

**Mars Rover Challenge**

The Mars Raiders

Eingereicht bei:

**Fachhochschule Kufstein Tirol**

**Studiengang Smart Products & Solutions**

Verfasser:

**Marco Gökmen, Bettina Oberloher, Alexander Mairhofer, Kaj Nocker, Philip Reitstätter, Nicolas Roth, Martin Schafferer, Stefan Sulzenbacher**

**SPS.bbM.18**

**Inhaltsverzeichnis**

[Einführung 1](#_Toc3756765)

[1.1 Beschreibung der Aufgabenstellung 1](#_Toc3756766)

[1.2 Restriktionen 2](#_Toc3756767)

[1.3 Analyse bestehender Konzepte 3](#_Toc3756768)

[Material und Methoden 5](#_Toc3756769)

[2.1 Fertigungstechnologien 5](#_Toc3756771)

[2.2 CAD-Design 6](#_Toc3756772)

[2.3 Projektmanagement 8](#_Toc3756773)

[2.3.1 Projektteam „Mars Raiders – Not fast, but furious“ 8](#_Toc3756774)

[2.3.2 Projektdefinition und -planung 9](#_Toc3756775)

[2.3.3 Krisenmanagement 11](#_Toc3756776)

[2.3.4 Resultate und Lessons Learned 11](#_Toc3756777)

[Resultate und Diskussion 13](#_Toc3756778)

[3.1 Mechanisches Konzept 13](#_Toc3756780)

[3.1.1 Konzeptfindung 13](#_Toc3756781)

[3.1.2 Entscheidung für das finale Konzept 13](#_Toc3756782)

[3.1.3 Konzeptausführung Räder 15](#_Toc3756783)

[3.1.4 Konzeptausführung Gehäuse 18](#_Toc3756784)

[3.2 Sensor- und Aktorkonzept 19](#_Toc3756785)

[3.2.1 Ultraschall Sensoren 19](#_Toc3756786)

[3.2.2 Linetracking Sensoren 20](#_Toc3756787)

[3.2.3 Kamera 20](#_Toc3756788)

[3.2.4 Drehzahlsensoren 20](#_Toc3756789)

[3.2.5 Gyro – und Beschleunigungssensor 21](#_Toc3756790)

[3.3 Softwarekonzept 21](#_Toc3756791)

[3.3.1 Konzeptfindung 21](#_Toc3756792)

[3.3.2 Plattformsoftware 22](#_Toc3756793)

[3.3.3 Applikationssoftware 24](#_Toc3756794)

[3.4 Printed Circuit Board PCB - Design 29](#_Toc3756795)

[3.5 Systems Engineering 32](#_Toc3756796)

[Schlussfolgerung 33](#_Toc3756797)

[A1 Stückliste 1](#_Toc3756798)

[A2 Laserzuschnitt Teile 2](#_Toc3756799)

[A3 Berechnung Getriebe 3](#_Toc3756800)

****Abbildungsverzeichnis****

[Abbildung 1: Vorderansicht Marsrover 7](#_Toc3756801)

[Abbildung 2: Auswahl Projektmanagement Methode 9](#_Toc3756802)

[Abbildung 3: Projektplan 10](#_Toc3756803)

[Abbildung 4: Rad mit Antrieb 15](#_Toc3756804)

[Abbildung 5: Antriebskraft Berechnung 16](#_Toc3756805)

[Abbildung 6: Befestigung Gabelstange 17](#_Toc3756806)

[Abbildung 7: Seitenansicht Marsrover 18](#_Toc3756807)

[Abbildung 8: Querschnitt Marsrover 19](#_Toc3756808)

[Abbildung 10: Ablauf des Skriptkonzepts 25](#_Toc3756809)

[Abbildung 11: Skriptlogik Track 1 und 2 26](#_Toc3756810)

[Abbildung 12: Skriptlogik Track 3 26](#_Toc3756811)

[Abbildung 13: Skriptlogik Track 4 29](#_Toc3756812)

[Abbildung 14: Layout der Fertigen Platine1 31](#_Toc3756813)

****Tabellenverzeichnis****

[Tabelle 1: Entscheidungsmatrix der Konzepte 14](#_Toc3756814)

[Tabelle 2: PCA9555 Ansteuerung 23](#_Toc3756815)

****Kurzbeschreibung****

Das Projekt beschäftigt sich mit der Planung, Entwicklung, Konstruktion, sowie das Testen eines fiktiven Mars Rovers. Das Projekt ist eine Kombination aus den Fächern: Software Engineering, Embedded Systems und Projektmanagement. Zusätzlich fließen Inhalte aus den Kursen Sensorik und Einführung in die Programmierung ein. Somit stellt die MarsRover Challenge eine Zusammenführung fast aller Kurse aus dem ersten Semester dar.

