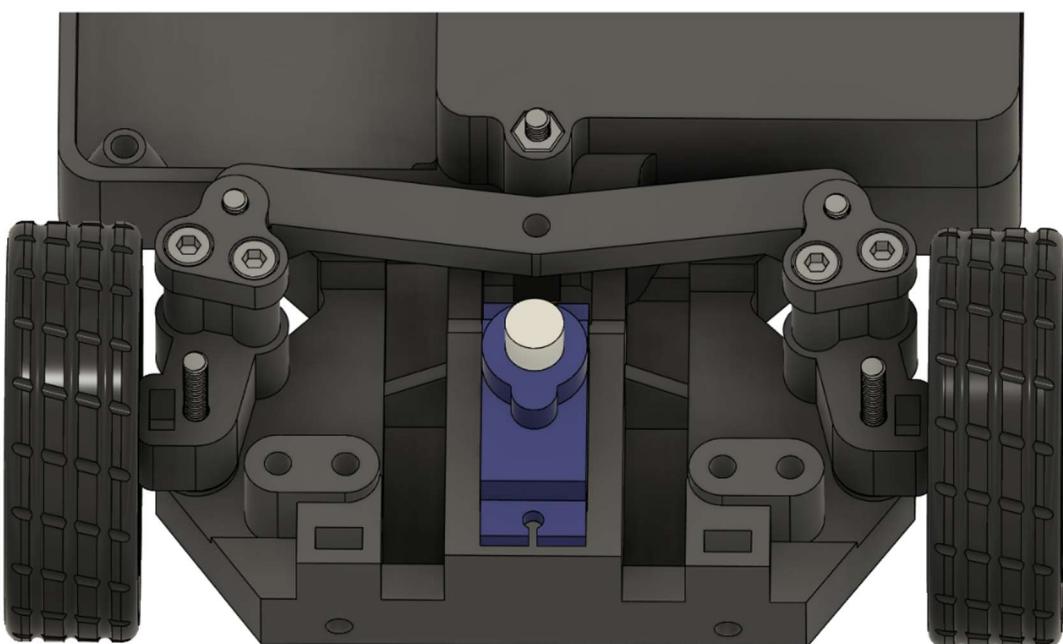
3



4

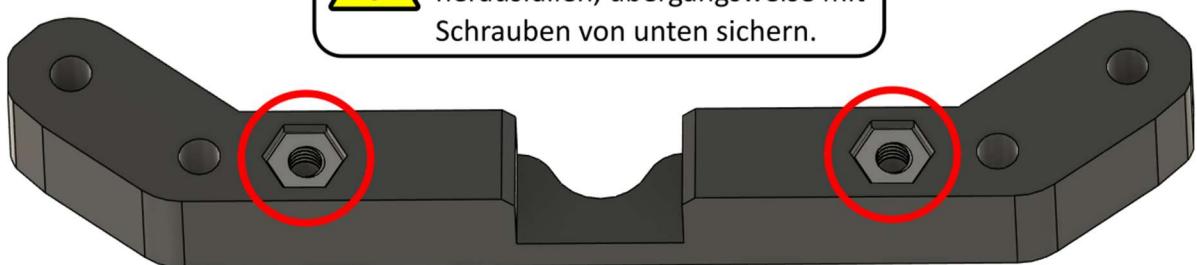


Vorbereitete Vorderräder mit Lenkgestänge
auf zwei M3x22 Schrauben aufsetzen. Die
Komponenten sitzen aktuell nur lose auf:
Bis zum nächsten Schritt in Position halten.

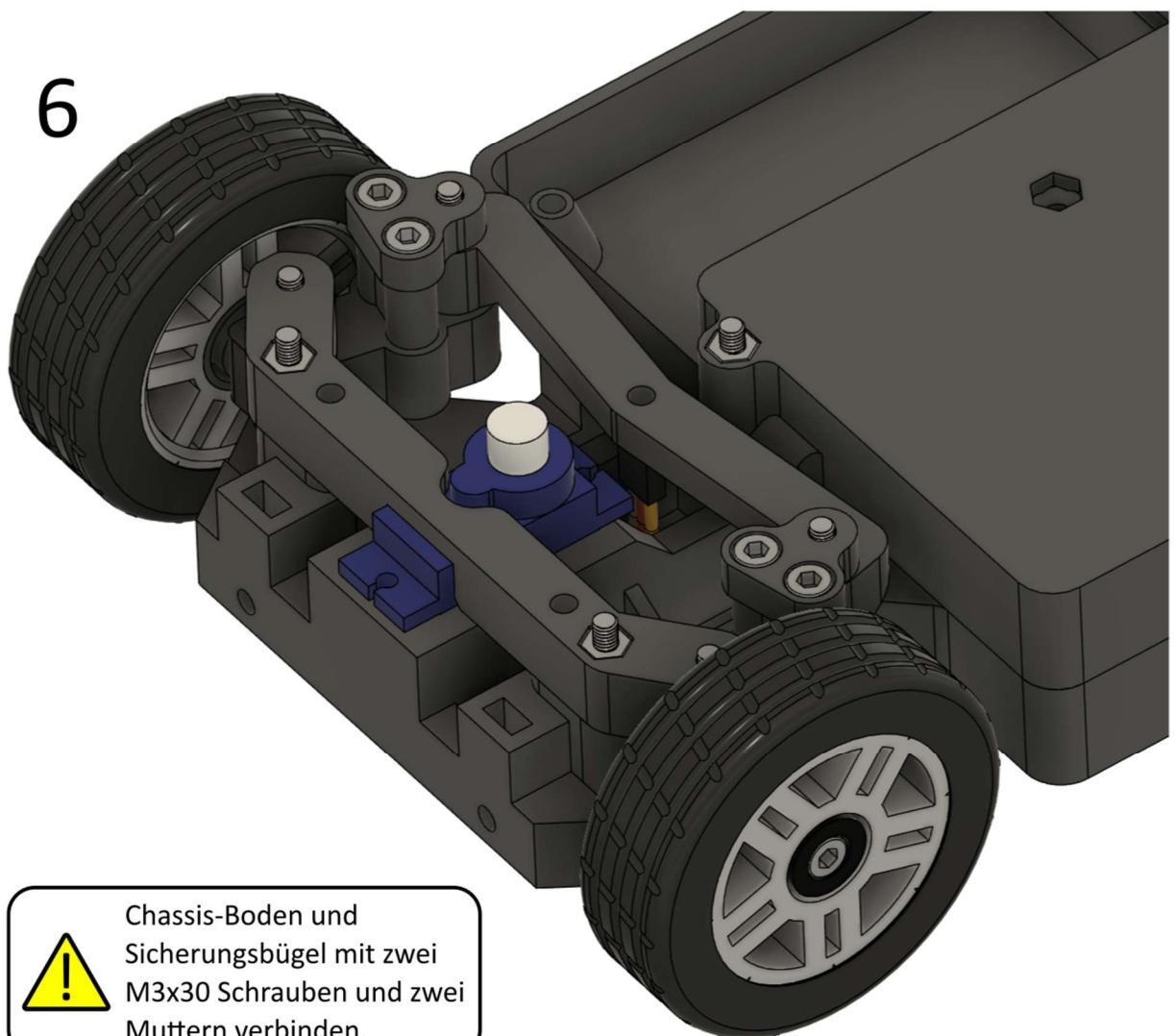


5

Muttern in den markierten Stellen einsetzen. Sollten sie von allein herausfallen, übergangsweise mit Schrauben von unten sichern.



6



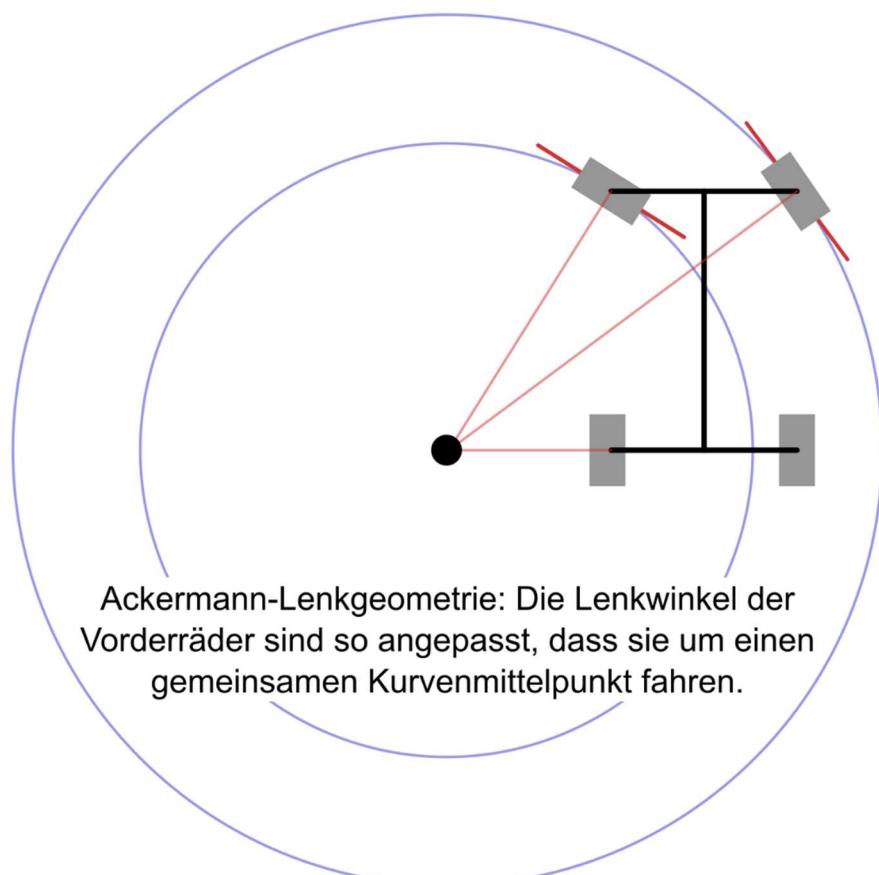
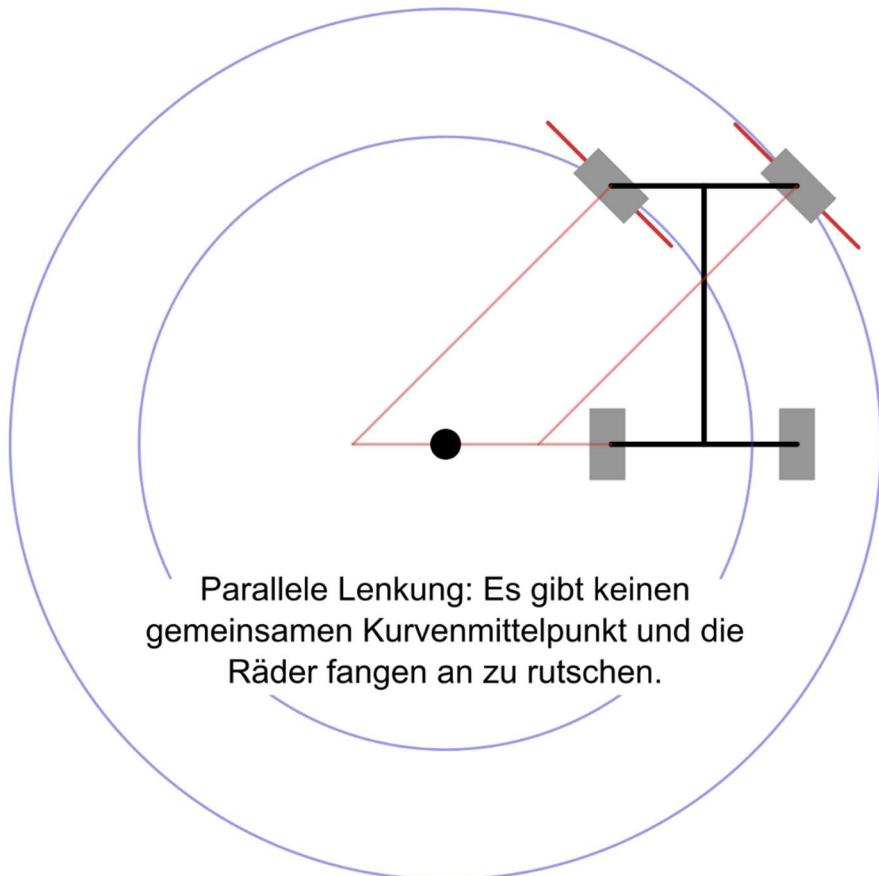
Chassis-Boden und Sicherungsbügel mit zwei M3x30 Schrauben und zwei Muttern verbinden.

Schon gewusst?

Wenn die Vorderräder zur Lenkung eingeschlagen, dann bewegt sich das innere Rad weiter als das äußere: Der Lenkungswinkel des inneren Rades ist größer als der des äußeren Rades. Diese Anordnung wird Ackermann-Lenkgeometrie genannt und unterstützt die Haftung des Fahrzeugs in engen Kurven: Da in einem Auto nur die Vorderräder gelenkt werden, muss es in einer Kurve auf einem Kreis fahren, dessen Mittelpunkt eine Verlängerung der Hinterachse ist, damit die Räder nicht anfangen zu rutschen. Das ist auch der Grund, wieso man in engen Parklücken besser rückwärts als vorwärts einparken kann: Der Wendepunkt befindet sich immer auf der Höhe der Hinterachse.

Werden die Vorderräder in dieser Konstellation mit dem exakt gleichen Winkel eingeschlagen, treffen sich ihre Winkel nicht in diesem Mittelpunkt. In der Konsequenz können sie nicht auf einer stabilen Kreisbahn rollen, sondern müssen entgegen ihrer Laufrichtung gezogen werden. Die Ackermann-Lenkgeometrie nähert beide Vorderräder dem idealen Kurvenmittelpunkt an, sodass das Auto in der Kurve so stabil wie möglich fahren kann. Dieses Konzept wurde für Pferdekutschen entwickelt und für Autos übernommen. Allerdings nutzen Autos oftmals keine vollständige Ackermann-Lenkgeometrie, da dieses Konzept vor allem bei langsamem Fahren die Stabilität erhöht. Schnelle Kurvenfahrten sorgen für verschiedene weitere Effekte, wie eine Verlagerung der Fahrzeugmasse auf die kurvenäußeren Räder und die Neigung des Fahrzeugs zur Kurvenaußenseite, was dafür sorgt, dass andere Lenkgeometrien besser funktionieren.

