

# Vorwort

## Über SunFounder

SunFounder ist ein Technologieunternehmen, spezialisiert auf die Entwicklung von Raspberry Pi und Arduino Open Source Community. Engagiert für die Förderung der Open Source Kultur, streben wir danach, Menschen auf der ganzen Welt Spass an der Eletronikherstellung zu bringen und es jedem zu erlauben ein Maker zu sein. Unsere Produkte umfassen Lernsets, Entwicklungsboards, Roboter, Sensormodule und Entwicklungswerkzeuge. Neben hochwertigen Qualitätsprodukten bietet SunFounder auch Video-Tutorials an, zur Unterstützung Ihrer eigenen Projekte. Wenn Sie nun Interesse an Open Source haben oder am Basteln von etwas Coolem, dann heißen wir Sie herzlich willkommen!

## Über dieses Set

Dieses Set eignet sich für das Raspberry Pi Modell B+, 2 Modell B and 3 Modell B.

In diesem Handbuch werden wir Ihnen zeigen, wie man einen Smart Car bildet, mit Hilfe von Erläuterungen, Illustrationen von Bauteilen und schematischen Schaltkreisdiagrammen, aus Sicht der Hardware und Software. Besuchen Sie unsere Webseite [www.sunfounder.com](http://www.sunfounder.com) für den Download von zugehörigen Codes und schauen Sie sich unsere Anleitung auch unter [LEARN > Get Tutorials](#) und Videos unter [VIDEO](#) an, oder holen sich den Code auf unserer Seite auf [github.com](https://github.com/sunfounder/Sunfounder Smart Video Car Kit for RaspberryPi)

<https://github.com/sunfounder/Sunfounder Smart Video Car Kit for RaspberryPi>

## Gratis Support



Für **TECHNISCHE Fragen**, fügen Sie ein Topic in der **FORUM** Sektion auf unserer Webseite hinzu, worauf wir sobald wie möglich antworten werden.



Für **NICHT-TECHNISCHE Fragen**, wie über Bestellungs- und Zustellungsprobleme, **senden Sie eine E-Mail an [service@sunfounder.com](mailto:service@sunfounder.com)**. Teilen Sie Ihre Projekte mit uns im FORUM.

# Verzeichnis

Einführung.....	3
Übersicht der Bauteile .....	4
i. Acrylplatten.....	4
ii. Mechanische Befestigungsteile .....	5
iii. Antriebsteile .....	6
iv. Elektrokomponenten.....	7
v. Kundenseitig .....	10
Montage .....	11
i. Mechanische Montage.....	11
1. Auto Montage.....	11
2. Montage der Kamerahalterung.....	25
3. Kamerahalterung + Auto .....	30
ii. Einbau der Elektrokomponenten .....	32
iii. Schaltkreise verbinden.....	33
Grundlage der Elektrokomponenten .....	43
i. Raspberry Pi.....	43
ii. DC-DC Abwärtswandler.....	44
iii. Servo.....	44
iv. Gleichstrommotortreiber .....	45
v. USB Wi-Fi Adapter .....	45
vi. Servosteuerung .....	46
vii. 18650*2 Batteriehalter .....	46
Zugehörige Software .....	47
i. Download und Installataion des Raspbian auf einer TF Karte.....	47
ii. Holen Sie sich den Quellcode.....	47
iii. Softwareumgebung.....	48
Ausführung auf PC .....	48
Bereiten Sie den PC für den Fernzugriff vor .....	48
Ausführung auf Raspberry Pi .....	49
Installation von python-dev, python-smbus.....	49
I2C-Port Einstellungen.....	49
MJPG-Streamer .....	49
iv. Kalibrierung .....	52
Vorbereitung.....	52

Kalibrierung beginnen.....	55
Motor-Einstellungen .....	55
Turning-Einstellungen .....	55
Mount-Einstellungen .....	56
Raus auf die Strasse! .....	57
v. Über das Programm.....	61
Abstrakt .....	61
Einführung zum Socket.....	61
1. Server .....	61
2. Client.....	63
Zusammenfassung .....	65

sunFounde

# Einführung

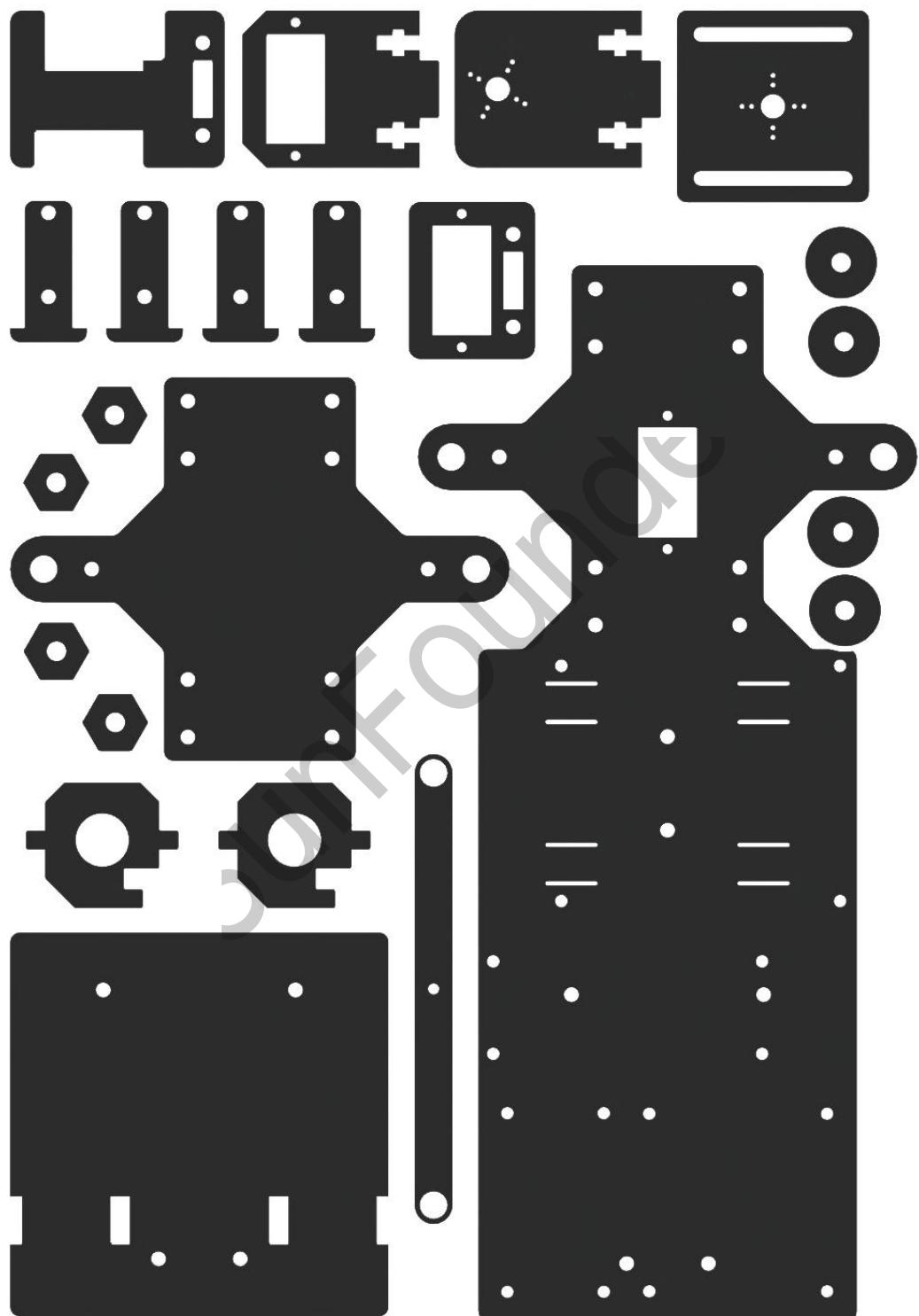
Das SunFounder Smart Video Car Set für Raspberry Pi besteht aus einer Raspberry Pi, DC-DC Abwärtswandler, USB-Kamera, Gleichstrommotortreiber, und einer auf PCA9685 basierenden Servosteuerung. Aus der Sicht der Software, ist das Smart Car von der Struktur ein Client-Server (C/S-Struktur). Der TCP Server läuft auf Raspberry Pi, für die direkte Steuerung vom Auto. Und die Videodaten werden über die Open Source Software MJPG-Streamer in Echtzeit erfasst und übermittelt. Der TCP Client läuft auf dem PC, für das Versenden eines Steuerbefehls. Client und Server sind beide in Python geschrieben.

Das Smart Car wurde basierend auf das Open Source Hardware Raspberry Pi entwickelt und integriert das Wissen der Mechanik, Elektronik und Computer, somit hat es eine tiefe pädagogische Bedeutung.



# Übersicht der Bauteile

## i. Acrylplatten

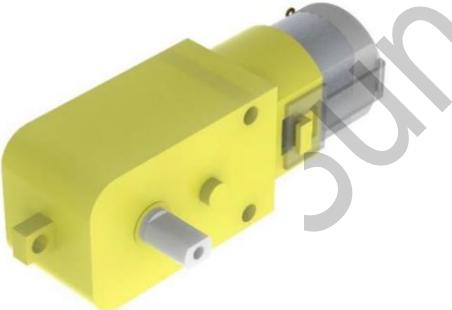


## ii. Mechanische Befestigungsteile

Teile	Name	Stk.
	M1.2*4 Selbstschneidende Schraube	8
	M2*8 Schraube	6
	M2.5*6 Schraube	16
	M3*10 Senkschraube	2
	M3*8 Schraube	8
	M3*10 Schraube	6
	M3*30 Schraube	4
	M4*25 Schraube	2
	M2.5*8 Kupfersäule	16
	M3*24 Kupfersäule	8

	M2 Mutter	6
	M2.5 Mutter	16
	M3 Mutter	20
	M4 Selbstverriegelungsmutter	2
	F694ZZ Bundlager	2

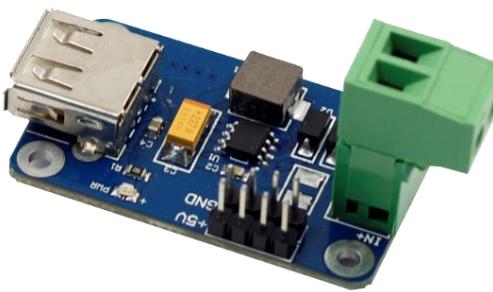
### iii. Antriebsteile

Teile	Name	Stk.
	TowerPro Micro Servo SG90	3
	Untersetzungsgtriebe	2
	Vorderrad	2

	Hinterrad	2
--	-----------	---

#### iv. Elektrokomponenten

Teile	Name	Stk.
	Raspberry Pi Modell B+	1
	16-Kanal 12-bit PWM Antrieb (Servosteuerung)	1
	L298N Gleichstrommotor- treiber	1

	DC-DC Abwärtswandler	1
	USB WiFi Adapter	1
	USB-Kamera	1
	18650*2 Batteriehalter	1
	Band	1
	USB-Kabel	1

	Kreuzsteck-schlüssel	1
	Kreuzschrauben-zieher	1
	Spiralschlauch	1
	20cm Schaltdraht (F - F)	4
	10cm Schaltdraht (F - F)	5
	10cm Schaltdraht (M - F)	2
	20cm Schaltdraht (M - M)	2

## v. Kundenseitig

Die unten angezeigten Teile sind NICHT im Set beinhaltet.

Teile	Name	Stk. (benötigt)
	18650 3.7V aufladbare Li-ion Batterie	2
	TF Karte	1

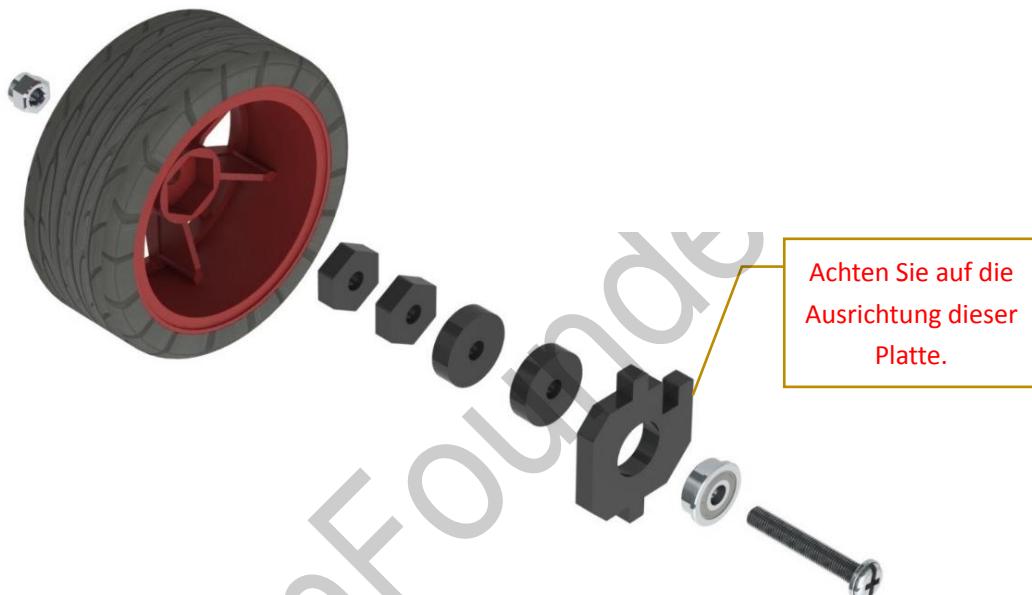
# Montage

## i. Mechanische Montage

### 1. Auto Montage

#### Vorderrad

- a) Schrauben Sie das F694ZZ Bundlager, Vorderrad und die folgenden Acrylplatten mit einer M4\*25 Schraube und einer M4 Selbstverriegelungsmutter zusammen, siehe Abbild (Abb.). Benutzen Sie dafür die Kreuzsteckschlüssel, um die M4 Selbstverriegelungsmutter zu befestigen.



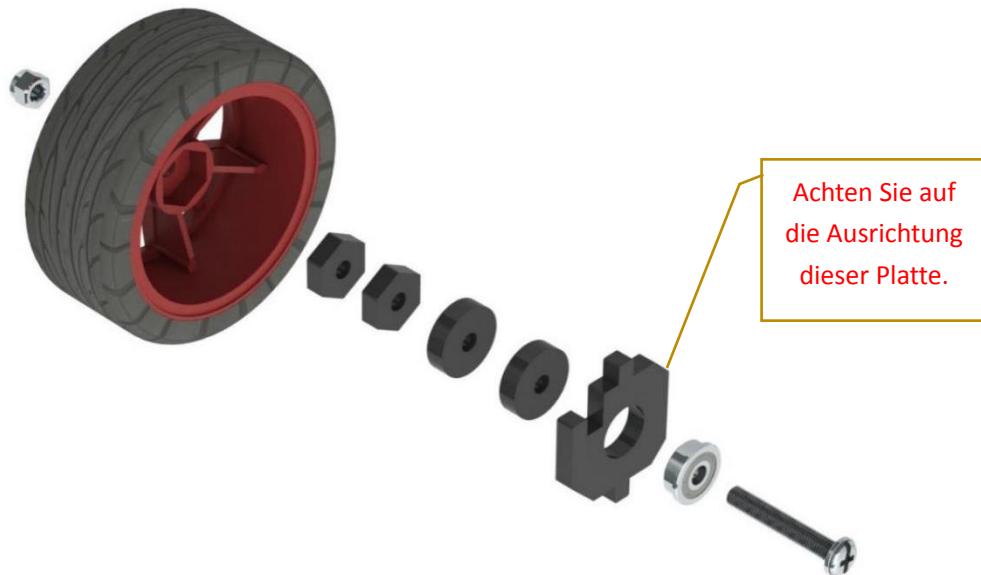
- b) Die zusammengesetzten Teile sollten, wie im unterem Abbild (Abb.) dargestellt aussehen.



- c) Bitte beachten Sie, dass die Mutter nicht zu fest zugedreht werden darf, ansonsten kann sich das Rad nicht flexibel drehen. Aber auch nicht zu locker, da der Abstand zwischen den Teilen nicht zu gross sein darf.



- a) Schrauben Sie das F694ZZ Bundlager, Vorderrad und die folgenden Acrylplatten mit einer M4\*25 Schraube und einer M4 Selbstverriegelungsmutter zusammen, wie folgt:



b) Die zusammengesetzten Teile sollten, wie im unterem Abb. dargestellt aussehen.



c) Bitte beachten Sie, dass die Mutter nicht zu fest zugedreht werden darf, ansonsten kann sich das Rad nicht flexibel drehen. Aber auch nicht zu locker, da der Abstand zwischen den Teilen nicht zu gross sein darf.

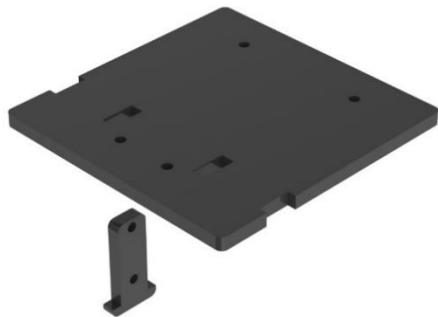


*Hinweis: Die Acrylplatten der zwei Räder werden in entgegengesetzten Richtungen eingebaut, siehe Abb. unten.*



## **Hintere Unterplatte des Fahrgestells + Hinterräder**

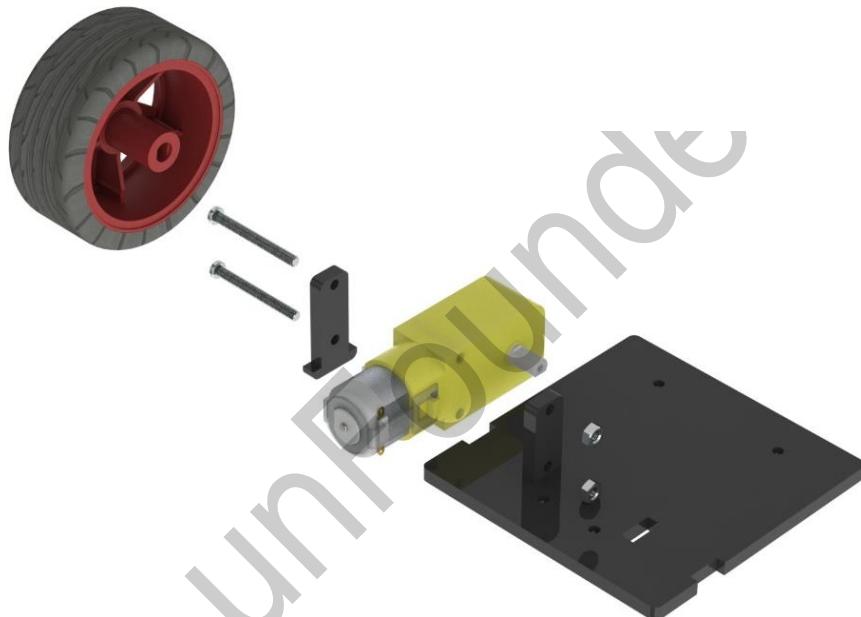
- a) Setzen Sie die folgenden zwei Acrylplatten zusammen.



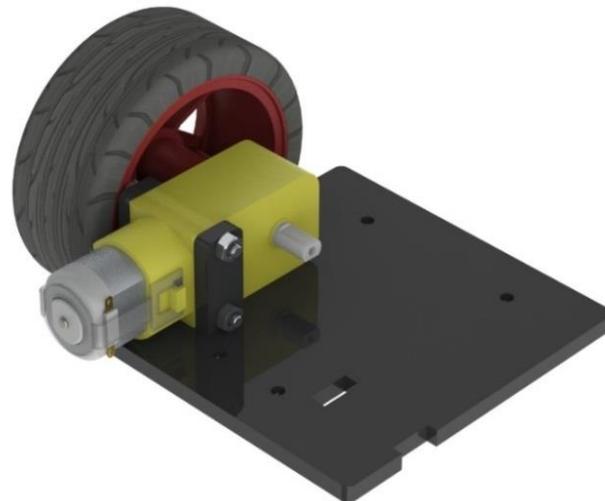
- b) Die zusammengesetzten Teile sollten, wie im unterem Abb. dargestellt aussehen.



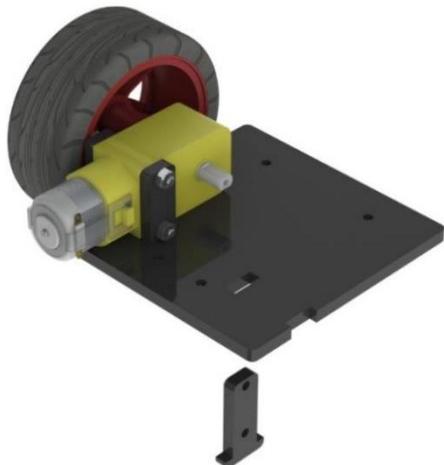
- c) Befestigen Sie das Untersetzungsgetriebe, das Hinterrad und die folgenden Acrylplatten mit zwei M3\*30 Schrauben und einer M3 Mutter.



