

**S E M I N A R A R B E I T**

Rahmenthema des Wissenschaftspropädeutischen Seminars:

*Astrophysik*

Leitfach: Physik

Thema der Arbeit:

**Mars-Rover – einen autonom fahrenden Prototyp entwickeln**

Verfasser/in:

Sebastian Hau

Kursleiter/in:

OStR Herr Koch

Abgabetermin:

07.11.2023

<b>Bewertung</b>	Note	Notenstufe in Worten	Punkte		Punkte
schriftliche Arbeit				x 3	
Abschlusspräsentation				x 1	
				Summe:	
Gesamtleistung (max. 30 Punkte) nach § 61 (7) GSO =				Summe:2	
				(gerundet)	

---

 Datum und Unterschrift der Kursleiterin bzw. des Kursleiters

## Gliederung

<b>1</b>	<b>EINFÜHRUNG IN DAS THEMA</b>	<b>1</b>
<b>2</b>	<b>DER MARS</b>	<b>2</b>
2.1	ALLGEMEINES ZUM MARS	2
2.1.1	<i>Atmosphäre</i>	2
2.1.2	<i>Oberfläche</i>	2
2.2	VERGANGENHEIT DES MARS	3
2.2.1	<i>Hinweise auf lebensfreundliche Bedingungen</i>	3
2.2.2	<i>Die wichtigsten Mars Missionen</i>	4
<b>3</b>	<b>MARS-ROVER PROTOTYP</b>	<b>6</b>
3.1	VORÜBERLEGUNGEN	6
3.2	ERSTER PROTOTYP	6
3.3	FINALER PROTOTYP	7
3.4	KOMPONENTEN	7
3.5	CODE	7
3.5.1	<i>Konstruktor</i>	7
3.5.2	<i>Setup</i>	8
3.5.3	<i>Loop</i>	9
3.5.4	<i>Move und Turn</i>	9
3.5.5	<i>Distance</i>	10
3.5.6	<i>checkDirection</i>	10
3.5.7	<i>turnDirection</i>	11
3.6	KOMPLIKATIONEN BEIM PROGRAMMIEREN	12
3.6.1	<i>Distanz Berechnung</i>	12
3.6.2	<i>Loop</i>	13
<b>4</b>	<b>MARS-ROVER</b>	<b>14</b>
4.1	PERSEVERANCE MARS-ROVER	14
4.2	VERGLEICHEN DES PERSEVERANCE MARS-ROVERS MIT DEM PROTOTYPEN	16
<b>5</b>	<b>FAZIT</b>	<b>17</b>
<b>6</b>	<b>ANHANG</b>	<b>18</b>
6.1	ABBILDUNGEN	18
6.2	TEILE	30
6.3	CODE	31
<b>7</b>	<b>QUELLEN</b>	<b>35</b>
7.1	LITERATURQUELLEN	35
7.2	INTERNETQUELLEN	35

# 1 Einführung in das Thema

Schon seit tausenden von Jahren interessieren sich die Menschen für den Mars. Während in der Antike der rote Planet als bedrohlich wahrgenommen wurde und er nach den Göttern des Kriegs oder der Unterwelt benannt wurde [1.1], ist der Himmelskörper in der Moderne ein sehr interessantes Forschungsobjekt, aufgrund seiner Ähnlichkeiten zur Erde. Seit Jahrhunderten träumen die Menschen davon Leben auf dem Mars zu finden, oder gar selbst auf ihm zu leben [1, 1]. Doch um in der heutigen Zeit unseren Nachbar-Planeten zu kolonialisieren, müssen zunächst unzählige Messungen und Überlegungen angestellt werden. Nun kann man aber noch keine Menschen zum Messen schicken, da die lange Reise logistisch noch unmöglich ist. Deshalb werden für die ersten Erkenntnisse über den Mars Sonden und Rover gesendet. Diese müssen in lebensfeindlichen Bedingungen ohne Kontakt zur Menschheit autonom funktionieren und Ergebnisse liefern.

In dieser Seminararbeit soll ein autonom fahrender Prototyp eines Rovers nachgebaut werden, um zu verstehen, wie schwer die Ansätze sind. Des Weiteren wird genauer erläutert, welchen Bedingungen Roboter auf dem Mars ausgesetzt sind und warum dieser trotzdem so interessant ist. Anschließend wird erläutert, was ein bereits eingesetzter Mars-Rover an Equipment verbaut hat, um zu funktionieren und Daten zu liefern, die uns weitere Erkenntnisse über den roten Planeten bringen. Dieser wird abschließend auch noch mit dem Prototyp verglichen, um zu verdeutlichen, wie komplex Mars Missionen sind.

---

<sup>1</sup> 1 steht für Internetquelle und wird auch fortführend so abgekürzt.

## 2 Der Mars

### 2.1 Allgemeines zum Mars

Der Mars ist mit einer mittleren Entfernung zur Sonne von 227,9 Millionen Kilometer [L<sup>2</sup>,1, S.160] der vierte Planet in unserem Sonnensystem und einer der zwei direkten Nachbarn unseres Heimatplaneten. Er ist ebenso wie die Erde, ein terrestrischer Planet, wobei der rote Planet nur rund ein Zehntel der Masse der Erde hat [I,2] und einen etwa halb so großen Durchmesser besitzt [I,3]. Er benötigt für eine Rotation um die eigene Achse ca. 24 Stunden 37 Minuten und für eine Umrundung der Sonne 1,88 Erdjahre [I,3]. Des Weiteren besitzt der Mars zwei Monde, die deutlich kleiner sind als der Erdmond [I,3]. Aufgrund der Neigung der Marsachse von 25,2° [I, 4] gibt es auf dem Mars Jahreszeiten, die jedoch durch die stark elliptische Bahn nicht von gleicher Dauer sind [I, 3]. So ist der Sommer auf der Nordhalbkugel etwas länger als der Winter [I, 3].

#### 2.1.1 Atmosphäre

Die Atmosphäre besteht zu 95,32% aus Kohlendioxid und lediglich zu 0,13% aus Sauerstoff [I,2]. Durch das fehlende Magnetfeld, die geringere Masse und die daraus resultierende geringere Schwerkraft beträgt der Atmosphärendruck nur 1% des Drucks der Erde [L,1,S.162-163]. Aufgrund der fehlenden Treibhausgase in der Atmosphäre, die dafür sorgen könnten die Energie der Sonne festzuhalten [L,1,S.162], liegt die mittlere Temperatur bei etwa -63°C [I,2].

#### 2.1.2 Oberfläche

Das Höhenprofil des Mars ist ein sehr Ungleichmäßiges, da die Landschaft im Süden größtenteils ein von Kratern überzogenes Hochland und der Norden hauptsächlich ein flaches Tiefland ist [I, 3 | Abb<sup>3</sup>. 1]. Des Weiteren ist erstaunlich, dass der rote Planet nicht nur den höchsten bekannten Vulkan [I, 5], sondern auch den größten Canyon des Sonnensystems [I, 6] beherbergt. Während sich der Olympus Mons über einen Durchmesser von 624 Kilometer bis zu 25 Kilometer in die Höhe bäumt [I, 7], erstreckt sich das Valles Marineris über 20% des Mars Umfangs, also 4000 Kilometer und ist bis zu 7

---

<sup>2</sup> L steht für Literaturquellen und wird auch fortführend so abgekürzt.

<sup>3</sup> Abb steht für Abbildung und vermerkt auf die Abbildungen im Anhang. Es wird auch fortführend so abgekürzt.

Kilometer tief [I, 8]. Man schließt nicht aus, dass der Mars noch vulkanisch aktiv ist, da vulkanische Ablagerungen entdeckt wurden, die innerhalb der letzten 50.000 Jahre ausgeschleudert worden sein müssen. Zudem deuten in der Region aufgezeichnete Marsbeben und die Präsenz von Wasser unter den Eiskappen der Pole des Planeten auf Magmabewegungen unter der Kruste hin [I, 9]. Jedoch wird es aufgrund der fortgeschrittenen Planetengeschichte und der fehlenden frischen Lavaströme, als sehr unwahrscheinlich erachtet, dass es noch Magma nahe der Oberfläche gibt. Eher wird vermutet, dass tiefe Magmareservoire und -ströme existieren, die womöglich für lokal mildere Bedingungen im Untergrund sorgen könnten [I, 9]. Die Farbe des roten Planeten lässt sich durch die Staubschicht erklären, die größtenteils aus Eisenoxid besteht. Diese Schicht kann bei Stürmen wochenlang den Mars einhüllen [I, 3].

## 2.2 Vergangenheit des Mars

### 2.2.1 Hinweise auf lebensfreundliche Bedingungen

Aufgrund der beträchtlichen Menge an Wassereisvorkommen an den Polen des Mars und den zahlreichen Oberflächenstrukturen, die Zeichen von Erosionen aufweisen, wird vermutet, dass es in der Vergangenheit große Mengen flüssigen Wassers auf dem roten Planeten gegeben haben muss [L, 1, S.55]. Ein weiterer Hinweis auf nicht gefrorenes Wasser ist das berechnete Abströmen des Edelgases Argon, bei welchem lediglich durch Entweichen ins All der Gehalt in der Atmosphäre abnehmen kann. Daraus konnte man schlussfolgern, dass der Mars früher eine erdähnlich dichte Kohlendioxidatmosphäre hatte, die durch den Treibhauseffekt für ein warmes und feuchtes Klima gesorgt haben kann [I, 10]. Auch wurde in dem Marsgestein eine hohe Konzentration an Manganoxid entdeckt, welches impliziert, dass vor Milliarden Jahren mehr Sauerstoff in der Atmosphäre war. Denn „die einzige Möglichkeit auf der Erde, die wir kennen, um solche Mangan Materialien herzustellen, involviert atmosphärischen Sauerstoff oder Mikroben“ [frei aus dem Englischen übersetzt, I, 11]. Die Atmosphäre des Mars wurde jedoch von Sonnenstürmen zerstört, da die geladenen Partikel, nicht wie auf der Erde, von einem Magnetfeld abgelenkt wurden [vgl. 2.1 Allgemeines zum Mars]. Diese rissen die zuvor durch die ultraviolette Strahlung der Sonne ionisierten Teilchen mit ins All hinaus [I, 10]. Weiterhin wurden durch eine elektromikroskopische Analyse eines Meteoriten (Meteorit ALH 8400), der vom Mars kam, Strukturen sichtbar, die versteinerten Mikroben ähnelten [L, 1, S.55]. All diese

Erkenntnisse über die Vergangenheit des roten Planeten lassen Wissenschaftler darüber streiten, ob es womöglich einmal Leben auf dem Mars gegeben haben könnte [I, 12], oder ob vielleicht sogar das Leben vom Mars auf die Erde kam [L, 1, S.55].

## 2.2.2 Die wichtigsten Mars Missionen

Um all diese Erkenntnisse zu gewinnen, wurden in der Vergangenheit zahlreiche Missionen zum Mars unternommen. Man mutmaßte schon im späten 19. Jahrhundert, nach der angeblichen Entdeckung von kanalähnlichen Strukturen, dass es auf dem Mars Leben gäbe und eine Zivilisation den Mars bewohne [I, 1]. Doch um die Fantasien der Menschen zu bestätigen, begann 1960 die Sowjetunion erste Versuche eine Sonde zum Mars zu schicken. Die folgenden fünf Jahre missglückte jede weitere Marsmission der NASA und der Sowjets. Erst 1965 gelang es der NASA-Sonde Mariner 4 den roten Planeten zu passieren und die ersten Bilder des Mars zu schießen. Diese waren ernüchternd, da die Oberfläche karg, öde und frei von jeder Spur von Leben war. 1969 wurden Mariner 6 und Mariner 7 zum Mars geschickt. Diese lieferten rund 200 Bilder, die gezeigt haben, dass die angeblichen Marskanäle nicht existieren. 1971 setzte das Sowjetische Landemodul von „Mars 3“ als erstes irdisches Objekt auf dem Mars auf, jedoch lieferte es nur wenig Daten. 1976 brachten die NASA-Lander von Viking 1 und Viking 2, ausgestattet mit zahlreichen Messgeräten, abermals die Erkenntnis, dass es keinerlei Hinweis auf Leben auf dem Mars gab. Bis 1997 gab es keine weiteren erfolgreichen Marsmissionen. Erst die Kartierung des Mars durch den Mars Global Surveyor brachte die Entdeckung von Wassereisvorkommen und war somit ein voller Erfolg. 2001 suchte man mit der NASA-Sonde Mars Odyssey nach früheren Spuren von Wasservorkommen und erfasste Oberflächen- und Mineralienzusammensetzungen. Die Sonde wurde zudem technisch so ausgerüstet, dass sie für nachfolgende Missionen als Funkrelais benutzt werden konnte. 2003 konnte mit der europäischen Mission Mars Express eine dreidimensionale Karte in nie dargewesener Auflösung erstellt werden. 2004 landeten die NASA-Mars-Rover Spirit und Opportunity auf dem Mars und brachten durch den Fund von Sulfaten, welche sich nur in flüssigem Wasser bilden können, eine wichtige Erkenntnis. 2006 entdeckte der Mars Reconnaissance Orbiter Spuren wechselhaften Klimas, in dessen Verlauf auch mehrere hundert Millionen Jahre Wasser vorhanden gewesen sein muss. Die NASA-Sonde Phoenix wies 2008 am Nordpol zweifelsfrei Wassereis nach. 2012 landete der NASA-Rover Curiosity auf dem Mars und fand zu Sedimenten zusammengebackene Flusskiesel [I, 13]. Der 2021 auf dem Mars gelandete Perseverance Rover ist zusammen mit

der in seinem Rumpf gelagerten kleinen Helikopter Drohne Ingenuity der neuste Rover auf dem Mars und sucht nach mikrobiellen Fossilien und kohlenstoffhaltigen Molekülen. Hierzu nimmt er auch Bodenproben, die mit weiteren Missionen, die bereits in Planung sind, zur Erde zurückgebracht werden sollen [1, 14].