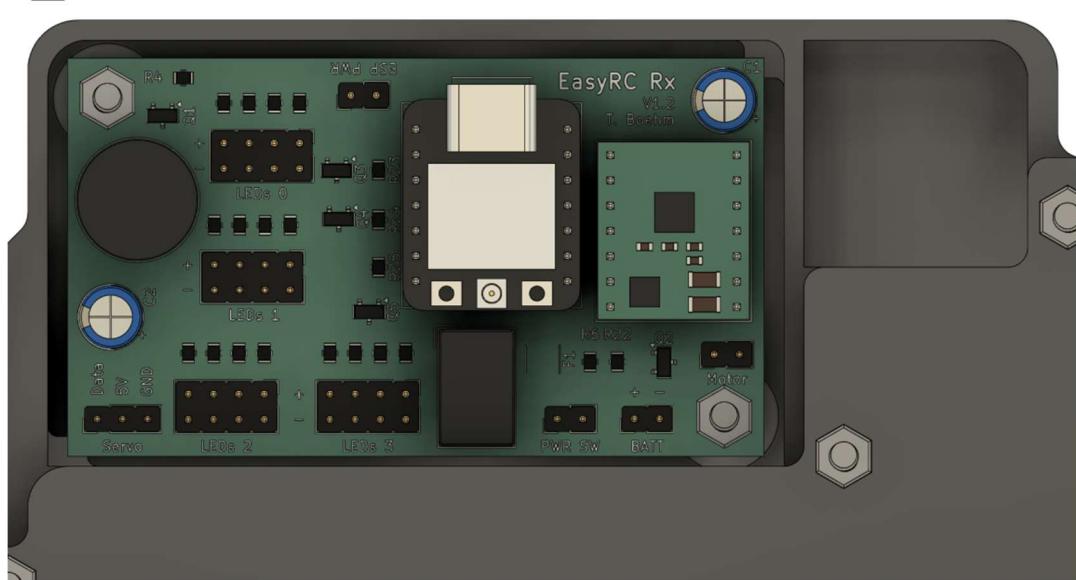
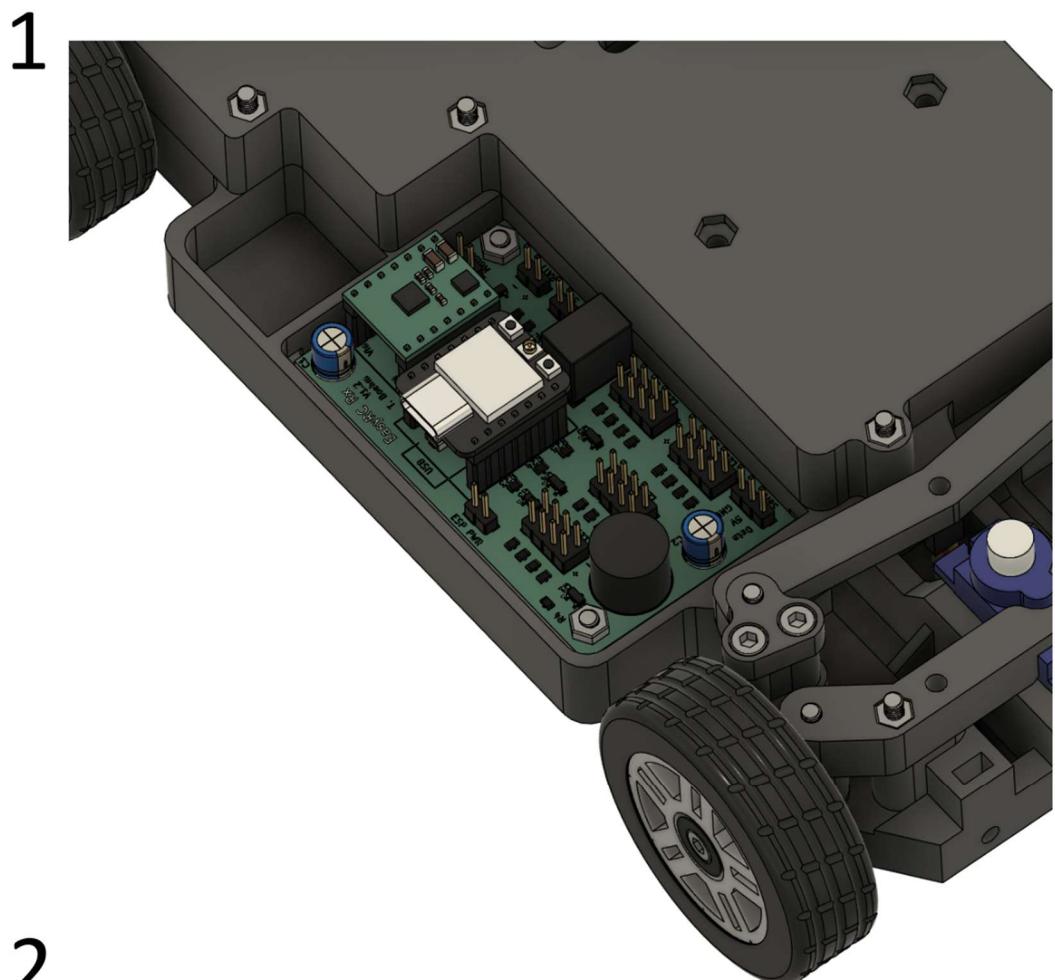
Den Extremfall stellen F1-Autos dar: Sie nutzen sogar negative Ackermann-Lenkgeometrie, also stärker eingeschlagene kurvenäußere als kurveninnere Räder, um die Reifentemperatur durch zusätzlichen Abrieb des kurveninneren Rades hochzuhalten. Denn im Hochgeschwindigkeitsbereich ist eine hohe Reifentemperatur notwendig, um beste Traktion zu gewährleisten. Die Nachteile dabei sind ein höherer Reifenabrieb und schlechtere Kurveneigenschaften bei langsamen Fahrten, was für diese Rennfahrzeuge allerdings keine Rolle spielt.



3.4 Finalisierung Chassis

3.4.1 Leiterplatte

Benötigte Bauteile: Vorbereitetes Chassis, EasyRC Rx-Platine, M3x12 mm Schraube (2x), M3 Mutter (2x)

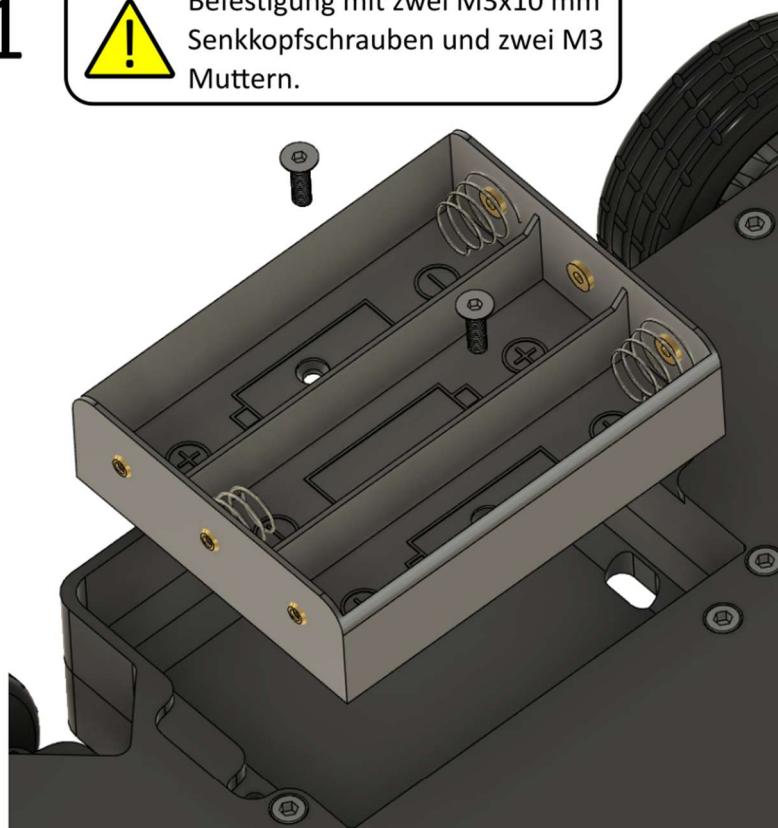


3.4.2 Batteriefach

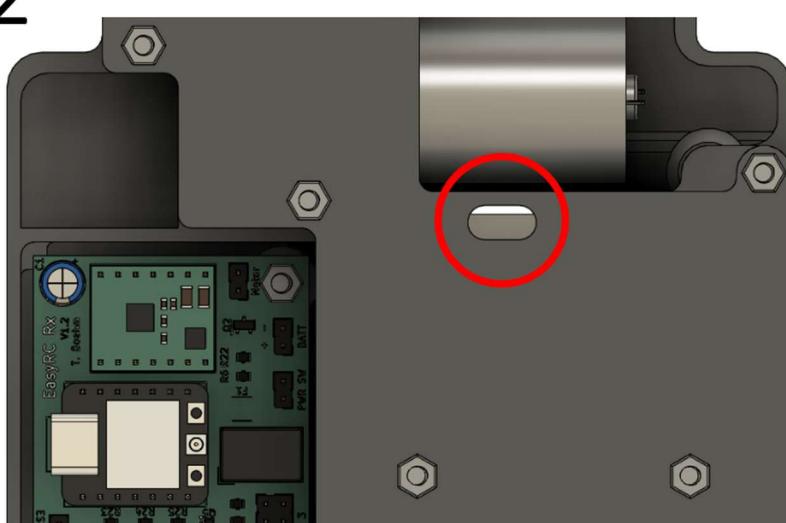
Benötigte Bauteile: Vorbereitetes Chassis, Batteriefach-Abdeckung, Batteriehalter für drei 18650 Lilon-Zellen, M3x12 mm Schraube (2x), M3x10 mm Senkkopfschraube (2x), M3 Mutter (2x)

1

 Befestigung mit zwei M3x10 mm Senkkopfschrauben und zwei M3 Muttern.



2



 Das Anschlusskabel muss vor dem Verschrauben durch die markierte Öffnung geführt werden.

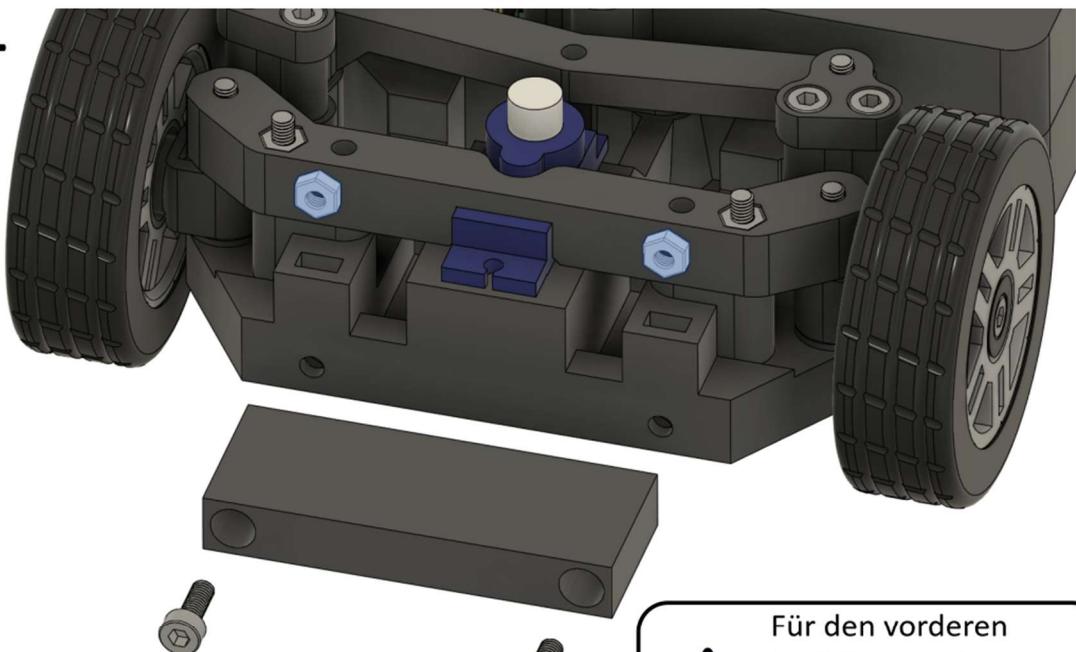
3



3.4.3 Stoßfänger

Benötigte Bauteile: Vorbereitetes Chassis, Chassis-Stoßfänger vorn, Chassis-Stoßfänger hinten, An/Aus-Schalter, M3x8 mm Schraube (4x), M3 Mutter (2x)

1



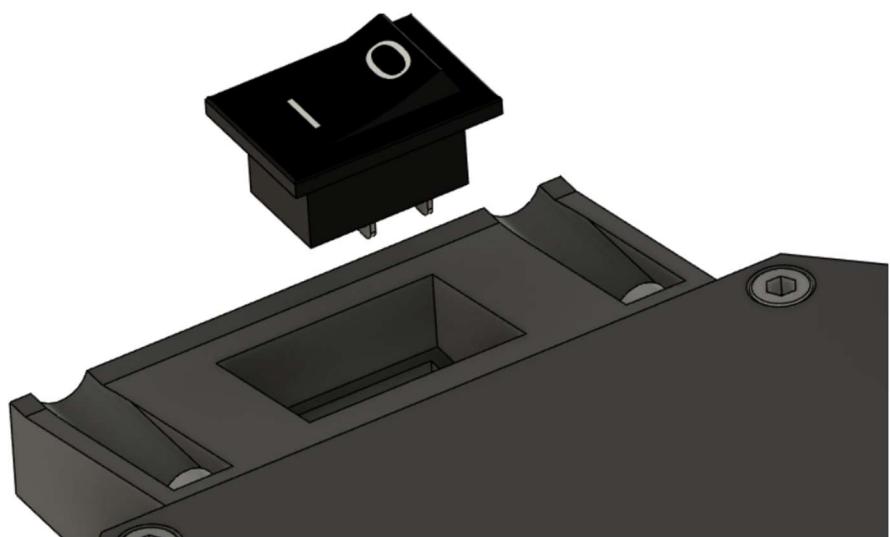
Für den vorderen
Stoßfänger müssen M3
Muttern in das Chassis
eingesetzt werden.

2



Die Aussparung für
die Schrauben muss
zum Fahrzeughoden
ausgerichtet sein!

3



4



Der An/Aus-Schalter wird in den hinteren
Stoßfänger eingesetzt (mit leichtem Druck,
der Schalter schnappt in seine Position).

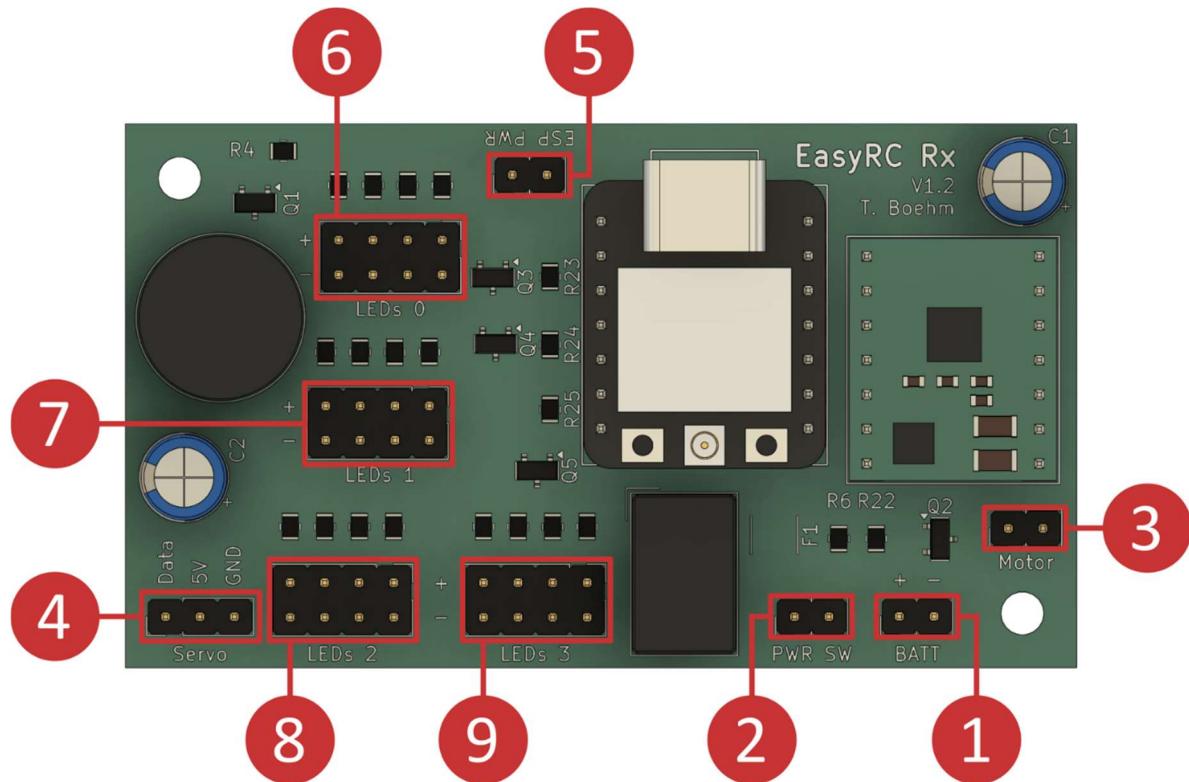


3.4.4 Verkabelung

Die Verkabelung des Chassis beinhaltet die Markierungen 1 bis 4: Batteriefach, An/Aus-Schalter, Antriebsmotor und Lenkung können bereits verkabelt werden. Unbedingt auf korrekte Polarität achten (siehe Tabelle)!