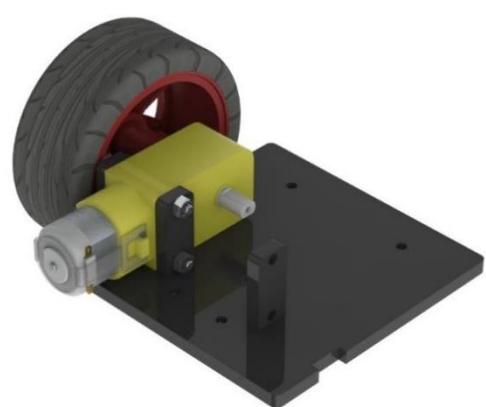
- d) Die zusammengesetzten Teile sollten, wie im unterem Abb. dargestellt aussehen.



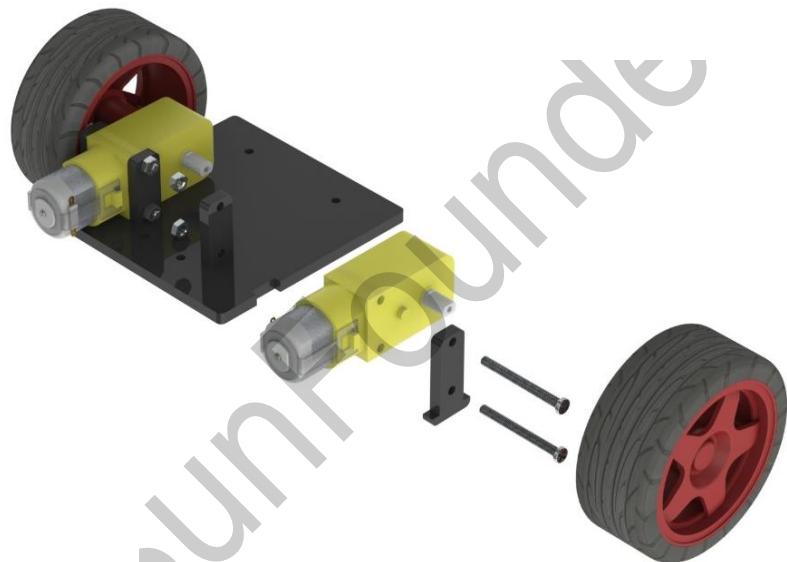
e) Setzen Sie folgende Acrylplatte wie zuvor ein.



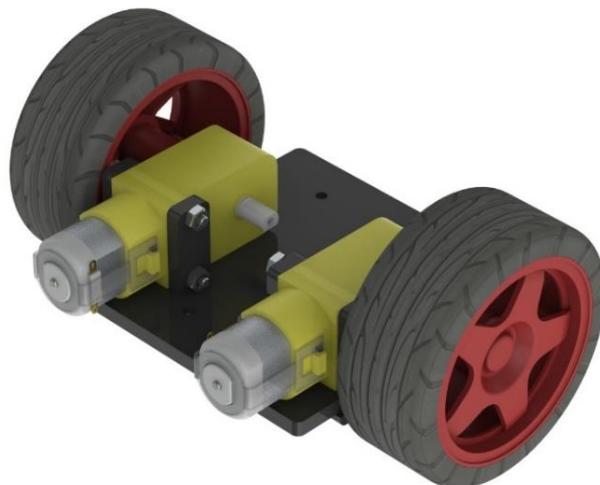
f) Die zusammengesetzten Teile sollten, wie im unterem Abb. dargestellt aussehen.



g) Wie zuvor befestigen Sie das zweite Untersetzungsgetriebe und Hinterrad auf der gegenüberliegenden Seite mit zwei M3\*30 Schrauben and M3 Mutter.

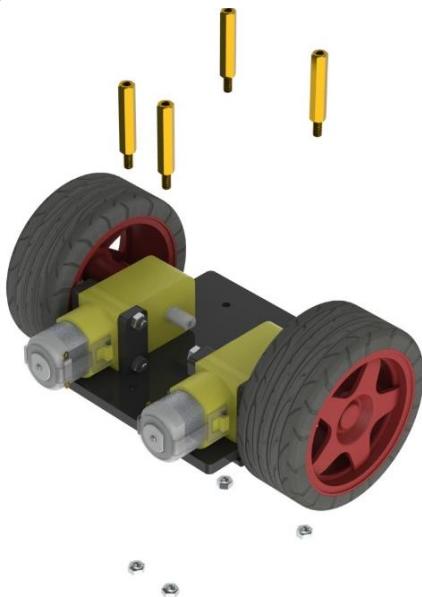


h) Die zusammengesetzten Teile sollten, wie im unterem Abb. dargestellt aussehen.

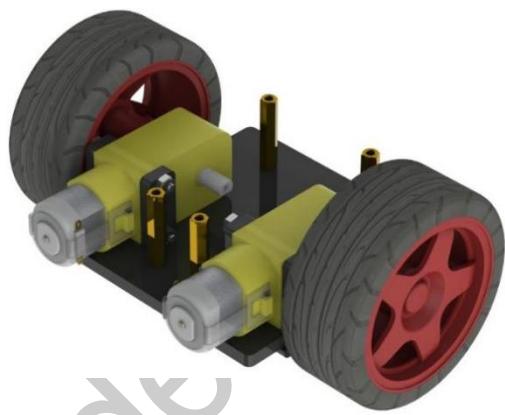


## Hintere Unterplatte des Fahrgestells + Kupfersäulen

- a) Befestigen Sie die vier M3\*24 Kupfersäulen mit vier M3 Muttern in die Acrylplatte, siehe Abb.:



- b) Die zusammengesetzten Teile sollten, wie im unterem Abb. dargestellt aussehen.



## Servo-Einstellungen

Vor dem Einbau der Servos, müssen wir den Winkel der Ausgangsachse auf 90° einstellen. Warum? Der Grund ist, dass die Servos die wir benutzen sich ausschliesslich in einem Bereich von 0°-180° drehen dürfen. Wenn es ausserhalb des zulässigen Bereiches liegt, dann kann es zu einem Durchbrennen der Servos führen. **Daher müssen wir vor der Montage die Achse auf 90° einstellen**, und bauen dann den Servo nach der gegebenen Anleitung ein.

### 1. Methode

Die einfachste Weise ist es, den Servo an einem Raspberry Pi und PCA9685 PWM Treiber, oder an Arduino-Boards zu verkabeln und geben dem Servo PWM Signale, sodass der Winkel der Ausgangsachse sich auf 90° einstellt.

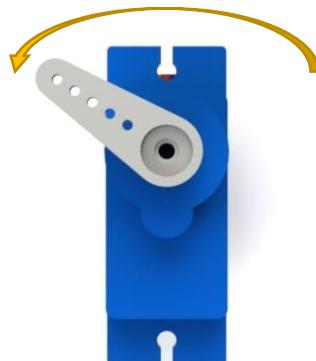
### 2. Methode

Die 2. Methode könnte sich schwierig aufweisen, wobei die 1. Methode zu kompliziert sein könnte. Bitte folgen Sie der Anleitung unten sorgfältig. Es könnte auf dem ersten Blick komplex aussehen, aber sobald Sie damit anfangen, werden Sie es ganz einfach finden.

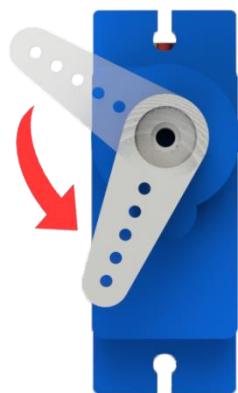
Bereiten Sie den Servo und einen einarmigen Servoarm vor.



a) Befestigen Sie den Servoarm an die Achse und drehen es gegen den Uhrzeigersinn bis es sich nicht mehr drehen lässt. Das könnte so aussehen:



b) Es kann sich in der unten angezeigten Position befinden:



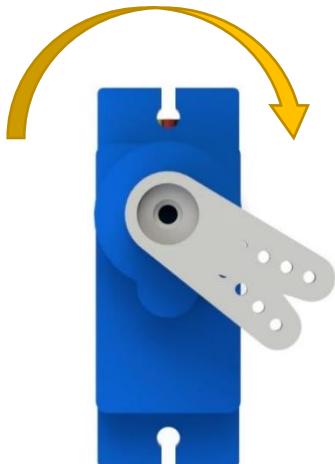
c) Unabhängig in welcher Position sich der Servoarm nun befindet, entnehmen Sie es vorsichtig, ohne daran zu drehen. Dann setzen Sie es wieder an, wie in die unten angezeigte Position. Es sollte sich in der Mitte des Servos oder ein wenig darunterliegend befinden:



Die Feineinstellungen erfolgen über die Software im Kalibrierungsabschnitt. Versuchen Sie es einfach in die ungefähre Position zu bringen.

d) Drehen Sie nun den Servoarm in den Uhrzeigersinn, bis es sich nicht mehr drehen lässt. Es sollte in einer naheliegenden, wie unten dargestellt, Position befinden:

Falls es ganz anders als abgebildet aussehen sollte, müssen Sie die vorherigen Schritte wiederholen.



e) Sollte es nun ungefähr in der Position sein, dann müssen Sie den Servoarm nochmal gegen den Uhrzeigersinn drehen bis der Hebel sich in einer aufrechten Position befindet.



Damit sollte der Winkel der Ausgangsachse sich in einer  $90^\circ$  Position befinden. Für die Feineinstellungen warten wir auf den zugehörigen Code.

Entnehmen Sie den Servoarm vorsichtig. Es ist passabel wenn die Achse sich minimal bewegen sollte, da kleine Fehler keine Korrektur benötigen.

Wir empfehlen, dass Sie alle 3 Servos aus der Verpackung, für die Einstellung vor der Montage, zu nehmen, sodass der ganze Prozess einfacher wird.

## Lenkstange + Servoarm

- a) Befestigen Sie folgende Acrylplatte mit einer M2\*8 Selbstschneidende Schraube, an das zweite Loch des Servoarms (siehe Abb).
- ◆ Die M2\*8 Selbstschneidende Schraube ist mit in der Servo Verpackung enthalten; eins der zwei längsten Schrauben.
  - ◆ Der Servoarm ist auch mit in der Servo Verpackung enthalten. *Hinweis:* Seien Sie vorsichtig mit der Schraube, Sie könnten sich daran verletzen.



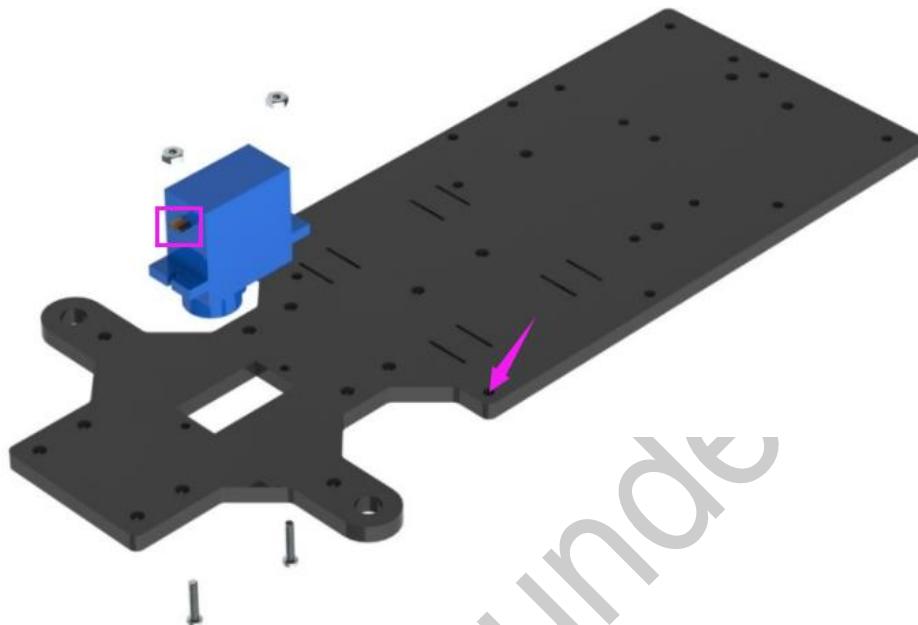
- b) Ziehen Sie die Schraube zuerst fest, dann lockern es wieder ein wenig, um die Bewegung der Lenktange zu gewährleisten. Die zusammengesetzten Teile sollten so, wie im unterem Abb. dargestellt aussehen.



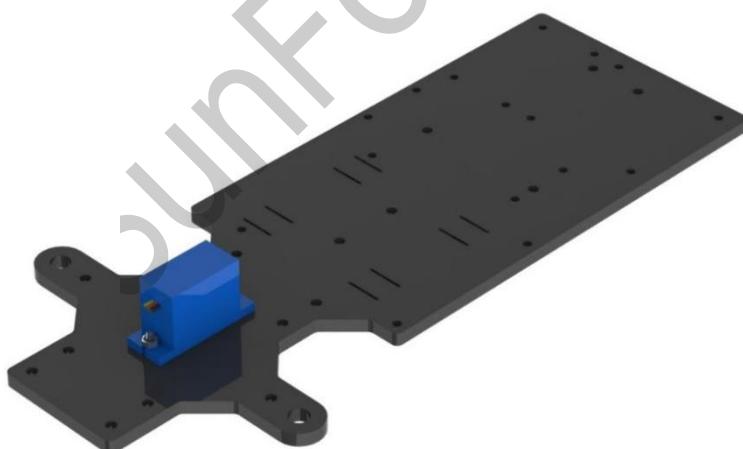
## Lenkservo + Oberplatte

a) Befestigen Sie nun den Servo auf die unten angezeigte Acrylplatte mit zwei M2\*8 Schrauben und M2 Muttern.

- ◆ Achten Sie auf die Oberfläche der Platte. Es wird im Abb. auf das durch den Pfeil angezeigte Loch in der Platte hingewiesen.



b) Die zusammengesetzten Teile sollten, wie im unterem Abb. dargestellt aussehen.



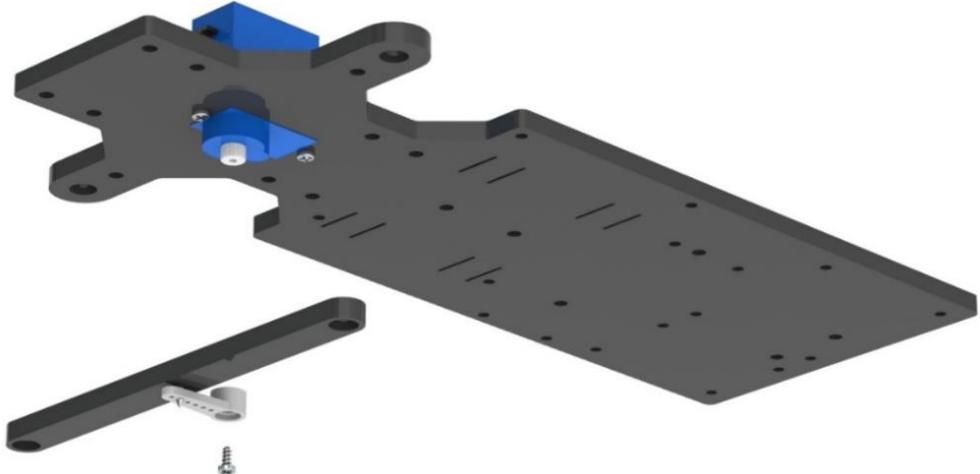
- ◆ **Hinweis:** Die Ausgangssachse des Servos sollte zur Vorderseite der Platte gerichtet sein (siehe Abb).



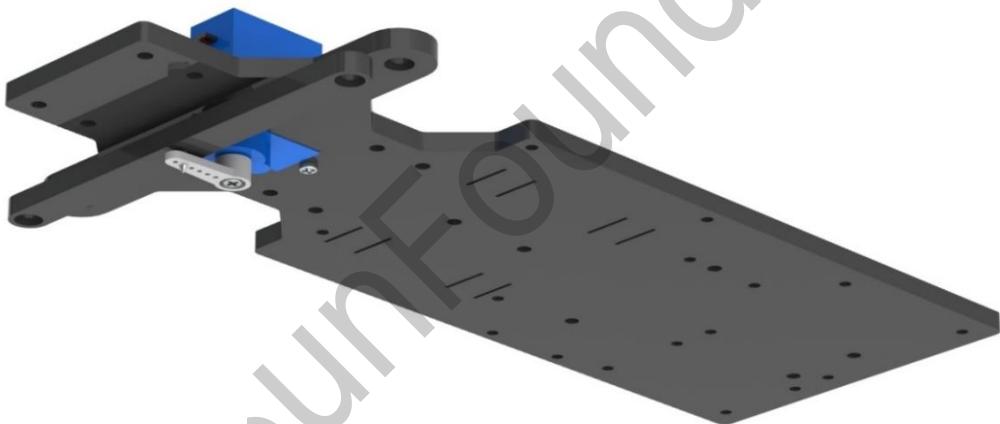
## Lenkservo + Lenkstange

a) Schrauben Sie die unten angezeigten Teile mit einer M2\*4 Schraube zusammen.

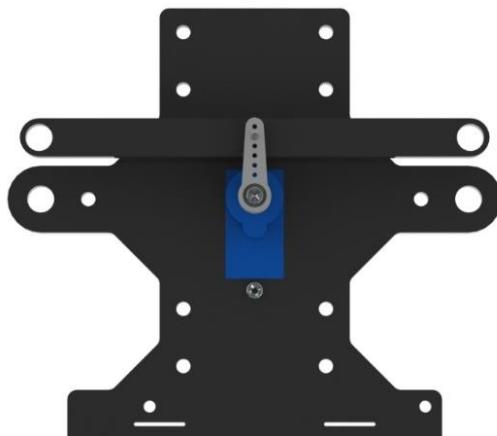
- ◆ Die M2\*4 Schraube ist mit in der Servo Verpackung enthalten; einer der kürzesten Schrauben in der Packung.



b) Die zusammengesetzten Teile sollten, wie im unterem Abb. dargestellt aussehen.

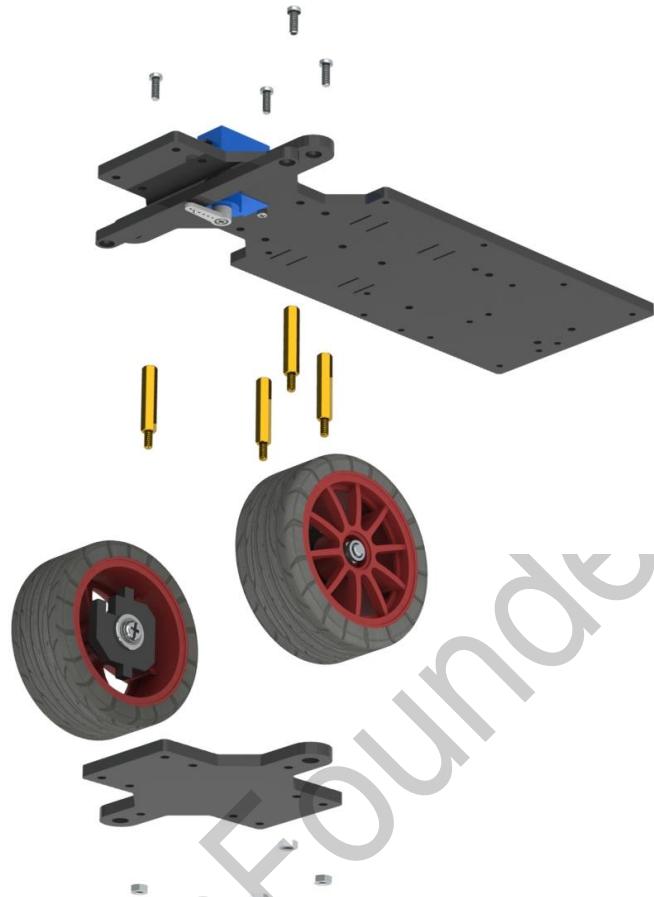


- ◆ Achten Sie auf die Ausrichtung während dem Zusammenbau. Falls Sie einen Fehler finden sollten, dürfen Sie nicht an der Ausgangsachse des Servos drehen. Stattdessen sollten Sie alles auseinander bauen und die Teile wieder richtig zusammenbauen.

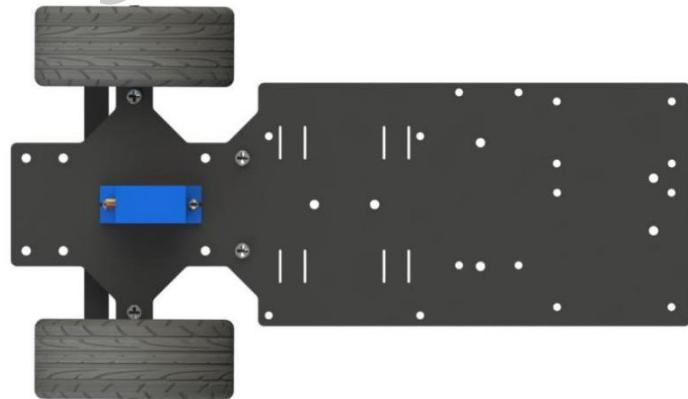


## Vordere Unterplatte des Fahrgestells + Oberplatte

- a) Schrauben Sie die unten angezeigten Teile und die Vorderräder mit vier M3\*8 Schrauben, vier M3\*24 Kupfersäulen und vier M3 Muttern zusammen.



- b) Die zusammengesetzten Teile sollten, wie im unterem Abb. dargestellt aussehen.