## 3 Mars-Rover Prototyp

### 3.1 Vorüberlegungen

Um den Mars zu erforschen, wurde bisher nur auf nicht bemannte Fahrzeuge gesetzt, da die große Entfernung zur Erde zu vielen Komplikationen führen kann, wenn man eine bemannte Mission starten wollen würde. Doch auch unbemannte Missionen bringen ihre Herausforderungen mit sich. Sie können nicht einfach, wie Roboter auf der Erde ferngesteuert werden, denn die an die Roboter geschickten Signale würden trotz Lichtgeschwindigkeit zwischen 3 Minuten 2 Sekunden und 22 Minuten 24 Sekunden [1, 15] brauchen, was in der Praxis das Bedienen nahezu Unmöglich macht. Ein weiteres Problem wäre die Sonne, die sich zwischen Erde und Mars befinden kann. Diese würde die Kommunikation für den betroffenen Zeitraum unmöglich machen. Alle Gefährte, die zum Mars geschickt werden, müssen demnach vollkommen autonom funktionieren. So muss also ein Prototyp eines Mars-Rovers z.B eigenständig Hindernissen ausweichen können, um sich nicht festzufahren. Um dies mit dem Mikrokontroller Arduino Uno [T<sup>4</sup>, 1] möglich zu machen, ist ein sparsamer Umgang mit den Anschlüssen nötig, da diese begrenzt sind. Um möglichst wenige Komponenten zu verwenden, wurde sich grundsätzlich überlegt, einen Entfernungsmesser auf einem Servomotor zu befestigen, um einen größtmöglichen Bereich durch die Entfernungsmessung abzudecken.

### 3.2 Erster Prototyp

Der Erste Prototyp sollte mit DC-Motoren [T, 2] fahren. Um diese herum wurde ein zweiteiliges Modell designt, dass man zusammenstecken kann [Abb. 2&3]. Auf der unteren Platte war geplant die Motoren festzuklebt, während auf der oberen der Mikrokontroller und ein Steckboard hätte liegen sollen. Zusätzlich wurden passende Reifen modelliert, die mit einem Profil versehen waren, um bessere Haftung zu haben [Abb. 4]. Jedoch wurde nach ersten Programmierungsversuchen und kurzer Recherche festgestellt, dass die DC-Motoren, um richtig angesteuert zu werden, nicht nur einen weiteren Kontroller, sondern auch eine extra Stromversorgung benötigt hätten. Deshalb wurde beschlossen, dass man diesen Prototypen aufgibt und mit einem anderen Ansatz das Modell zum Laufen bringen wird.

---

<sup>4</sup> T steht für Teile und vermerkt auf die Teile im Anhang. Es wird auch fortführend so abgekürzt.



## 3.3 Finaler Prototyp

Daher wurde dann mit Stepper Motoren [T, 3] gearbeitet. Da diese jedoch jeweils vier Steckplätze beim Arduino Uno in Anspruch nehmen, wurden nur zwei, diagonal zueinander liegende Stepper Motoren verbaut. Die verbleibenden zwei Räder, die benötigt wurden, um das Gefährt stabil zu machen, wurden an Kugellagern [T, 4] mit dem Gehäuse verbunden. Für diesen neuen Prototypen wurde erneut ein Rumpf erstellt [Abb. 5], der alle Komponenten miteinander verbinden kann. Zudem wurden neue Räder für die Stepper-Motoren [Abb. 6] und für die Kugellager [Abb. 7] designed. Da die Räder der Stepper Motoren auf glatten Oberflächen wenig Bodenhaftung hatten, wurden Ausschnitte einer Anti-Rutsch-Matte aufgeklebt und die passiven Räder wurden vergrößert [Abb.8]. Außerdem wurde noch ein Verbindungsstück [Abb. 9] zwischen dem Servomotor [T, 5] und dem Ultraschallsensor [T, 6] modelliert.

## 3.4 Komponenten

Das Herzstück des Modell Mars-Rovers ist der Mikrokontroller Arduino Uno, der durch Pins Strom fließen lassen kann und auch Stromflüsse auslesen kann. Dieser versorgt alle Komponenten mit Strom und führt den zuvor auf ihm gespeicherten Code aus. Für die Stromversorgung des Mikrokontrollers selbst sind sechs AA-Batterien zuständig, die in einem Batteriefach liegen, das über einen Niedrigvoltstecker mit dem Arduino Uno verbunden ist. Für den primären Stromkreislauf zuständig sind die 5V und GND Pins, die an einer Steckplatine so angesteckt sind, dass alle weiteren Teile mit dem Kreislauf verbunden sind. Die restlichen Steckplätze am Arduino sind vom jeweilig angeschlossenen Teil abhängig vergeben (siehe Steckplan [Abb. 10]). Der Rumpf des Rovers, sowie alle Reifen und der Adapter, wurden aus PLA [T, 7] 3D-gedruckt, welches ein einfach druckbarer, jedoch trotzdem stabiler Kunststoff ist.

## 3.5 Code

### 3.5.1 Konstruktor

Im Konstruktor werden zu Beginn des Programms Werte von Variablen festgelegt und zugewiesen. Auch werden sogenannte Bibliotheken hinzugefügt und aktiviert. Diese dienen dazu, komplexere Komponenten mit einfachen Befehlen anzusteuern, anstatt komplizierte Programmcodes zu schreiben. Sie übersetzten also einfache Anweisungen in solche, die die

jeweiligen Komponenten verstehen. Als erstes wurde mit dem *#include* Befehl die Servo Bibliothek hinzugefügt [C<sup>5</sup>, 1, Z.1]. Anschließend wird der Servomotor *servo* genannt [C, 1, Z.2]. Darauffolgend wird die Stepper Bibliothek hinzugefügt [C, 1, Z.3] und im Zuge dessen der *int SPU* der Wert 2048 gegeben [C, 1, Z.4]. SPU ist kurz für Schritte pro Umdrehung, wofür ein vorgegebener Wert verwendet wird. Mit *int* kann man einer Variable, die beliebig benannt werden kann, eine ganze Zahl zuweisen, die maximal 32-bits belegt, also zwischen -2.147.483.648 und 2.147.483.648 liegt [I, 16]. *int* wird also bei kleineren Werten genommen, um Speicherplatz zu sparen. In Zeile fünf und sechs [C, 1, Z.5-6] des Codes werden den zwei Stepper Motoren die entsprechenden Pins zugeordnet. Anschließend werden *trigger* und *echo*, um das spätere Programmieren einfacher und übersichtlicher zu machen, die Werte 11 und 12 gegeben, welche jeweils den Pin angeben, an dem die entsprechenden Gegenstücke des Ultraschallsensors an das Arduino Uno Board angeschlossen werden [C, 1, Z.7-8]. Als nächstes wurde die Stopp-Distanz in Zentimetern definiert [C, 1, Z.9]. Als Stopp-Distanz wurde 20 festgelegt, da der Rover so genug Platz hat, um sich zu drehen. Der Datentyp *long* wurde für die *pulseTime* und die *distance* [C, 1, Z.10-11] verwendet, da es bei diesen Variablen möglich ist, dass die Werte, beispielsweise durch Messfehler, größer sein können. Denn wenn alle Bits belegt sind, also der größtmögliche Wert einer Variable zugewiesen ist und der Wert noch größer wird, werden die Bits resettet und es wird wieder bei 0 angefangen. Dieses Verhalten kann durch das Binärsystem eines Computers erklärt werden. So sind die Bits entweder an, oder aus, was durch 0, für aus, und 1 für an, dargestellt wird. Es wird im Binären, beispielsweise bei 8 Bit, also wie folgt gezählt: 00000001 (1), 00000010 (2), 00000011 (3), 00000100 (4), 00000101 (5), 00000110 (6), 00000111 (7), 00001000 (8), ... [I, 17]. Wenn alle Bits an sind, also z.B. alle 32 Bits auf 1 sind, werden alle wieder auf 0 gesetzt, obwohl die Zahl eigentlich noch größer werden würde. Um dies zu vermeiden, kann man mit *long* bis zu 64 Bits belegen, also fast 5 Milliarden Mal so hoch, bzw. niedrig, zählen. Als letztes wird im Konstruktor der Variable *turnDir* noch der Wert 1 gegeben [C, 1, Z.12].

### 3.5.2 Setup

Im Setup wird zunächst dem Servomotor der Pin 13 des Arduino Unos zugewiesen [C, 2, Z.16]. Anschließend werden die Stepper Motoren auf die Geschwindigkeit 12 gesetzt

---

<sup>5</sup> C steht für Code und vermerkt auf den Code im Anhang. Es wird auch fortführend so abgekürzt.

[C, 2, Z.17-18]. In Zeile 19 wird der Pin, an dem der Trigger des Ultraschallsensors angeschlossen ist, auf den Output Modus gestellt, was bedeutet, dass ein Signal aus diesem Pin hinausgeht [C, 2, Z.19]. Der Pin, an dem das Echo des Sensors angeschlossen ist, wird wiederum auf den Input Modus gestellt, dort wird also ausgelesen, wenn ein Signal kommt [C, 2, Z.20].

### 3.5.3 Loop

Die Loop ist der Teil des Programmcodes, der sich dauerhaft wiederholt. Hier stehen nur die Funktionen, die beim Fahren des Rovers gebraucht werden. So wird zunächst der Servomotor auf 90 gestellt, damit er zu Beginn nach vorne gerichtet ist und dementsprechend die Entfernungsmessung in Fahrtrichtung stattfindet [C, 3, Z.25]. Bevor der Rover sich bewegt, wird immer die aktuelle Distanz zum nächsten Objekt vor sich in Zentimetern durch die *getDistance* Funktion auf die Variable *distance* überschrieben [C, 3, Z.26]. Anschließend wird durch den if Befehl überprüft, ob der gemessene Abstand nach vorne größer oder gleich der Stopp-Distanz ist. Nur wenn dem so ist wird *moveForward* ausgeführt [C, 3, Z.27-30]. Gegensätzlich überprüft die nächste if Bedingung, ob der in der Variable *distance* gespeicherte Wert kleiner oder gleich der Stopp-Distanz ist. Wenn dem so ist, wird zunächst der Variable *turnDir* ein neuer Wert von der Funktion *checkDirection* zugewiesen und anschließend der Befehl *turnDirection* ausgeführt [C, 3, Z.31-35].

### 3.5.4 Move und Turn

Die move und turn Funktionen sind ausschließlich für die Bewegung des Prototypen zuständig. Der *moveForward* Befehl besteht aus zwei nacheinander folgenden Handlungen. Die Stepper Motoren drehen sich um einen Step in entgegengesetzte Richtung, da sie diagonal zueinander in andere Richtungen zeigen und sich so in die gleiche Richtung bewegen [C, 4, Z.38-42]. Vorwärts ist hierbei dort, wo beim Rover der Ultraschallsensor verbaut ist. Diese Bewegung ist nur effektiv, da die loop sich so schnell wiederholt, dass die Steps nahezu nahtlos ausgeführt werden. Für das Drehen des Rovers sind die turn Methoden zuständig. Die *turnLeft* Funktion lässt hierbei die Stepper Motoren jeweils -2100 Schritte machen, was in einer ca. 90° Drehung des Rovers nach links resultiert [C, 4, Z.43-47]. Ebenso lässt der *tunRight* Befehl die Motoren jeweils 2100 Steps machen, was den Rover 90° nach rechts drehen lässt [C, 4, Z.48-52]. Die letzte Methode, die direkt

die Stepper Motoren ansteuert, ist die *turnAround* Methode, die lediglich zweimal hintereinander die *turnLeft* Funktion ausführt [C, 4, Z.53-57].