Die Kurzschlussbrücke (Markierung 5) muss vor der Programmierung des Fahrzeugs entfernt werden! Wird der Mikrocontroller programmiert, während die Kurzschlussbrücke aufgesetzt ist, kann es durch einen zu hohen Stromfluss über den USB-Anschluss zur Platine zu Defekten am PC kommen.

Die LEDs (Markierungen 6 bis 9) werden in Kapitel 5 beim finalen Zusammenbau des Fahrzeugs angeschlossen. Es können bis zu vier LEDs je Kanal angeschlossen werden. Im Auslieferungszustand des Fahrzeugs werden zwei LEDs je Kanal eingesetzt. Diese können an beliebigen Positionen der vier Steckplätze angeschlossen werden.



Zahl	Bezeichnung	Polarität
1	BATT (Batterieanschluss)	Rot zu +; Schwarz zu -
2	PWR SW (An/Aus-Schalter)	Egal
3	Motor (Antriebsmotor)	Egal (Drehrichtung wird per Software eingestellt)
4	Servo (Lenkungsmotor)	Gelb zu Data; Rot zu 5V; Schwarz zu GND
5	ESP PWR (Kurzschlussbrücke)	Egal
6	LEDs 0 (Frontscheinwerfer)*	Rot zu +; Schwarz zu -
7	LEDs 1 (Bremslichter)*	Rot zu +; Schwarz zu -
8	LEDs 2 (Blinker links)*	Rot zu +; Schwarz zu -
9	LEDs 3 (Blinker rechts)*	Rot zu +; Schwarz zu -

*Die LED-Anschlussleisten bestehen aus vier Anschlusspaaren: Es können bis zu vier LEDs je Kanal angeschlossen werden. In dieser Ansicht sind die oberen Kontakte + und die unteren Kontakte -. Je zwei LED pro Kanal sind im Kit enthalten.

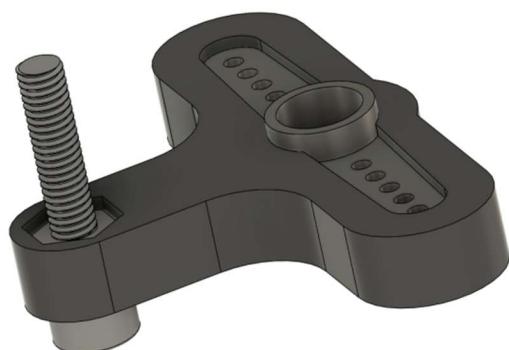
3.4.5 Vorbereitung Servohorn

Benötigte Bauteile: Servohorn-Aufsatz, Servohorn, M3x16 mm Schraube, M3 Mutter

1



2



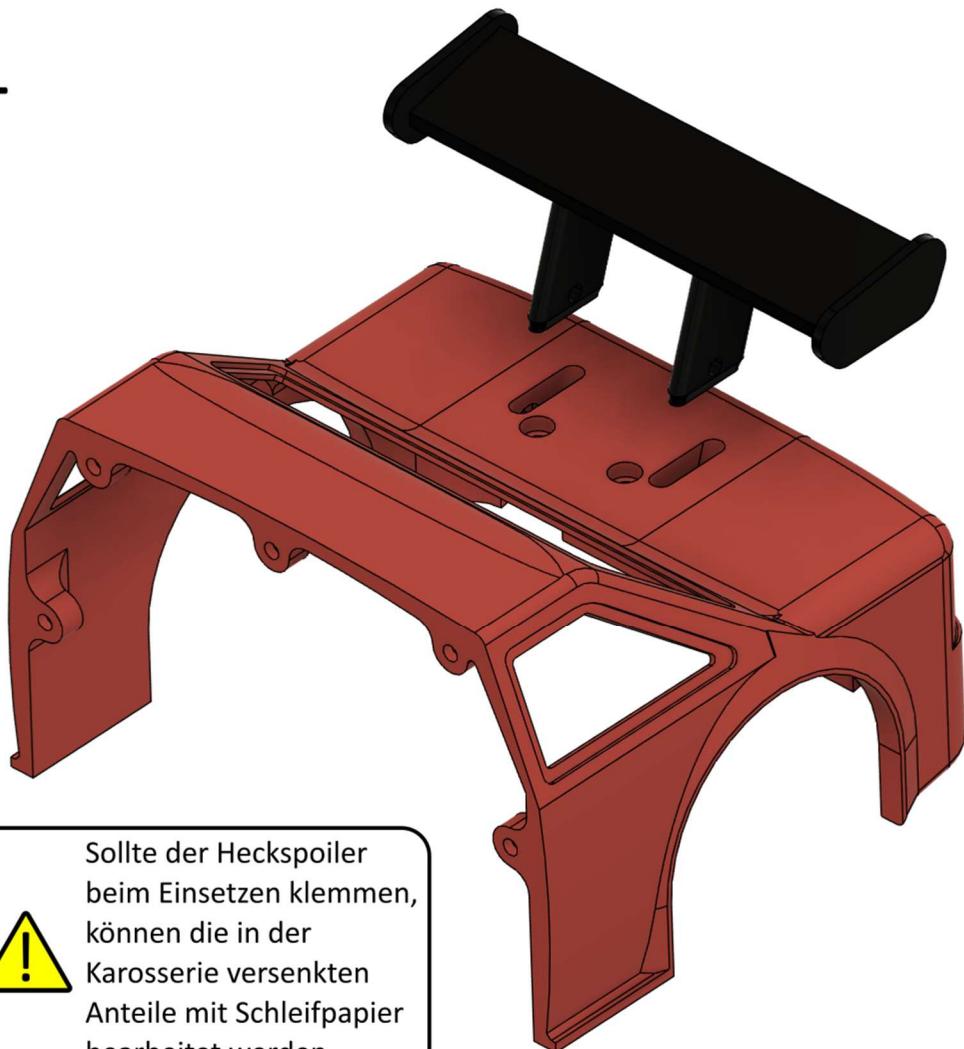
Das Servohorn wird erst nach der
Programmierung des Fahrzeugs
auf den Servomotor gesetzt.

3.5 Karosserie

3.5.1 Heckspoiler

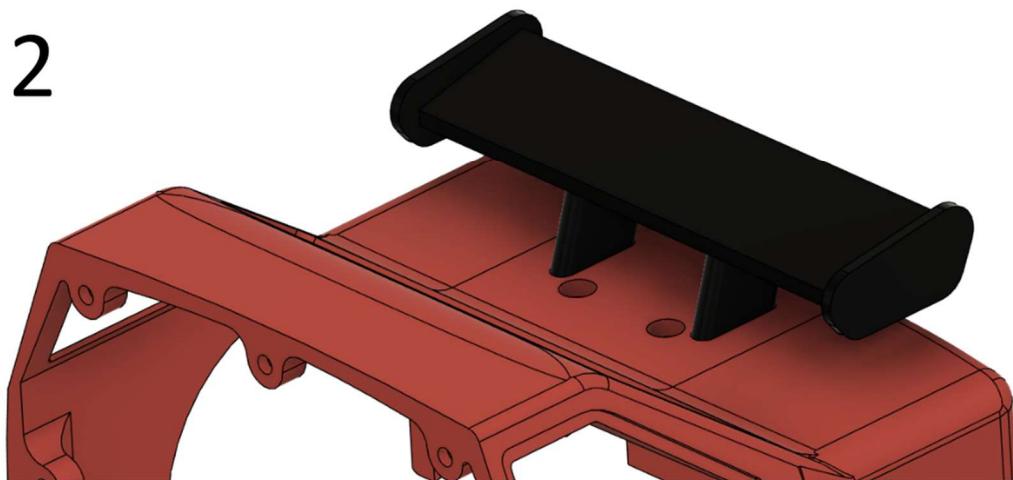
Benötigte Bauteile: Karosserie, Heck, Heckspoiler, Heckspoiler-Halterung, M3x8 mm Schraube (2x), M3x16 mm Schraube (2x), M3 Mutter (2x)

1

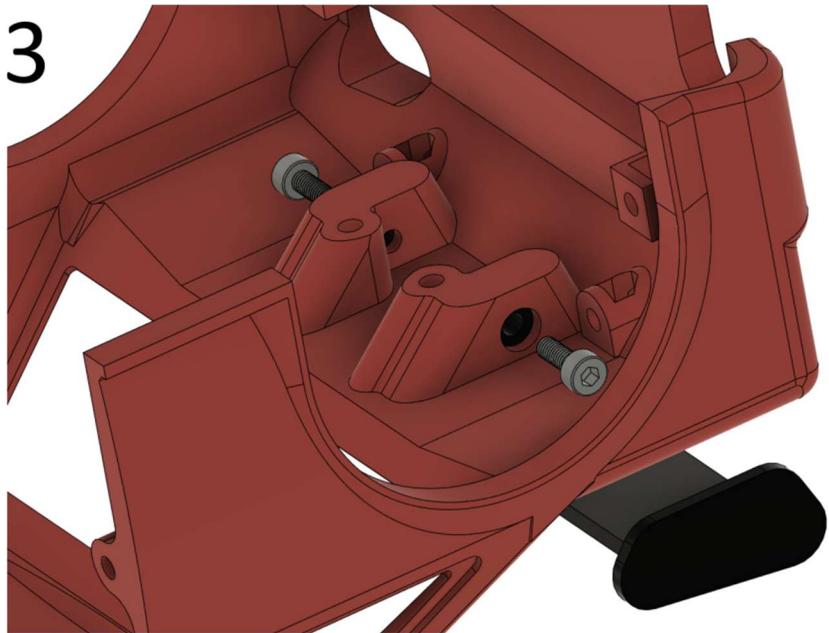


Sollte der Heckspoiler
beim Einsetzen klemmen,
können die in der
Karosserie versenkten
Anteile mit Schleifpapier
bearbeitet werden.

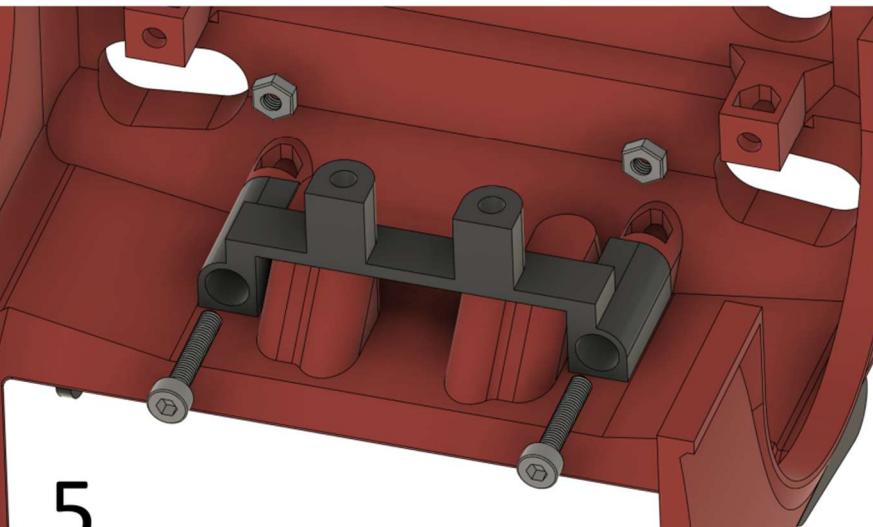
2



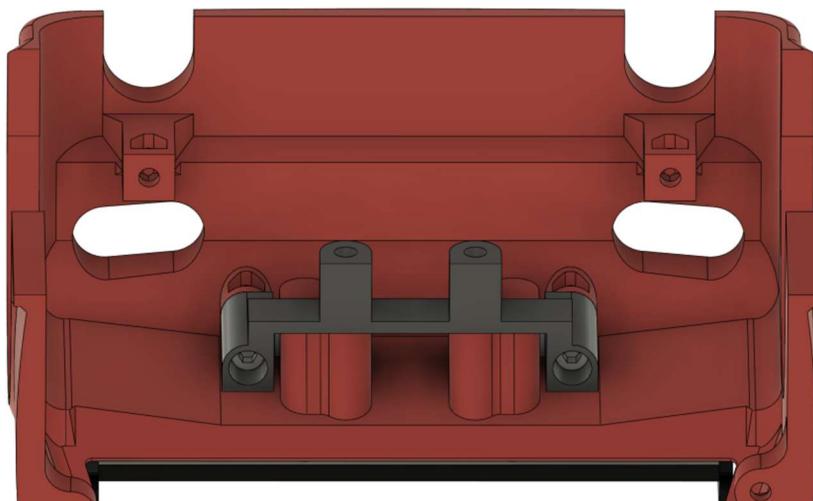
3



4



5



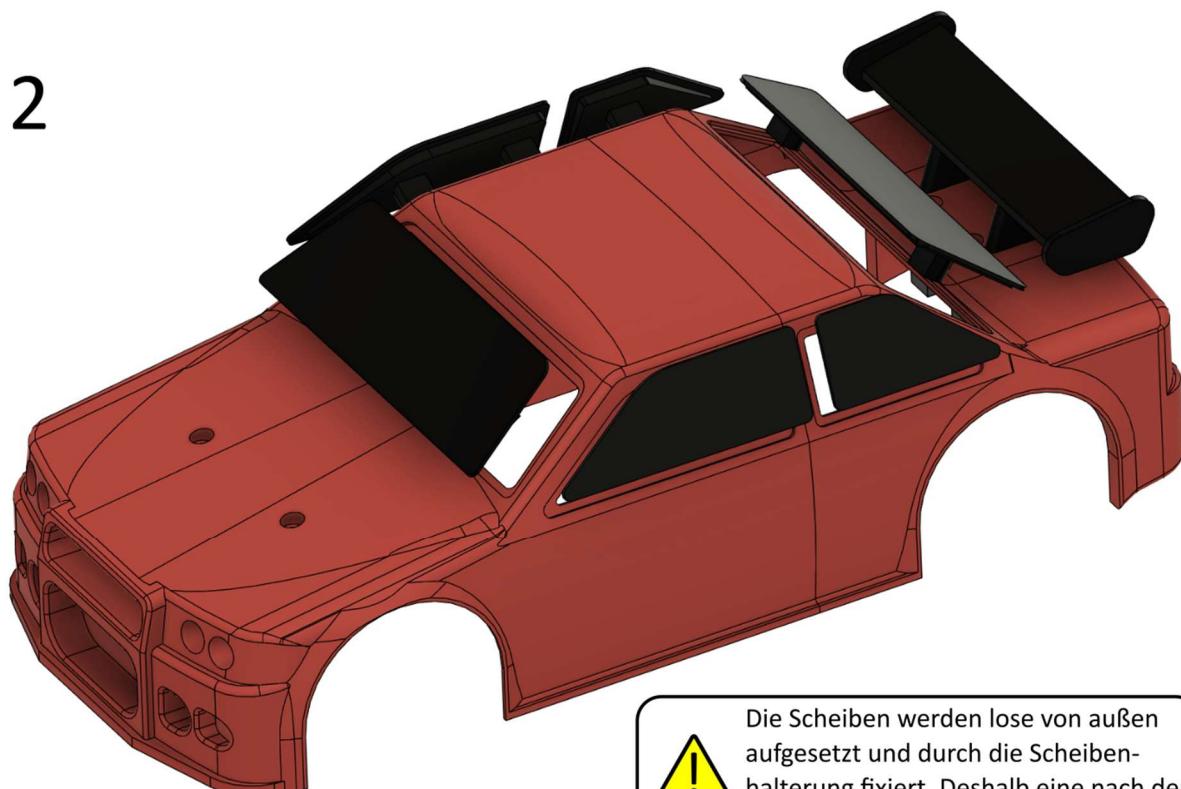
3.5.2 Karosserie-Verbindung und Scheiben