## Obere Platte + Kupfersäulen

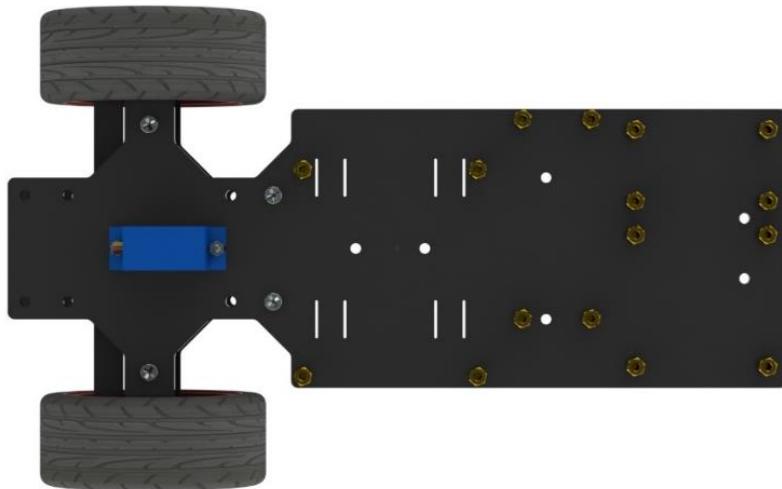
- a) Befestigen Sie die 16 M2.5\*8 Kupfersäulen und 16 M2.5 Muttern in die Acrylplatte, siehe Abb.:



- b) Die zusammengesetzten Teile sollten, wie im unterem Abb. dargestellt aussehen.

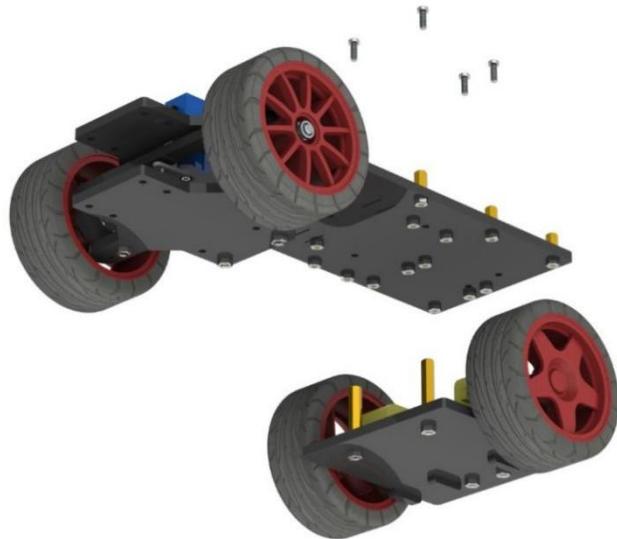


- c) Sicht von oben auf die hintere Seite der Platte:



## **Hintere Unterplatte des Fahrgestells + Oberplatte**

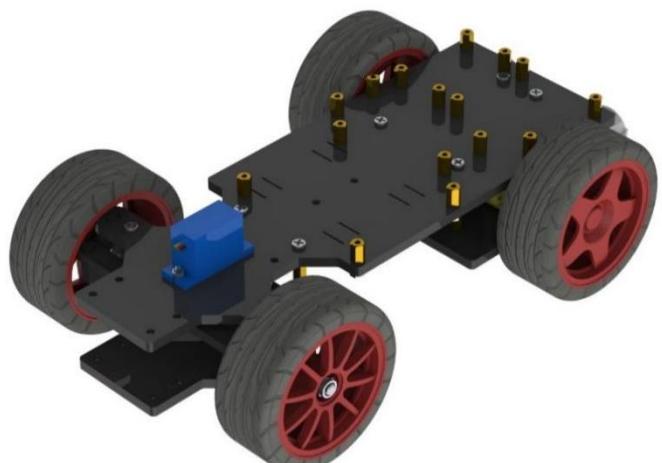
- a) Schrauben Sie die zwei fertiggestellten Teile mit vier M3\*8 Schrauben zusammen.



- b) Die zusammengesetzten Teile sollten, wie im unterem Abb. dargestellt aussehen.



- c) Hintere Seite der Platte:



## Batteriehalter

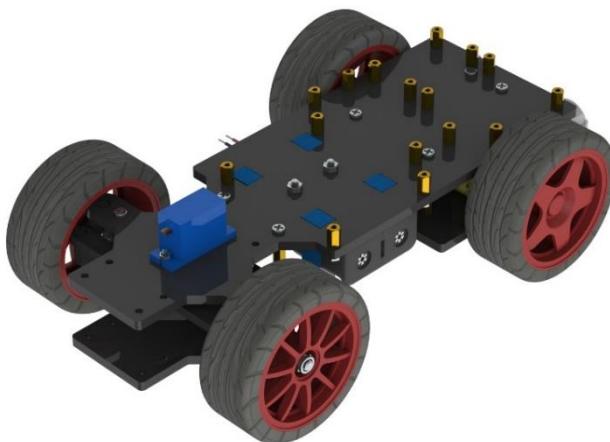
- a) Befestigen Sie den Batteriehalter in die untere Seite der Platte mit zwei M3\*10 Senkkopfschrauben und zwei M3 Muttern.
- ◆ Wenn Sie möchten, können Sie das Band durch die untere Seite der Platte ziehen, womit das Entnehmen der Batterien einfacher wird.



- b) Die zusammengesetzten Teile sollten, wie im unterem Abb. dargestellt aussehen.



- c) Sicht auf die obere Seite des Gestells. Nun ist die Auto-Montage abgeschlossen.



## 2. Montage der Kamerahalterung

### Platten + Servoarm

a) Befestigen Sie den Servoarm auf die folgende Acrylplatte mit vier M1.2\*4 Schrauben.

- ◆ Der Servoarm ist in der Servo Verpackung enthalten.



b) Die zusammengesetzten Teile sollten, wie im unterem Abb. dargestellt aussehen.

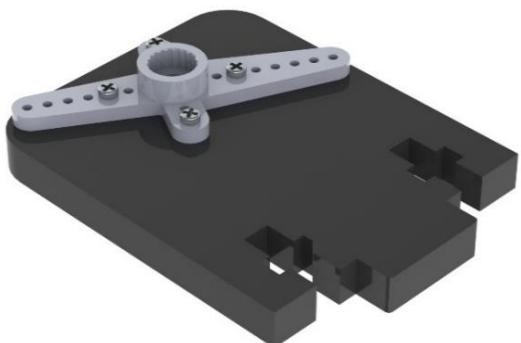


c) Befestigen Sie den Servoarm auf die folgende Acrylplatte mit vier M1.2\*4 Schrauben.

- ◆ Der Servoarm ist in der Servo Verpackung enthalten.

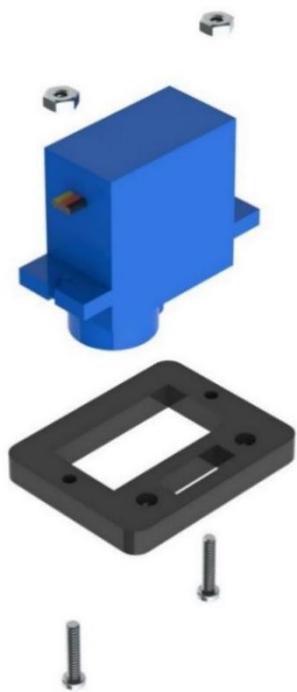


d) Die zusammengesetzten Teile sollten, wie im unterem Abb. dargestellt aussehen.

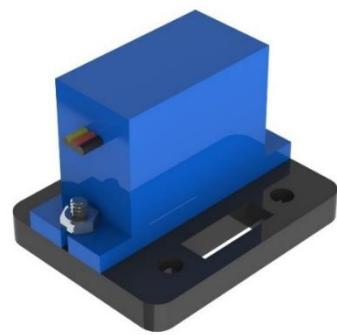


## Schwenk-/Neigungsservo + Platte

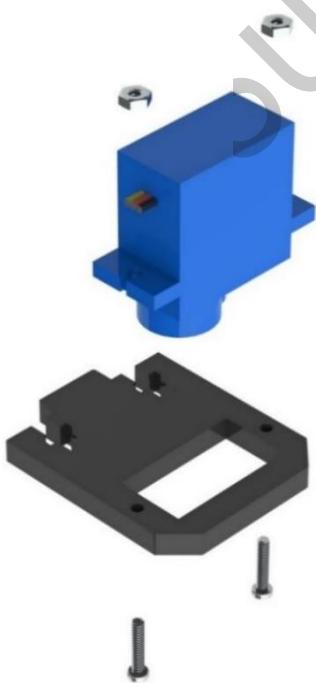
- a) Befestigen Sie das Servo auf folgende Acrylplatte mit zwei M2\*8 Schrauben und zwei M2 Muttern, dies ist das "Schwenkservo".



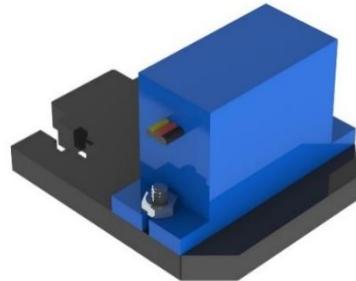
- b) Die zusammengesetzten Teile sollten, wie im unterem Abb. dargestellt aussehen.



- c) Befestigen Sie das Servo auf folgende Acrylplatte mit zwei M2\*8 Schrauben und zwei M2 Muttern, dies ist das "Neigungsservo".

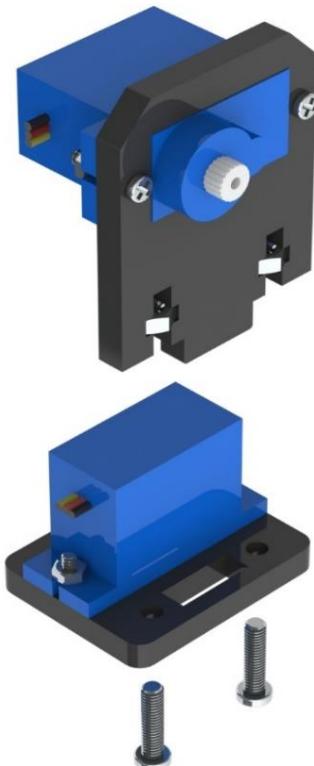


- d) Die zusammengesetzten Teile sollten, wie im unterem Abb. dargestellt aussehen.

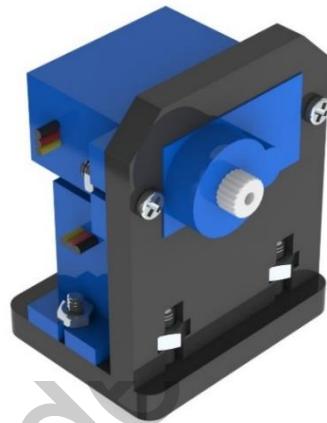


## Platte mit Schwenkservo + Platte mit Neigungsservo

- a) Schrauben Sie die zwei Plattenteile mit zwei M3\*10 Schrauben und zwei M3 Muttern zusammen. Die beiden Platten sollten senkrecht zueinander stehen.



- b) Die zusammengesetzten Teile sollten, wie im unterem Abb. dargestellt aussehen.



- c) Befestigen Sie die folgenden zwei Teile in einem rechten Winkel miteinander durch zwei M3\*10 Schrauben and zwei M3 Muttern.



- d) Die zusammengesetzten Teile sollten, wie im unterem Abb. dargestellt aussehen.



## Servos + Servoarmplatten

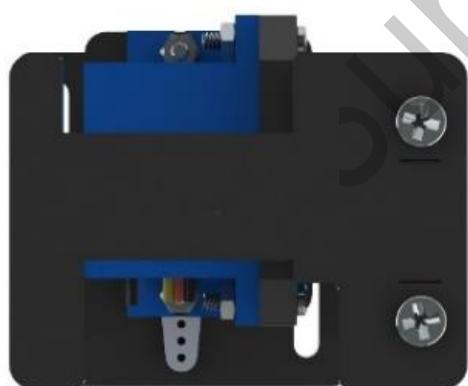
- a) Bringen Sie folgende Teile **ohne jegliche Schrauben** zusammen.



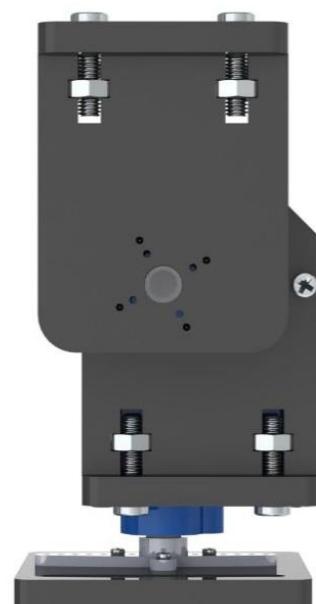
- b) Die zusammengesetzten Teile sollten, wie im unterem Abb. dargestellt aussehen.



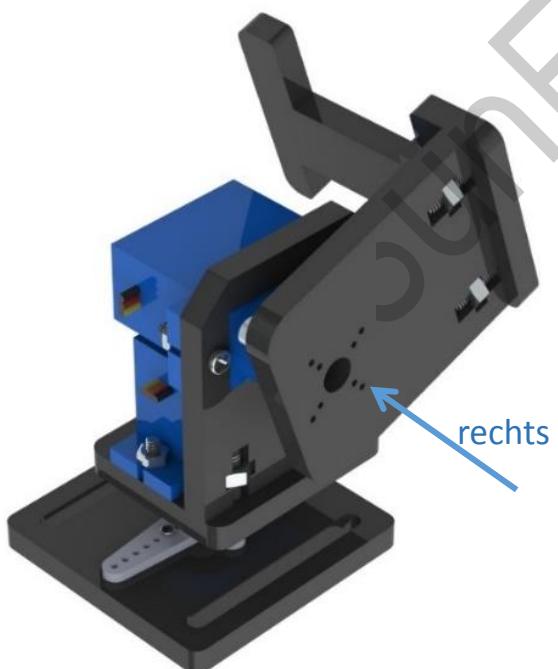
◆ Ansicht von oben:



◆ Vorderansicht:



- ◆ Vor dem Zusammenschrauben müssen Sie zuerst den Drehwinkel überprüfen. Drehen Sie die Ausrichtung der Achse des (oberen) Neigungsservos zu Ihrer rechten Seite, siehe Abb. Das (untere) Schwenkservo sollte in der Lage sein, im Bereich von 0°-180°, von links nach rechts zu drehen; das Neigungsservo sollte ebenfalls im Bereich von 0°-180° Grad, von vorne nach hinten drehbar sein.
- ◆ Achten Sie auf die Ausrichtung während dem Zusammenbau. Falls Sie einen Fehler finden sollten, dürfen Sie nicht an der Achse des Servos drehen. Stattdessen sollten Sie alles auseinander bauen und die Teile wieder richtig zusammensetzen. Die Kalibrierung hierfür ist dieselbe wie zuvor in der **Servo-Einstellung**.



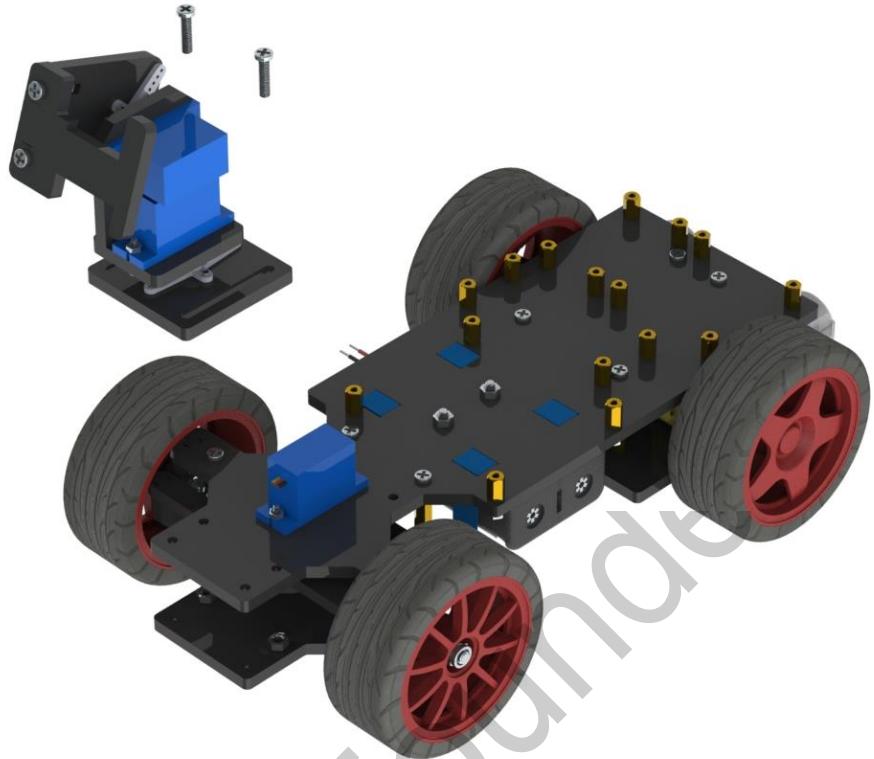
- c) Nun schrauben Sie alles zusammen mit M2\*4 Schrauben.

Die M2\*4 Schraube ist mit in der Servo Verpackung enthalten und ist die Kürzeste aller Schraube.

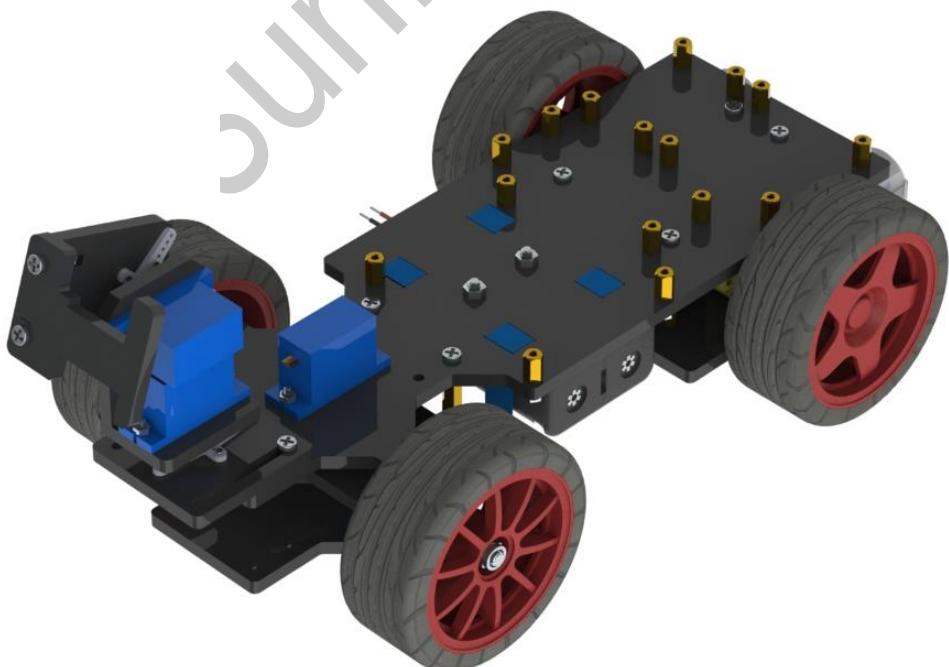


### 3. Kamerahalterung + Auto

- a) Befestigen Sie die Kamerahalterung auf das Auto mit zwei M3\*10 Schrauben und M3 Muttern.



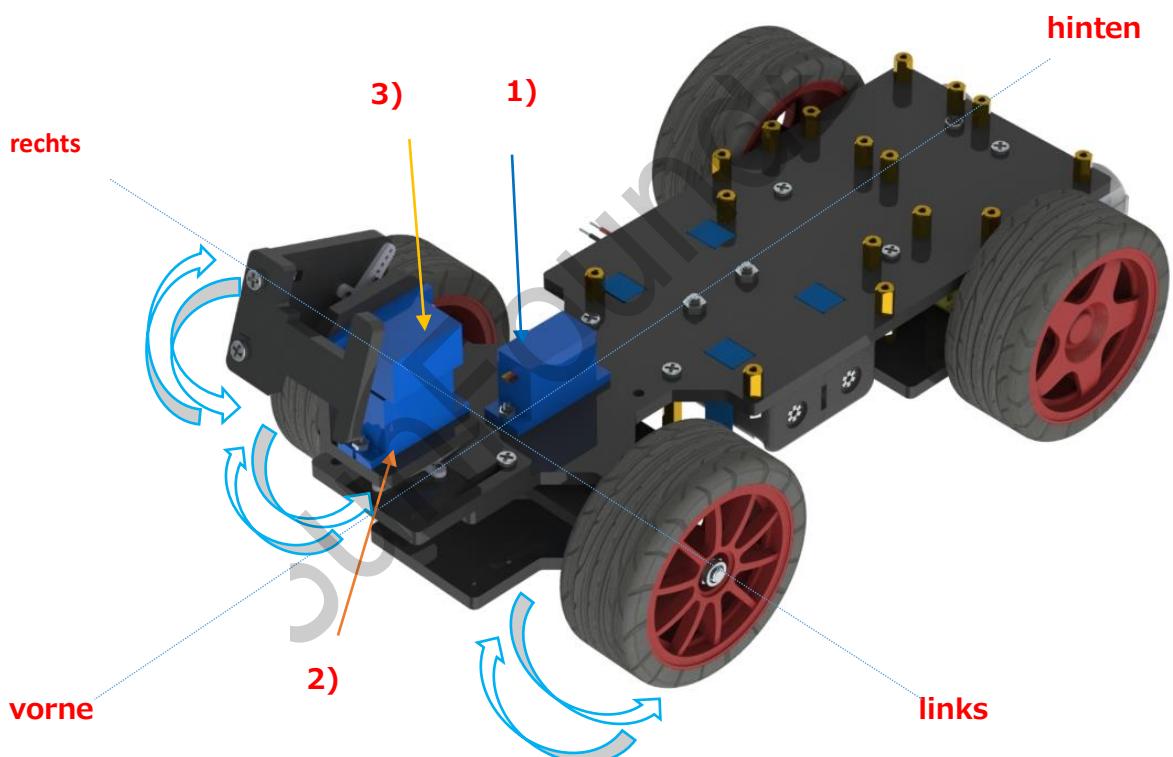
- b) Die zusammengesetzten Teile sollten, wie im unterem Abb. dargestellt aussehen.