### 3.5.5 Distance

Um die Distanz zu messen, wird zunächst sichergestellt, dass auf dem Trigger-Pin kein Strom fließt, anschließend wird 5 Millisekunden gewartet, bis für 10 Millisekunden ein Signal gegeben wird, das einen Ultraschall auslöst, ehe das Signal wieder gestoppt wird [C, 5, Z.61-65]. Anschließend wird mit *pulseIn* die Zeit in Millisekunden gemessen, bis der Ultraschall, der von Objekten reflektiert wird, wieder gemessen wird, also ein Signal am echo Pin ankommt. Dieser Wert wird auf die *pulseTime* überschrieben. [C, 5, Z.66]. Die Entfernung wird anschließend berechnet, indem der Wert der Variable *pulseTime* zunächst halbiert wird, um lediglich die einfache Strecke zum Objekt zu bekommen und nicht noch zusätzlich die, die nach der Reflektion zurückgelegt wurde. Anschließend wird sie mit 0,03432, dem Wert Schallgeschwindigkeit in Zentimeter pro Mikrosekunde, multipliziert [I, 18]. Als Ergebnis erhält man den Wert der Entfernung zum Objekt in Zentimetern, welcher der Variable *distance* zugewiesen wird [C, 5, Z.67]. Anschließend wird *distance* returned, da int Funktionen nur gültig sind, wenn eine Variable mit Wert zurückgegeben wird [C, 5, Z.68].

### 3.5.6 checkDirection

Mit *int checkDirection* wird festgestellt, auf welcher Seite des Rovers die jeweils gemessene Distanz unter der Stopp-Distanz liegt und darauf basierend wird die Variable *turnDir* mit unterschiedlichen Werten zurückgegeben. Diese Funktion wird, wie in 3.5.3 beschrieben, erst aufgerufen, wenn die nach vorne gemessene Distanz kleiner als die Stopp-Distanz ist. Um die verschiedenen Distanzen festzuhalten, wird zunächst eine Matrix *distances* mit zwei Feldern erstellt, die beide den Wert 0 haben [C, 6, Z.73]. Mit einer solchen Matrix kann man mehrere Werte in unterschiedlichen Feldern speichern, ohne sehr viele Variablen erstellen zu müssen. In diesem Fall hat die Matrix nur zwei Felder. Dies wird durch den Wert in den eckigen Klammern angegeben. Müsste man weitaus mehr Werte zuordnen, könnte man mehrere Reihen machen, also wie bei einer Tabelle Felder in die gedachte Vertikale und Horizontale anlegen. Dies ist jedoch bei nur zwei Werten nicht nötig. Für die erste Entfernung wird der Servomotor auf 180 gestellt. Dies bedeutet, dass der darauf verbaute Ultraschallsensor nach links zeigt. Anschließend wird kurz gewartet, bevor die Distanz mit

der *getDistance* Funktion erfasst wird und der Wert in das erste Feld der *distances* Matrix eingetragen wird [C, 6, Z.74-76]. Anschließend wird das gleiche für die rechte Seite des Rovers gemacht, indem der Servomotor auf die 0 Stellung geht, also nach rechts zeigt. Nach einem Delay wird die gemessene Distanz im zweiten Feld der Matrix eingetragen [C, 6, Z.77-79]. Anschließend wird mit der ersten if Funktion geprüft, ob sowohl nach links als auch nach rechts sehr viel Platz ist. Hierzu werden die Bedingungen gestellt, dass beide Felder einen größeren Wert als 200 haben müssen, also in beide Richtungen mehr als 2 Meter Platz sein muss. Wenn dies so ist, wird der Variablen *turnDir* der Wert 0 gegeben [C, 6, Z.80-81]. Falls diese Bedingung erfüllt wurde, werden die folgenden else if Methoden nicht mehr aufgerufen, da else nur die Alternative ist, falls die if Gegebenheit nicht erfüllt worden ist. Somit werden alle nachfolgenden else if Funktionen bis zum nächsten Befehl, also *return turnDir*, übersprungen. Dies gilt nach jeder if Funktion, auf die eine else-Funktion folgt. Somit können die *turnDir* Werte nicht von folgenden else if Funktionen überschrieben werden, auch wenn eine der folgenden else if Bedingungen erfüllt wäre. In der nächsten if-Bedingung darf keiner der beiden Felder einen Wert haben, der kleiner ist als die Stopp-Distanz, da in dem Fall auf beiden Seiten und vor dem Rover ein Objekt wäre. Es wird bei Erfüllung der if Methode der *turnDir* Variable der Wert 1 gegeben [C, 6, Z.82-83]. In den folgenden zwei else if Methoden wird geprüft, welcher Wert größer ist. Wenn der Wert der linken Seite größer ist, als der auf der rechten, wird *turnDir* der Wert 0 gegeben. Wenn die Distanz auf der rechten Seite größer ist, als die auf der linken, wird der Variablen *turnDir* der Wert 2 gegeben [C, 6, Z.84-87]. Abschließend wird *turnDir* zurückgegeben [C, 6, Z.88].

### 3.5.7 turnDirection

Die *turnDirection* Funktion wird nach Beenden des *checkDirection* Befehls ausgeführt und bestimmt nun basierend auf den Ergebnissen dieser, wie der Rover sich bewegen muss. Dies wird mit einer switch case Funktion bewerkstelligt. Diese kann aufgrund der unterschiedlichen Werte (case) einer Variable (switch) verschiedene Funktionen ausführen. Sie ist im Grunde ein vereinfachter und deutlich übersichtlicherer Command als eine Aneinanderreihung von if Funktionen. In diesem Fall ist der switch die Variable *turnDir* [C, 7, Z.93]. Beim ersten Case, also *turnDir* hat den Wert 0, wird die Funktion *turnLeft* ausgeführt [C, 7, Z.95-97], da die 0, wie in 3.5.6 beschrieben, vergeben wird, wenn nach links mehr Platz ist, oder in beide Richtungen gleich viel Platz ist. Der break Befehl, ist dazu

da, um zu bedeuten, wo das auszuführende zu Ende ist und entweder der nächste case ist, oder die Funktion vorbei ist [C, 7, Z. 97&100&103]. Im darauffolgenden case, 1, wird die *turnAround* Funktion aufgerufen [C, 7, Z.98-100], da lediglich hinter dem Rover kein Objekt ist [vgl. 3.5.6]. Der Letzte Case, 2, ruft den *turnRight* Befehl auf [C, 7, Z.101-103], da rechts vom Rover am meisten Platz ist [vgl. 3.5.6].

## 3.6 Komplikationen beim Programmieren

Es gab bei der Entwicklung des Programmcodes nur zwei große Probleme [2.6.1 & 3.6.2]. Durch Try and Error wurden die Werte der turn Funktionen so bestimmt, dass sie den Rover um ca. 90° drehen.

### 3.6.1 Distanz Berechnung

Zunächst wurde bei der Berechnung der Distanz anstatt eines Punkts, wie es im Englischen benutzt wird, ein Komma bei der Schallgeschwindigkeit in Zentimeter pro Mikrosekunde gesetzt, wie es im Deutschen benutzt wird [C, 5, Z.67]. Es stand also statt 0.03432 die Zahl 0,03432 dort. Dies hatte zur Folge, dass das Programm die *pulseTime* immer mit 0 multipliziert und dementsprechend der Variable *distance* immer der Wert 0 gegeben wurde. Somit versuchte der Rover sich immer zu drehen, da das Programm festgestellt hat, dass vor dem Rover ein Objekt ist. Im Zuge dessen drehte sich der Rover aber immer nur um 180°, da beim *checkDirection* immer der zweite Fall, also *turnDir* gleich 1, eintrat, weil die dort gemessenen Distanzen auch 0 waren. Um diesen Fehler im Code zu entdecken, wurde zunächst im setup der *Serial.begin(9600)* Befehl ausgeführt, um die Kommunikation mit dem seriellen Monitor zu beginnen. Anschließend wurde am Ende der loop der Befehl *Serial.print(distance)* geschrieben, um am Ende jeder einzelnen Loop den Wert der Distanz auf dem seriellen Monitor sehen zu können. Dieser ist in der zugehörigen Software zum Programmieren des Arduino einsichtig. Es wurde festgestellt, dass immer der Wert 0 gegeben wurde, jedoch wurde nicht direkt bemerkt, dass dies an der falschen Zeichensetzung lag. Also wurde zunächst beim *Serial.print* Befehl die Variable *distance* durch *pulseTime* ersetzt, um zu überprüfen, ob diese richtige Werte ausgibt. Nach Bestätigen der Richtigkeit der *pulseTime*, wurde der *getDistance* Command noch einmal sehr genau inspiziert, mit einer Musterlösung daneben. Es wurde festgestellt, dass der Punkt mit dem Komma verwechselt wurde. Um nun die Richtigkeit der gesamten *getDistance* Funktion zu überprüfen, wurde erneut der *Serial.print* Befehl mit der Variable

*distance* ausgeführt, wobei die korrekten Werte ausgegeben wurden. Die Commands, die zur Überprüfung genutzt wurden, konnten anschließend wieder entfernt werden, da diese Speicherplatz eingenommen hätten.

### 3.6.2 Loop

Das weitere große Problem war, dass zu Beginn alle Funktionen in der Loop standen. Dies hatte zur Folge, dass der Rover immer alle *move* und *turn* Befehle ausführte und sich somit nicht so fortbewegte, wie er sollte. Es wurden also zunächst diese Commands außerhalb der Loop geschrieben. Jedoch gab es das weitere Problem, das durch das Ausführen des *getDistance* und der *Direction* Funktionen sehr viele Delays vorhanden waren, die in jeder Loop ausgeführt wurden. Dadurch waren die Zeitabstände zwischen den einzelnen Ausführungen der *moveForward* Funktion so groß, dass die Bewegung sehr langsam war. Nach Erkennen dieses Fehlers wurden die betroffenen Funktionen außerhalb der Loop untergebracht, wodurch sie nur aufgerufen werden, wenn sie wirklich benötigt werden.

## 4 Mars-Rover

Wie bereits in 2.2.2 erwähnt, gab es in der Vergangenheit schon mehrere Rover, die auf dem Mars gelandet sind. Um nun den selbstgebauten Prototypen zu vergleichen, wird zunächst der aktuellste, der Perseverance Mars-Rover, genauer betrachtet.

### 4.1 Perseverance Mars-Rover

Der Perseverance Mars-Rover ist unter dem Mission Namen Mars 2020 im Februar 2021 im Jezero Krater auf dem Mars gelandet. Die Hauptmission des Rovers ist es, wie bereits in 2.2.2 erwähnt, nach Anzeichen vergangenen Lebens zu suchen und Marsgesteinsproben für Untersuchungen auf der Erde zu entnehmen [1, 19]. Perseverance ist mit rund 3 Meter Länge, 2,7 Meter Breite und 2,2 Meter Höhe in etwa so groß wie ein PKW, wiegt aber mit 1025 Kilogramm weniger als ein kompaktes Auto [1, 20]. Der Body des Rovers schützt die zwei Computer und die Elektronik, mit einer stabilen Außenwand und durch die Temperaturregelung im Inneren [1, 21]. Der Body ist an, aus Titan-Rohren gefertigten, Beinen [Abb. 11] montiert, an denen wiederum 6 Aluminiumreifen mit Stollen und Titanfelgen [Abb.12] befestigt sind. Somit kann der Mars-Rover über Hindernisse hinweg fahren und hat eine gute Bodenhaftung [1, 22]. Der Rover verwendet Front [Abb. 13] und Rear HazCams [Abb. 14], also Kameras, die nach vorne und hinten ausgerichtet sind und nur dafür zuständig sind, autonom Hindernisse zu erkennen und zu vermeiden. Außerdem werden sie benutzt, um die Bewegungen der Arme zu sehen. Zusätzlich sind im Mast Navigationskameras [Abb. 15] verbaut, die die gleiche Aufgabe wie die HazCams haben und außerdem dafür zuständig sind, Entfernungen zu messen, wenn der Rover „blind fährt“, also ihm eine bestimmte Distanz in eine bestimmte Richtung zum Fahren gegeben wird. Im Inneren des Rovers befindet sich eine CacheCam, die die Bodenentnahmen fotografiert, bevor sie verschlossen werden. Die ebenfalls im Mast verbaute Mastcam-Z [Abb. 16] ist ein Paar hochauflösender Farb-Kameras, die Fotos, Videos und eine 3D Ansicht, ähnlich dem der menschlichen Augen aufnehmen kann. Des Weiteren besitzen sie starke Zoom Linsen. Die letzte im Mast verbaute Kamera ist die SuperCam [Abb. 17], welche durch beschießen von Materialien in einer Entfernung von mindestens 7 Metern mit einem Laser das dadurch entstehende Plasma mit einem Spektrographen aufnehmen kann. Mit dem Spektrum des Plasmas kann man bestimmen, wie das gemessene Material zusammengesetzt ist. Das PIXL-Modul [Abb. 18] am Arm des Rovers, benutzt Röntgenfluoreszenz zum Bestimmen