Benötigte Bauteile: Karosserie-Heck (vorbereitet), Karosserie-Front, Scheiben (1x Front, 2x links, 2x rechts, 1x Heck), Scheibenhalterung, M3x8 mm Schraube (15x), M3 Mutter (15x)

1

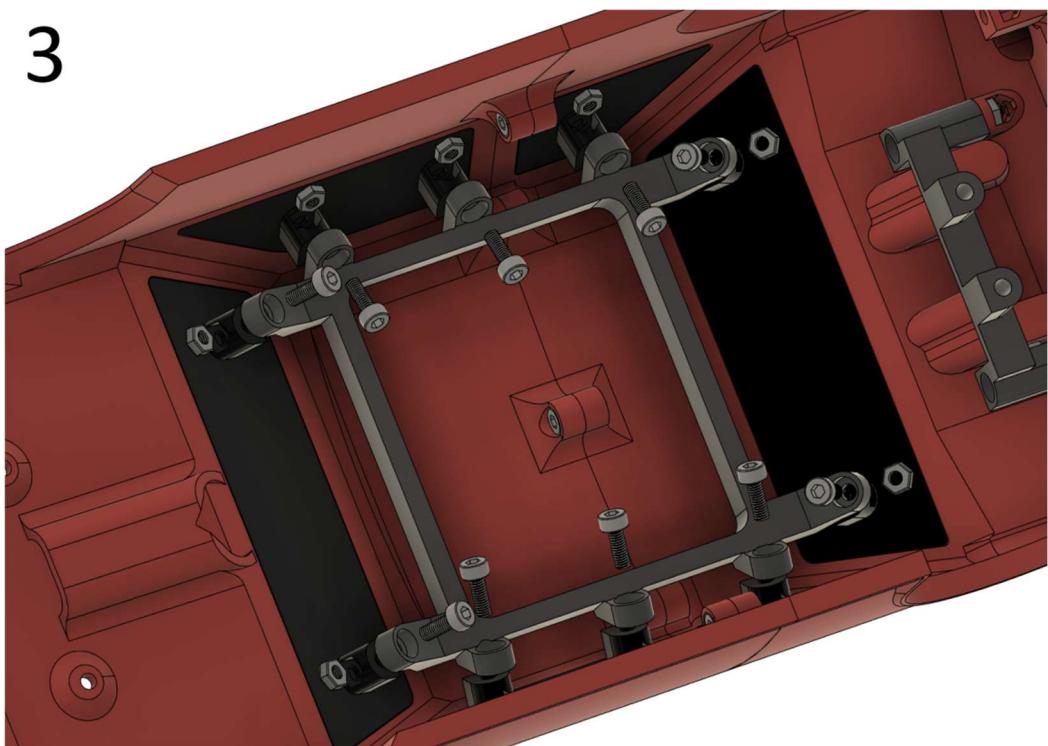


2

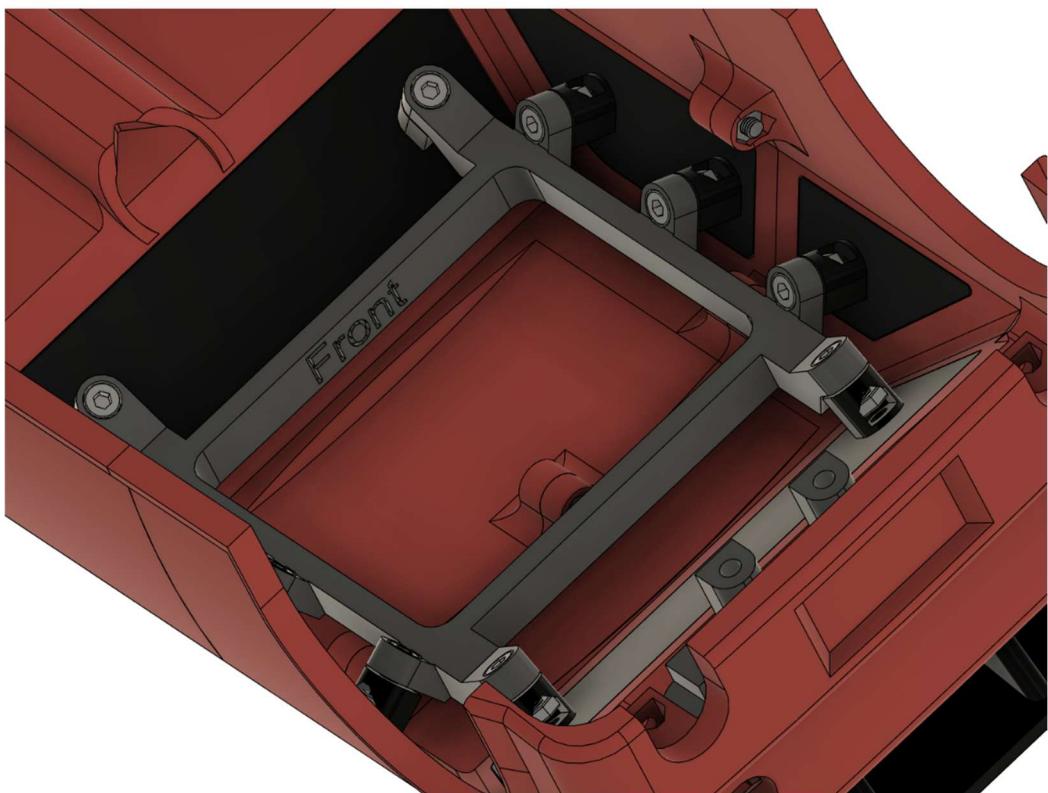


Achtung! Die Scheiben werden lose von außen aufgesetzt und durch die Scheibenhalterung fixiert. Deshalb eine nach der anderen einsetzen und verschrauben.

3



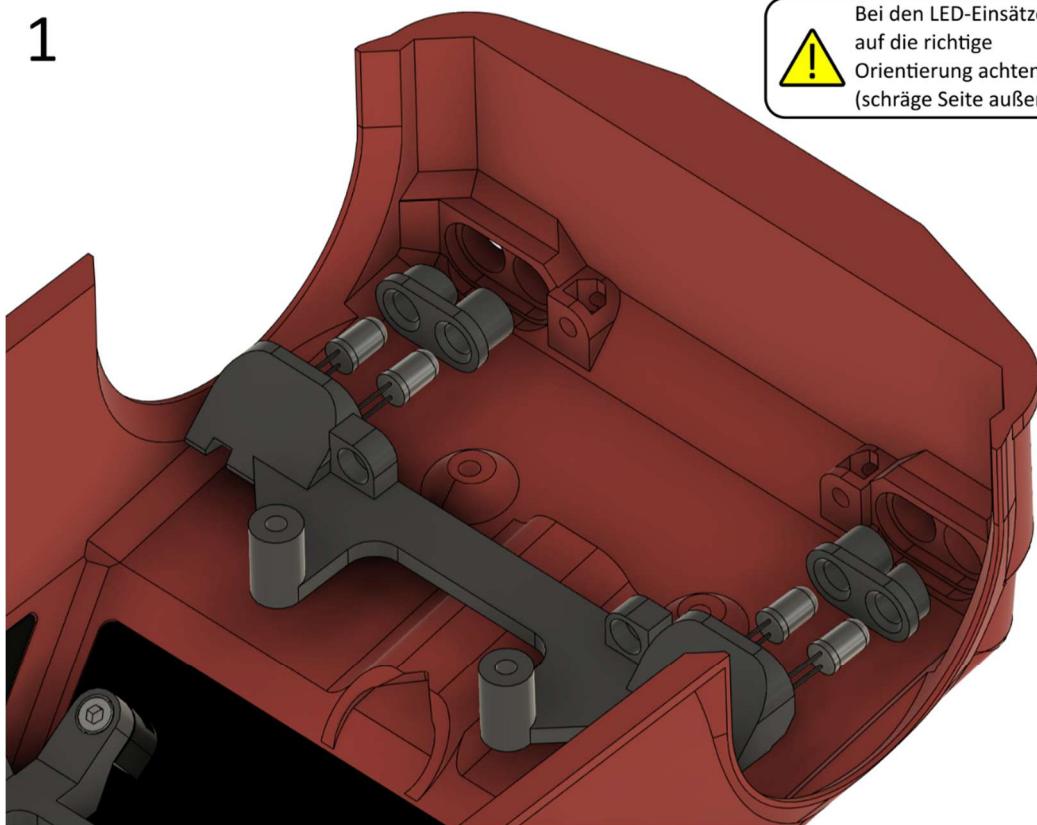
4



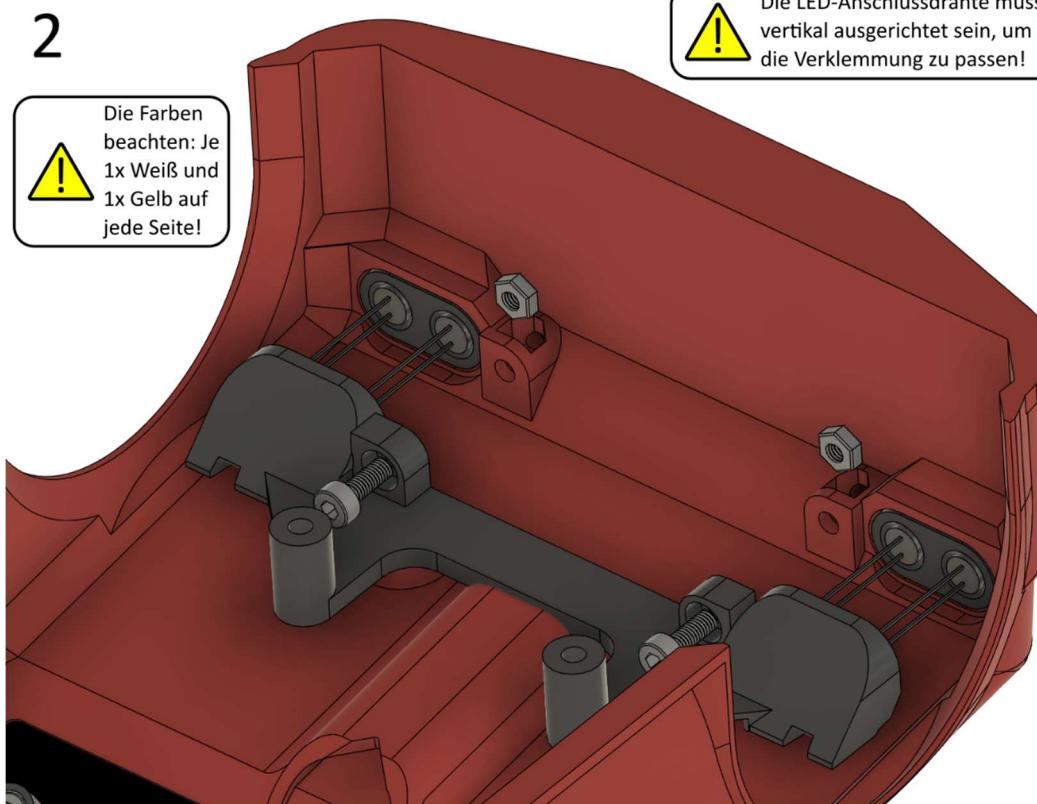
3.5.3 Frontscheinwerfer

Benötigte Bauteile: Karosserie (vorbereitet), Frontlicht-Verklemmung, LED weiß (2x), LED gelb (2x), M3x8 mm Schraube (2x), M3 Mutter (2x)

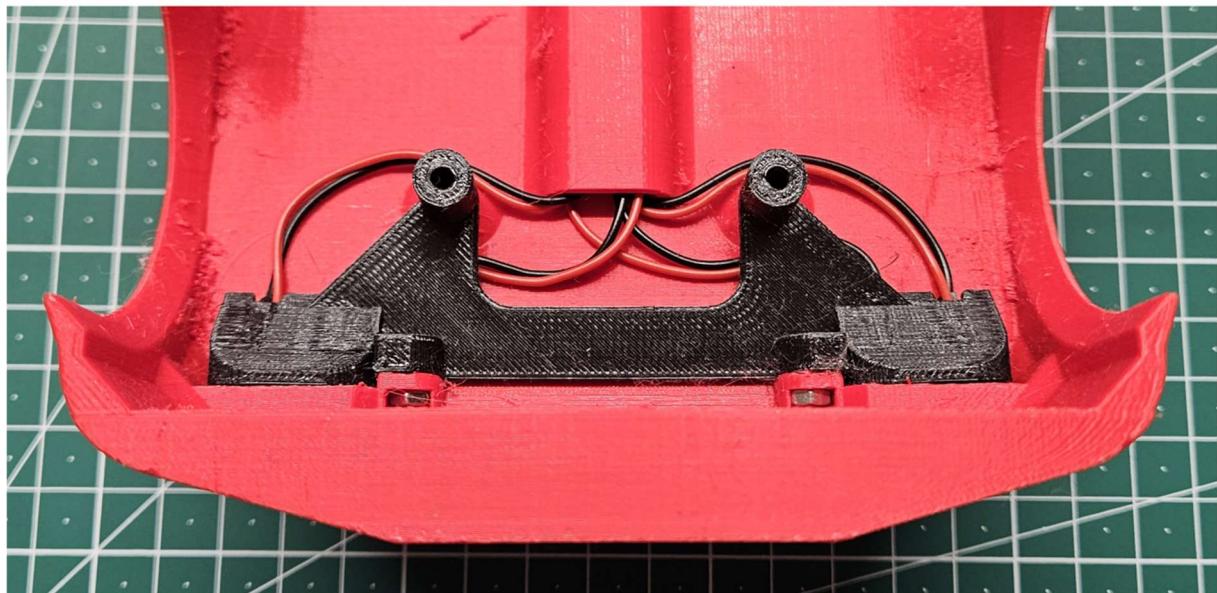
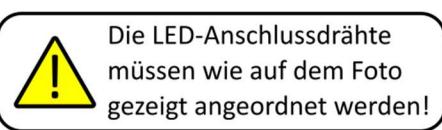
1



2



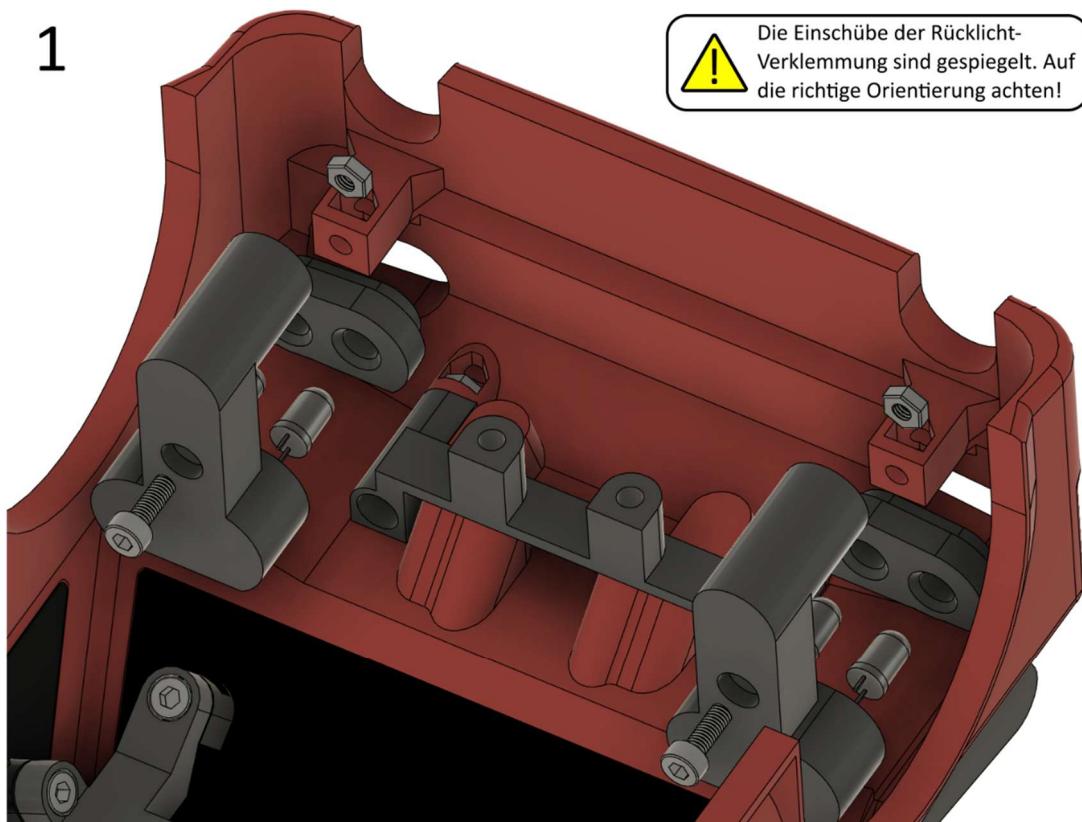
3



3.5.4 Heckscheinwerfer

Benötigte Bauteile: Karosserie (vorbereitet), Rücklicht-Verklemmung (2x), LED rot (2x), LED gelb (2x), M3x8 mm Schraube (2x), M3 Mutter (2x)

1



Die Einschübe der Rücklicht-Verklemmung sind gespiegelt. Auf die richtige Orientierung achten!