Das autonome Fahrzeug muss bei diesem Projekt die Tracks: Climbing the Wall, Bridge over Troubled Water, Make it Rain und Rock ‘n’ Roll (genauere Beschreibung der einzelnen Tracks sind in der Einführung) absolvieren.

Zur Bewältigung der einzelnen Aufgaben musste für das autonome Fahrzeug das Chassis konstruiert werden, die Software entwickelt, die Elektronik geplant und eine Platine erstellt werden. Aufgrund der zeitlichen Restriktion wurde das Team in Subteams eingeteilt, um die Fertigstellung gewährleisten zu können. Für die Projektdurchführung standen die Räumlichkeiten der Fachhochschule Kufstein und das Fablab der Werkstätte Wattens zur Verfügung. Für die Konstruktion stand ein Lasercutter und 3D-Drucker zur Verfügung. Für die Software wurde eine Raspberry Pi Model B 3 verwendet und dieser mit der erstellten Platine verbunden.

Für das Projektmanagement wurde eine Kombination von traditionellem und agilem Projektmanagement eingesetzt, um das Projekt in der vorgegebenen Zeit umzusetzen. Für die Projektkommunikation wurde die Dropbox, sowie Github verwendet.

Zusammenfassend konnte das Projekt aufgrund eines sehr gut zusammenarbeitenden Teams und einer Arbeitsteilung in der vorgegebenen Zeit umgesetzt werden. Das Team plante, konstruierte und programmierte ein funktionsfähiges autonomes Fahrzeug. Auch mit abschließenden technischen Problemen konnte das Team an den Challenges teilnehmen. Abschließend hat das Team mit diesem Projekt die Kurse aus dem ersten Semester kombiniert und jeder individuelle Lernerfahrungen aus den Kursen Embedded Systems, System Engineering, Projektmanagement, Sensorik und Einführung in die Programmierung mitnehmen können.

# Einführung

Die “MarsRover Challenge” stellt ein kursübergreifendes Projekt dar, bei dem die Teammitglieder ihr erarbeitetet Wissen aus den Kursen:

* Embedded Systems
* Systems Engineering
* Projektmanagement

anwenden und vertiefen. Dabei wird das Team mit den Herausforderungen:

* Konstruktion
* Elektronik
* Platinenplanung
* Softwareentwicklung

konfrontiert. Die einzelnen Teammitglieder erweitern ihre Fähigkeiten im Bereich der Planung, Umsetzung, Problembehandlung und Teamwork. Besonders im Fokus steht dabei, dass erweitern der Kompetenzen in technischen und informationstechnologischen Projekten.

## Beschreibung der Aufgabenstellung

Die Teams müssen sich mit 4 verschiedenen Tracks befassen:

* Climbing Up the Walls
* Bridge Over Troubled Water
* Make it Rain
* Rock ‘n‘ Roll

Alle Tracks müssen dabei vom Fahrzeug autonom, d.h. ohne Einwirkung der Teammitglieder, absolviert werden.

Climbing Up the Walls

Die erste Aufgabe beschäftigt sich mit dem Überwinden einer Rampe. Das Fahrzeug muss eine Gesamthöhe von 1m mit einer maximalen Steigung von 45° absolvieren.

Bridge over Troubled Water

Bei dieser Challenge muss eine Brücke überquert werden. Die Brücke besteht aus zwei 30mm dicken Kanthölzern, die in einem Abstand von 350mm von der Holzmitte auseinander liegen. Das Fahrzeug muss dabei eine Distanz von 1m zurücklegen.

Make It Rain

Der MarsRover wird dabei vor die Herausforderung gestellt einen Korridor mit der Länge von 3m und einer Breite von 500mm zu überwinden. Beim Verlassen des Korridors dürfen die außerhalb liegenden Fahrzeugteile mit Wasser besprüht werden.