chemischer Elemente in sehr kleinen Bereichen. Das Modul besitzt zudem eine Mikrokontext Kamera, um die Karte der Elementzusammensetzung mit den sichtbaren Eigenschaften der Zielgebiete zueinander in Beziehung zu setzen. Ebenfalls auf dem Arm montiert ist der SHERLOC Context Imager [Abb. 19], welcher mit einer Macro Kamera einen Kontext zu den mit dem Laser und Spektrometer gemessenen Werten liefert. Die letzte Kamera auf dem Arm ist die WATSON Kamera [Abb. 20], welche wie ein Vergrößerungsglas für die SHERLOC Kamera ist und fein skalierte Strukturen und Texturen erfassen kann. Außerdem ist diese Kamera dafür zuständig, Bilder des Rovers zu machen [1, 23]. Ebenfalls am Arm befestigt ist der Coring Drill [Abb. 21], welcher für die Entnahme von Bodenproben zuständig ist. Hierzu hat der Bohrer drei verschiedene Aufsätze, sogenannte Bits. Die coring und regolith Bits nehmen lose Proben direkt von der Mars Oberfläche in eine versiegelbare Röhre, mit einem Durchmesser von 13 Millimetern und einer Länge von 60 Millimeter, auf. Der abrader Bit kratzt erst die obersten Schichten der Steine ab, um nicht vom Wetter beeinflusste Proben zu entnehmen [1, 24]. Für die Energieversorgung [Abb. 22] sind die zwei Lithium-Ionen-Batterien zuständig, die dauerhaft durch das MMRTG geladen werden. Der Multi-Mission Radioisotope Thermoelectric Generator, kurz MMRTG, ist ein Generator, der die Hitze der natürlichen Zersetzung des Plutoniums in elektrische Energie umwandelt. Als Treibstoff wird 4,8 Kilogramm Plutonium Dioxid verwendet, welches 14 Jahre halten soll [1, 25]. Der Perseverance Rover sendet und empfängt mit drei Antennen Signale von und zur Erde. Mit der Ultra-High Frequency (UHF) Antenna [Abb. 23] übermittelt der Rover Daten über Sonden, die als Relais dienen, an die Erde. Die High-Gain Antenna [Abb. 24] dient dazu, Daten direkt zur Erde zu schicken, bzw. zu empfangen. Sie hat jedoch eine deutlich geringere Datenübertragungsrate als die UHF-Antenne, mit 800 Bits pro Sekunde, gegen 2 Megabits pro Sekunde. Die kleinste Antenne, die Low-Gain Antenna [Abb.25], dient lediglich dazu Daten von der Erde zu empfangen und ist mit einer Übertragungsrate von maximal 30 Bits pro Sekunde auch die Langsamste [1, 26]. Der Mars-Rover beherbergt des weiteren eine Wetterstation, MEDA [Abb. 26], die verschiedene Daten, wie Luftdruck, Temperatur, Luftfeuchtigkeit und Wind misst [1, 20]. Um die Geologie unter der Oberfläche des Mars zu untersuchen, ist am Heck des Perserverance Rovers das sogenannte RIMFAX Modul [Abb.27] verbaut, welches ein bodendurchdringendes Radar ist [1, 20]. Das letzte Modul ist das MOXIE Experiment, welches im Inneren des Bodys aus Kohlenstoff Dioxid aus der Atmosphäre Sauerstoff herstellt [1, 20].

## 4.2 Vergleichen des Perseverance Mars-Rovers mit dem Prototypen

Wenn man nun den Prototyp und den Perseverance Mars-Rover vergleicht, stellt man fest, dass der Modell Rover eine komplett andere Funktionsweise hat. Denn während Perseverance kameragesteuert fährt und Hindernisse erkennt, muss sich der Prototyp nur auf eine Entfernungsmessung verlassen. Dies liegt an der Komplexität der Programmierung einer Objekterkennungssoftware und der damit verbundenen Rechenleistung, die sowohl den Umfang einer Seminararbeit als auch die Möglichkeiten des Arduino Uno überschreiten würden. Des Weiteren werden vom Ultraschallsensor besonders bodennahe Objekte nicht erfasst. Dies kann aufgrund der geringen Bodenfreiheit und des Fehlens von zwei weiteren Rädern dazu führen, dass sich der Nachbau festfährt. Es gibt auch keinerlei Backup Systeme, wie Kameras, die überprüfen können, ob alles so funktioniert, wie es soll. Generell lässt sich feststellen, dass der Nachbau keinerlei Messgeräte verbaut hat. Dies war jedoch nicht das Ziel der Seminararbeit. Aufgrund dessen fehlt dem Modell auch jegliche Kommunikationsmöglichkeit, da keine Daten empfangen oder versendet werden können. Somit können auch keine Ziele gegeben werden und der Weg wird immer nur nach demselben Muster ausgewählt. Es war logischerweise auch nicht möglich einen kleinen Nuklearreaktor zu verbauen und es wurde als Energiequelle auf Batterien zurückgegriffen.

## 5 Fazit

Diese Seminararbeit hat sich mit der Komplexität eines autonom fahrenden Gefährtes beschäftigt, welches die Grundlage für einen Rover sein könnte. Es wurde dargelegt, dass nicht nur die Auswahl der zu verbauenden Komponenten entscheidend ist, sondern auch, dass der Programmcode fehlerfrei sein muss, damit der Prototyp problemlos und zuverlässig funktioniert.

Des Weiteren wurde erläutert, warum der Mars ein so interessantes Forschungsobjekt ist und welche Herausforderungen er zunächst den Robotern, die auf seiner Oberfläche forschen und möglicherweise in der Zukunft auch den Menschen, die ihn besiedeln, bietet. Um die Simplizität des Modell Rovers darzustellen, wurde als Vergleich der neuste Mars-Rover vorgeführt. Hierbei wurde erkannt, dass der Prototyp noch gänzlich ungeeignet dazu wäre, auf den Mars geschickt zu werden und noch sehr vieles verändert werden hätte müssen. Abschließend stellt man fest, dass solche Marsmissionen hochkomplex sind, jedoch unser Nachbar sehr viele potentielle Erkenntnisse, nicht nur zur frühen Erdgeschichte, sondern womöglich sogar zum Ursprung des Lebens liefern kann.

## 6 Anhang

### 6.1 Abbildungen

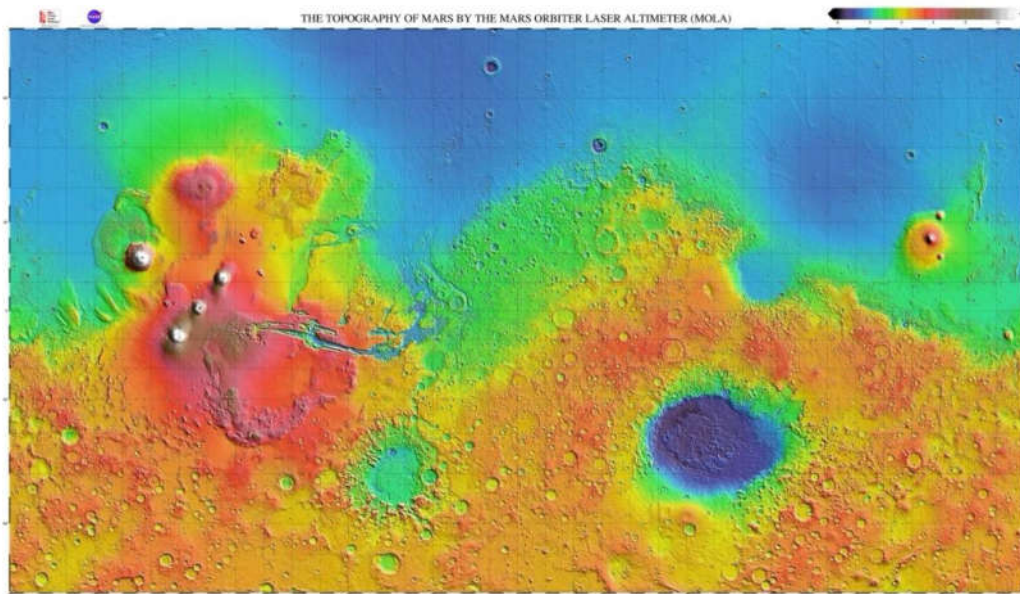


Abbildung 1,

unter: [https://marsoweb.nas.nasa.gov/globalData/images/fullscale/MOLA\\_mercat.jpg](https://marsoweb.nas.nasa.gov/globalData/images/fullscale/MOLA_mercat.jpg)

Stand: 1.11.2023

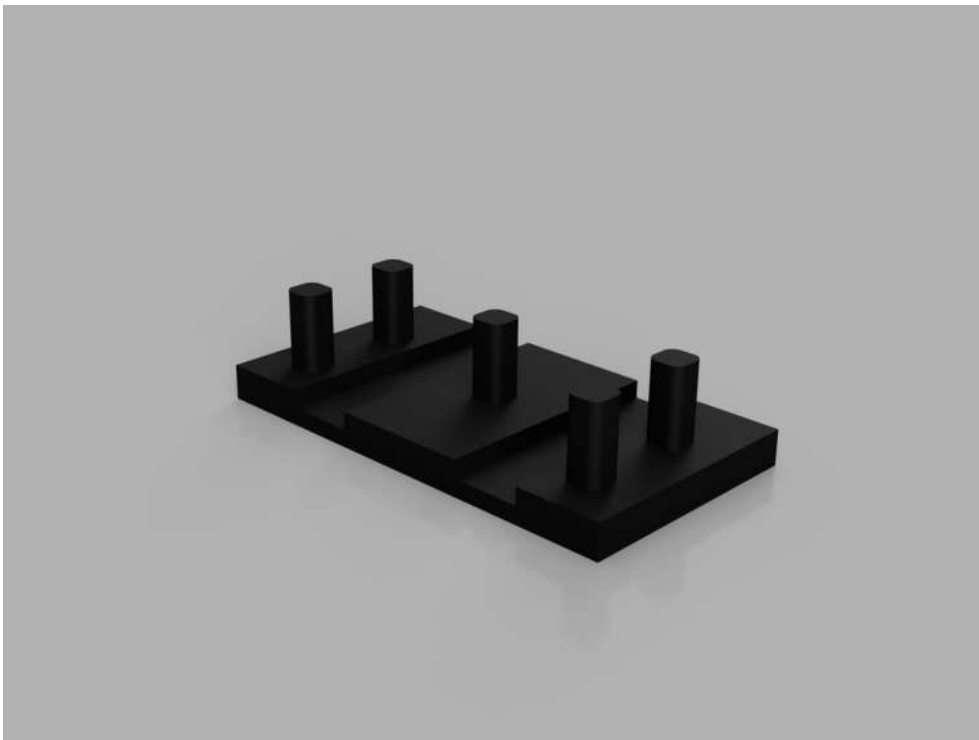


Abbildung 2,

Render des Modells der unteren Platte aus Autodesk Fusion 360 (Software)