2

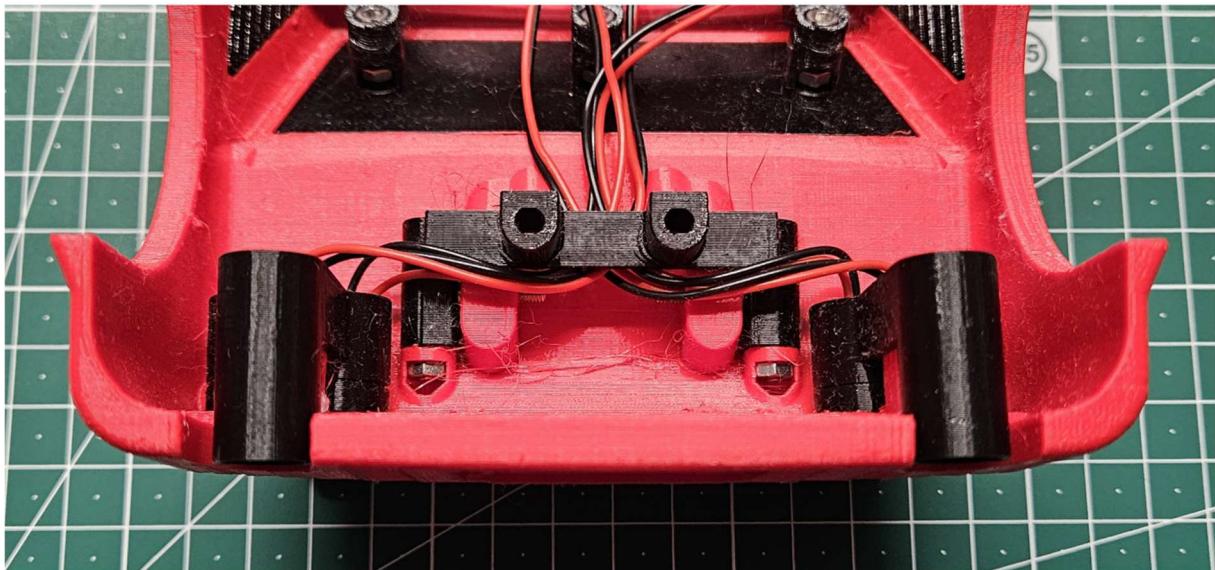
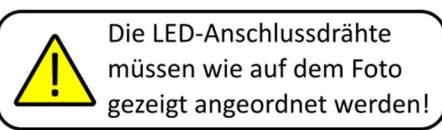


Die Muttern müssen vor dem
Aufsetzen der Auspuff-Attrappen
in die Karosserie gesetzt werden!

Die LED-Anschlussdrähte müssen
vertikal ausgerichtet sein, um in
die Verklemmung zu passen!

Die Farben
beachten: Je
1x Rot und
1x Gelb auf
jede Seite!

3



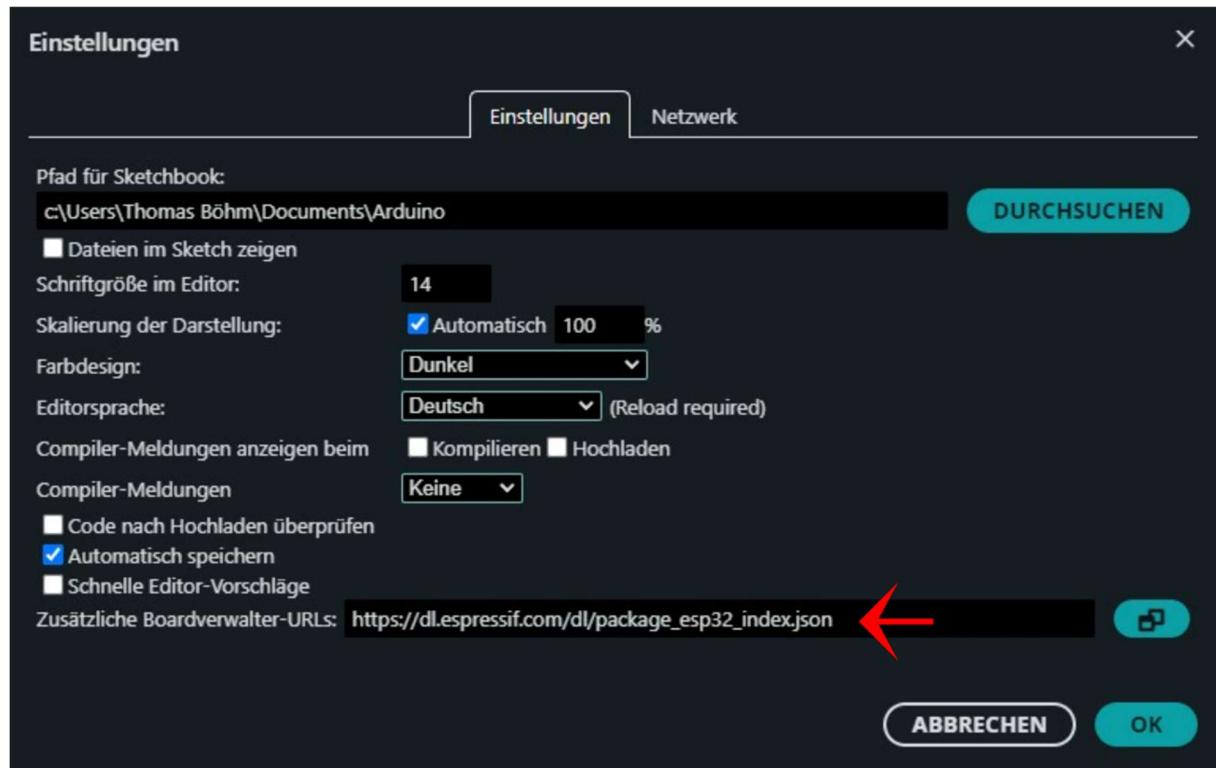
4. Programmierung

4.1 Installation Arduino IDE

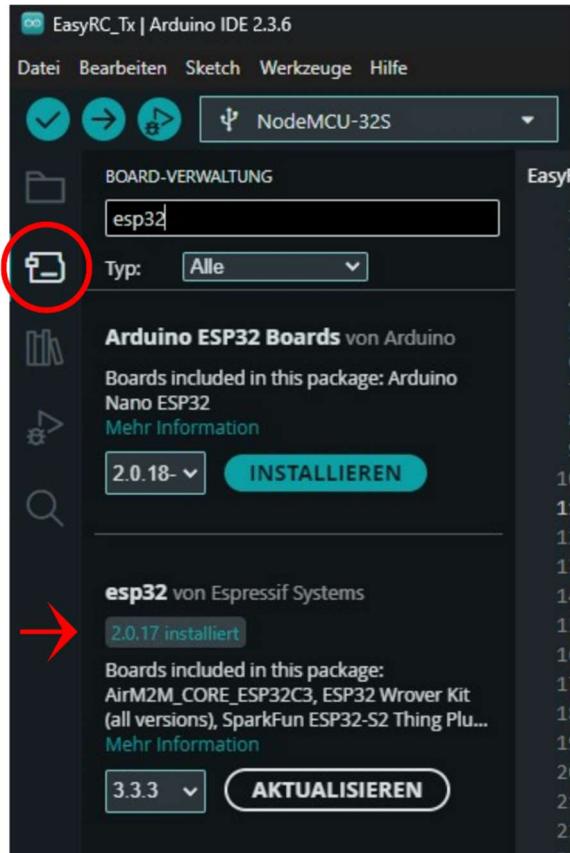
EasyRC verwendet Mikrocontroller aus der ESP32-Familie der Firma Espressif Systems. Diese Plattform kann frei über die Arduino-Entwicklungsumgebung programmiert werden. Für die Programmierung wird deshalb die Software Arduino IDE (getestet mit Version 2.3.6) verwendet:

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

Nach der Installation der Software muss sie für die Verwendung der ESP32-Mikrocontroller eingerichtet werden. Unter Datei -> Einstellungen muss unter „Zusätzliche Boardverwalter-URLs“ die Adresse https://dl.espressif.com/dl/package_esp32_index.json eingegeben werden.



Als nächstes muss der ESP32-Boardverwalter installiert werden. Dazu wird in der Symbolspalte am linken Rand des Anwendungsfensters das zweitoberste Symbol angeklickt. Dadurch öffnet sich die Board-Verwaltung als Sidebar. Hier muss „esp32“ eingegeben werden und im Anschluss „esp32 von Espressif Systems“ ausgewählt und installiert werden. Dabei muss die Version 2.0.17 gewählt werden (nicht die aktuelle Version!). Der Hintergrund dafür ist, dass bestimmte Programmbibliotheken im Generationswechsel von V2 auf V3 eine andere Syntax verwenden. V3 ist aber nicht so stabil wie V2, weshalb EasyRC für die ältere Version programmiert worden ist.



Um eine Code-Datei in Arduino IDE zu öffnen, muss sie in einem Ordner abgelegt sein, der denselben Namen wie die Datei trägt (ohne die .ino-Endung der Datei). Der Standard-Pfad für Arduino IDE ist in Windows Dokumente -> Arduino. Sofern keine Änderungen daran vorgenommen werden sollen, können einfach in diesem Ordner die zwei notwendigen Ordner für die EasyRC-Codes der Fernsteuerung und des Codes erstellt werden und anschließend die eigentlichen Codes dort abgelegt werden. Die Codes sind frei zum Download verfügbar unter <https://github.com/TRD-B/EasyRC>.

Dokumente > Arduino >			
	Sortieren	Anzeigen	...
Name	Änderungsdatum	Typ	Größe
EasyRC_Rx	02.09.2025 21:33	Dateiordner	
EasyRC_Tx	09.09.2025 20:26	Dateiordner	

Dokumente > Arduino > EasyRC_Rx			
	Sortieren	Anzeigen	...
Name	Änderungsdatum	Typ	Größe
EasyRC_Rx.ino	11.10.2025 08:37	INO-Datei	12 KB

4.2 Mögliche und benötigte Code-Modifikationen

EasyRC ist so konzipiert, dass der Code des Empfängers (des Modelfahrzeugs) (EasyRC_Rx.ino) unverändert auf den Mikrocontroller hochgeladen werden kann, sofern keine Änderung der Geräte-Adresse (MAC-Adresse) notwendig ist. Die Geräte-Adresse muss angepasst werden, wenn mehr als ein EasyRC-System am gleichen Ort betrieben werden soll. Die Geräte-Adresse legt fest, welche Fernsteuerung welches Fahrzeug steuert (siehe dazu Kapitel 7.1). Der Code des Senders (der Fernsteuerung) (EasyRC_Tx.ino) beinhaltet einige Variablen, die modifiziert werden müssen, um das Fahrzeug zu kalibrieren. Diese sind unter dem Abschnitt „//calibration data“ im Code zu finden:

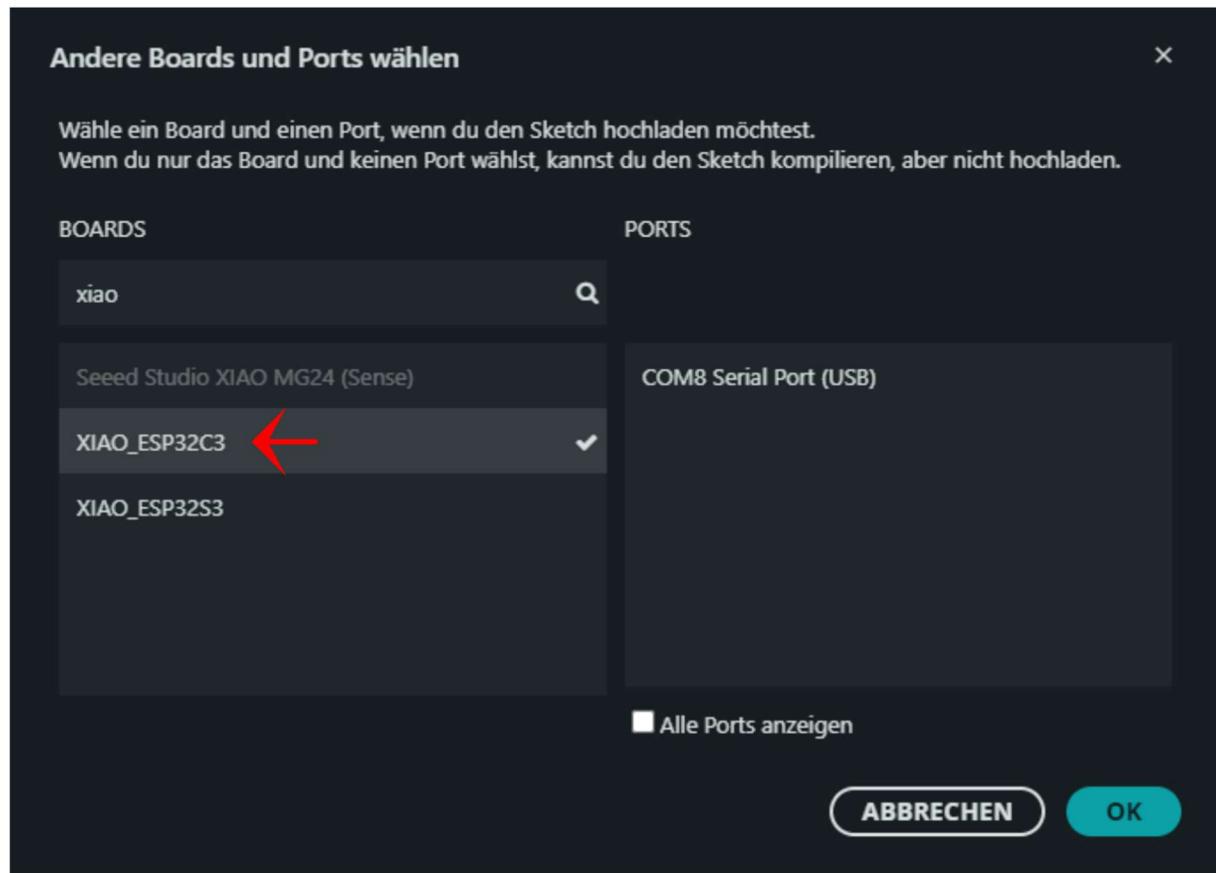
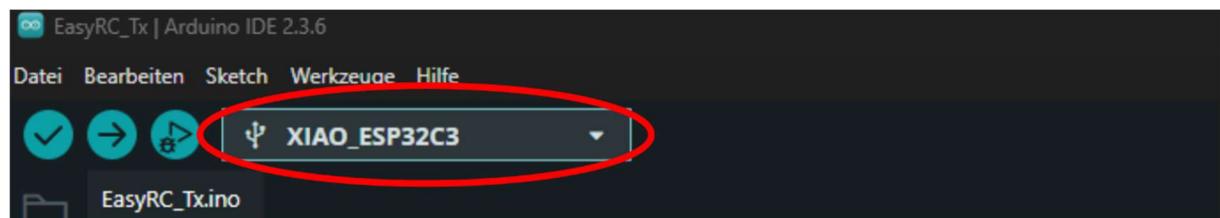
```
EasyRC_Tx.ino
1 //Libraries
2 #include <WiFi.h>
3 #include <esp_now.h> //library for bidirectional wireless communication between two ESP32
4 #include <esp_wifi.h> //needed to adjust the MAC address
5 #include <Wire.h>
6
7 //Pin definitions
8 #define vsense 2
9 #define joyx 3
10 #define joyy 4
11 #define button 5
12 #define led 20
13
14 //calibration data
15 bool motordirection = false; //change this variable to false to invert the motor spin direction
16 int motorint = 10; //sets the delay time between motor speed changes (in ms). Higher values result in
//better traction but slower responses. Values between 5 and 10 are recommended.
17 int speedlimit = 127; //this variable sets the maximum speed of the car (values between 0 and 255)
18 int leftlimit = 358; //steering limit calculation: leftlimit = x/20ms*2^12 (default: 205 (x = 1ms))
19 int rightlimit = 258; //steering limit calculation: rightlimit = x/20ms*2^12 (default: 410 (x = 2ms))
20 //RC servos run at 50 Hz (20 ms intervals), with the center at 1.5 ms and a range of +0.5 ms around the center
21 //starting values should be 205 to 410 (left to right or right to left, depends on the servo)
22 //adjust these values to a smaller or larger interval depending on the actual steering angles
23 //individually decrease/increase one of the two values to trim the center of the steering
24
25 //MAC addresses
26 uint8_t CarMAC[] = {0xAA, 0xA0, 0xBE, 0xEF, 0xFE, 0xED}; //we will set this MAC address for the car
27 uint8_t RemoteMAC[] = {0xB0, 0xAD, 0xBE, 0xEF, 0xFE, 0xEF}; //we will set this MAC address for the remote control
28 uint8_t broadMAC[] = {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF}; //using this MAC address as a receiver for sending
//packages means broadcasting (ignoring if the data has been received)
29
30
```

Zeile	Code	Mögliche Werte	Erklärung
15	bool motordirection = false;	true, false	Diese Variable kehrt die Drehrichtung des Motors um. Sollte das Fahrzeug beim Bewegen des Joysticks nach vorn rückwärtsfahren, kann hier true gegen false (oder umgekehrt) ausgetauscht werden.
16	int motorint = 10;	1 - 20	Diese Variable verändert, wie schnell das Fahrzeug Änderungen der Geschwindigkeit basierend auf der Joystickposition umsetzt. Ein niedrigerer Wert sorgt für eine höhere Reaktionsfreudigkeit. Ein höherer Wert verhindert ein Durchdrehen der Reifen, lässt das Fahrzeug jedoch trüger auf Brems- und Beschleunigungseingaben reagieren.

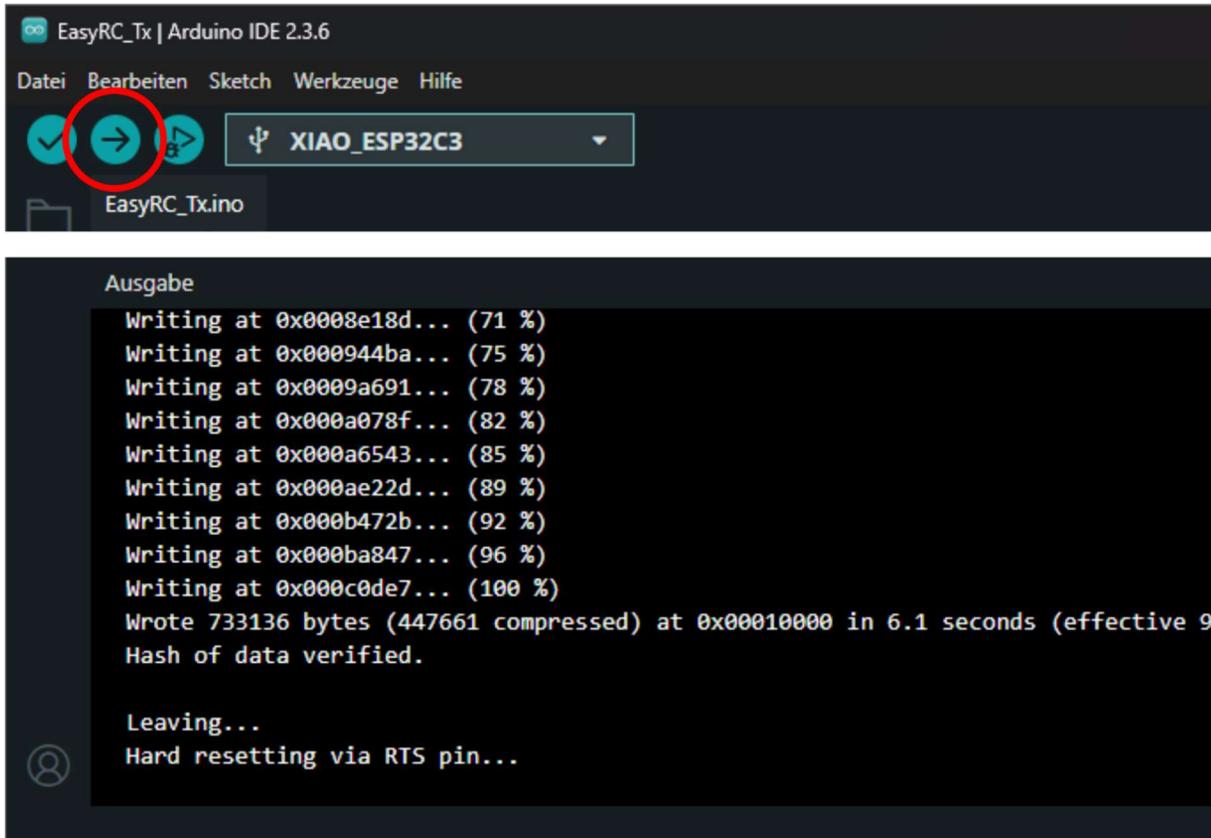
18	int speedlimit = 127;	51 - 255	Diese Variable limitiert die maximale Geschwindigkeit des Fahrzeugs. 255 stellt die Maximalgeschwindigkeit dar. Werte kleiner als 51 (20% der Maximalgeschwindigkeit) sind zwar möglich, aber nicht zu empfehlen, da der Motor so nicht genug Kraft hat, um das Fahrzeug tatsächlich zu bewegen.
19	int leftlimit = 358;	308 - 410	Diese Variable definiert den maximalen Lenkungswinkel des Fahrzeugs nach links. Der Wert ergibt sich aus der Ansteuerung des Servomotors. Ein Wert von 308 entspricht der neutralen Position (Zentrum) des Servomotors. Beim mitgelieferten Servomotor entspricht ein höherer Wert einem Ausschlag der Lenkung nach links; das kann je nach Servomotor-Typ allerdings auch umgekehrt sein. Ein Wert von 410 entspricht dem theoretischen Maximalausschlag des Servomotors. Es sollte nicht blind der Maximalwert eingetragen werden, sondern stufenweise der maximale Ausschlag der Lenkung ermittelt werden. Der maximale Lenkwinkel ist der Winkel, bei dem die Räder noch nirgendwo an der Karosserie streifen.
20	int rightlimit = 258;	205 - 308	Diese Variable definiert den maximalen Lenkungswinkel des Fahrzeugs nach rechts (analog zum Lenkungswinkel nach links). Die neutrale Position der Lenkung ergibt sich aus dem Mittelwert des linken und rechten Lenkungswinkels. Das bedeutet, dass bei der Erhöhung eines Werts (größer als 308) der andere entsprechend verringert werden muss (kleiner als 308), um den Geradeauslauf des Fahrzeugs nicht zu verändern. Das theoretische Intervall zwischen 205 und 410 für die Servo-Ansteuerung kann je nach Servomotor abweichen. Deshalb können auch etwas größere bzw. kleinere Werte funktionieren.

4.3 Programmieren der Mikrocontroller

Zum Hochladen eines Codes auf einen Mikrocontroller muss dieser per USB an den PC angeschlossen werden. **Wenn der Mikrocontroller des Fahrzeugs programmiert werden soll, muss dafür der „ESP PWR“ Jumper vorher abgezogen werden (andernfalls kann der USB-Anschluss des PCs beschädigt werden!).** Die Fernsteuerung kann direkt über den von außen zugänglichen USB-Anschluss programmiert werden. Nachdem der Mikrocontroller mit dem PC verbunden worden ist, muss in Arduino IDE das richtige Modell und der richtige USB-Anschluss ausgewählt werden. Dazu muss in der oberen Leiste in das Feld mit dem USB-Verbindungssymbol geklickt werden. Daraufhin öffnet sich ein Dropdown-Menü, in dem auf „Wähle ein anderes Board und einen anderen Port ...“ geklickt werden muss. Um den passenden Mikrocontroller zu finden, kann in der linken Spalte nach „Xiao“ gesucht werden. Das richtige Modell ist der „XIAO_ESP32C3“. In der rechten Spalte werden angeschlossene USB-Komponenten angezeigt. Die Nummer des Ports (COM8 im Bild) variiert je nach verwendetem USB-Anschluss des Computers. Sollte hier mehr als ein Port angezeigt werden, obwohl nur ein Mikrocontroller angeschlossen ist, hilft es, nicht benötigte USB-Geräte vom PC abzustecken und erneut nach aktiven Ports zu suchen.



Sobald das Mikrocontroller-Modell und der Port ausgewählt sind, kann der Code hochgeladen werden. Dazu wird in der oberen Leiste in Arduino IDE auf den nach rechts zeigenden Pfeil geklickt. Der Computer prüft daraufhin den Code auf Syntaxfehler, kompiliert ihn (Umwandlung in maschinentauglichen Code) und überträgt ihn via USB auf den Mikrocontroller.