- b) Vor der nächsten Montage, sollten Sie die Lenkbarkeit des Servos überprüfen.
- 1) Zuerst drehen Sie die Vorderräder nach rechts und links und achten auf den höchstmöglichen Grad der Drehung: die Räder sollten auf beiden Seiten im gleichen Grad drehbar sein.
  - 2) Drehen Sie am Schwenkservo auf dem Objektträger nach rechts und links, um zu versichern, dass es normal funktioniert – prüfen Sie es nochmal genau durch das Drehen von links nach rechts, oder umgekehrt.
  - 3) Dasselbe mit dem Neigungsservo. Prüfen Sie es genau durch das Drehen von vorne nach hinten oder umgekehrt.

Falls einer der 3 Servos die vorher genannten Bedingungen nicht erfüllen sollte, nehmen Sie es auseinander und bauen es, wie in den **Servo-Einstellungen** beschrieben wieder zusammen.

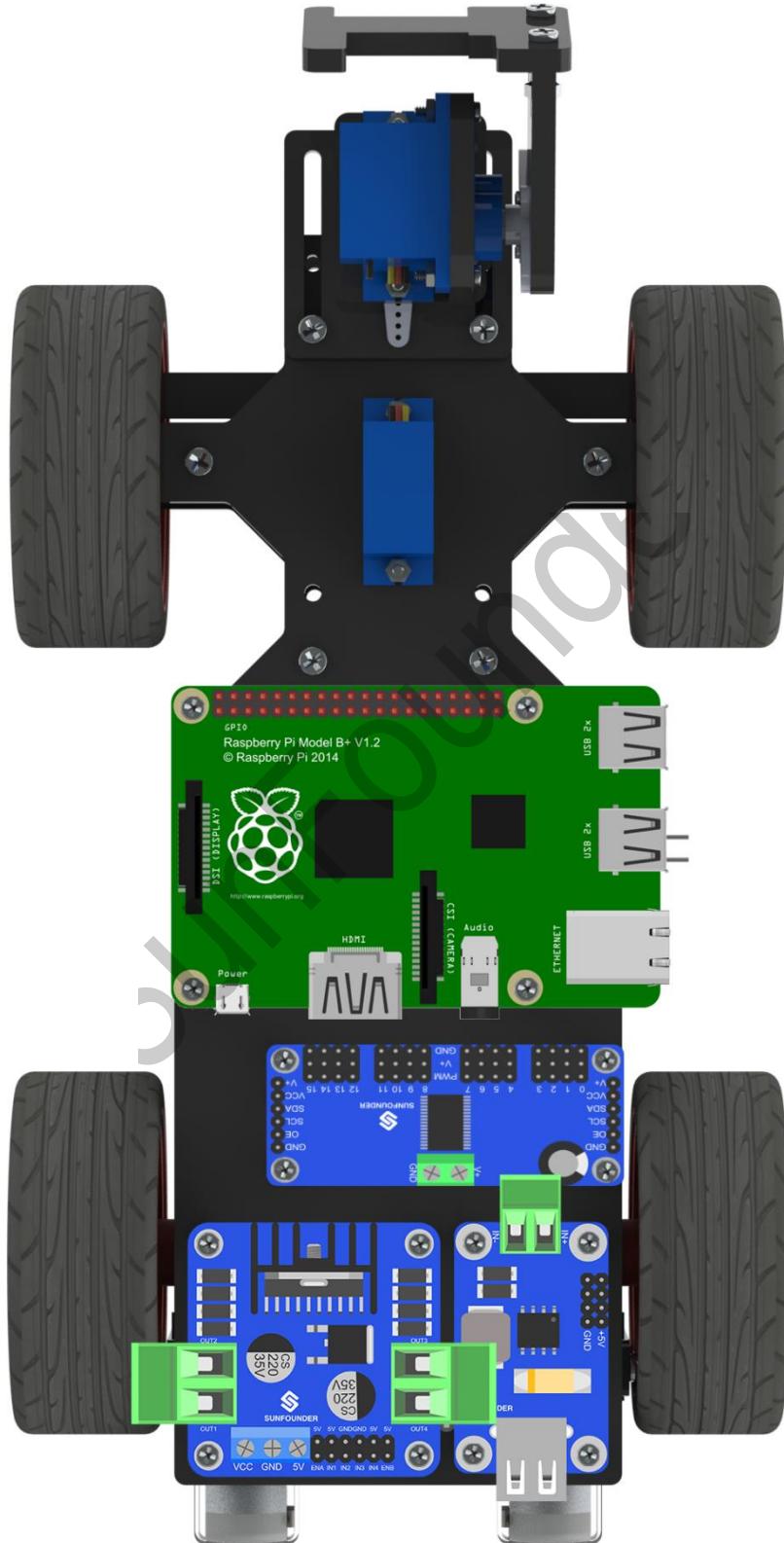
**Bitte folgen Sie GENAU der Anleitung in Servo-Einstellungen, um ein Durchbrennen des Servos zu verhindern!**



## ii. Einbau der Elektrokomponenten

- Befestigen Sie die Elektrokomponenten an das Auto mit M2.5\*6 Schrauben.

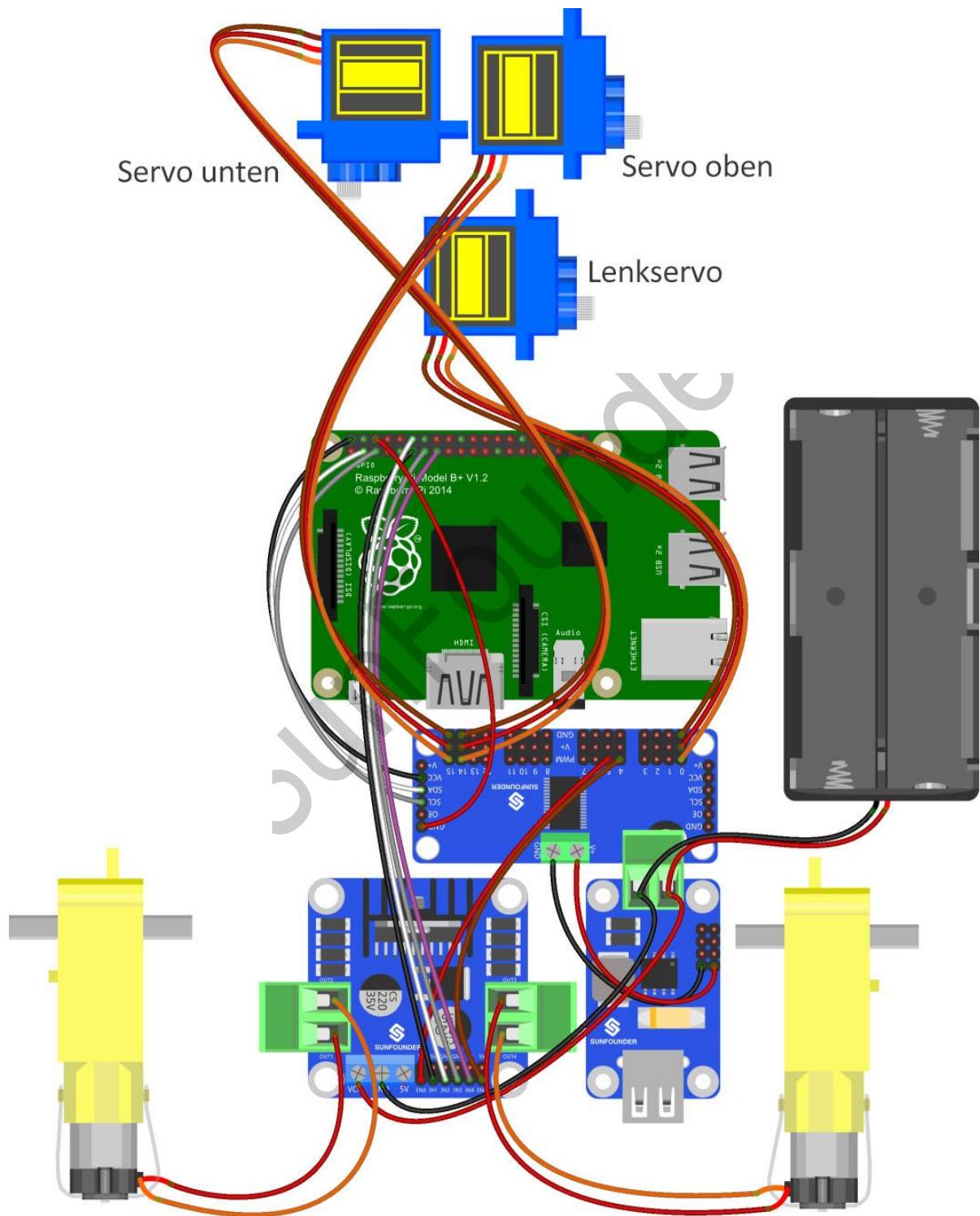
Siehe Abb.:



### iii. Schaltkreise verbinden

#### Vorschau:

Das Endergebnis wird folgendermaßen aussehen, wenn alle Schaltkreise miteinander verbunden sind. Es sieht komplizierter aus als es ist, also keine Sorgen! Sie werden in den nächsten Schritten eine detaillierte Anweisung vorfinden. Schritt für Schritt wird alles erklärt.

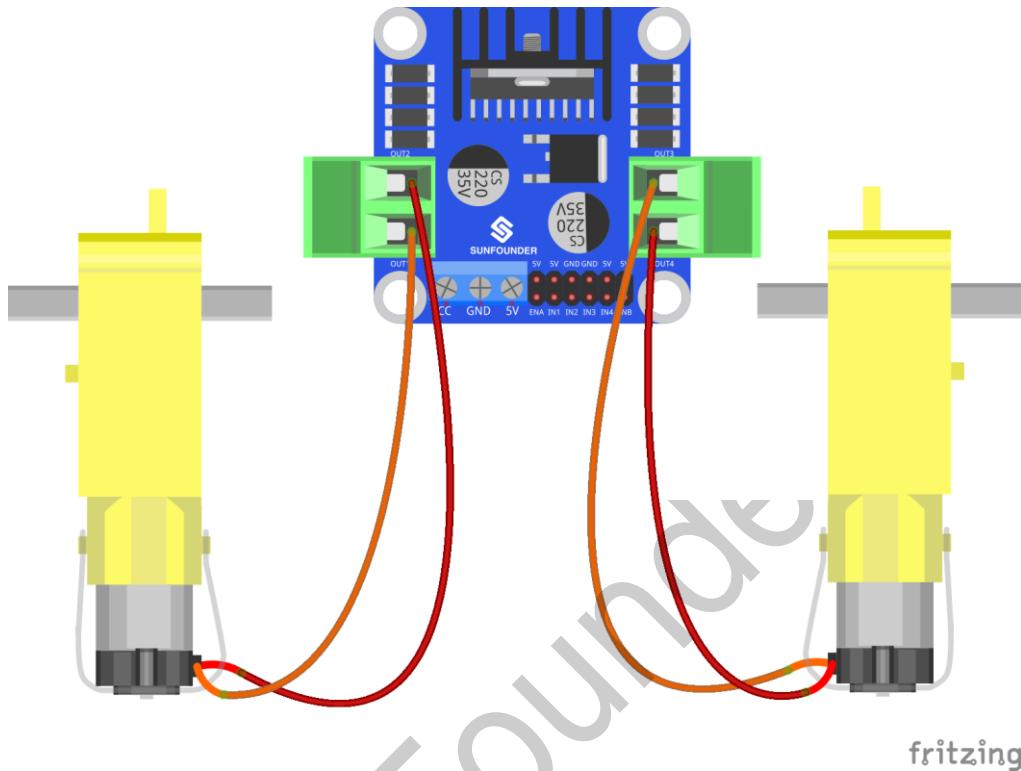


fritzing

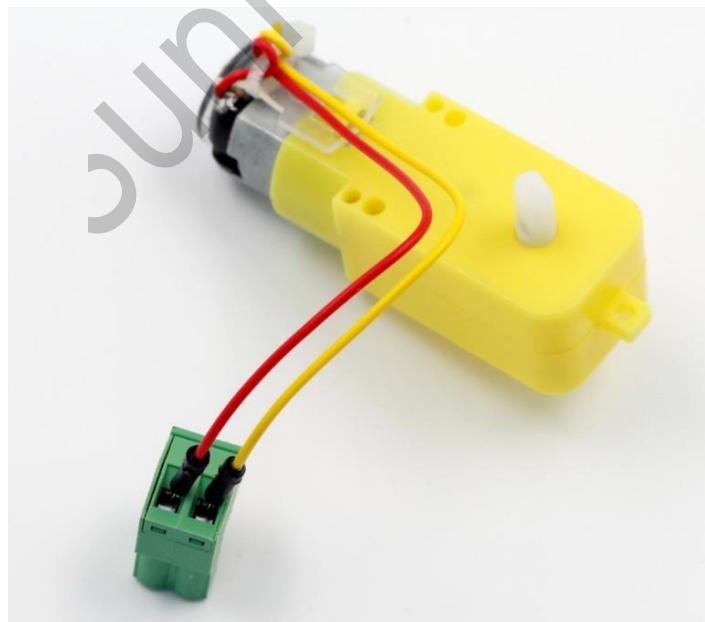
**1. Schritt:** Verbinden Sie die 2 Gleichstrommotoren mit dem Motortreiber.

Sie können den L-förmigen PCB Verbindung entfernen und später wieder anschliessen.

Hinweis: Nachdem alles fertig verkabelt ist, und das Auto in die entgegengesetzte, als die gewünschte Richtung fährt, müssen Sie nur die Kabeln der 2 Motoren umtauschen.

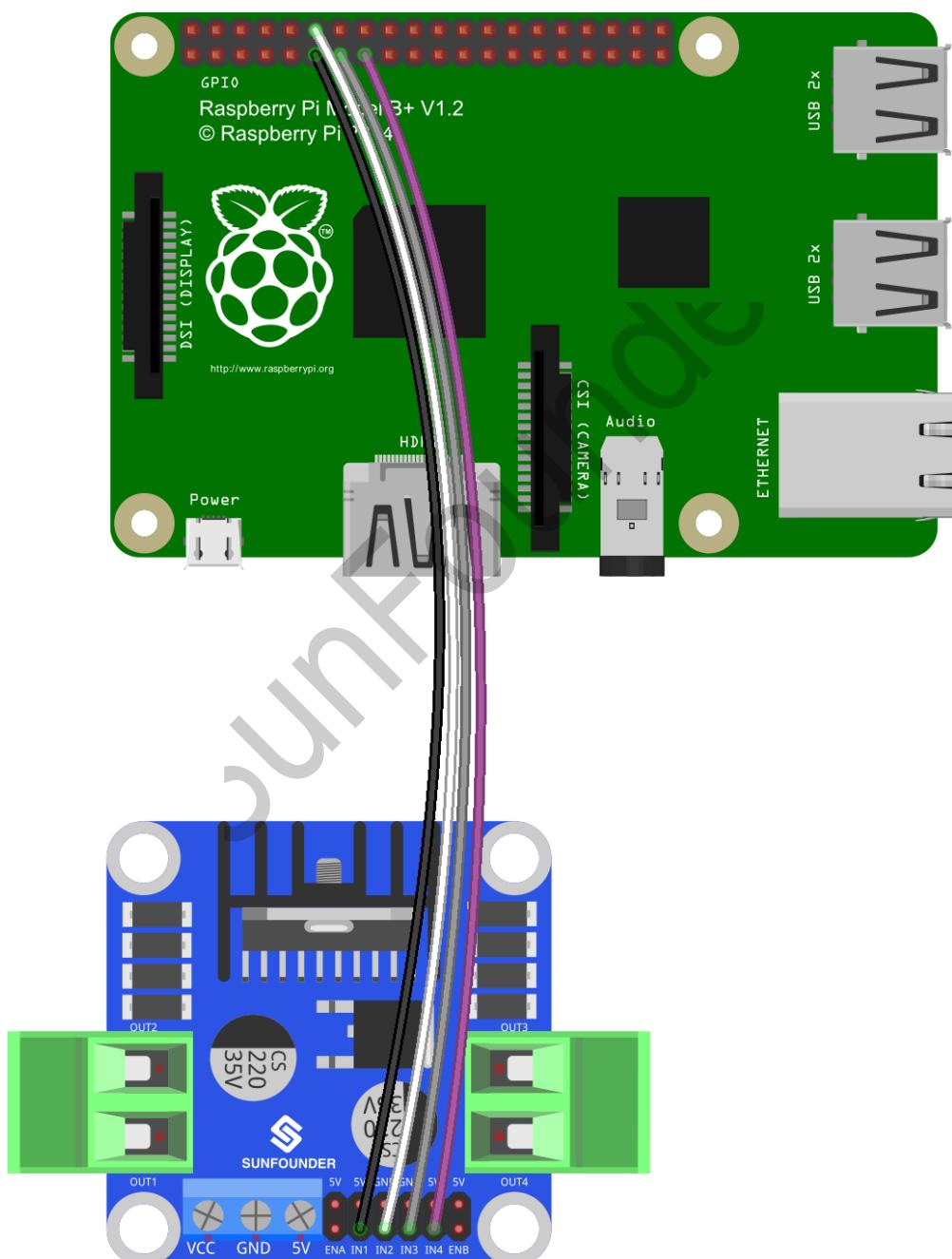


Nach der Verkabelung sollte es so aussehen:



**2. Schritt:** Verbinden Sie den Motortreiber mit dem Raspberry Pi GPIO Port basierend auf die folgende Tafel.

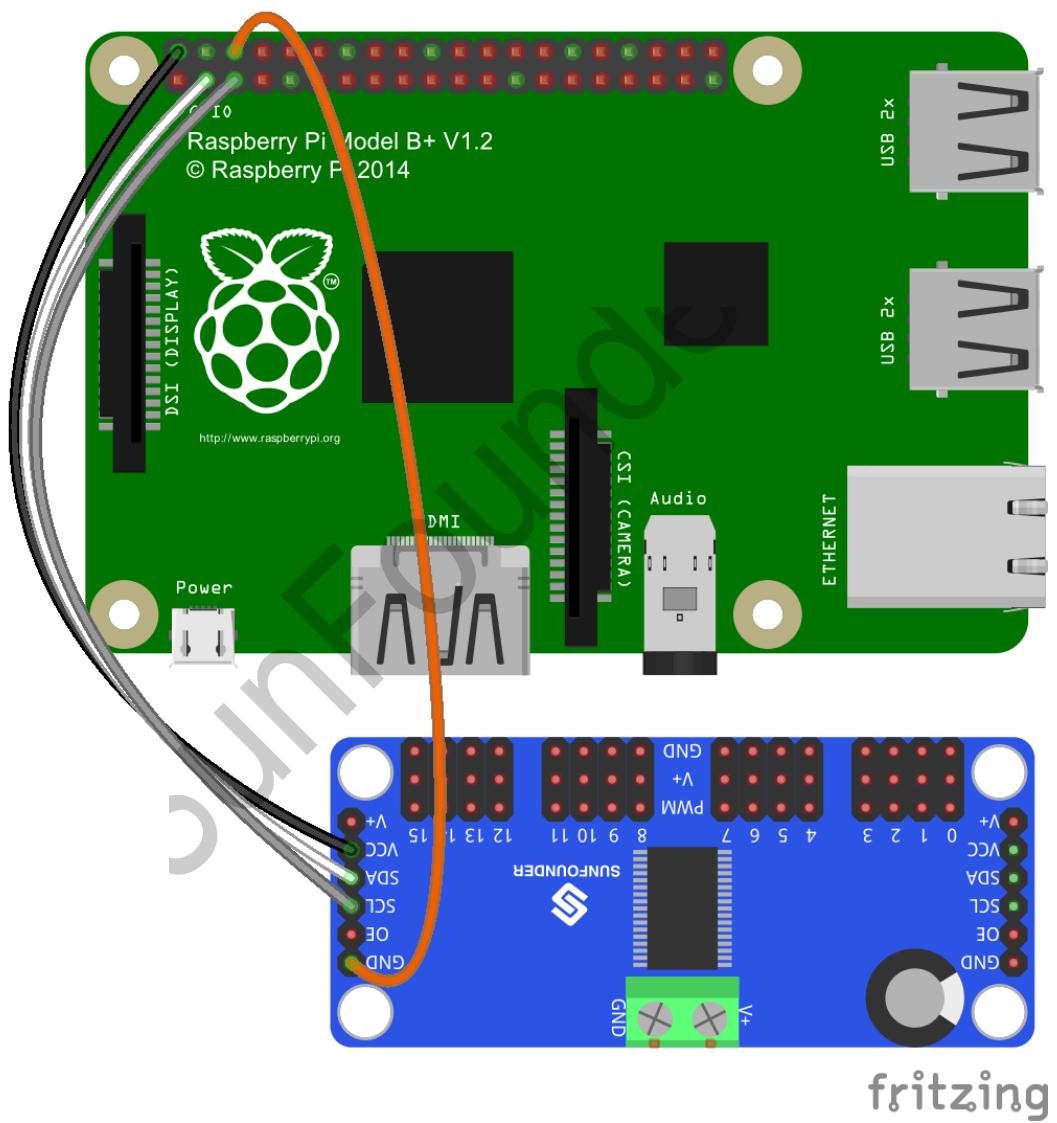
Raspberry Pi GPIO Port	Gleichstrommotortreiber
Pin11	IN1
Pin12	IN2
Pin13	IN3
Pin15	IN4