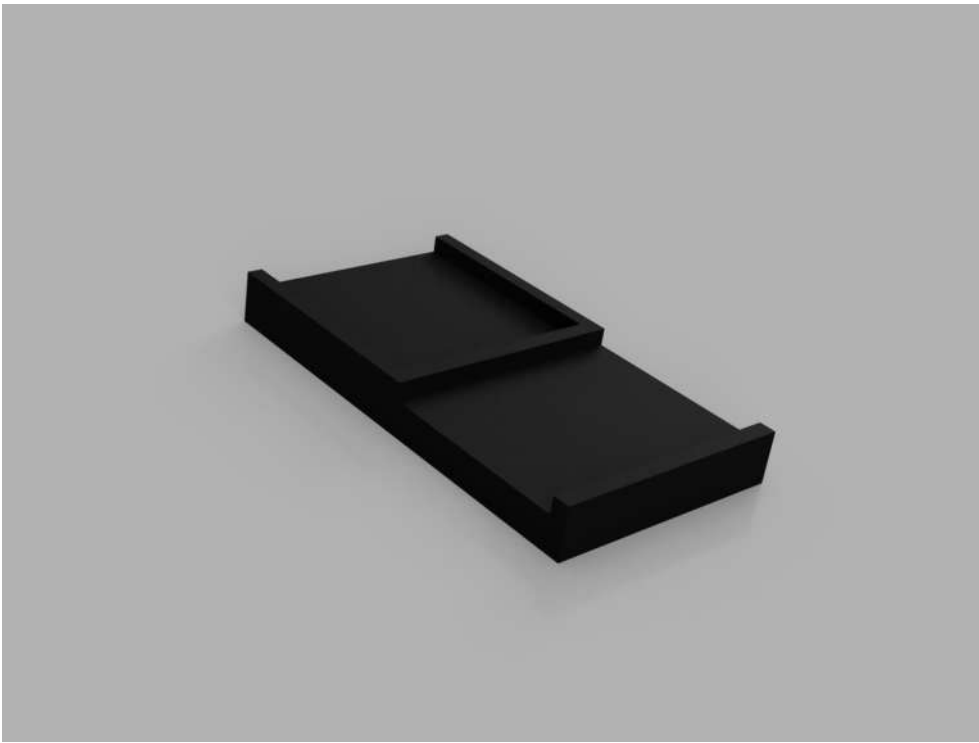


Abbildung 3

Render des Modells der oberen Platte aus Autodesk Fusion 360 (Software)



Abbildung 4

Render des Modells des Rads für die DC-Motoren aus Autodesk Fusion 360 (Software)

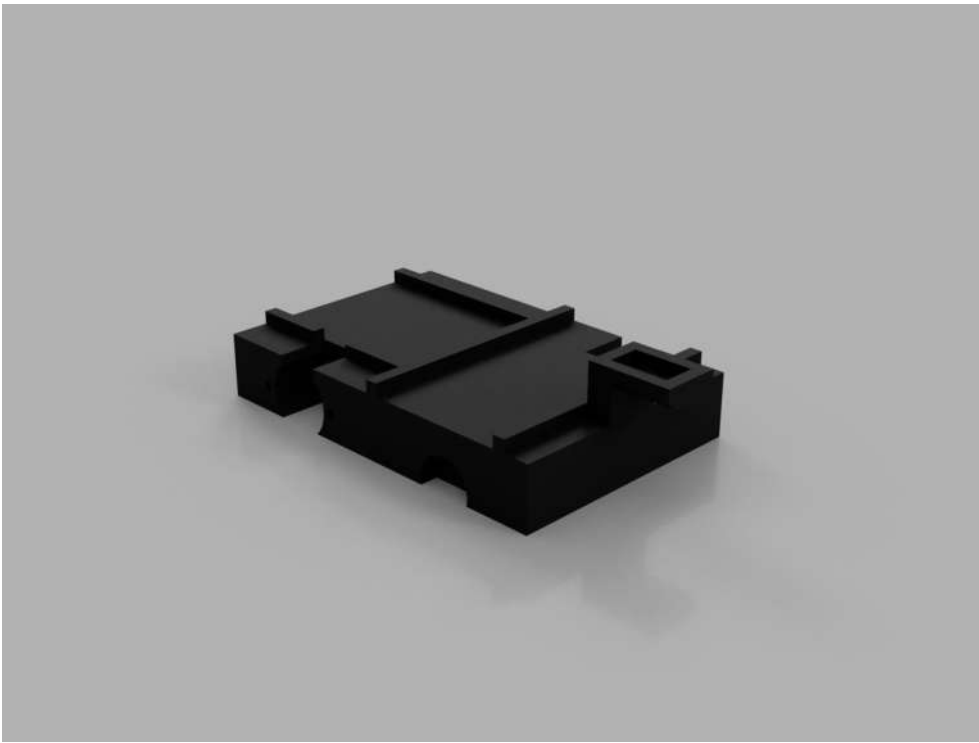


Abbildung 5

Render des Modells des Rumpfs aus Autodesk Fusion 360 (Software)



Abbildung 6

Render des Modells des Rads für die Stepper-Motoren aus Autodesk Fusion 360 (Software)



Abbildung 7

Render des Modells des Rads für die Kugellager aus Autodesk Fusion 360 (Software)



Abbildung 8

Bild der aufgeklebten Anti-Rutsch-Matte auf einem Reifen

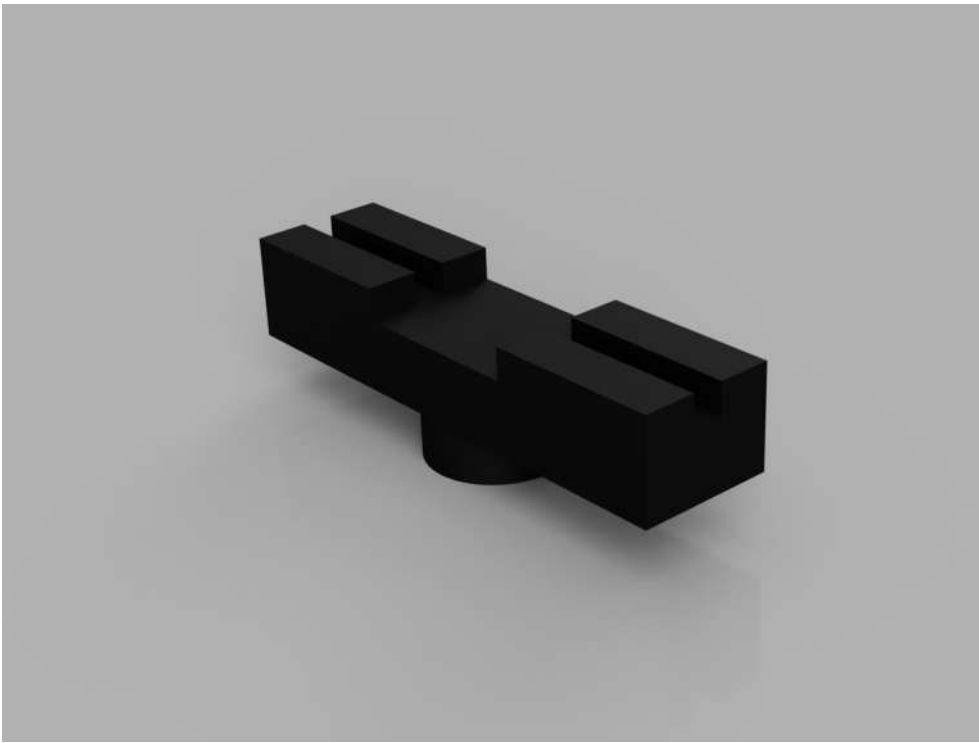


Abbildung 9

Render des Modells des Adapters zwischen Servomotor und Ultraschallsensor aus Autodesk Fusion 360 (Software)

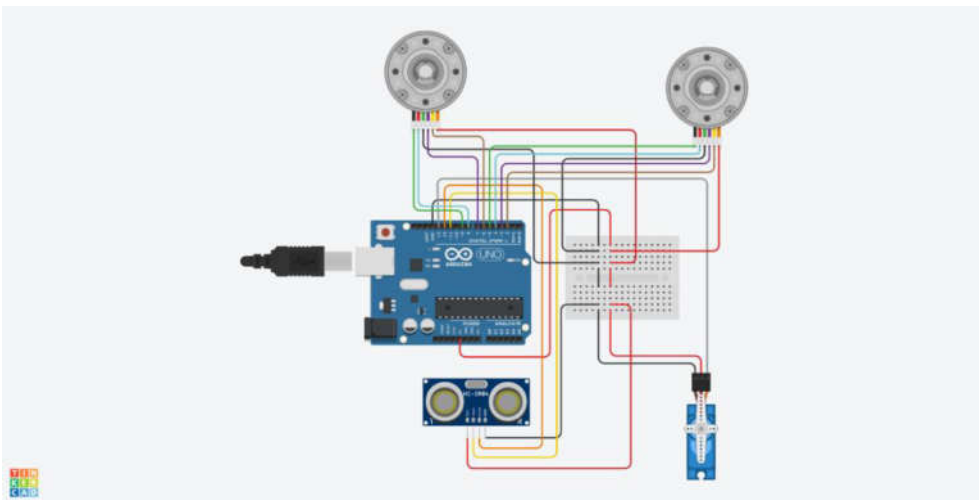


Abbildung 10

Steckplan des Marsrovers, erstellt mit Autodesk Tinkercad ( <https://www.tinkercad.com> )