The screenshot shows the Arduino IDE interface. At the top, the title bar reads "EasyRC_Tx | Arduino IDE 2.3.6". Below it is a menu bar with "Datei", "Bearbeiten", "Sketch", "Werkzeuge", and "Hilfe". To the right of the menu is a dropdown menu showing "XIAO_ESP32C3". On the left, there are icons for file operations: a checkmark, a blue arrow pointing right (circled in red), a green arrow pointing right, and a refresh symbol. The main area displays the code "EasyRC_Tx.ino". Below the code, the "Ausgabe" (Output) window is open, showing the progress of the upload:

```
Writing at 0x0008e18d... (71 %)
Writing at 0x000944ba... (75 %)
Writing at 0x0009a691... (78 %)
Writing at 0x000a078f... (82 %)
Writing at 0x000a6543... (85 %)
Writing at 0x000ae22d... (89 %)
Writing at 0x000b472b... (92 %)
Writing at 0x000ba847... (96 %)
Writing at 0x000c0de7... (100 %)
Wrote 733136 bytes (447661 compressed) at 0x00010000 in 6.1 seconds (effective 9
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

Wenn der Prozess erfolgreich ist, zeigt das Ausgabefenster schließlich den in der Abbildung gezeigten Text („Leaving... Hard resetting via RTS pin...“) an. Die USB-Übertragung ist allerdings fehleranfällig: Sollte die Übertragung abbrechen oder mit einer Fehlermeldung gar nicht erst starten, sollte der Mikrocontroller ab- und wieder angesteckt werden sowie geprüft werden, ob der richtige USB-Port ausgewählt worden ist. Besteht der Fehler nach mehreren Versuchen weiterhin, kann auch ein Neustart des PCs Abhilfe schaffen.

5. Erste Schritte und Bedienung

Wenn die Mikrocontroller der Fernbedienung und des Fahrzeugs programmiert worden sind, kann es schon fast losgehen! Zunächst muss aber noch die Lenkung des Fahrzeugs getrimmt werden und die Karosserie aufgesetzt werden.

5.1 Batterien einsetzen und Geräte anschalten

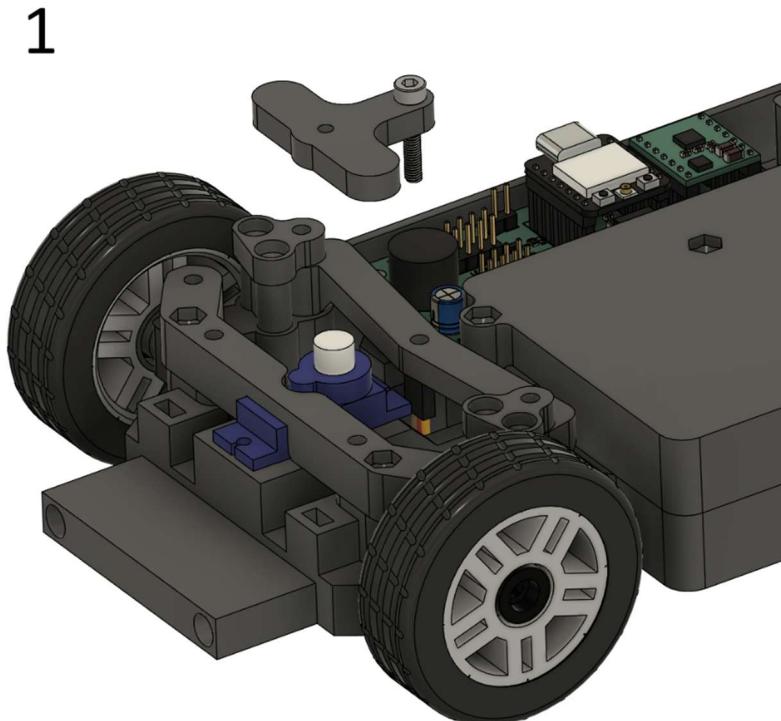
Als erstes werden Batterien in das Fahrzeug und die Fernbedienung eingesetzt. Die EasyRC-Platinen sind zwar vor Verpolung der Eingangsspannung geschützt, aber dennoch sollte man es nie auf solche Sicherungen ankommen lassen: Beim Einbau bitte immer zweimal prüfen, ob die Batterien richtigerum eingesetzt wurden. Anschließend kann die Fernbedienung angeschaltet werden. Die Status-LED der Fernbedienung leuchtet durchgehend, wenn sie ordnungsgemäß funktioniert. Sollte die LED blinken, ist die Batteriespannung niedrig. Dann muss die Batterie zunächst aufgeladen werden, bevor das Aufsetzen des Fahrzeugs weitergeht. Das ist wichtig: Lilon-Batterien reagieren empfindlich auf Tiefentladung. Sollte die Batteriespannung zu tief fallen, wird die Batterie beschädigt.

Wenn die Fernbedienung angeschaltet wird, kalibriert sich der Joystick automatisch. Die Position, die der Joystick zum Zeitpunkt des Einschaltens innehat, wird als neutrale Position hinterlegt. Deshalb darf der Joystick beim Anschalten der Fernbedienung nicht bewegt werden. Diese Kalibration wird bei jedem Start der Fernbedienung durchgeführt. Sollte man also aus Versehen den Joystick bewegt haben, muss die Fernbedienung einfach nur aus- und wieder angeschaltet werden, um wieder eine korrekte Kalibration zu erhalten. Als zweites wird das Fahrzeug angeschaltet. Die ESP_PWR-Kurzschlussbrücke muss davor installiert sein, denn andernfalls bekommt der Mikrocontroller keinen Strom. Solange die Karosserie noch nicht aufgesetzt ist, gibt es noch kein Feedback zum Status des Fahrzeugs über die LEDs. Das Fahrzeug hat aber einen Piezobuzzer, der nach dem Anschalten dreimal piept (der dritte Ton ist etwas länger und höher als die ersten beiden Töne). Anschließend piept der Buzzer alle zwei Sekunden, sofern das Fahrzeug nicht mit einer Fernsteuerung verbunden ist. Sobald das Fahrzeug „seine“ Fernsteuerung findet, folgt ein Signalsequenz mit einem längeren und einem kürzeren Piepton.

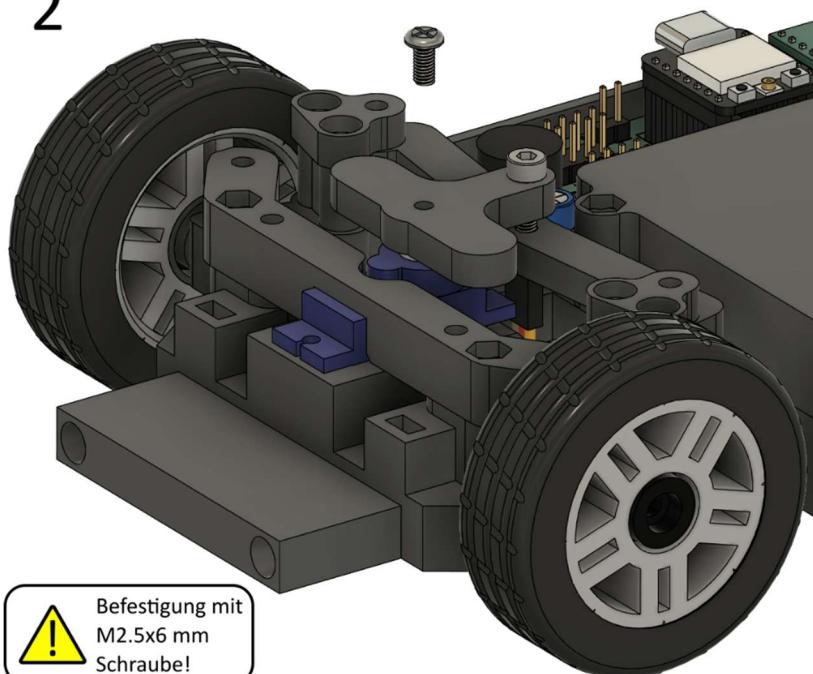
5.2 Lenkung fertig montieren

Zunächst zur Lenkung: Wie im Abschnitt zur Programmierung erklärt worden ist, gibt es im Code zwei Variablen, die den maximalen Lenkausschlag nach links und rechts festlegen. Diese müssen angepasst werden, sodass das Fahrzeug bei neutraler Joystick-Position auch tatsächlich geradeaus fährt und außerdem beim Einschlagen der Lenkung die Vorderräder nicht am Rahmen streifen (der maximale Lenkausschlag darf nicht zu groß sein). Doch ein Schritt nach dem anderen. Als erstes muss das Servohorn auf den Servomotor gesetzt und mit dem Lenkgestänge verbunden werden. Dieser Schritt wird erst jetzt durchgeführt, da zunächst der Servomotor seine neutrale Position einnehmen muss. Der Servomotor fährt automatisch seine neutrale Position an, wenn der Joystick der Fernbedienung nicht bewegt wird und der Mittelwert zwischen linkem und rechtem maximalen Lenkwinkel bei 308 liegt. Das ist die Ausgangslage, wenn der Code der Fernsteuerung unverändert übernommen worden ist.

Nun wird das Servohorn so auf den Servomotor gesetzt, dass die Linie zwischen dem Kontaktpunkt zum Lenkgestänge und der Motorwelle möglichst parallel zur Längsachse des Fahrzeugs ist. Da die Abtriebswelle des Servomotors ein kleines Zahnrad ist und das Servohorn in dieses Zahnrad greifen muss, kann es sein, dass das Servohorn nicht in dieser Idealline aufgesetzt werden kann. Das ist kein Problem; es sollte nur so nahe wie möglich an dieser Idealline ausgerichtet sein. Die bereits vorinstallierte M3x16 mm Schraube des Servohorns muss hierfür gleich in die entsprechende Aussparung des Lenkgestänges geschoben werden. Jetzt muss das Servohorn nur noch auf dem Motor fixiert werden. Damit ist die erste Grundeinstellung der Lenkung abgeschlossen. Die feine Trimmung wird ausschließlich über die Software der Fernsteuerung durchgeführt, nachdem das Fahrzeug mit Karossiere fertig montiert ist.

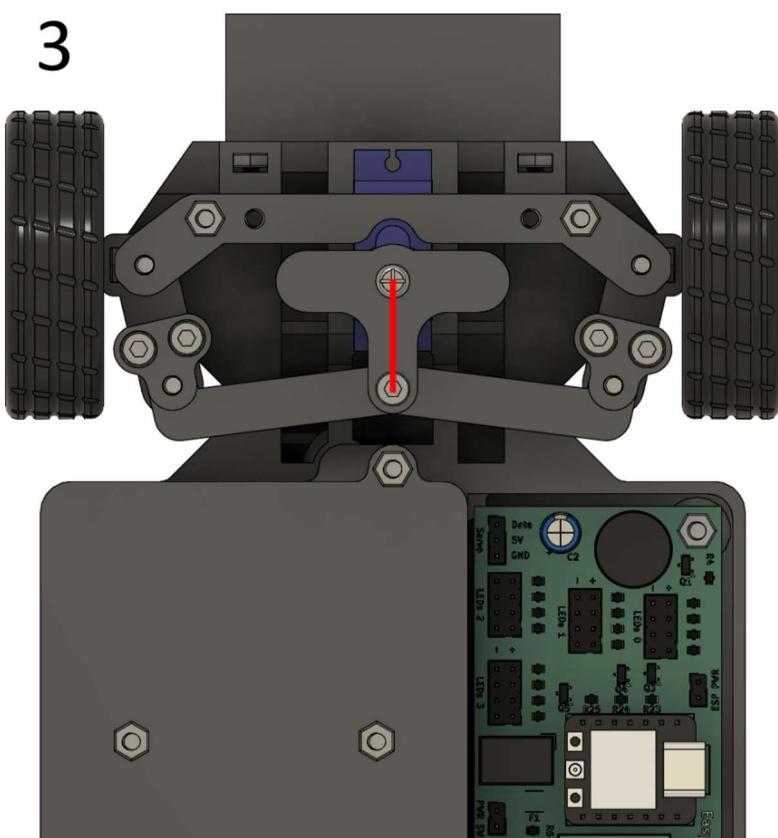


2



 Befestigung mit
M2.5x6 mm
Schraube!

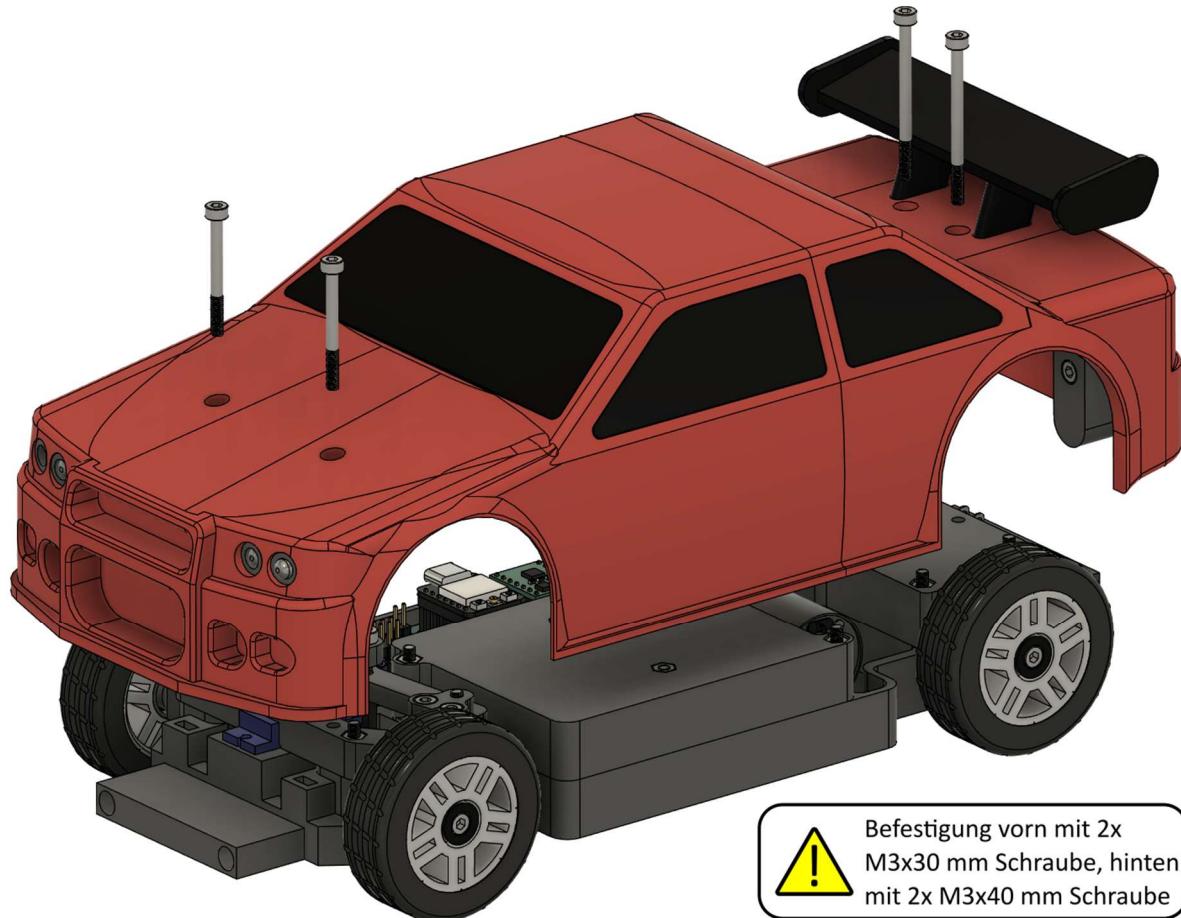
3



 Nach der Einstellung der Neutralposition
des Servos muss das Servohorn so
aufgesetzt werden, dass es möglichst
parallel zur roten Linie ausgerichtet ist.

5.3 Fahrzeug fertig montieren

Nun werden die LEDs der Karosserie mit der Platine verbunden und anschließend die Karosserie auf das Chassis aufgesetzt. Die LEDs werden wie in Kapitel 3.4.4 beschrieben angeschlossen. Wichtig: Es dürfen keine Kabel beim Verschrauben der Karosserie eingeklemmt werden. Außerdem müssen die Kabel so verlegt werden, dass sie nicht an den Rädern schleifen – dabei bitte auch den Lenkausschlag der Vorderräder beachten.



5.4 Steuerungsprinzip und Funktionen des Fahrzeugs

Das Fahrzeug wird über den einzelnen Joystick der Fernsteuerung gesteuert. Wenn der Joystick nach vorn bewegt wird, beschleunigt das Fahrzeug vorwärts. Wird der Joystick nach hinten bewegt, fährt das Fahrzeug rückwärts. Sollte diese Zuordnung vertauscht sein, muss die Variable „motordirection“ im Code der Fernsteuerung modifiziert werden (siehe Kapitel 4.2).

Wenn der Joystick nach links bewegt wird, lenkt das Fahrzeug nach links; analog lenkt das Fahrzeug nach rechts, wenn der Joystick nach rechts bewegt wird. Sollte das Fahrzeug in die falsche Richtung lenken, müssen die Werte der Variablen „leftlimit“ und „rightlimit“ im Code der Fernsteuerung vertauscht werden (siehe Kapitel 4.2). Bevor die Lenkung korrekt getrimmt ist, kann es sein, dass die neutrale Lenkposition nicht korrekt ausgerichtet ist. Das wird im nächsten Schritt eingestellt.

Die weißen Frontscheinwerfer-LEDs leuchten durchgehend, sobald das Fahrzeug angeschaltet wird. Die roten Rücklichter leuchten mit voller Intensität, solange das Fahrzeug steht oder rückwärtsfährt. Sie leuchten mit reduzierter Intensität, wenn das Fahrzeug vorwärtsfährt. Die Blinker blinken automatisch, wenn der Joystick nach links bzw. nach rechts bewegt wird. Sollte die Zuordnung der Lichter nicht passen, sollte geprüft werden, ob alle LEDs korrekt angeschlossen sind.

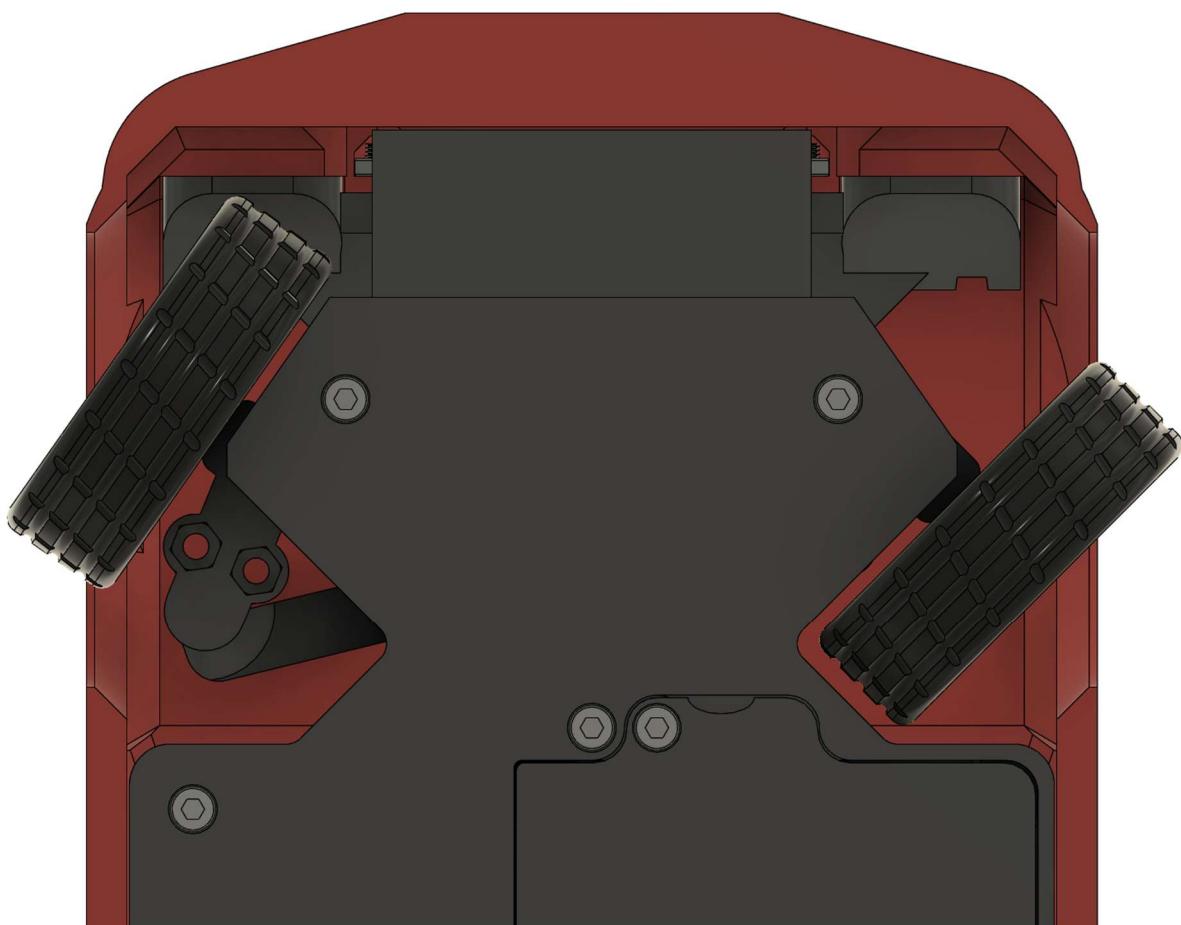
Der Piezobuzzer des Fahrzeugs fungiert außerdem als Hupe. Wird der Joystick gedrückt, ertönt ein Signalton, solange der Joystick gedrückt bleibt. Das Fahrzeug hupt allerdings nur, während es stillsteht. Die Funktion ist deaktiviert, sobald das Fahrzeug fährt.

Die Hupe und die Blinklichter dienen außerdem zur Statusanzeige des Fahrzeugs. Neben der initialen Tonsequenz beim Anschalten des Fahrzeugs und der Verbindung mit der Fernbedienung (siehe Kapitel 5.1) ertönt außerdem eine Tonsequenz kombiniert mit dem Leuchten der Blinker, wenn die Batteriespannung zu niedrig ist, das Fahrzeug keine Verbindung mehr zur Fernsteuerung hat, oder aber seit längerer Zeit keine Nutzereingaben über die Fernbedienung mehr erfolgt sind. In allen drei Fällen wird der Antriebsmotor deaktiviert und kein Datensignal mehr an den Servomotor der Lenkung übertragen. Die letzten beiden Fälle sind reversibel, der Alarmzustand kann also durch eine erneute Nutzereingabe bzw. ein Anschalten/in-Reichweite-bringen der Fernbedienung deaktiviert werden. Die Warnung zur niedrigen Batteriespannung kann nur durch einen Neustart des Fahrzeugs zurückgesetzt werden.

Ursache	Ton- und Blinkfrequenz
Batteriespannung sinkt auf <10,2 V.	Blinker leuchten und Hupe ertönt alle drei Sekunden für 0,5 Sekunden.
Keine Nutzereingaben für wenigstens 60 Sekunden mehr erfolgt	Blinker leuchten und Hupe ertönt alle drei Sekunden für eine Sekunde.
Signalverlust zur Fernbedienung für wenigstens eine Sekunde	Blinker leuchten und Hupe ertönt alle drei Sekunden für 1,5 Sekunden.

5.5 Trimmung der Lenkung

Zum Einstellen der Lenkung wird das Fahrzeug geradeaus bewegt, indem der Joystick nach vorn geschoben wird. Wenn das Fahrzeug auf eine Seite zieht, müssen die Variablen „leftlimit“ und „rightlimit“ angepasst werden. Der maximale Lenkausschlag wird angepasst, indem die Werte für „leftlimit“ und „rightlimit“ gleichzeitig weiter vom Servo-Mittelwert entfernt werden. Der maximale Lenkausschlag darf nicht mehr als 45° für das kurveninnere Rad betragen (siehe Abbildung). Solange der maximale Lenkausschlag noch nicht erreicht ist, kann ein Drall nach links ausgeglichen werden, indem der Wert für „rightlimit“ weiter vom Servo-Mittelwert (308) entfernt wird (analog für einen Rechtsdrall mit „leftlimit“). Wenn der maximale Lenkausschlag bereits erreicht ist, wird ein Linksdrall durch eine Annäherung des Werts für „leftlimit“ an den Servo-Mittelwert (308) ausgeglichen (analog für einen Rechtsdrall mit „rightlimit“). Die Modifikation der Lenkung kann bei aktivem Fahrzeug direkt im Code der Fernbedienung erfolgen (siehe Kapitel 4.2). Das erlaubt eine schnelle Anpassung.



5.6 Feintuning der Fahreigenschaften

Wenn die Lenkung eingestellt ist, ist das Fahrzeug grundsätzlich vollständig fahrbereit. Es gibt aber noch weitere Variablen, die das Fahrverhalten beeinflussen. Zunächst ist hier die Maximalgeschwindigkeit zu nennen. Diese wird ebenfalls im Code der Fernsteuerung eingestellt, über die Variable „speedlimit“ (siehe Kapitel 4.2). Der Maximalwert liegt bei 255, was einer Geschwindigkeit von bis zu ca. 10 km/h entspricht. Der Wert kann theoretisch bis auf 0 reduziert werden, allerdings skaliert die Geschwindigkeit im niedrigen Bereich nicht mehr linear mit der Vorgabe, da diese die maximale Motorspannung (eingestellt über Pulsweitenmodulation) definiert. Ein Wert von ca. 120 bis 150 stellt einen sinnvollen Kompromiss aus hoher Maximalgeschwindigkeit und guter Kontrollierbarkeit dar: Zu hohe Geschwindigkeiten bedeuten, dass Kurven und Bremsen schwierig zu meistern sind.

Eine weitere Modifikationsmöglichkeit ist die Empfindlichkeit, mit der der Motor auf Eingaben durch den Joystick reagiert. Die Variable „motorint“ im Code der Fernsteuerung (siehe Kapitel 4.2) gibt vor, wie viele Millisekunden zwischen Änderungen der Sollgeschwindigkeit vergehen. Ein niedrigerer Wert bedeutet, dass das Fahrzeug auf abrupte Joystick-Eingaben sehr schnell reagiert. Dieses sehr responsive Verhalten birgt den Nachteil, dass die Räder zum Durchdrehen neigen, was die Kontrollierbarkeit, insbesondere beim Bremsen, deutlich verschlechtert. Eine Erhöhung des Werts lässt den Motor hingegen träger auf veränderte Eingaben reagieren, sodass die Traktion besser wird, das Fahrzeug aber länger zum Beschleunigen und Bremsen benötigt. Als Startwert ist ein Wert von 10 voreingestellt. Dieser kann schrittweise erhöht oder reduziert werden, bis der individuell bevorzugte Kompromiss aus Reaktionsfreudigkeit und Kontrollierbarkeit gefunden ist.

Die Einstellung der Werte erfolgt in der Fernbedienung, sodass das Fahrzeug wie bei der Trimmung der Lenkung nicht abgeschaltet oder zerlegt werden muss, wenn die Änderungen vorgenommen werden.

6. Mögliche Fehler und Lösungsansätze

Fehlerbeschreibung	Ursache	Lösungsmöglichkeiten
Status-LED der Fernbedienung leuchtet nach dem Anschalten nicht.	Batterie falschherum eingesetzt.	Auf korrekte Polarität achten. Anschlüsse zwischen Platine und Batteriehalter ebenfalls prüfen!
Status-LED der Fernbedienung blinkt zweimal pro Sekunde.	Batteriespannung niedrig (<3.5 V).	Batterie laden.
Status-LED der Fernbedienung flackert unruhig oder mit reduzierter Helligkeit.	Batteriespannung zu niedrig für korrekten Betrieb der Fernsteuerung (<3.0 V).	Batteriespannung prüfen. Sollte die Batterie weniger als 2.5 V haben, wurde sie tiefentladen und muss ersetzt werden. Andernfalls laden.
Es ertönt kein Signalton/es leuchtet keine LED des Fahrzeugs nach dem Anschalten.	Wenigstens eine der drei Batterien falschherum eingesetzt.	Auf korrekte Polarität achten. Anschlüsse zwischen Platine und Batteriehalter ebenfalls prüfen!
Es ertönt kein Signalton/es leuchtet keine LED des Fahrzeugs nach dem Anschalten.	Die Batterien sind defekt (alle drei zusammen liefern < 7 V, also weniger als 2,3 V/Zelle).	Batterien einzeln prüfen und tiefentladene/defekte Zellen ersetzen.
Die Tonsequenz für die Verbindung mit der Fernbedienung ertönt nicht (das Fahrzeug piept im Zweisekundentakt, obwohl eine angeschaltete Fernbedienung daneben ist).	Geräte-Adressen zwischen Fernbedienung und Fahrzeug nicht korrekt (gleich) definiert.	Die Codezeilen 27 und 28 in der Fernsteuerung und 23 und 24 im Fahrzeug müssen identisch sein. Siehe Kapitel 7.1.
Die Tonsequenz für die Verbindung mit der Fernbedienung ertönt nicht (das Fahrzeug piept im Zweisekundentakt, obwohl eine angeschaltete Fernbedienung daneben ist).	Antennen nicht ordnungsgemäß in den Anschläßen der Mikrocontroller.	Die Antennenanschlüsse auf korrekten Sitz prüfen. Die Funkreichweite ohne die externen Antennen ist extrem gering.
Die Tonsequenz für die Verbindung mit der Fernbedienung ertönt nicht (das Fahrzeug piept im Zweisekundentakt, obwohl eine angeschaltete Fernbedienung daneben ist).	Starke Interferenzen durch WLAN.	Funkkanal der EasyRC-Plattform wechseln. Siehe Kapitel 7.2.
Das Fahrzeug fährt, obwohl der Joystick der Fernsteuerung nicht bewegt wird.	Joystick nicht korrekt kalibriert.	Fernsteuerung aus- und wieder anschalten. Dabei den Joystick nicht aus der Ruheposition bewegen.
Das Fahrzeug fährt eine Kurve, obwohl der Joystick nur nach vorn/hinten bewegt wird.	Lenkung nicht korrekt getrimmt.	Anweisungen in Kapitel 5 zur Trimmung der Lenkung folgen.

7. Anhang

7.1 Änderung der Geräte-Adresse der EasyRC-Plattform

Die Geräte-Adresse der EasyRC-Plattform kann geändert werden, um mehrere EasyRC-Systeme gleichzeitig betreiben zu können. Wichtig ist, dass immer ein Paar aus Sender und Empfänger die gleichen Geräte-Adressen hinterlegt hat: Das Fahrzeug hat die Geräte-Adresse der zugehörigen Fernsteuerung hinterlegt und prüft beim Erhalt neuer Datenpakete, ob diese von einer Fernsteuerung mit der korrekten Geräte-Adresse verschickt worden sind. Deshalb müssen die Variablen CarMAC und RemoteMAC sowohl im Code des Mikrocontrollers der Fernsteuerung als auch des Fahrzeugs geändert werden, wenn ein zweites EasyRC-System eingesetzt werden soll.

Die Geräte-Adressen sind in der Fernsteuerung in den Codezeilen 27 und 28 hinterlegt. Im Fahrzeug sind sie in den Zeilen 23 und 24 zu finden. Die Geräte-Adresse besteht aus sechs Zeichenpaaren, die sich aus Zahlen und Buchstaben zusammensetzen können. Die Abbildung zeigt die Geräte-Adressen in der Fernsteuerung. Zur Änderung der Geräte-Adressen reicht es bereits aus, lediglich ein Zeichen eines Zeichenpaares zu ändern.