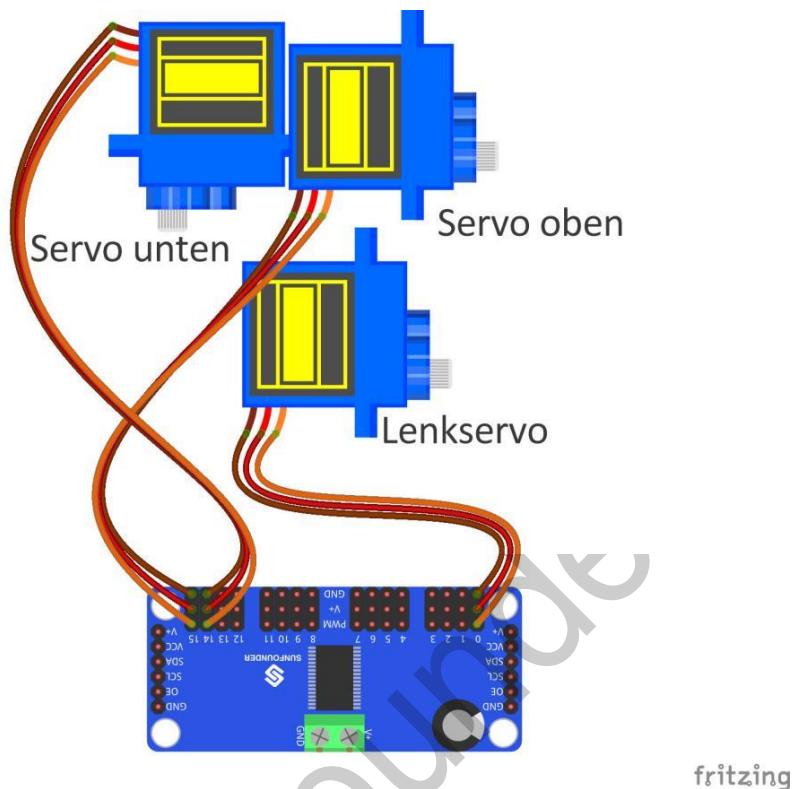
fritzing

**3. Schritt:** Verbinden Sie die Servosteuerung mit Raspberry Pi GPIO Port wie folgt:

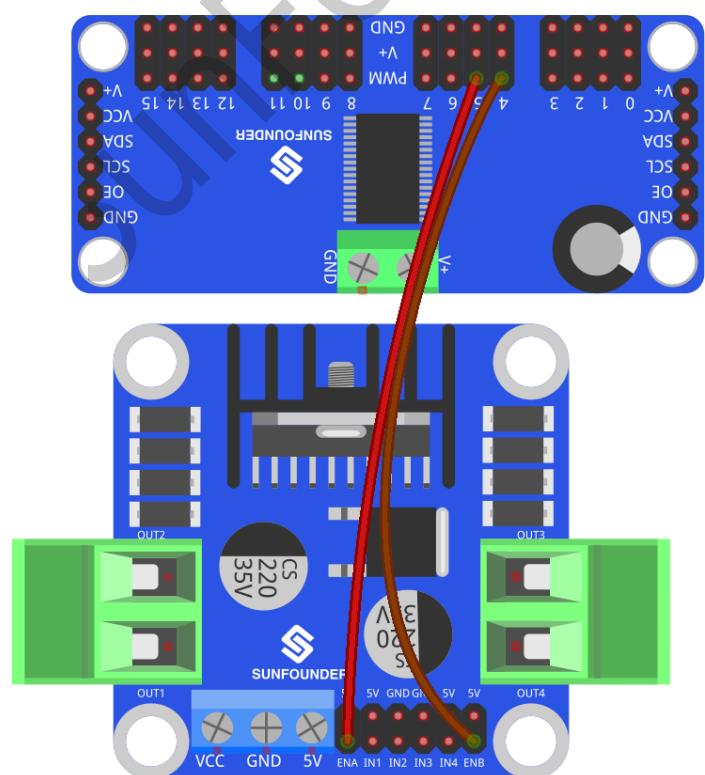
Raspberry Pi GPIO Port	Servosteuerung
Pin 2 (5V)	VCC
Pin 3 (SDA)	SDA
Pin 5 (SCL)	SCL
GND	GND



**4. Schritt:** Verkabeln Sie den Lenkservo, mit dem die Fahrtrichtung gesteuert wird, zu CH0 auf der Servosteuerung, und die untere/obere Servo, mit dem die Perspektive der Kamera gesteuert wird, zu CH14 und CH15. Siehe Abb.:

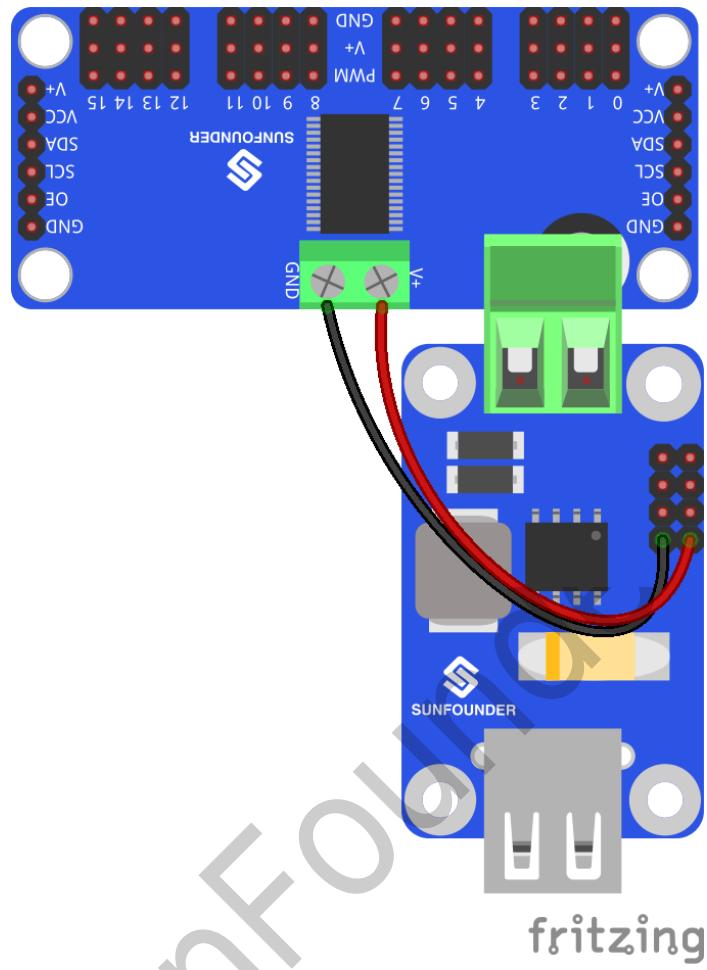


**5. Schritt:** Verkabeln Sie den Motorantrieber mit der Servosteuerung.

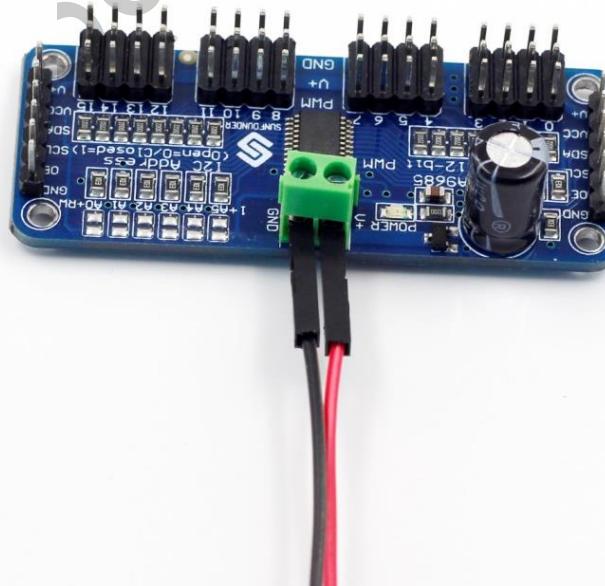


fritzing

**6. Schritt:** Verbinden Sie die Servosteuerung mit dem DC-DC Abwärtswandler. Für diese Verkabelung müssen Sie erst die Schrauben (siehe Abb.) etwas lockern, dann die Kabeln einstecken, und die Schrauben mit einem Schraubenzieher wieder befestigen.

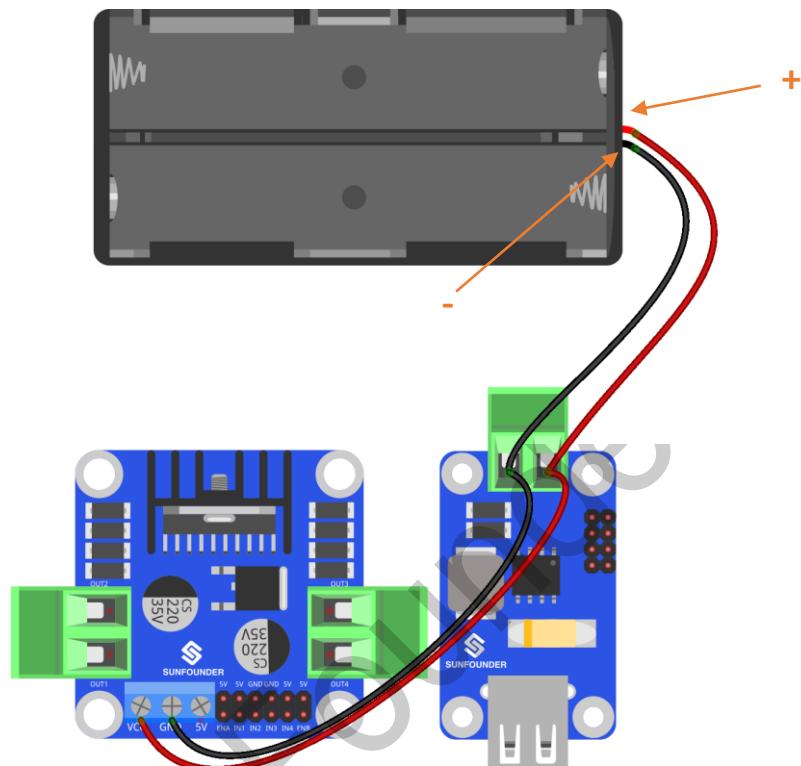


Nach der Verkabelung sollte es so aussehen:



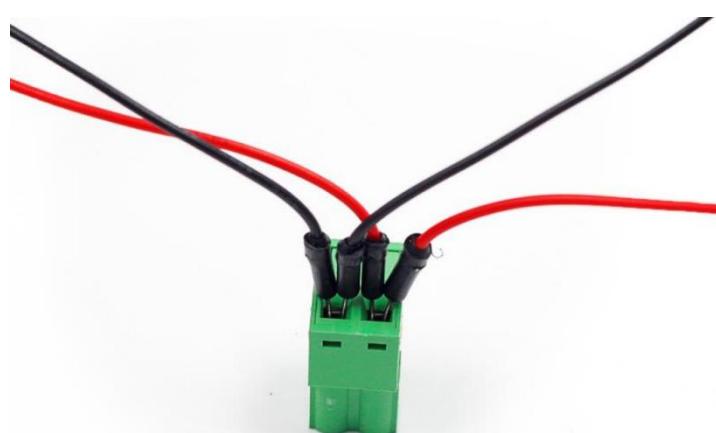
**7. Schritt:** Verbinden Sie den Batteriehalter mit dem DC-DC Abwärtswandler und dem Gleichstrommotortreiber.

**Hinweis:** Bitte legen Sie NOCH KEINE Batterien ein, sondern erst wenn Sie mit den ganzen Verkabelungen fertig sind. Und achten Sie darauf, dass Sie das rote Kabel (Strom) mit der Anode des Halters verbinden, und das schwarze Kabel (GND) mit der Kathode verbinden.

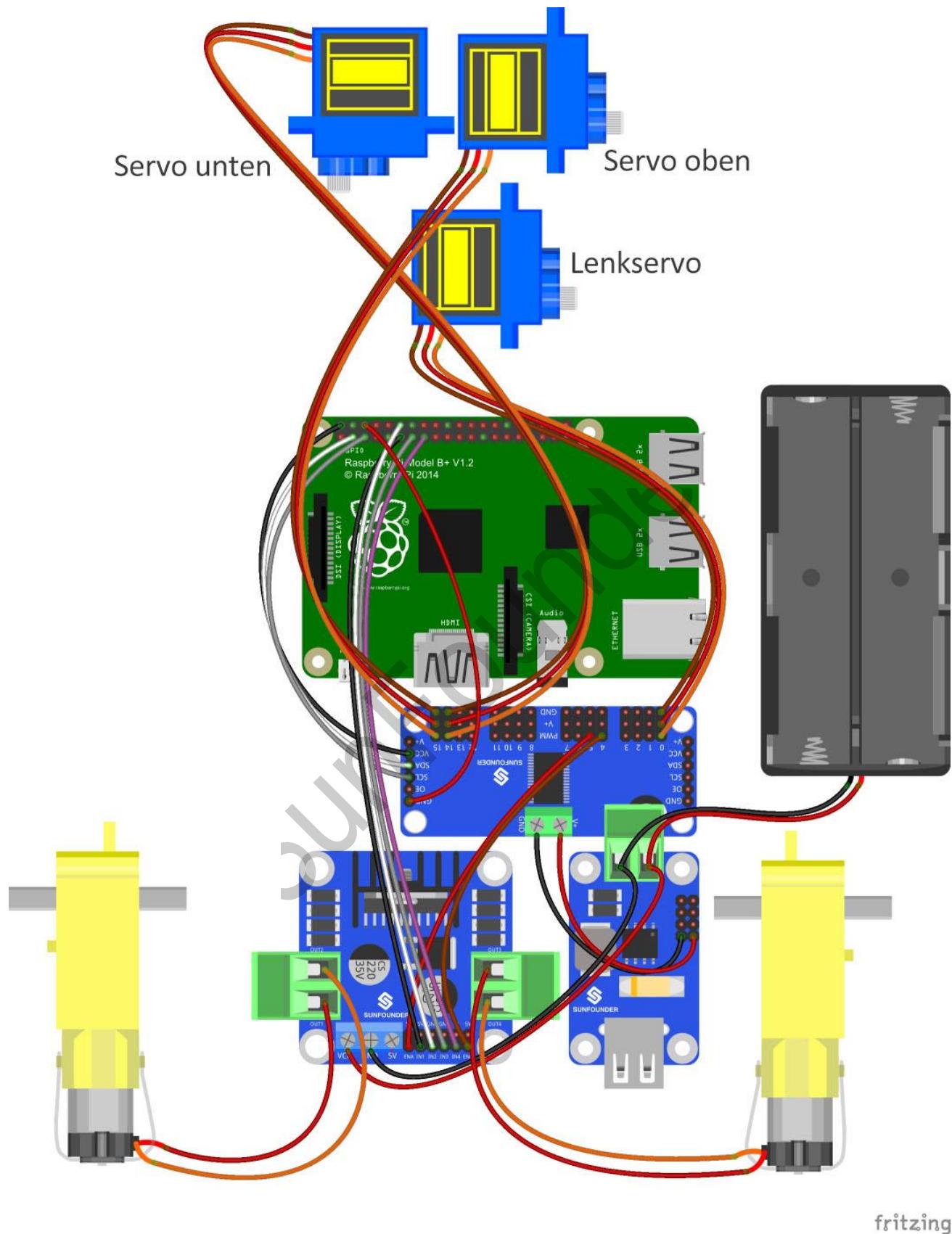


fritzing

An der Anschlussstelle des Abwärtswandlers, stecken Sie die 4 Kabelenden in die zwei Büchsen, jeweils zwei in einer. Beachten Sie: die roten Kabeln in die Anode und die schwarzen in die Kathode. Stecken Sie AUF KEINEN FALL einen roten mit einem schwarzen Kabel zusammen in eine Buchse, ansonsten entsteht ein Kurzschluss!



Das Ganze mit den Verkabelungen sollte so aussehen:



**8. Schritt:** An das Raspberry Pi verbinden Sie nun den DC-DC Abwärtswandler, USB Wi-Fi Adapter und die USB-Kamera.



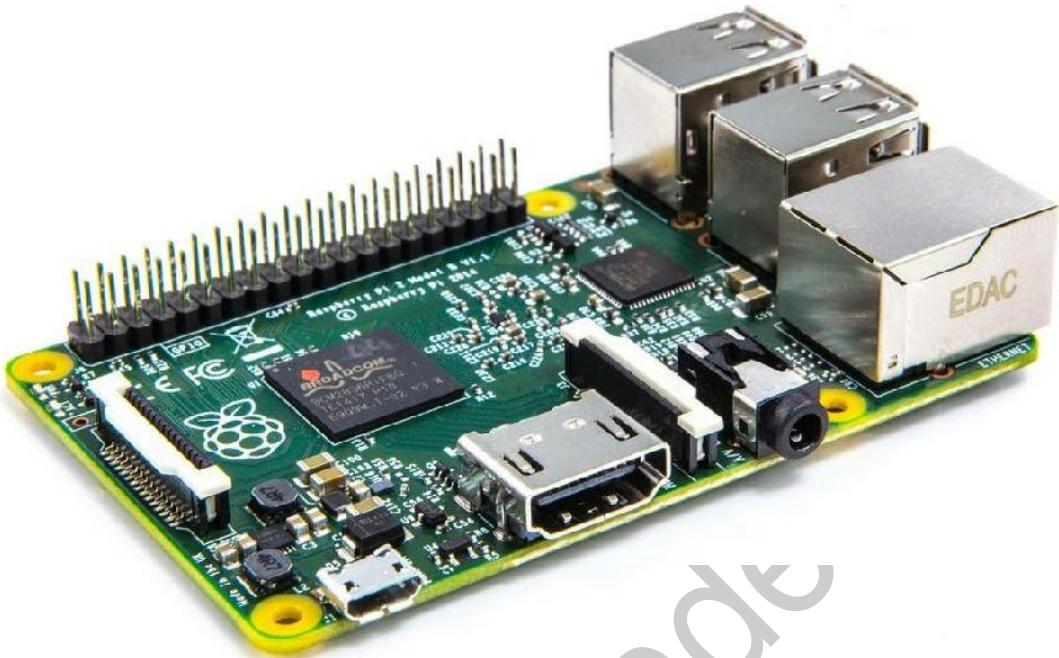
----- Glückwunsch! Die Verkabelung ist vollendet. -----

Das erfolgreich zusammengesetzte Auto sollte wie folgt aussehen:



# Grundlage der Elektrokomponenten

## i. Raspberry Pi

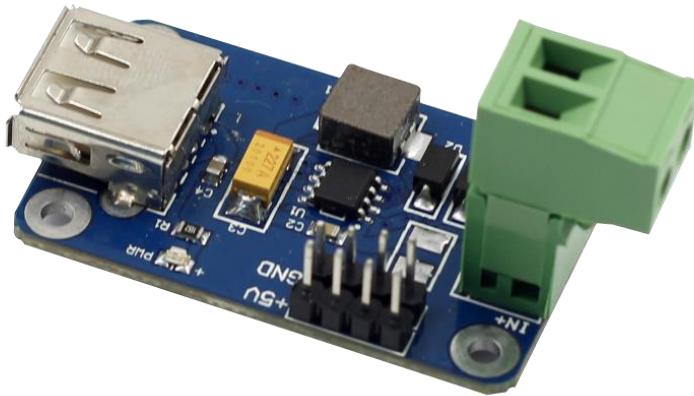


Das Raspberry Pi ist ein kostengünstiger, kreditkartengrosser Computer, welches man an einem Computermonitor oder TV stecken kann, und ist verwendbar mit einer Standardtastatur und Maus. Dieses Gerät ermöglicht Menschen aller Altersstufen das Computing zu erkunden und Programmsprachen wie Scratch und Python zu erlernen. Es ist Leistungsfähig und kann , vom Surfen im Internet, Abspielen von Hochqualitätsfilmen bis hin zu Kalkulationstabellen, Textverarbeitungen und Spiele spielen, alles was man von einem Desktop Computer erwarten würde.

Darüber hinaus ist Raspberry Pi in der Lage mit der Außenwelt zu interagieren und wird in vielen digitalen Maker-Projekten verwendet, von Musikmaschinen und Eltern-Detektoren bis hin zu Wetterstationen und zwitschernden Vogelhäusern mit Infrarotkameras.