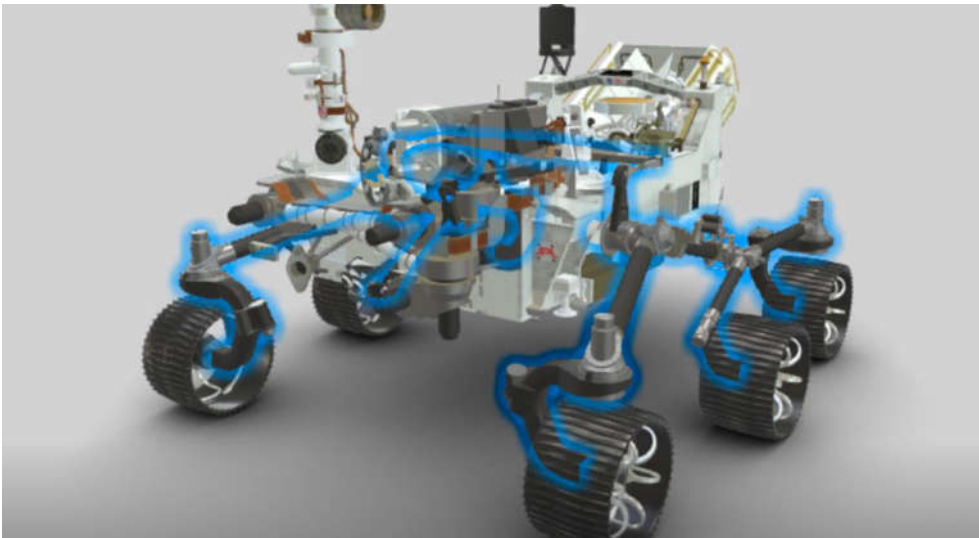


Abbildung 11

Suspension, Screenshot von <https://mars.nasa.gov/mars2020/spacecraft/rover/>

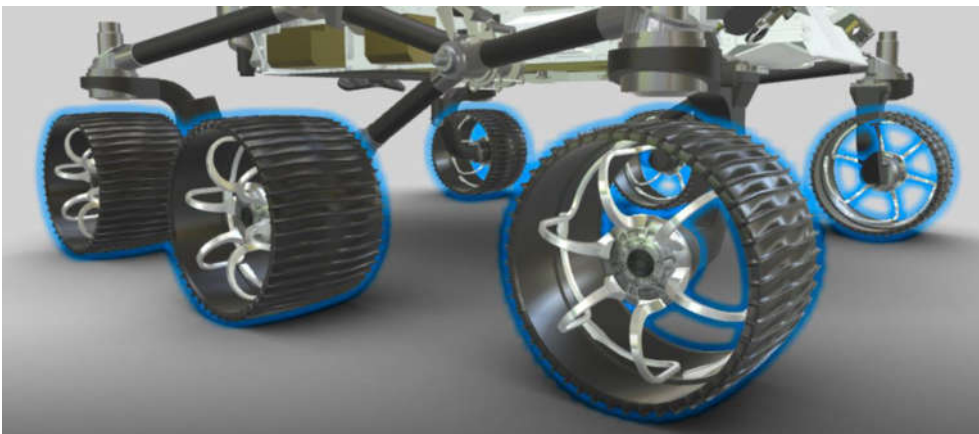


Abbildung 12

Wheels, Screenshot von <https://mars.nasa.gov/mars2020/spacecraft/rover/>

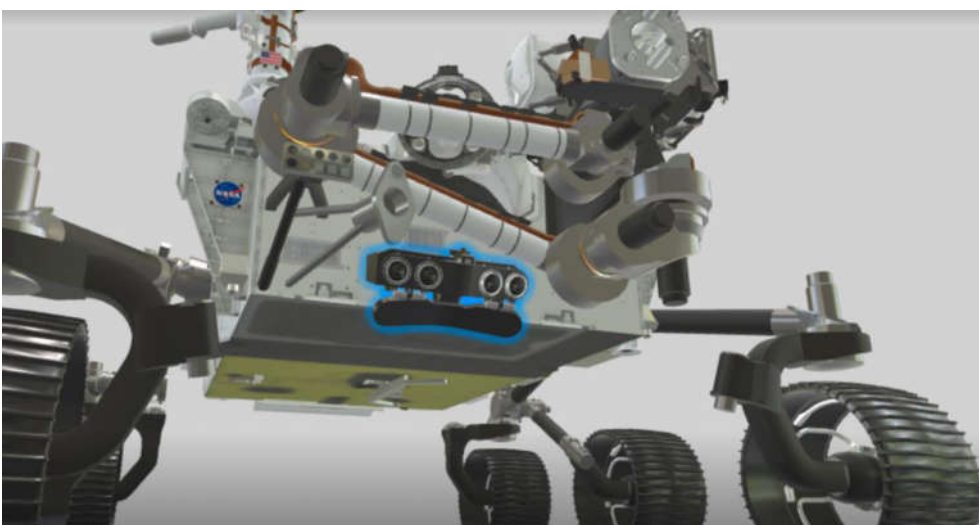


Abbildung 13

Fron HazCams, Screenshot von <https://mars.nasa.gov/mars2020/spacecraft/rover/>



Abbildung 14

Rear HazCams, Screenshot von <https://mars.nasa.gov/mars2020/spacecraft/rover/>

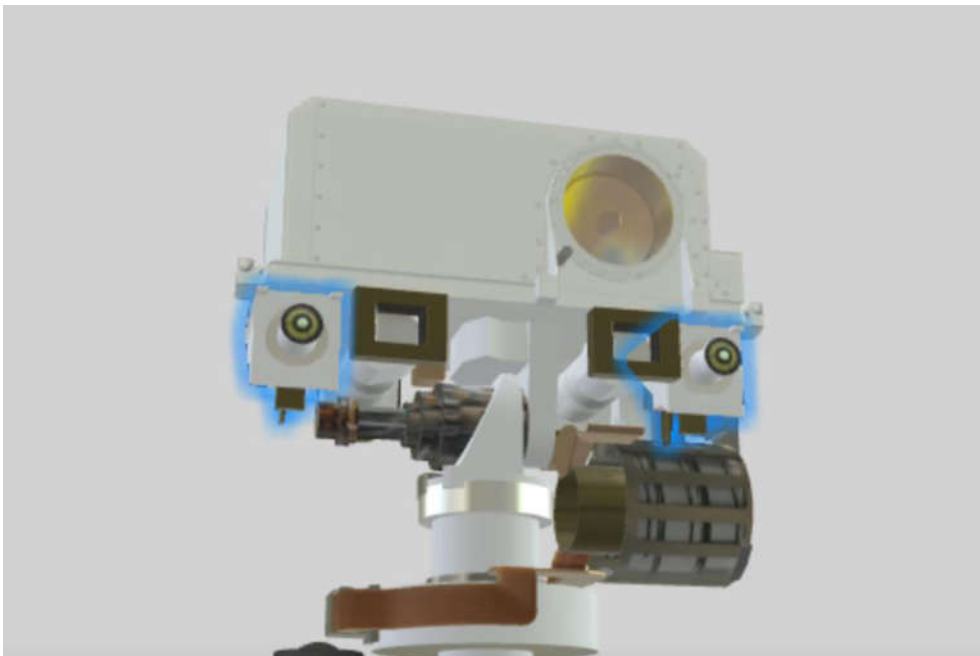


Abbildung 15

NavCams, Screenshot von <https://mars.nasa.gov/mars2020/spacecraft/rover/>

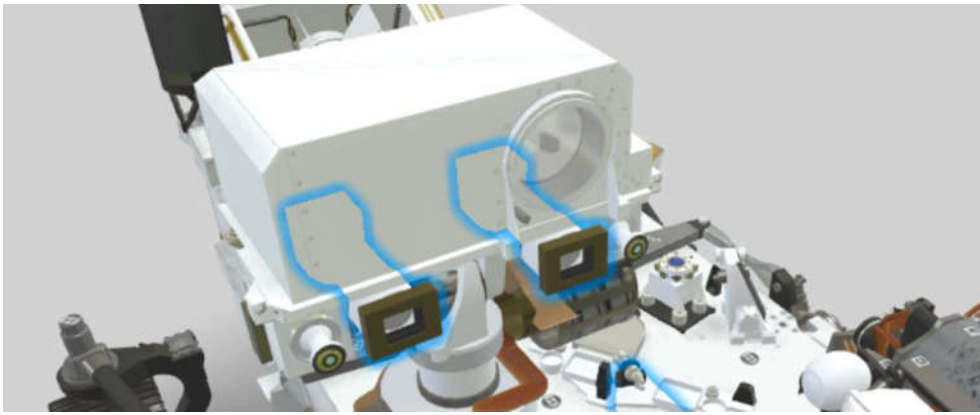


Abbildung 16

Mastcam-Z, Screenshot von <https://mars.nasa.gov/mars2020/spacecraft/rover/>

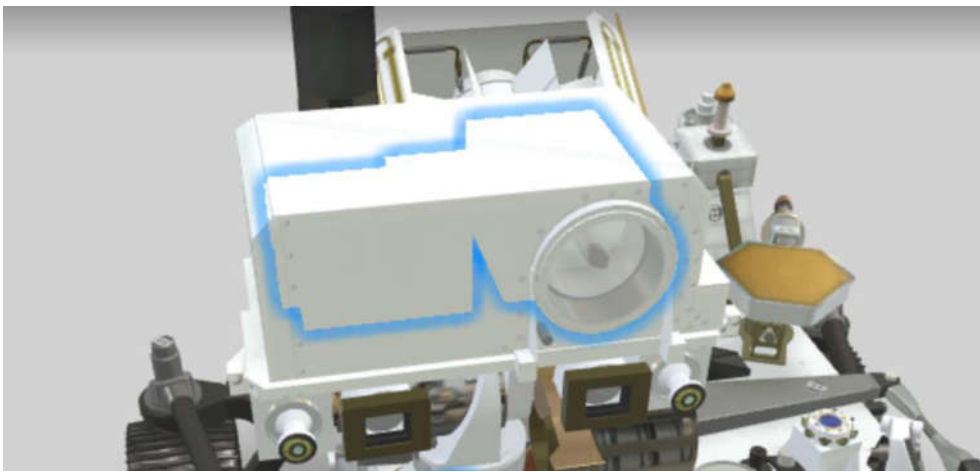


Abbildung 17

SuperCam-Z, Screenshot von <https://mars.nasa.gov/mars2020/spacecraft/rover/>

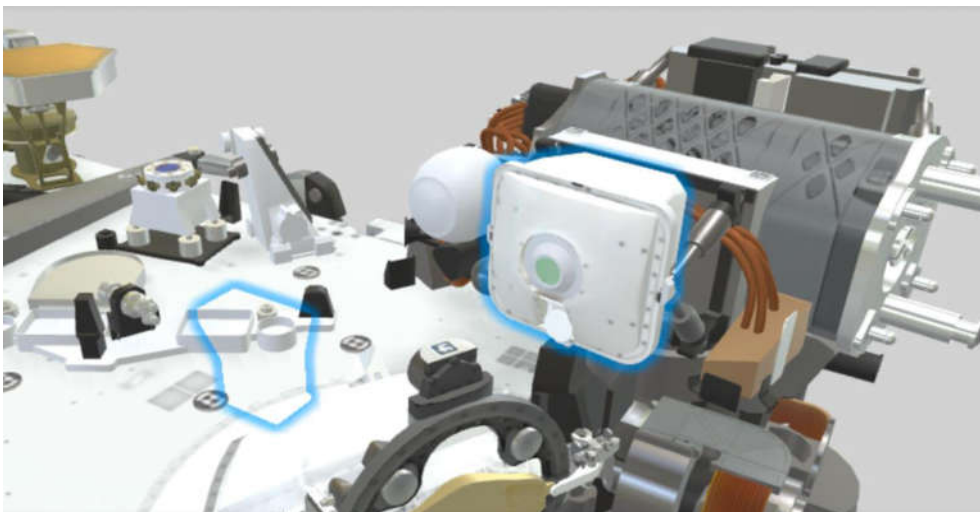


Abbildung 18

PIXL, Screenshot von <https://mars.nasa.gov/mars2020/spacecraft/rover/>



Abbildung 19

SHERLOC, Screenshot von <https://mars.nasa.gov/mars2020/spacecraft/rover/>

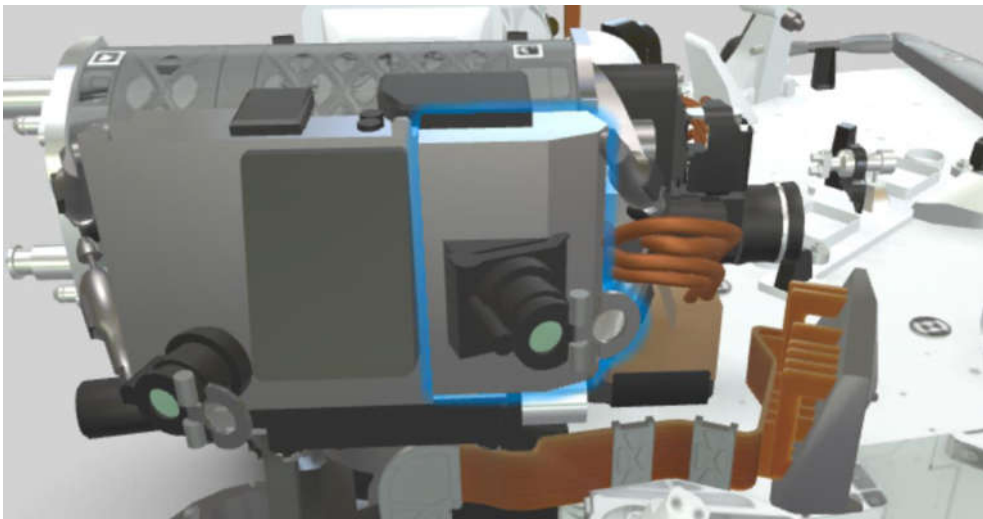


Abbildung 20

WATSON, Screenshot von <https://mars.nasa.gov/mars2020/spacecraft/rover/>



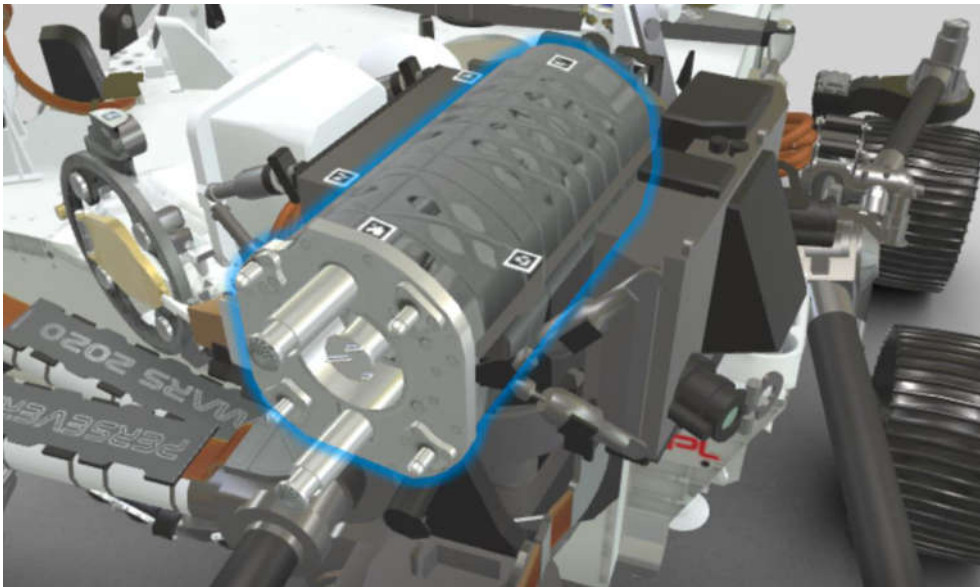


Abbildung 21

Coring Drill, Screenshot von <https://mars.nasa.gov/mars2020/spacecraft/rover/>

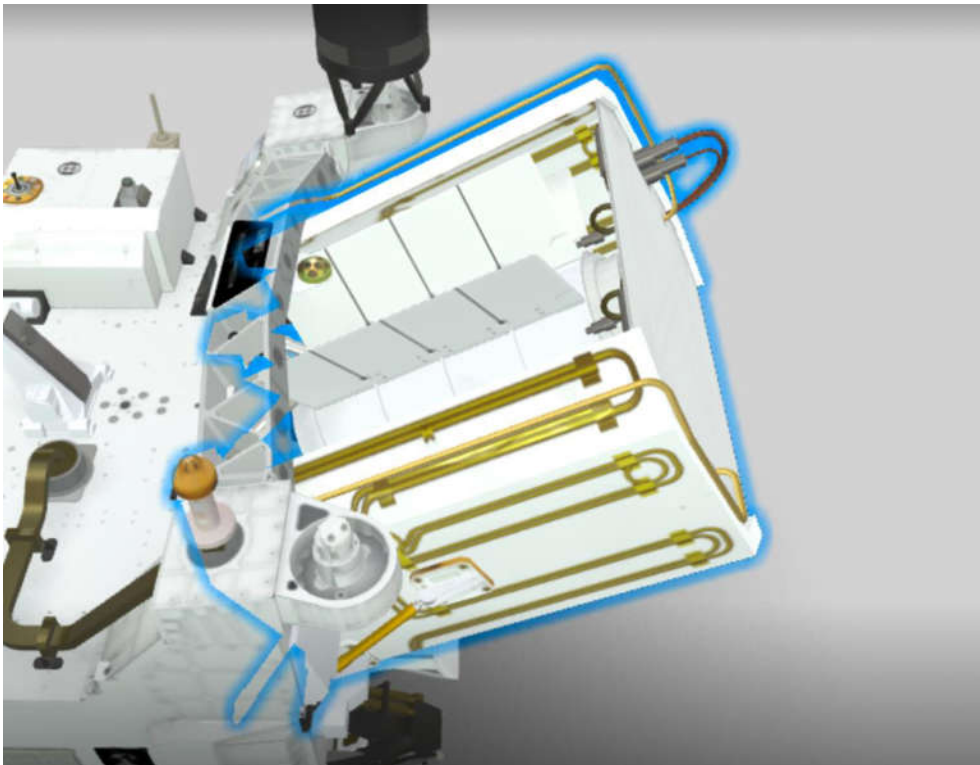


Abbildung 22