```
26 //MAC addresses
27 uint8_t CarMAC[] = {0xAA, 0xA0, 0xBE, 0xEF, 0xFE, 0xED}; //we will set this MAC address for the car
28 uint8_t RemoteMAC[] = {0xB0, 0xAD, 0xBE, 0xEF, 0xFE, 0xEF}; //we will set this MAC address for the remote control
29 uint8_t broadMAC[] = {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF}; //using this MAC address as a receiver for sending
30 //packages means broadcasting (ignoring if the data has been received)
31
```

Hier ist eine Auswahl an möglichen Geräteadressen, die die Zeilen im Code direkt ersetzen können. Das oberste Paar entspricht den Geräte-Adressen im Auslieferungszustand. Das ausgewählte Paar muss sowohl in der Fernsteuerung als auch im Fahrzeug hinterlegt werden.

uint8_t CarMAC[] = {0xAA, 0xA0, 0xBE, 0xEF, 0xFE, 0xED}; //we will set this MAC address for the car
uint8_t RemoteMAC[] = {0xB0, 0xAD, 0xBE, 0xEF, 0xFE, 0xEF}; //we will set this MAC address for the remote control
uint8_t CarMAC[] = {0xA0, 0xA0, 0xBE, 0xEF, 0xFE, 0xED}; //we will set this MAC address for the car
uint8_t RemoteMAC[] = {0xBA, 0xAD, 0xBE, 0xEF, 0xFE, 0xEF}; //we will set this MAC address for the remote control
uint8_t CarMAC[] = {0xAA, 0xAA, 0xBE, 0xEF, 0xFE, 0xED}; //we will set this MAC address for the car
uint8_t RemoteMAC[] = {0xB0, 0xAA, 0xBE, 0xEF, 0xFE, 0xEF}; //we will set this MAC address for the remote control
uint8_t CarMAC[] = {0xA0, 0xAA, 0xBE, 0xEF, 0xFE, 0xED}; //we will set this MAC address for the car
uint8_t RemoteMAC[] = {0xBA, 0xAA, 0xBE, 0xEF, 0xFE, 0xEF}; //we will set this MAC address for the remote control

7.2 Änderung des Funkkanals der EasyRC-Plattform

Der Funkkanal wird im Code in den Zeilen 90 und 92 (Fernbedienung, in der Abbildung gezeigt) sowie 124 und 126 (Fahrzeug) eingestellt, indem die rot markierte Zahl verändert wird. Die Voreinstellung der Plattform ist Kanal 1. Es kann ein Wert zwischen 1 und 13 ausgewählt werden. Wichtig: Ein verbundenes Paar aus Fahrzeug und Fernsteuerung muss denselben Kanal nutzen. Andernfalls ist keine stabile Verbindung möglich!

```
83 // Activate ESP-NOW
84 WiFi.mode(WIFI_STA); //Set device as a Wi-Fi Station
85 esp_wifi_set_mac(WIFI_IF_STA, RemoteMAC); //Overwrite h
86 esp_wifi_set_protocol(WIFI_IF_STA, WIFI_PROTOCOL_LR); /
87 esp_now_init(); // Initialize ESP-NOW
88
89 esp_wifi_set_promiscuous(true); //set WiFi channel
90 esp_wifi_set_channel(1, WIFI_SECOND_CHAN_NONE);
91 esp_wifi_set_promiscuous(false);
92 peerInfo.channel = 1;
93 peerInfo.encrypt = false;
```

8. Impressum

EasyRC ist eine open-source-Plattform, entwickelt von Thomas Böhm. Alle relevanten Informationen sowie Updates zum Projekt sind im zugehörigen GitHub-Repository zu finden:
<https://github.com/TRD-B/EasyRC>

Der Autor übernimmt keine Haftung für Schäden oder Verletzungen, die durch die Nutzung der zur Verfügung gestellten Hard- und Software entstehen können!