In diesem Set benutzen wir Raspberry Pi als Core-Controller von Treibgeräten wie Gleichstrommotor und Servo. Mit Hilfe des Gerätes und einer Kamera lassen sich Videodaten in Echtzeit aufzeichnen und über das Wi-Fi Netzwerk übermitteln.

## ii. DC-DC Abwärtswandler



Dieses Modul wurde basierend auf dem Chip XL1509 entwickelt. Es wandelt den Output der Batterie von 7.4V zu 5V, so dass Raspberry Pi und das Servo mit ausreichendem Strom versorgt werden. Als ein DC zu DC Abwärtswandler-IC (integrierter Schaltkreis), weist der Chip eine Eingangsspannung von 4.5V bis 40V auf und erzeugt eine Ausgangsspannung von 5V bei einer Stromstärke von 2A. Bitte beachten Sie: nur wenn die Eingangsspannung bis zu 6.5V beträgt, kann eine 5V Ausgangsspannung unterstützt werden.

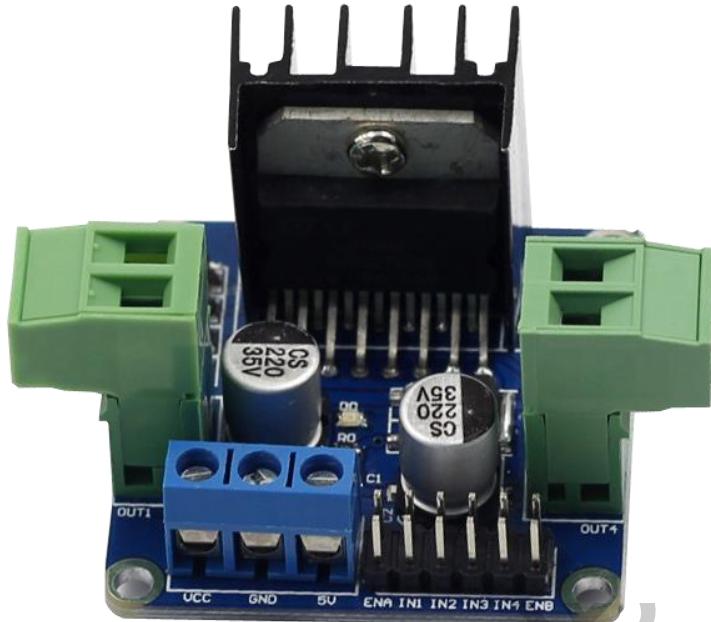
## iii. Servo



Bei diesem Smart Car, steuert ein Servo die Fahrtrichtung des Autos und die beiden anderen die Bewegung der Kamera zwischen X- und Y-Achse, womit die Abdeckung der Kamera definiert ist. Ein Servo ist ein automatisches Regelsystem, zusammengesetzt aus Gleichstrommotor, Untersetzungsgetriebe, Sensor und Regelkreis. Es definiert den Drehwinkel von der Ausgangsachse durch die Übertragung spezifischer PWM-Signale.

Im allgemeinen unterstützt das Servo den maximalen Drehwinkel von der Achse (etwa 180°). Es unterscheidet sich von einem gewöhnlichen Gleichstrommotor durch die Rotation: Gleichstrommotoren drehen sich in einem Vollkreis wobei sich Servos nur in einem bestimmten Grad drehen. Außerdem wird das erstere durch seine Vollkreisdrehung für die Energieversorgung eingesetzt, während das letztere nur für die Drehwinkelsteuerung eines Objektes verwendet wird (Gelenke eines Roboters).

#### **iv. Gleichstrommotortreiber**



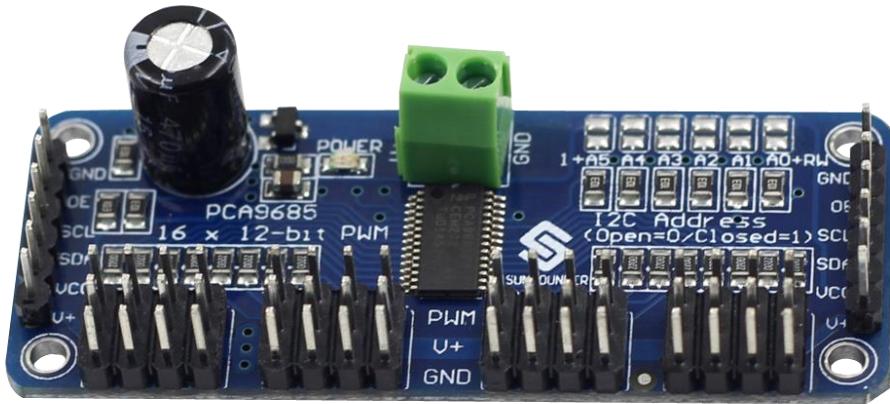
Wie der Name schon sagt, wird das Modul zum Antrieb von Gleichstrommotoren verwendet. Der Treiber wurde basierend auf L298N entwickelt. Als ein Hochspannungs- und Großstromchip für Motortreiber, gekapselt mit 15 Pins, besitzt es eine maximale Betriebsspannung von 46V und erreicht im Instant einen Spitzenstromspannung von 3A, bei einem Betriebsstrom von 2A und einer Nennleistung von 25W. Somit ist es vollständig in der Lage, 2 Gleichstrommotoren mit Niedrigleistung anzutreiben.

#### **v. USB Wi-Fi Adapter**



Der Adapter hilft dem Raspberry Pi sich mit einem Wi-Fi Hotspot zu verbinden.

## vi. Servosteuerung



Die Servosteuerung wurde basierend auf PCA9685 gebaut. PCA9685 ist eine 16-Kanal LED-Steuerung mit I2C-Busschnittstelle. Das Auflösungsverhältnis jedes Kanals beträgt 12 Bits ( $2^{12}=4096$  Pegel). Die Steuerung arbeitet in einer Frequenz zwischen 40Hz und 1000Hz und das Tastverhältnis kann in einem Bereich von 0 to 100% eingesetzt werden. Es überliefert PWM-Signale an das Servo und steuert somit auch den Drehwinkel. Das Modul steuert auch das Arbeitszyklus der Rechteckwellen, die vom Kanal 14 und 15 ausgesendet werden, was zur Regulation der Drehzahl des Gleichstrommotors dient, um so die Geschwindigkeit des Fahrzeugs zu kontrollieren.

## vii. 18650\*2 Batteriehalter



Um die Module und Servos am Auto mit Strom zu versorgen, benötigen wir 18650 Batterien.

**Hinweis:** Bitte achten Sie auf die Beschriftungen von Kathode (-) und Anode (+) innerhalb des Batteriehalters (nahe am Pol). Dementsprechend werden die 18650 Batterien eingelegt: Batteriekathode zum Halter -, und Batterieanode zum Halter +.

# Zugehörige Software

## i. Download und Installation des Raspbian auf einer TF Karte

Wenn Sie das Raspbian System schon installiert haben, dann können Sie diesen Schritt überspringen.

Suchen Sie die Installationsanleitung für Raspbian auf der Raspberry Pi Webseite <https://www.raspberrypi.org/downloads/>. Downloaden Sie das Raspbian auf Ihre TF Karte und installieren Sie es.

Nach der Installation, benötigen Sie möglicherweise grundlegende Einstellungen für das Raspberry Pi. Sehen Sie sich dafür die Anleitung auf unserer Webseite [www.sunfounder.com](http://www.sunfounder.com) an.

## ii. Holen Sie sich den Quellcode

**1. Schritt** Den Quellcode direkt aus Github auf Ihr Raspberry Pi herunterladen.

```
cd ~  
git clone https://github.com/sunfounder/Sunfounder_Smart_Video_Car_Kit_for_RaspberryPi.git
```

**2. Schritt** Den Quellcode direkt aus Github auf Ihr Linux herunterladen.

### 1. Methode

Öffnen Sie ein Terminal in Ihrem Linux. Downloaden Sie **git**:

Für Ubuntu/Debian:

```
sudo apt-get install git-core git
```

Für Redhat/Fedora:

```
sudo yum install git-core git
```

Kopieren Sie sich die Quelle aus github:

```
git clone https://github.com/sunfounder/Sunfounder_Smart_Video_Car_Kit_for_RaspberryPi.git
```

### 2. Methode

Falls Ihr Linux die Installation von **git** durch **yum** oder **apt-get** nicht unterstützen sollte, dann aus folgendem Link herunterladen:

[https://github.com/sunfounder/Sunfounder\\_Smart\\_Video\\_Car\\_Kit\\_for\\_RaspberryPi](https://github.com/sunfounder/Sunfounder_Smart_Video_Car_Kit_for_RaspberryPi)

oder nach Sunfounder auf Github suchen und finden Sie folgende Repository:  
[Sunfounder\\_Smart\\_Video\\_Car\\_Kit\\_for\\_RaspberryPi](https://github.com/sunfounder/Sunfounder_Smart_Video_Car_Kit_for_RaspberryPi).

Klicken Sie auf **Download ZIP**, wie folgt:

Sunfounder Smart Video Car Kit for Raspberry Pi <http://www.sunfounder.com> — Edit

110 commits 2 branches 0 releases 2 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

Cavon add trouble shooting no  
client merge cali\_client\_win to cali\_client  
datasheet modify this file  
html\_server fix format  
mjpg-streamer writing calibration  
server modified pwm driver to PCA9685.py  
.gitignore ignore config  
README.md add trouble shooting no

Clone with HTTPS Use SSH  
https://github.com/sunfounder/Sunfounder\_Smart\_Video\_Car\_Kit\_for\_RaspberryPi

Open in Desktop Download ZIP

Nach dem Download:

- Gehen Sie zur Datei und entpacken es durch **Extract here**
- Oder entpacken Sie es durch:

```
unzip Sunfounder_Smart_Video_Car_Kit_for_RaspberryPi-master.zip
```

Dann ändern Sie den Ordnernamen

```
mv ./Sunfounder_Smart_Video_Car_Kit_for_RaspberryPi-master ~/Sunfounder_Smart_Video_Car_Kit_for_RaspberryPi
```

### iii. Softwareumgebung

#### Ausführung auf PC

##### Bereiten Sie den PC für den Fernzugriff vor

Installieren Sie `python-tk` für das Remote Interface

```
sudo install apt-get install python-tk
```

Für Windows:

Gehen Sie auf die Python Webseite [www.python.org](http://www.python.org), dort finden Sie die neueste Python 2 vor und installieren Sie es. Nach der Installation, STARTEN Sie den Computer NEU.

## Ausführung auf Raspberry Pi

### Installation von python-dev, python-smbus

Installieren Sie `python-dev` und `python-smbus`:

```
sudo apt-get update  
sudo apt-get upgrade  
sudo apt-get install python-dev  
sudo apt-get install python-smbus
```

### I2C-Port Einstellungen

Führen Sie folgenden Befehl aus, um **Raspberry Pi Software Konfigurations-Tool (raspi-config)** zu öffnen

```
sudo raspi-config
```

Aktivieren Sie I2C:

Wählen Sie **Advanced Options => I2C => <Yes> => <Ok> => <Yes>**

Wählen dann **<Finish>** und schließen das Fenster.

Wenn eine Meldung des Neustarts erscheint, klicken Sie auf **<No>**. Vor dem Neustart müssen noch einige Konfigurationen durchgeführt werden.

## MJPG-Streamer

### Einführung

Die Erfassung und Übertragung von Videodaten durch das SunFounder Smart Video Car erfolgt auf der Basis des MJPG-Streamers.

Der MJPG-Streamer besteht aus Kommandozeilenanwendungen, die das JPG-Frame von einem einzigen Eingabe-Plugin zu mehreren Ausgabe-Plugins kopiert. Es kann dazu verwendet werden JPEG-Dateien über ein IP-basierendes Netzwerk, von der Webcam zu einem Viewer wie Firefox, Cambozola und Videolanclient oder sogar über Windows Mobilgeräte mit einem TCPMP-Player zu streamen.

Es wurde für eingebettete Systeme mit begrenzten Ressourcen in Bezug auf RAM und CPU geschrieben. Sein Ursprung "uvc\_streamer" wurde geschrieben, da Linux-UVC kompatible Kameras direkt JPEG-Daten erzeugen, somit eine schnelle und performante MJPEG-Streams ermöglichen, sogar auch vom eingebetteten Systemen wie OpenWRT. Das Eingangsmodul "input\_uvc.so" erfasst solche JPG-Frames von einer angeschlossenen Webcam.

### Installation

Verbinden Sie das USB-Kamera mit Raspberry Pi, und führen das Kommando `lsusb` aus. Das GEMBIRD repräsentiert die USB-Kamera; wenn es auf dem Bildschirm angezeigt wird, bedeutet es, dass es vom System angenommen wurde.

```
lsusb
```

Sie werden folgendes sehen:

```
pi@raspberrypi:~ $ lsusb
Bus 001 Device 004: ID 1908:2310 GEMBIRD
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp. SMSC9512/9514 Fast Ethernet Adapter
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Prüfen Sie, ob der Treiber für die Kamera normal funktioniert:

```
ls /dev/vid*
```

Sie werden folgendes sehen::

```
pi@raspberrypi:~ $ ls /dev/vid*
/dev/video0
```

Wenn /dev/video0 gedruckt wird, ist der Treiber im Normalzustand.

Dann installieren Sie die benötigte Software, wie folgt:

```
sudo apt-get install subversion
sudo apt-get install libv4l-dev
sudo apt-get install libjpeg8-dev
sudo apt-get install imagemagick
```

Kompilieren Sie den Quellcode vom MJPG-Streamer:

```
cd /home/pi/Sunfounder_Smart_Video_Car_Kit_for_RaspberryPi/mjpg-streamer/mjpg-streamer
sudo make USE_LIBV4L2=true clean all
```

Installieren Sie:

```
sudo make DESTDIR=/usr install
```

## Testen

Tippen Sie `sudo sh start.sh` ein und klicken auf Enter.

Anschließend tippen Sie die Adresse (ersetzen Sie 192.168.0.126 mit Ihrer Raspberry Pi IP-Adresse) in die Adressleiste auf Ihrem Browser ein (Firefox wird empfohlen):

<http://192.168.0.126:8080/stream.html>

DHinteren Sie auf **Enter** und Sie werden die Echtzeitaufzeichnung von der Kamera auf Ihrem Bildschirm sehen können.

192.168.0.126:8080/stream.html

## MJPG-Streamer Demo Pages

a ressource friendly streaming application

Home  
Static  
Stream  
Java  
Javascript  
VideoLAN  
Control

Version info:  
v0.1 (Okt 22, 2007)

# Display the stream

## Hints

This example shows a stream. It works with a few browsers like Firefox for example. To see a simple example click [here](#). You may have to reload this page by pressing F5 one or more times.

## Source snippet

```

```



© The [MJPG-streamer team](#) | Design by [Andreas Viklund](#)

## iv. Kalibrierung

### Vorbereitung

#### Serverausführung

Führen Sie den Kalibrierungsserver auf Ihrem Raspberry Pi aus und warten Sie bis eine Verbindung vom Client hergestellt wird.

Stellen Sie sicher, dass der Stromkreis richtig angeschlossen ist. Schalten Sie nun das Smart Car ein und öffnen das Terminal im Linux. Verbinden Sie es mit Raspberry Pi über SSH. Danach gehen Sie auf das Datenverzeichnis von Sunfounder\_Smart\_Video\_Car\_Kit\_for\_RaspberryPi/server und führen den Server cali\_server.py aus. Falls das Servo mit anormalen Geräuschen stecken bleibt, trennen Sie sofort alle Kabeln und wiederholen Sie die Anweisungen in den Servo-Einstellungen für die Kalibrierung, um weitere Schäden zu vermeiden.

```
cd ~/Sunfounder_Smart_Video_Car_Kit_for_RaspberryPi/server  
sudo python cali_server.py
```

```
pi@raspberrypi ~ $ cd Sunfounder_Smart_Video_Car_Kit_for_Ras  
pberryPi/server/  
pi@raspberrypi ~/Sunfounder_Smart_Video_Car_Kit_for_Raspberr  
yPi/server $ sudo python cali_server.py  
offset_x = 0  
offset_y = 0  
offset = 0  
turning0 = True  
turning1 = True  
Waiting for connection...
```

Der Inhalt zur Konfiguration wird angezeigt und die letzte Zeile sollte folgendermaßen lauten:  
**Waiting for connection...**

Nun könnte sich zu dieser Zeit das Auto ein bisschen bewegen (in die Original Position)

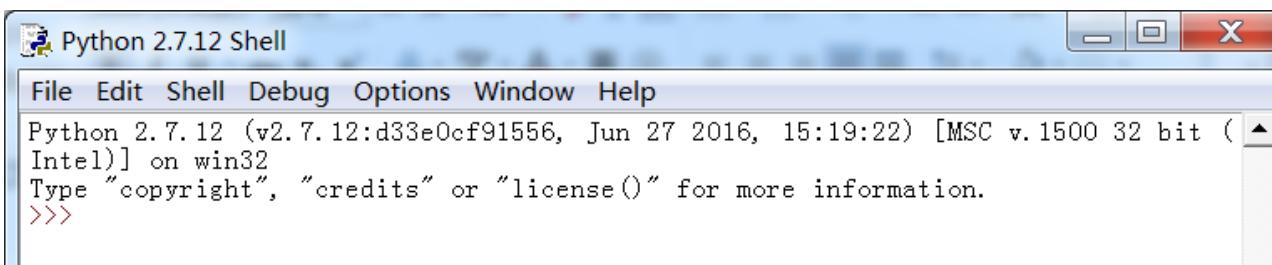
#### Ausführung vom Client

Führen Sie die Kalibrierung vom Client auf dem PC aus, um sich mit dem Raspberry Pi Server in Verbindung zu setzen.

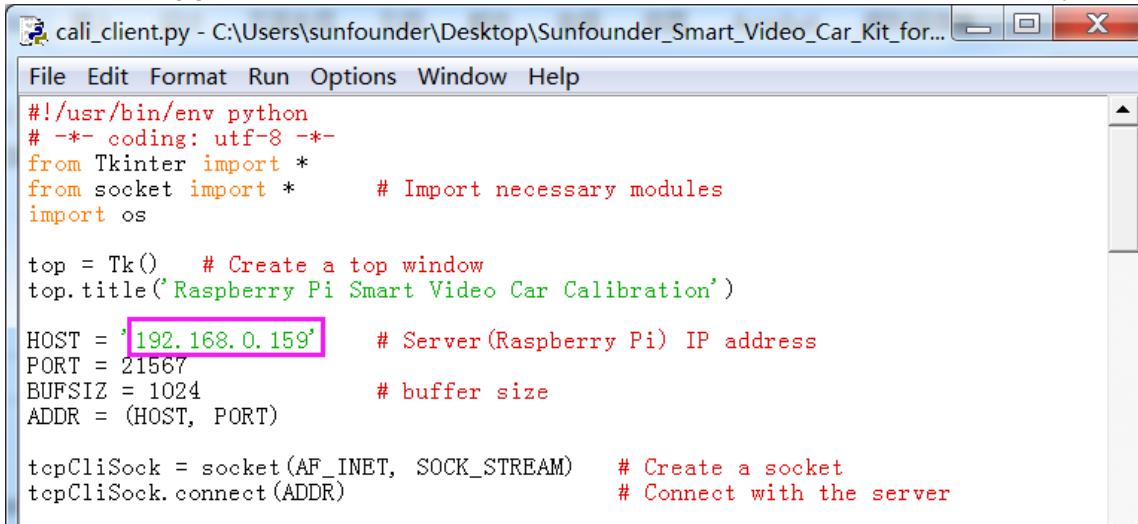
#### Für Windows:

Entpacken Sie das heruntergeladene Code-Paket.

Klicken Sie auf Ihrem Computer auf **Start**. Wenn Sie nun Python in die Suchleiste eintippen, werden Sie IDLE (Python GUI) finden. Sobald Sie auf dies klicken wird dieses Fenster erscheinen:



Öffnen Sie die Datei **File > Open > Sunfounder\_Smart\_Video\_Car\_Kit\_for\_RaspberryPi-master > client > cali\_client.py** und ändern Sie den Wert von **HOST** für die IP-Adresse des Raspberry Pis.