Power Source, Screenshot von <https://mars.nasa.gov/mars2020/spacecraft/rover/>

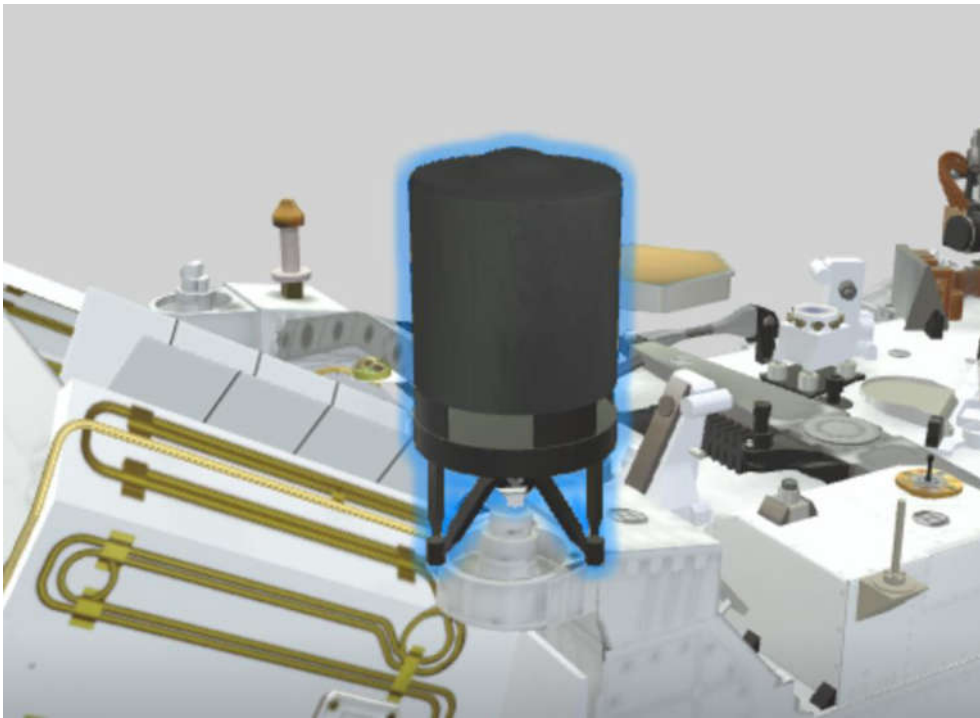


Abbildung 23

Ultra-High Frequency Antenna, Screenshot von

<https://mars.nasa.gov/mars2020/spacecraft/rover/>



Abbildung 24

High-Gain Antenna, Screenshot von <https://mars.nasa.gov/mars2020/spacecraft/rover/>

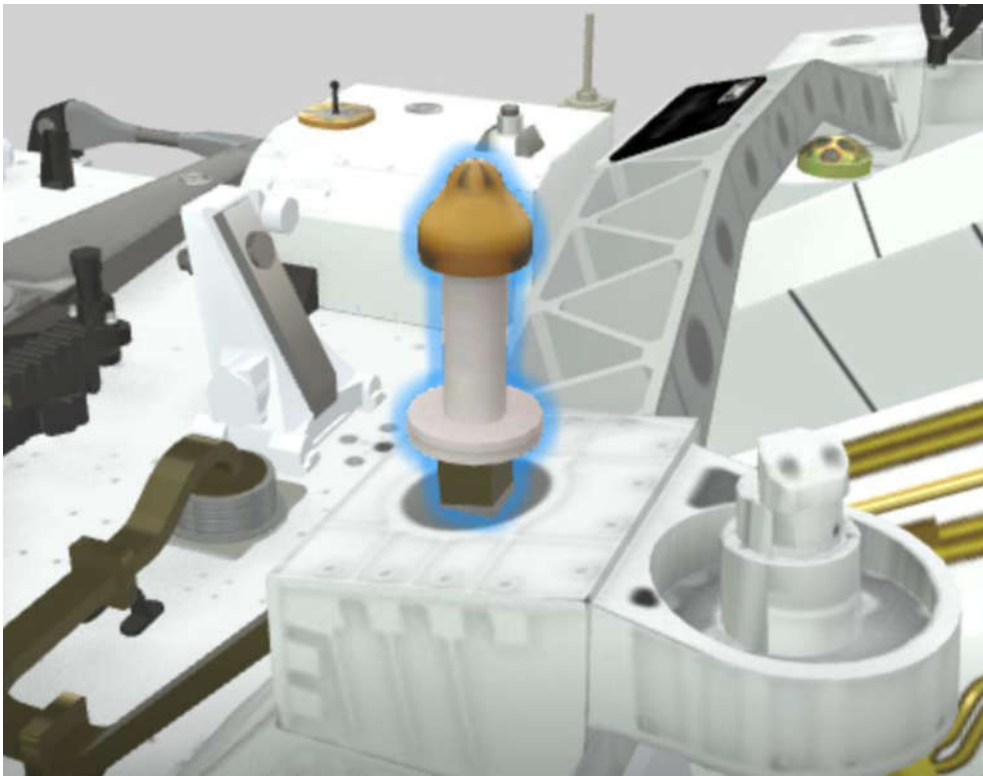


Abbildung 25

Low-Gain Antenna, Screenshot von <https://mars.nasa.gov/mars2020/spacecraft/rover/>

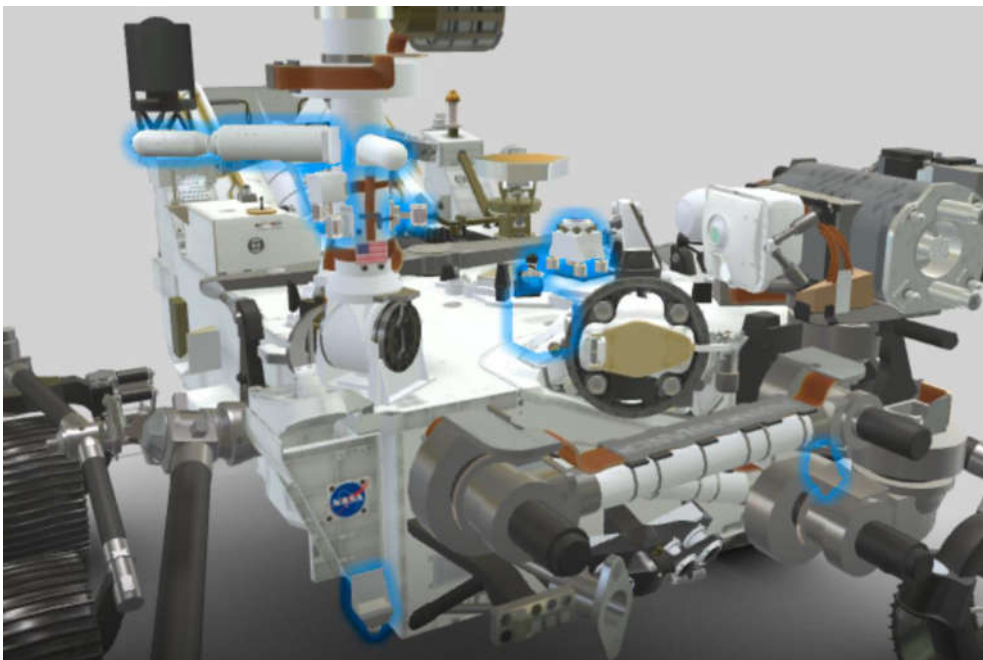


Abbildung 26

MEDA, Screenshot von <https://mars.nasa.gov/mars2020/spacecraft/rover/>

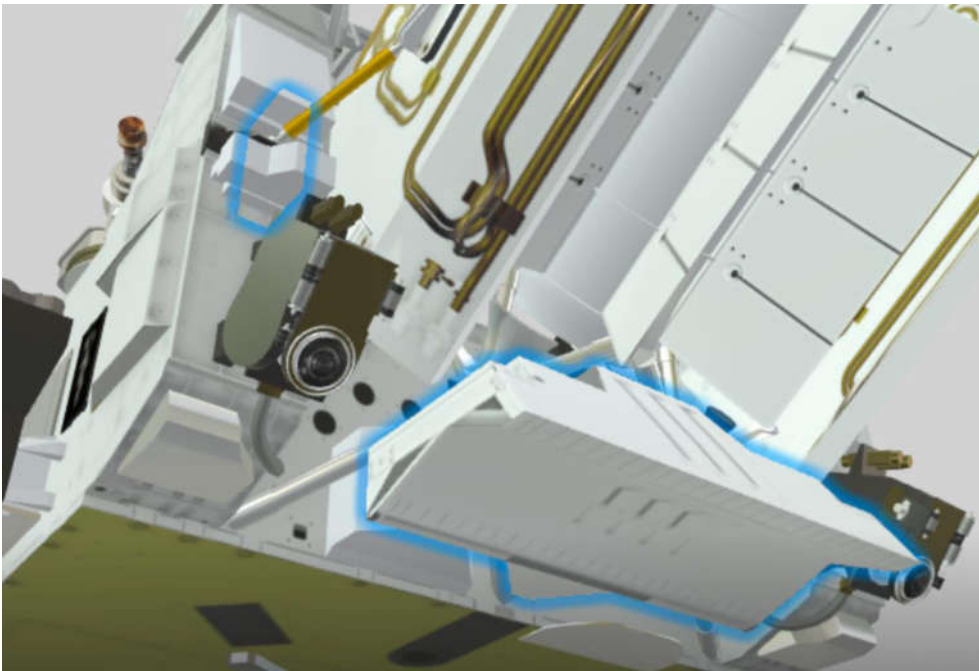


Abbildung 27

RIMFAX, Screenshot von <https://mars.nasa.gov/mars2020/spacecraft/rover/>

## 6.2 Teile

[1] Arduino Uno: <https://store.arduino.cc/products/arduino-uno-rev3>

[2] ANGEER 10x Mini Motor DC 1.5V-6V Micro130 Elektromotor 23000RPM Starker Magnetischer Motor für Spielzeug Auto Roboter DIY Fan Wissenschaft Projekt STEM

[https://www.amazon.de/ANGEER-Stücke-Mini-Motor-Arduino/dp/B07VCMZCCY/ref=sr\\_1\\_5?\\_mk\\_de\\_DE=ÅMÅŽÕÑ&crd=3I03UQUS7ZM9B&keywords=dc+motoren&qid=1698928969&srefix=dc+motoren%2Caps%2C151&sr=8-5](https://www.amazon.de/ANGEER-Stücke-Mini-Motor-Arduino/dp/B07VCMZCCY/ref=sr_1_5?_mk_de_DE=ÅMÅŽÕÑ&crd=3I03UQUS7ZM9B&keywords=dc+motoren&qid=1698928969&srefix=dc+motoren%2Caps%2C151&sr=8-5)

[3] ELEGOO 5er Stepper Motor Schrittmotor 5V 28BYJ-48 ULN2003 und 5 Stück Treiberplatine ULN2003 für Arduino

[https://www.amazon.de/Elegoo-Stepper-Schrittmotor-28BYJ-48-Treiberplatine/dp/B01MEGIHLF/ref=sr\\_1\\_2\\_sspa?crd=3I4VAEW5729N9W&keywords=stepper+motor+arduino&qid=1698939930&srefix=stepper+motor+ar%2Caps%2C135&sr=8-2-spons&sp\\_csd=d2lkZ2V0TmFtZT1zcF9hdGY&psc=1](https://www.amazon.de/Elegoo-Stepper-Schrittmotor-28BYJ-48-Treiberplatine/dp/B01MEGIHLF/ref=sr_1_2_sspa?crd=3I4VAEW5729N9W&keywords=stepper+motor+arduino&qid=1698939930&srefix=stepper+motor+ar%2Caps%2C135&sr=8-2-spons&sp_csd=d2lkZ2V0TmFtZT1zcF9hdGY&psc=1)



[4] Kugellager 8mm\*22mm\*7mm

[https://www.amazon.de/Ruesious-Kugellager-Shielded-Miniatur-Rillenkugellager/dp/B07VDRZDSC/ref=sr\\_1\\_5?\\_mk\\_de\\_DE=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crd=IC3FJTDL653M&keywords=kugellager%2B8x22x7%2B20%2Bst%C3%BCck%2Briesious&qid=1699219759&spreffix=kugellager%2B8x22x7%2B20%2Bst%C3%BCck%2Briesious%2Caps%2C83&sr=8-5&th=1](https://www.amazon.de/Ruesious-Kugellager-Shielded-Miniatur-Rillenkugellager/dp/B07VDRZDSC/ref=sr_1_5?_mk_de_DE=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crd=IC3FJTDL653M&keywords=kugellager%2B8x22x7%2B20%2Bst%C3%BCck%2Briesious&qid=1699219759&spreffix=kugellager%2B8x22x7%2B20%2Bst%C3%BCck%2Briesious%2Caps%2C83&sr=8-5&th=1)

[5] Feetech Mini Servo motor 120 degrees 9g

[https://store.arduino.cc/products/feetech-mini-servo-motor-120-degrees-9g?queryID=d7b32310ae73f4e288a83b2292670b9d&\\_gl=1\\*\\_skt8m9\\*\\_ga\\*MTA2MzQ5NTk0NC4xNjk4OTMwNjY4\\*\\_ga\\_NEXN8H46L5\\*MTY5ODk0MTE3My4zLjEuMTY5ODk0MTE4NS4wLjAuMA..](https://store.arduino.cc/products/feetech-mini-servo-motor-120-degrees-9g?queryID=d7b32310ae73f4e288a83b2292670b9d&_gl=1*_skt8m9*_ga*MTA2MzQ5NTk0NC4xNjk4OTMwNjY4*_ga_NEXN8H46L5*MTY5ODk0MTE3My4zLjEuMTY5ODk0MTE4NS4wLjAuMA..)

[6] Ultraschallsensor

[https://www.amazon.de/ARCELI-HC-SR04-Ultraschallmodul-Entfernungsmesswandler-Arduino/dp/B07MPZR59P/ref=sr\\_1\\_3?crd=1R61TNTJZ0E24&keywords=ultraschallsensor+arduino&qid=1698941417&spreffix=ultraschallsensor%2Caps%2C130&sr=8-3](https://www.amazon.de/ARCELI-HC-SR04-Ultraschallmodul-Entfernungsmesswandler-Arduino/dp/B07MPZR59P/ref=sr_1_3?crd=1R61TNTJZ0E24&keywords=ultraschallsensor+arduino&qid=1698941417&spreffix=ultraschallsensor%2Caps%2C130&sr=8-3)

[7] Extruder PLA NX 2 (Datenblatt extra)

[https://www.extruder.com/de/produkte/catalogue/pla-nx2-schwarz\\_1966/](https://www.extruder.com/de/produkte/catalogue/pla-nx2-schwarz_1966/)

## 6.3 Code

[1] Konstruktor

```
1  #include <Servo.h>
2  Servo servo;
3  #include <Stepper.h>
4  int SPU = 2048;
5  Stepper MotorL(SPU, 2, 4, 3, 5);
6  Stepper MotorR(SPU, 6, 8, 7, 9);
7  int trigger = 11;
8  int echo = 12;
9  int stopDist = 20;
10 long pulseTime = 0;
11 long distance = 0;
12 int turnDir = 1;
```

[2] setup

```
14 void setup()
15 {
16     servo.attach(13);
17     MotorL.setSpeed(12);
18     MotorR.setSpeed(12);
19     pinMode(trigger, OUTPUT);
20     pinMode(echo, INPUT);
21 }
```

[3] loop

```
23 void loop()
24 {
25     servo.write(90);
26     distance = getDistance();
27     if (distance >= stopDist)
28     {
29         moveForward();
30     }
31     if (distance <= stopDist)
32     {
33         turnDir = checkDirection();
34         turnDirection();
35     }
36 }
```

[4] move und turn

```
38 void moveForward()  
39 {  
40     MotorL.step(1);  
41     MotorR.step(-1);  
42 }  
43 void turnLeft()  
44 {  
45     MotorL.step(-2100);  
46     MotorR.step(-2100);  
47 }  
48 void turnRight()  
49 {  
50     MotorL.step(2100);  
51     MotorR.step(2100);  
52 }  
53 void turnAround()  
54 {  
55     turnLeft();  
56     turnLeft();  
57 }
```

[5] getDistance

```
59 int getDistance()  
60 {  
61     digitalWrite(trigger, LOW);  
62     delay(5);  
63     digitalWrite(trigger, HIGH);  
64     delay(10);  
65     digitalWrite(trigger, LOW);  
66     pulseTime = pulseIn(echo, HIGH);  
67     distance = (pulseTime / 2) * 0.03432;  
68     return distance;  
69 }  
70
```

#### [6] checkDirection

```
71  int checkDirection()
72  {
73      int distances[2] = { 0, 0 };
74      servo.write(180);
75      delay(500);
76      distances[0] = getDistance();
77      servo.write(0);
78      delay(1000);
79      distances[1] = getDistance();
80      if (distances[0] >= 200 && distances[1] >= 200)
81          | turnDir = 0;
82      else if (distances[0] <= stopDist && distances[1] <= stopDist)
83          | turnDir = 1;
84      else if (distances[0] >= distances[1])
85          | turnDir = 0;
86      else if (distances[0] <= distances[1])
87          | turnDir = 2;
88      return turnDir;
89  }
```

#### [7] turnDirection

```
91  void turnDirection()
92  {
93      switch (turnDir)
94      {
95          case 0:
96              | turnLeft();
97              | break;
98          case 1:
99              | turnAround();
100             | break;
101          case 2:
102              | turnRight();
103              | break;
104             }
105 }
```

# 7 Quellen

## 7.1 Literaturquellen

[1] Hawking Stephen & Lucy: Das Universum: Was unsere Welt zusammenhält, 4.Auflage, München, 2020, ISBN 978-3-570-17815-7

## 7.2 Internetquellen

[1] John Updike, Besuch auf dem Mars, <https://www.nationalgeographic.de/wissenschaft/besuch-auf-dem-mars> , Stand: 1.11.2023

[2] Matthias Böhm, Die Geologie des Mars, <https://web.archive.org/web/20030817005210/http://www.uni-bonn.de/~uzsrcj/index/Geologie/Arbeiten/Mars/Mars-Referat.htm> , Stand 1.11.2023

[3] Mars Society Deutschland e.V., Allgemeine Informationen zum Mars, <https://www.marssociety.de/de/der-mars/allgemeine-informationen-zum-mars.html> , Stand 1.11.2023

[4] Max-Planck-Institut, Marsforschung am Max-Planck-Institut für Sonnensystemforschung, <https://www2.mps.mpg.de/dokumente/publikationen/poster/marsforschung.pdf> , Stand: 1.11.2023

[5] Jochen Zielke, Steckbrief Mars, <https://www.planetwissen.de/natur/weltall/mars/pwiesteckbriefmars100.html#:~:text=und%20trostlosen%20Planeten., Unser%20Nachbarplanet,auf%2055%20Millionen%20Kilometer%20an.> , Stand: 1.11.2023

[6] Frankfurter Allgemeine, Panoramaflug über den größten Canyon des Sonnensystems, <https://www.faz.net/aktuell/wissen/weltraum/mars-missionen-panoramaflug-ueber-den-groessten-canyon-des-sonnensystems-1303671.html> , Stand: 1.11.2023

[7] National Aeronautics and Space Administration, Olympus Mons, <https://mars.nasa.gov/gallery/atlas/olympus-mons.html> , Stand; 1.11.2023

[8] National Aeronautics and Space Administration, Valles Marineris, <https://mars.nasa.gov/gallery/atlas/valles-marineris.html> , Stand: 1.11.2023

- [9] Nadja Podbregar, Ist der Mars doch vulkanisch aktiv?, <https://www.scinexx.de/news/kosmos/ist-der-mars-doch-vulkanisch-aktiv/> , Stand: 1.11.2023
- [10] Rainer Kayser, Wie der Mars seine Atmosphäre verlor, <https://www.weltderphysik.de/gebiet/universum/nachrichten/2017/wie-der-mars-seine-atmosphaere-verlor/> , Stand: 1.11.2023
- [11] Mike Wall, Mars' Atmosphere Was Likely More Oxygen-Rich Long Ago, <https://www.space.com/33296-mars-atmosphere-oxygen-curiosity-rover.html> , Stand: 1.11.2023
- [12] The European Space Agency, Neue Indizien für Leben auf dem Mars, [https://www.esa.int/Space\\_in\\_Member\\_States/Germany/Neue\\_Indizien\\_fuer\\_Leben\\_auf\\_dem\\_Mars](https://www.esa.int/Space_in_Member_States/Germany/Neue_Indizien_fuer_Leben_auf_dem_Mars) , Stand: 1.11.2023
- [13] Jochen Zielke und Uwe Gradwohl, Geschichte der Marsmissionen, <https://www.planetwissen.de/natur/weltall/mars/pwiegeschichtedermarsmissionen100.html> , Stand: 1.11.2023
- [14] The Planetary Society; Perseverance, NASA's newest Mars rover; <https://www.planetary.org/space-missions/perseverance> , Stand: 1.11.2023
- [15] Nola Taylor und Daisy Dobrijevic, How long does it take to get to Mars?, <https://www.space.com/24701-how-long-does-it-take-to-get-to-mars.html> , Stand 2.11.2023
- [16] Tech Differences, Difference Between int and long, <https://techdifferences.com/difference-between-int-and-long.html> , Stand: 2.11.2023
- [17] Freie Universität Berlin, Binärzählen, <https://www.osa.fu-berlin.de/mathematik/lehramt/aufgaben/binaerzahlen/index.html> , Stand: 2.11.2023
- [18] Funduino, Nr.11 Entfernungen messen, <https://funduino.de/nr-10-entfernung-messen> , Stand: 2.11.2023
- [19] National Aeronautics and Space Administration, Quick Facts, <https://mars.nasa.gov/mars2020/> , Stand: 3.11.2023
- [20] National Aeronautics and Space Administration, Rover, <https://mars.nasa.gov/mars2020/spacecraft/rover/> , Es wurden Informationen aus dem interaktiven Modell benutzt, Stand: 3.11.2023

- [21] National Aeronautics and Space Administration, Body, <https://mars.nasa.gov/mars2020/spacecraft/rover/body/> , Stand: 3.11.2023
- [22] National Aeronautics and Space Administration, Wheels and Legs, <https://mars.nasa.gov/mars2020/spacecraft/rover/wheels/> , Stand: 3.11.2023
- [23] National Aeronautics and Space Administration, The Cameras on the Mars 2020 Perseverance Rover, <https://mars.nasa.gov/mars2020/spacecraft/rover/cameras/#NavCams-Engineering-Cameras> , Stand: 3.11.2023
- [24] National Aeronautics and Space Administration, Robotic Arm, <https://mars.nasa.gov/mars2020/spacecraft/rover/arm/#Drill> , Stand: 3.11.2023
- [25] National Aeronautics and Space Administration, Electrical Power, <https://mars.nasa.gov/mars2020/spacecraft/rover/electrical-power/> , Stand: 3.11.2023
- [26] National Aeronautics and Space Administration, Communications, <https://mars.nasa.gov/mars2020/spacecraft/rover/communications/#Low-Gain-Antenna> , Stand: 3.11.2023

## Erklärung zur Seminararbeit

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe.

Insbesondere versichere ich, dass ich alle wörtlichen und sinngemäßen übernahmen aus anderen Werken als solche gekennzeichnet habe.

Bruckmühl, den \_\_\_\_\_

\_\_\_\_\_

Unterschrift