```
cali_client.py - C:\Users\sunfounder\Desktop\Sunfounder_Smart_Video_Car_Kit_for...
File Edit Format Run Options Window Help
#!/usr/bin/env python
# -*- coding: utf-8 -*-
from Tkinter import *
from socket import *      # Import necessary modules
import os

top = Tk()    # Create a top window
top.title('Raspberry Pi Smart Video Car Calibration')

HOST = '192.168.0.159'    # Server(Raspberry Pi) IP address
PORT = 21567
BUFSIZ = 1024            # buffer size
ADDR = (HOST, PORT)

tcpCliSock = socket(AF_INET, SOCK_STREAM)    # Create a socket
tcpCliSock.connect(ADDR)                      # Connect with the server
```

Nach der Änderung, speichern Sie die Datei, und klicken danach auf die **Run**-Option und wählen **Run Module** aus.

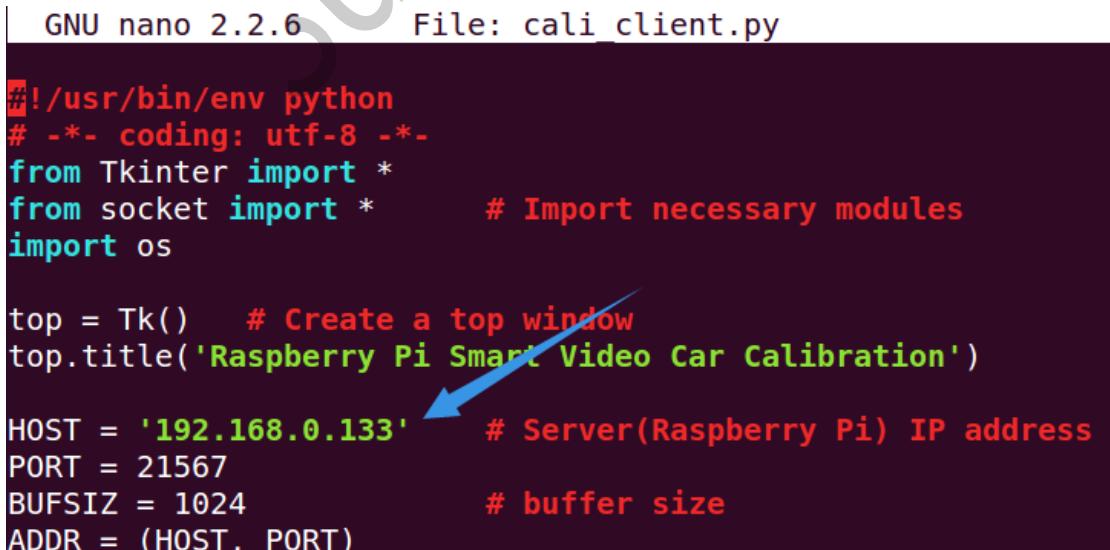
Für Linux:

Öffnen Sie ein zusätzliches Terminal im Linux (nicht über SSH auf Ihrem Pi). Suchen Sie den heruntergeladenen Sketch und editieren Sie *client/cali\_client.py*:

```
cd client/
sudo nano cali_client.py
```

```
pi@cavon:~$ cd Sunfounder_Smart_Video_Car_Kit_for_RaspberryPi/client/
pi@cavon:~/Sunfounder_Smart_Video_Car_Kit_for_RaspberryPi/client$ sudo nano cali_client.py
```

Suchen Sie den Wert von *HOST*:



```
GNU nano 2.2.6      File: cali_client.py

#!/usr/bin/env python
# -*- coding: utf-8 -*-
from Tkinter import *
from socket import *      # Import necessary modules
import os

top = Tk()    # Create a top window
top.title('Raspberry Pi Smart Video Car Calibration')

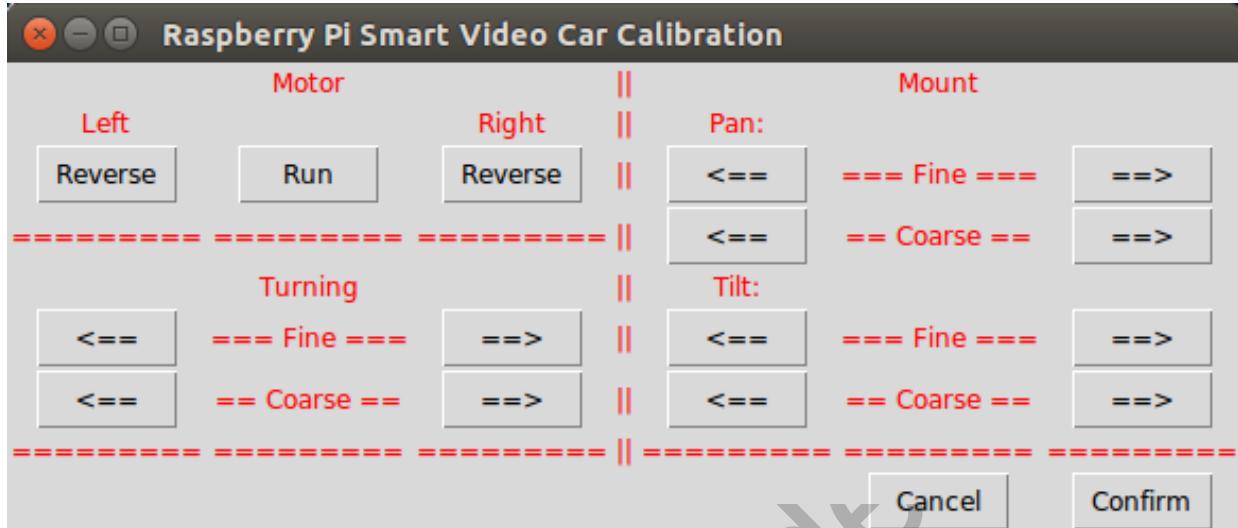
HOST = '192.168.0.133'    # Server(Raspberry Pi) IP address
PORT = 21567
BUFSIZ = 1024            # buffer size
ADDR = (HOST, PORT)
```

Tippen Sie Ihre eigene Adresse von Raspberry Pi dort ein (siehe blauer Pfeil). Betätigen Sie **Ctrl+O** für das Speichern und **Ctrl+X** für den Exit.

Führen Sie aus `cali_client.py` aus:

```
sudo python cali_client.py
```

**Unabhängig vom welchem System, Linux oder Windows**, sobald Sie `cali_client.py` ausführen wird dieses Fenster mit **Raspberry Pi Smart Video Car Calibration** (Kalibrierung des Smart Cars) erscheinen:



Im Terminal, welches per Fernzugriff mit Raspberry Pi verbunden ist, wird die IP Adresse vom PC angezeigt.

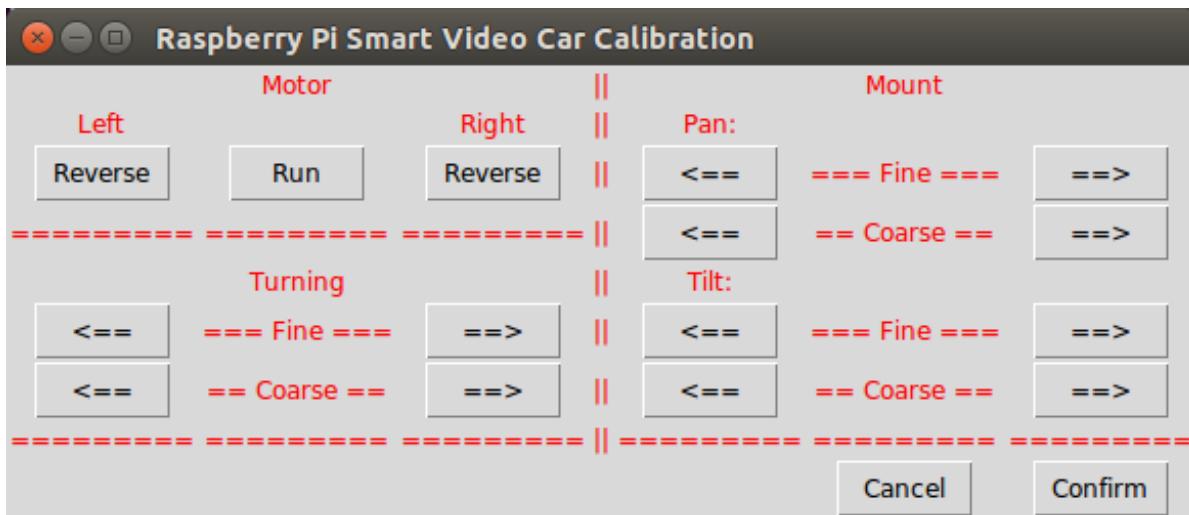
```
pi@raspberrypi ~/$ sudo python cali_server.py
offset_x = 0
offset_y = 0
offset = 0
turning0 = True
turning1 = True
Waiting for connection...
...connected from : ('192.168.0.124', 54220)
```

Nun können Sie mit der Kalibrierung beginnen. Davor nehmen Sie bitte die Verpackung und legen es senkrecht mit der Seitenfläche zuerst, auf dem Tisch. Setzen Sie das Auto auf die Verpackung und halten es waagerecht. Die Absicht ist, die Räder des Autos vom Tisch fern zu halten.

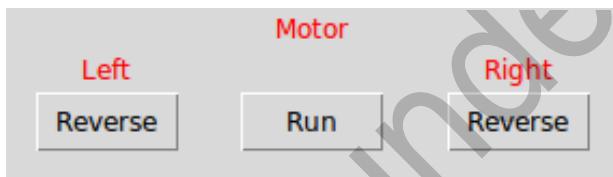
Standartgemäß sollten die Vorderräder gerade nach vorne gerichtet sein; die Kamera auf dem Neigungsservo sollte, unabhängig in welche Richtung der Schwenkservo geneigt ist, nach oben gerichtet sein. Wenn das nicht der Fall ist, können Sie die Richtung in **Raspberry Pi Smart Video Car Calibration** (siehe Abschnitt: Kalibrierung beginnen) einstellen.

## Kalibrierung beginnen

In der Kalibrierung UI gibt es drei Abteilungen: **Motor**, **Turning** und **Mount**.

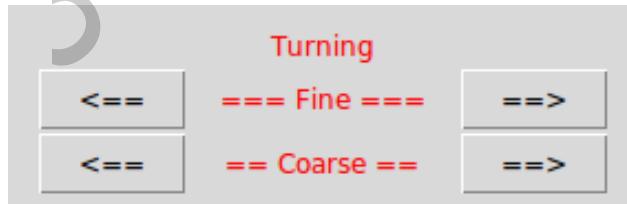


### Motor-Einstellungen



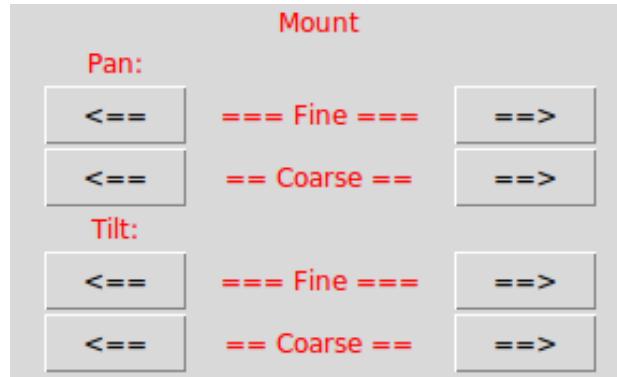
Klicken Sie auf **Run** und das Auto wird sich nach vorne bewegen. Prüfen Sie, ob beide Hinterräder zur gleichen Zeit sich bewegen. Wenn das nicht der Fall sein sollte, kann es an der Verkabelung liegen. Aber keine Sorge! Sie brauchen es nicht neu verkabeln; klicken Sie nur auf entsprechende Schaltfläche **Reverse** (Abb. oben). Und prüfen jetzt, ob die Räder sich nach vorne drehen. Wenn diese Schaltfläche normal funktioniert, klicken Sie wieder auf **Run**, um die Räder zum Halten zu bringen.

### Turning-Einstellungen



Zur gleichen Zeit sollten die Vorderräder genau nach vorne gerichtet sein. Wenn das nicht der Fall ist, müssen Sie einige Einstellungen durchführen. In der **Turning**-Abteilung klicken Sie auf Links- (**<==**) und Rechtspfeile (**==>**) in der oberen Reihe, um Feineinstellungen durchzuführen, und die darunter liegenden sind für Grobeinstellungen der Lenkrichtungen. Führen Sie die Einstellungen durch bis die Räder genau nach vorne zeigen. Nun können Sie das Auto auf dem Tisch stellen und klicken auf **Run**, um zu überprüfen, ob es sich nun geradlinig bewegen wird. Falls nicht, dann wiederholen Sie die Einstellungen zuvor.

## Mount-Einstellungen



Das Schwenkservo auf dem Auto sollte genau nach vorne gerichtet und die Kamera nach oben geneigt sein. Wenn das nicht der Fall ist, müssen Sie ähnliche Einstellungen wie zuvor vornehmen. In der **Mount**-Abteilung kann man das Schwenkservo und Neigungsservo einstellen. Wie zuvor gibt es für beide Fein- und Grobeinstellungen. Wiederholen Sie die Einstellungen bis es in die richtige Richtung geneigt ist.

Nachdem alle Einstellungen vorgenommen wurden, bestätigen Sie mit **Confirm**.

Und das Programm wird beendet.

Dann kehrt das Raspberry Pi zuHinter zum Status: **Waiting for connection...**



Betätigen Sie **Ctrl+C** für den Exit.

-----NUN IST IHR FAHRZEUG IST BEREIT FÜR EIN ABENTEUER!-----

## Raus auf die Strasse!

### Serverausführung

Führen Sie den TCP-Server auf dem Raspberry Pi aus, und warten bis der Client eine Verbindung mit dem Auto hergestellt hat.

Sie benötigen zwei Terminals für den Server, eins für die Ausführung von `tcp_client.py`, um Steuerungsbefehle für das Auto zu empfangen, und das andere für die Ausführung von MJPG-Streamer für die Video-Streams. Für den Serverstart in Windows, können Sie PuTTY benutzen, um sich per Fernzugriff in das Raspberry Pi zu loggen.

Öffnen Sie ein Terminal.

Das Raspberry Pi befindet sich möglicherweise noch im Serververzeichnis:

```
pi@raspberrypi ~/Sunfounder_Smart_Video_Car_Kit_for_RaspberryPi/server $
```

Falls nicht, dann gehen Sie zum Verzeichnis mit `cd`.

```
cd ~/Sunfounder_Smart_Video_Car_Kit_for_RaspberryPi/server
```

Dann führen Sie `tcp_server.py` aus:

```
cd ~/Sunfounder_Smart_Video_Car_Kit_for_RaspberryPi/server
```

Das Server-Programm auf Raspberry Pi wird ausgeführt bleiben und auf die Verbindung vom Client zu Raspberry Pi warten.

```
pi@raspberrypi ~/Sunfounder_Smart_Video_Car_Kit_for_RaspberryPi/server $ sudo python tcp_server.py
Waiting for connection...
```

Öffnen Sie ein zusätzliches Terminal. Loggen Sie sich ein in Raspberry Pi per Fernzugriff, und wechseln zu dem Verzeichnis, wo sich das Programm vom MJPG-Streamer befindet:

```
cd ~/Sunfounder_Smart_Video_Car_Kit_for_RaspberryPi/mjpg-streamer/mjpg-streamer
```

```
pi@cavon:~$ ssh pi@192.168.0.133
pi@192.168.0.133's password:
Linux raspberrypi 3.18.11+ #781 PREEMPT Tue Apr 21 18:02:18 BST 2015 armv6l
```

```
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.
```

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Jul 17 02:56:23 2015 from 192.168.0.118
pi@raspberrypi ~ $ cd Sunfounder_Smart_Video_Car_Kit_for_RaspberryPi/mjpg-streamer/mjpg-streamer
pi@raspberrypi ~/Sunfounder_Smart_Video_Car_Kit_for_RaspberryPi/mjpg-streamer/mjpg-streamer $
```

Führen Sie das Programm aus:

```
sudo sh start.sh
```

Die Videodatenerfassung sollte so starten:

```
UVCIIOC_CTRL_MAP - Error: Inappropriate ioctl for device  
mapping control for Pan Reset  
UVCIIOC_CTRL_MAP - Error: Inappropriate ioctl for device  
mapping control for Tilt Reset  
UVCIIOC_CTRL_MAP - Error: Inappropriate ioctl for device  
mapping control for Pan/tilt Reset  
UVCIIOC_CTRL_MAP - Error: Inappropriate ioctl for device  
mapping control for Focus (absolute)  
UVCIIOC_CTRL_MAP - Error: Inappropriate ioctl for device  
mapping control for LED1 Mode  
UVCIIOC_CTRL_MAP - Error: Inappropriate ioctl for device  
mapping control for LED1 Frequency  
UVCIIOC_CTRL_MAP - Error: Inappropriate ioctl for device  
mapping control for Disable video processing  
UVCIIOC_CTRL_MAP - Error: Inappropriate ioctl for device  
mapping control for Raw bits per pixel  
UVCIIOC_CTRL_MAP - Error: Inappropriate ioctl for device  
o: www-folder-path....: ./www/  
o: HTTP TCP port.....: 8080  
o: username:password.: disabled  
o: commands.....: enabled
```

Anschließend tippen Sie die Adresse (ersetzen Sie 192.168.0.xxx mit Ihrer Raspberry Pi IP-Adresse) in die Adressleiste auf Ihrem Browser ein (Firefox wird empfohlen):

<http://192.168.0.xxx:8080/stream.html>

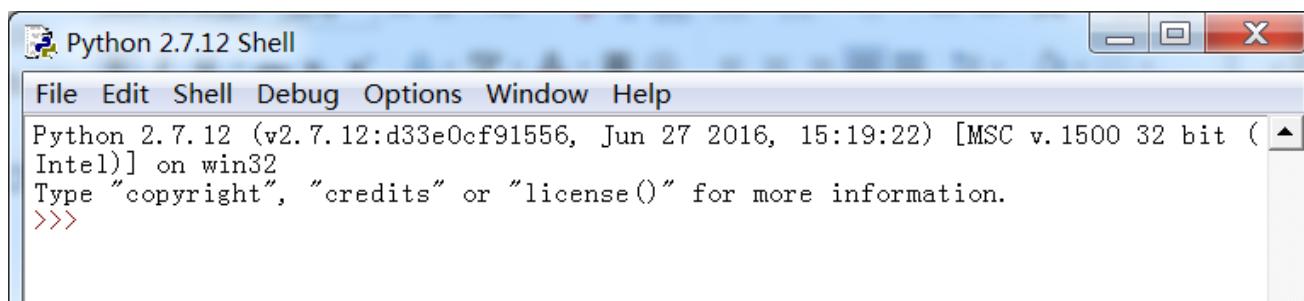
DHinteren Sie auf **Enter** und Sie werden die Echtzeitaufzeichnung von der Kamera auf Ihrem Bildschirm sehen können.

### Ausführung vom Client

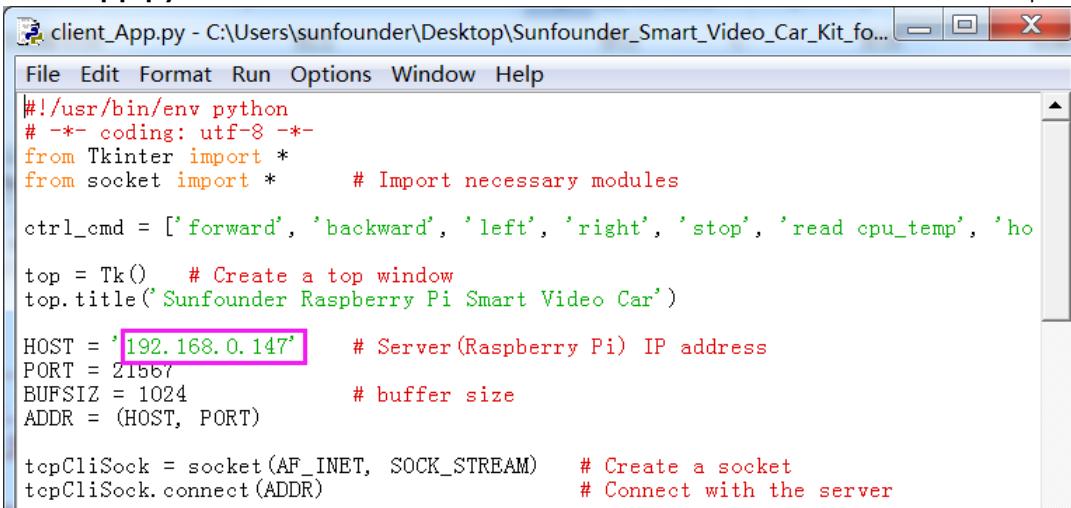
Führen Sie Client auf dem PC aus, um sich mit dem Server auf Raspberry Pi zu verbinden.

Für Windows:

Öffnen Sie IDLE



Öffnen Sie die Datei **File > Open > Sunfounder\_Smart\_Video\_Car\_Kit\_for\_RaspberryPi-master > client > client App.py** und ändern Sie den Wert von **HOST** für die IP Adresse des Raspberry Pis.



```

client_App.py - C:\Users\sunfounder\Desktop\Sunfounder_Smart_Video_Car_Kit_for_RaspberryPi\client\client_App.py
File Edit Format Run Options Window Help
#!/usr/bin/env python
# -*- coding: utf-8 -*-
from Tkinter import *
from socket import *      # Import necessary modules

ctrl_cmd = ['forward', 'backward', 'left', 'right', 'stop', 'read cpu_temp', 'home']

top = Tk()    # Create a top window
top.title('Sunfounder Raspberry Pi Smart Video Car')

HOST = '192.168.0.147'    # Server(Raspberry Pi) IP address
PORT = 21567
BUFSIZ = 1024            # buffer size
ADDR = (HOST, PORT)

tcpCliSock = socket(AF_INET, SOCK_STREAM)    # Create a socket
tcpCliSock.connect(ADDR)                      # Connect with the server

```

Nach der Änderung, speichern Sie die Datei und wählen **Run > Run Module** aus.

Für Linux:

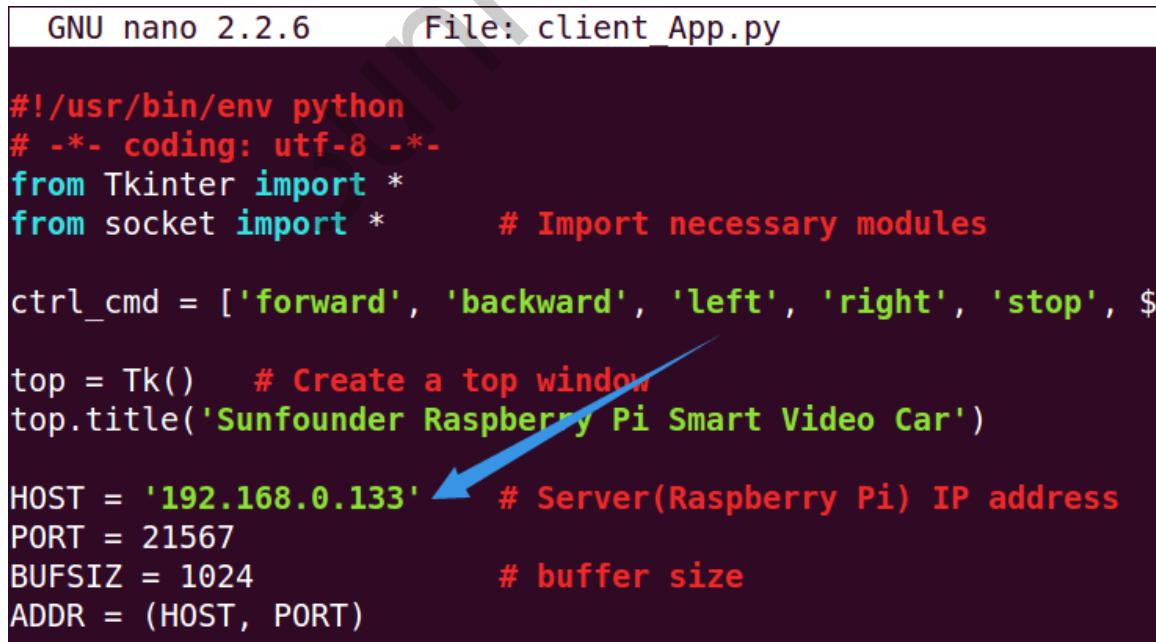
Öffnen Sie wie zuvor ein zusätzliches Terminal. Aber loggen Sie sich in hier nicht per Fernzugriff in Raspberry Pi ein. Gehen Sie zum Client mit cd und bearbeiten die Datei client.py:

```

pi@cavon:~$ cd Sunfounder_Smart_Video_Car_Kit_for_RaspberryPi
pi@cavon:~/Sunfounder_Smart_Video_Car_Kit_for_RaspberryPi/client$ sudo nano client_App.py

```

Ähnlich Änderungen vornehmen wie bei der Kalibrierung (siehe blauer Pfeil), wechseln Sie die IP-Adresse im Clientprogramm auf Ihrem PC:



```

GNU nano 2.2.6      File: client_App.py

#!/usr/bin/env python
# -*- coding: utf-8 -*-
from Tkinter import *
from socket import *      # Import necessary modules

ctrl_cmd = ['forward', 'backward', 'left', 'right', 'stop', '$

top = Tk()    # Create a top window
top.title('Sunfounder Raspberry Pi Smart Video Car')

HOST = '192.168.0.133'    # Server(Raspberry Pi) IP address
PORT = 21567
BUFSIZ = 1024            # buffer size
ADDR = (HOST, PORT)

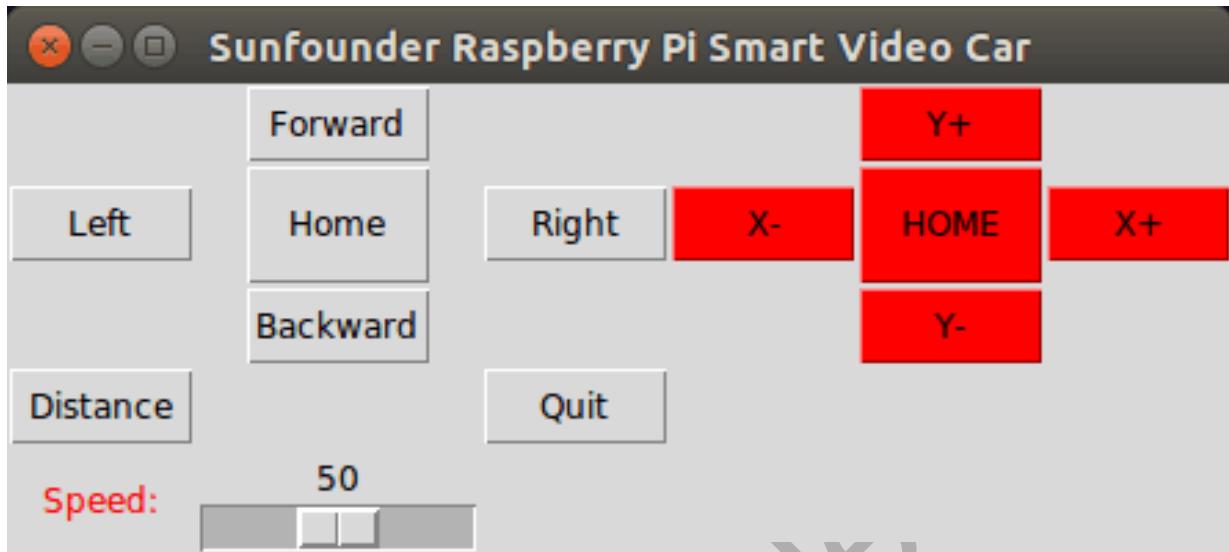
```

Nach der Änderung, betätigen Sie **Ctrl+O** für das Speichern und **Ctrl+X** für den Exit.

Nun führen Sie das Clientprogramm aus:

```
sudo python client_App.py
```

Unabhängig vom welchem Betriebssystem, Linux oder Windows, sobald Sie `client_App.py` ausführen wird folgendes Fenster erscheinen:



Sie können auf die Schaltflächen **Forward** und **Backward** dHinteren, um das Auto fern zusteuern. Oder klicken Sie auf **X+**, **X-**, **Y+** und **Y-** um die Kamera zu bedienen.

**Hinweis:**

Das Serverprogramm muss **vor** dem Clientprogramm ausgeführt werden. Einige Einstellungen des Servers müssen fertig gestellt werden bevor der Dienst ausgeführt wird. Es muss ein Kommunikationsendpunkt erstellt werden damit der Server die Befehle von Client empfangen kann. Benutzen Sie hierfür den Server als Empfänger oder einen Betreiber des Bustelefons eines Unternehmens. Sobald die Telefon- und Gerätinstallation abgeschlossen ist und der Empfänger oder Betreiber am Ort ist, kann der Dienst beginnen.

## v. Über das Programm

### Abstrakt

Aus Sicht der Software, ist die Struktur des Smart Cars eine C/S-Struktur. Das TCP Serverprogramm läuft auf Raspberry Pi zur Befehlsaufnahme von Client und steuert dementsprechend das Auto. Das Clientprogramm läuft auf dem PC und ist verbunden mit dem Server durch das TCP, und stellt dem Benutzer eine grafische Benutzeroberfläche (GUI) zur Verfügung, womit man bequem das Raspberry Pi fernsteuern kann. Beides, sowohl Client- als auch Serverprogramme werden in Python geschrieben.

Stellen Sie sicher, dass der Stromkreis richtig angeschlossen ist. Schalten Sie nun das Smart Car ein, loggen Sie sich per Fernzugriff ins Raspberry Pi ein. Gehen Sie zum Verzeichnis Sunfounder\_Smart\_Video\_Car\_Kit\_for\_RaspberryPi und prüfen die darunter liegenden Dateien.

```
cd ~/Sunfounder_Smart_Video_Car_Kit_for_RaspberryPi  
ls
```

```
pi@raspberrypi ~/Sunfounder_Smart_Video_Car_Kit_for_RaspberryPi $ ls  
client datasheet html_server i2cHelper.py mjpg-streamer README.md server
```

Sie werden 7 Dateien im Verzeichnis vorfinden: **client**, **datasheet**, **html\_server**, **i2cHelper.py**, **mjpg-streamer**, eine Datei **README.md** und **server**.

**client** das Client auf Ihrem PC

**datasheet** beinhaltet einige PDF-Dateien über den Chip (sehen Sie es sich auf Ihrem PC an)

**html\_server** Webserver läuft auf Raspberry Pi für den Android App Client

**i2cHelper** ein einfaches Skript zur Hilfe der Einrichtung von i2c auf Raspberry Pi

**mjpg-streamer** Treiber für die Kamera, um die Bilder zu erfassen und hochzuladen

**README.md** Einführungsdatei mit Information über das Update

**server** Server läuft auf Raspberry Pi

### Einführung zum Socket

Das C/S-Struktur-Programm von SunFounder Smart Car Raspberry Pi-basierten Smart Cars ist auf Grundlage vom Socket-Modul der Programmiersprache Python geschrieben. Das Socket steht für das Ende einer Verbindung. Beim TCP/IP-Protokoll wird ein Socket durch IP-Adresse und Portnummer bestimmt, was von Programmen verwendet wird, um Daten mit anderen Programmen auszutauschen. Das andere Programm kann sich dabei auf demselben Computer (Interprozesskommunikation) oder einem anderen, via Netzwerk erreichbaren Computer befinden.

### 1. Server

Hier haben wir einen Pseudocode bereit gestellt, welches einen universellen TCP Server für Erläuterungen erstellt. Beachten Sie, dass dies nur eine der Methoden des Server-Designs ist. Nachdem Sie sich Wissen darüber angeeignet haben, können Sie den Pseudocode nach Ihren Wünschen ändern:

```

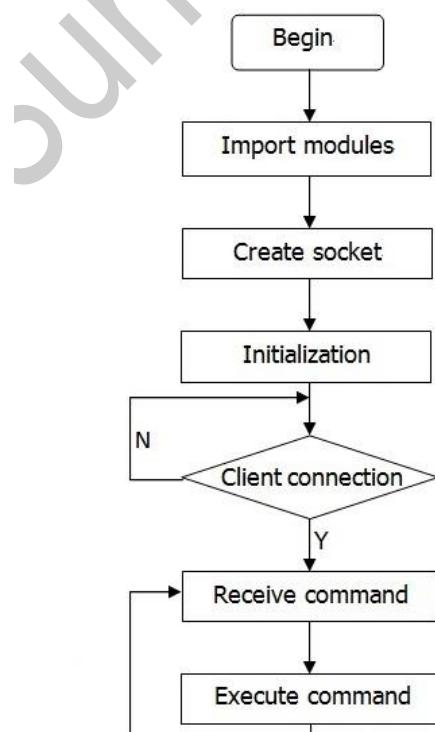
s = socket( )                      # Create a socket for the server.
s.bind( )                           # Bind the address to the socket.
s.listen( )                          # Listen to the connection.
inf_loop:                            # Indefinite Loop of the server.
    c = s.accept( )                  # Accept the connection from the client.
comm_loop:                            # Communication Loop.
    c.recv( )/c.send( )              # Dialog (receiving or sending data)
c.close( )                           # Close the socket of the client.
s.close( )                           # Close the socket of the server (optional).

```

Alle Arten von Sockets können über die Funktion `socket.socket()` erstellt werden und mit der Funktion `bind()` an eine IP-Adresse und Portnummer gebunden werden. Da das TCP ein verbindungsorientiertes Kommunikationssystem ist, sollten einige Einstellungen fertig gestellt werden, noch bevor das TCP anfängt zu operieren. Der TCP Server muss auf die Befehle vom Client "lauschen".

Nachdem die Einstellungen vollendet sind, tritt der Server in eine Endlosschleife ein. Ein einfacher, wie ein einzelner Thread, Server wird die Funktion `accept()` aufrufen und auf die kommende Verbindung warten. Standardmäßig blockiert die Funktion `accept()`, sodass bevor eine Verbindung entsteht, diese Funktion vorübergehend deaktiviert wird. Sobald eine Verbindung aufgebaut wurde, wird die Funktion `accept()` einen separaten Client-Socket eingeben für die nachfolgende Kommunikation. Nach der Erstellung des temporären Sockets, beginnt die Kommunikation. Sowohl der Server als auch Client werden im neuen Socket Daten versenden und empfangen. Die Kommunikation wird nicht enden bis einer der Enden die Verbindung schließt oder bis eine Nullzeichenfolge gesendet wird.

## Prozessdiagramm vom Serverprogramm



## 2. Client

Es ist einfacher einen TCP-Client zu erstellen als einen Server. Benutzen Sie den folgenden Pseudocode:

```
c = socket( )           # Create a client socket.  
c.connect( )            # Try to connect a server.  
comm_loop:              # Communication Loop.  
    c.send( )/c.recv( )   # Dialog (sending out and receiving data)  
c.close( )              # Close the client socket.
```

Wie zuvor erwähnt, werden alle Sockets über die Funktion `socket.socket()` erstellt. Anschließend kann die Funktion `connect()` aufgerufen werden, um sich mit dem Server zu verbinden. Nach der Verbindung wird der Dialog zwischen Client und Server aktiviert. Sobald der Dialog endet, wird der Client den Socket und die Verbindung schliessen.

## Einführung zu Tkinter

Entwickelt auf der Basis von Tkinter, besitzt unser Clientprogramm eine grafische Benutzeroberfläche. Tkinter ist ein GUI-Widget in der Programmiersprache Python, das für die schnelle Entwicklung von Anwendungsprogrammen mit grafischen Benutzeroberflächen dient. Es ist ganz einfach mit Tkinter, Sie müssen nur das Modul in Python importieren.

Erstellung und Ausführung eines GUI-Programms, gehen Sie folgendermaßen vor:

- a) Importieren Sie das Modul Tkinter (über `import Tkinter` oder `from Tkinter import *`).
- b) Erstellen Sie ein oberes Window-Objekt, das das ganze GUI-Programm enthalten wird.
- c) Erstellen Sie das Modul GUI, das für das Objekt benötigt wird und aktivieren Sie die Funktionen.
- d) Verbinden Sie die GUI-Module mit dem Code am Back-End System.
- e) In die Hauptereignisschleife eintreten.

Nehmen Sie ein einfaches GUI-Programm:

Erstellen Sie eine Datei `Tk_test.py` unter `/home`:

```
chmod +x Tk_test.py
```

Fügen Sie der Datei ein ausführbares Privileg an:

```
chmod +x Tk_test.py
```

Öffnen Sie die Datei:

```
vim Tk_test.py
```

Tippen Sie den unten angezeigten Code ein:

```
#!/usr/bin/env python
from Tkinter import *

top = Tk()      # Create a top window
top.title('Sunfounder.com')

label = Label(top, text='Hello Geeks !', fg='blue') # Create a Label and set its foreground
color as blue
label.pack()    # Layout

top.mainloop()  # main Loop
```

Speichern sie den Code und Exit.

Ausführen:

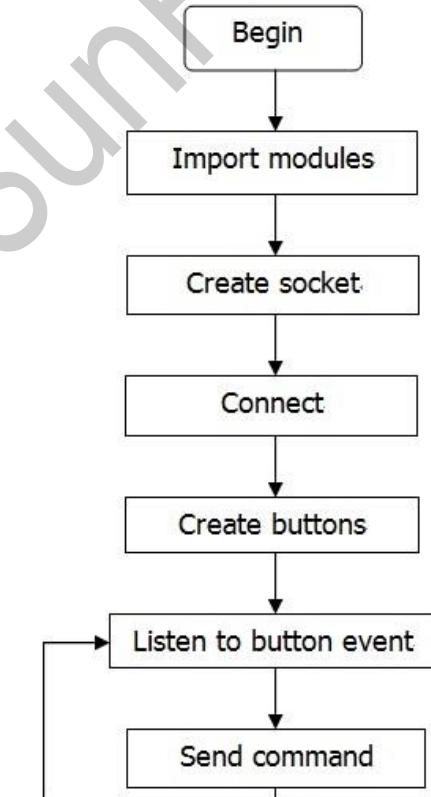
```
./Tk_test.py
```

Das unten angezeigte Bild sollte auf Ihrem Bildschirm erscheinen:



Klicken Sie auf zum Schliessen des Programms.

## Prozessdiagramm vom Client-Programm

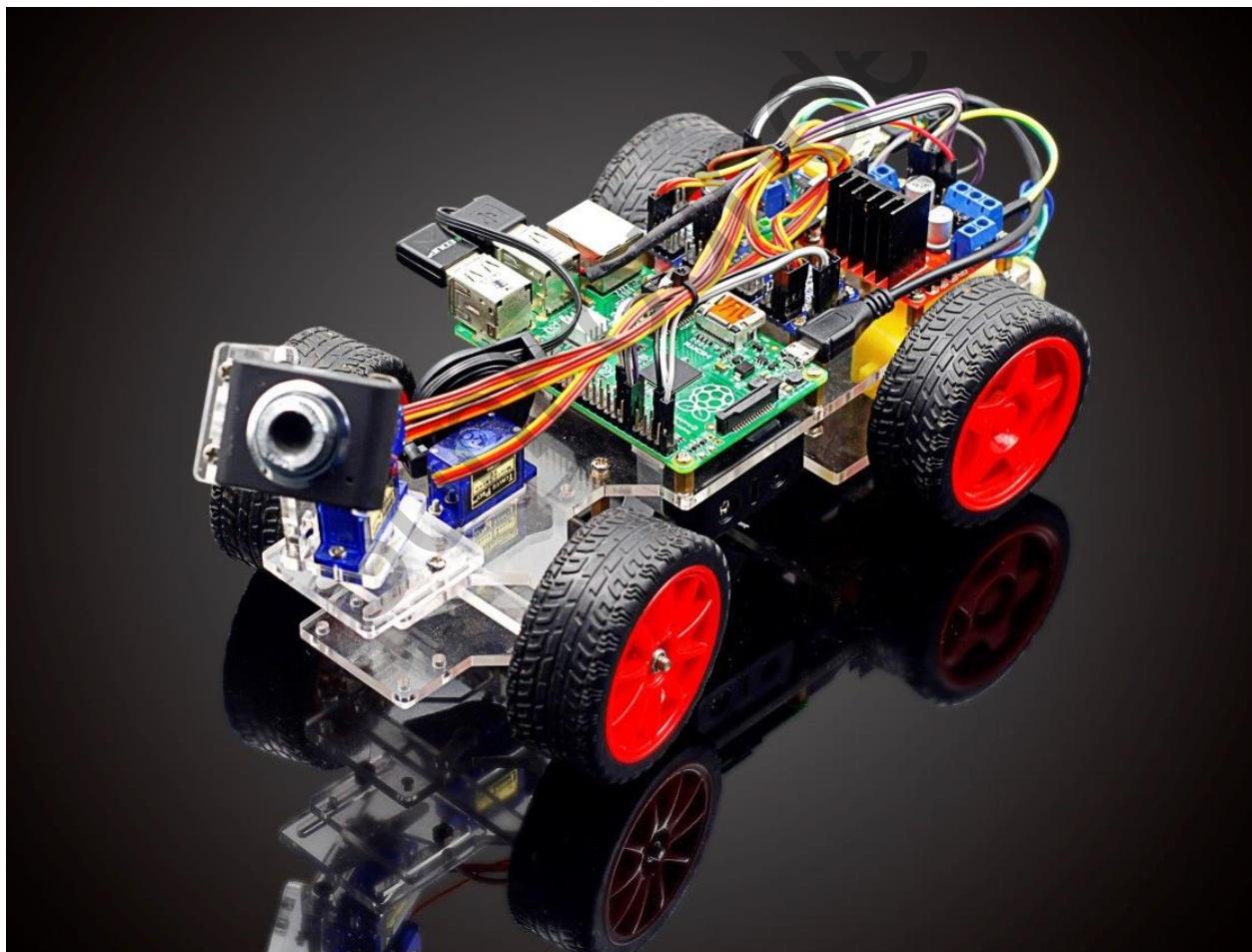


# Zusammenfassung

---

In diesem Handbuch haben wir gelernt, was für Bauteile für den Aufbau dieses Autosets benötigt werden, und durch die Einsetzung der mechanischen Komponenten und den Elektromodulen, Wissen über das Raspberry Pi, sowohl als auch durch kleine Einführungen über die Hauptkomponenten wie das Servo, Wi-Fi Adapter, usw., sich angeeignet. Darüber hinaus haben wir auch viel über Software und Codierung erfahren, welches als solide Grundlage für zukünftige Erforschungen im Open Source Bereich dienen wird.

Das **SunFounder Smart Video Car für Raspberry Pi** ist nicht nur ein Spielzeug, sondern mehr ein sinnvolles Entwicklungsset für Raspberry Pi. Nach all den Studien und Praxis an diesem Set, sollten Sie nun ein besseres Verständnis vom Raspberry Pi haben. Nun liegt es an Ihnen, es noch weiter zu verbessern!



## **Copyright Hinweis**

Alle Inhalte in diesem Handbuch, einschließlich aber nicht beschränkt auf Text, Bilder und Code ist das Eigentum von SunFounder Company. Dieses Handbuch ist nur für persönliche Studien-, Erforschungs-, Vergnügungs-, andere nicht kommerzielle oder gemeinnützige Zwecke, im Rahmen der einschlägigen Verordnungen und Uhrheberrecht Gesetze, und ohne die gesetzlichen Rechte des Autors und der jeweiligen Rechtinhaber zu verletzen. Für Personen oder Organisationen, die diese ohne Genehmigung für den kommerziellen Gewinn nutzen, behält sich das Unternehmen vor, rechtliche Schritte einzuleiten.

sunFounder