Rock ‘n‘ Roll

In einem Bereich von 5m Länge und 2m Breite werden Ramps und Obstacles aufgebaut. Die Ramps stellen dabei überfahrbare Hindernisse mit einer maximalen Höhe von 100mm und einer Steigung zwischen 30° bis 45° dar. Die Obstacles sind nicht-überfahrbare Hindernisse mit einer Mindesthöhe von 100mm. Die Obstacles sind rote Hindernisse (RGB: 255,0,0), damit diese erkannt werden können.

## Restriktionen

Für den finalen Prototypen sind folgende Restriktionen vorgesehen:

* Länge < 65cm
* Breite < 48 cm
* Höhe < 30 cm
* Gewicht < 3kg
* Kein Breadboard erlaubt
* Staub-/wasserfest

Für die Fertigung des Prototyps stehen ein gewisses Materialkontingent, sowie eine gewisses Zeitkontingent für die Fertigungsmaschinen im Fablab Wattens zur Verfügung.

## Analyse bestehender Konzepte

Konzept: Sand Flea Boston Dynamics

<https://www.youtube.com/watch?v=6b4ZZQkcNEo>

<https://www.bostondynamics.com/sandflea>

Das Konzept des Sand Flea baut auf einem 4 Rad DC Fahrzeug auf. Zur Überwindung von Hindernissen ist ein Sprungmechanismus eingebaut, der Hindernisse bis zu 10m Höhe überwinden kann. Die Räder werden dabei als Gyro verwendet, um eine sichere Landung zu gewährleisten. Bei diesem Konzept ergeben sich 2 Probleme bei der Umsetzung der MarsRover Challenge. Das erste Problem stellt der Sprungmechanismus dar. Es ist unklar, wie sich dieser Mechanismus verhält auf unterschiedlichen Bodengegebenheiten. Anhand der vorhandenen Informationen ist der Mechanismus nur auf festem Untergrund getestet, daher kann nicht analysiert werden, ob diese Funktion auch auf weichem Boden funktioniert. Das zweite Problem stellt das Gewicht dar. Bei einem Gewicht von 5kg ist das Fahrzeug um 2kg zu schwer für die MarsRover Challenge.

Die Ansteuerung des Sprungmechanismus stellt eine weitere Herausforderung für die Umsetzung dar. Dabei muss der Sprungmechanismus rechtzeitig ausgeführt werden, damit die Flugbahn des Fahrzeugs auch das Hindernis überwinden kann.

Konzept: Grappling Hook

<https://www.youtube.com/watch?v=mCGoqTE3uYo&feature=youtu.be>

Dieses Konzept baut auf den Ansatz auf, dass das Fahrzeug Hindernisse durch Verwendung eines Hackens überwindet. Das zusätzliche Gewicht, was für den zusätzlichen Motor und den Hacken entstehen können ein Problem für die MarsRover Challenge darstellen. Für die Verwendung des Seils, müssen einige Testversuche durchgeführt werden, um zu gewährleisten, dass die Stabilität gegeben ist. Ein weiteres Problem bei diesem Konzept ist, dass der Hacken sich verkeilen muss, ansonsten ist dieses Konzept für die Überwindung von Hindernissen nicht anwendbar.

In der Software muss berücksichtigt werden, wann der Hacken automatisch ausgelöst wird und in welchem Winkel die Flugbahn des Hackens verläuft, um zu gewährleisten, dass das Hindernis überwunden werden kann.

Konzept: ARTI Stair Climbing Mobile Robot

<https://www.youtube.com/watch?v=sohpgx6j_lk>

<https://www.solvelight.com/product/arti3-mobile-robot-platform/>

Dieses Konzept baut auf einem 3-moduligen Kettenfahrzeug auf. Die Ketten des vordersten Moduls sind als ein rechtwinkliges Dreieck aufgebaut, um die Hindernisse anzufahren. Die beiden hinteren Module schieben dabei das vordere an, um das Hindernis zu überwinden. Probleme bei dieser Konstruktion stellen Elektronik und Stabilität zwischen den einzelnen Modulen dar. Aufgrund der Bauweise könnte die Einhaltung der Gewichtsrestriktion eventuell nicht eingehalten werden.

Konzept: SAR URC 2018 Impuls Team

<https://www.youtube.com/watch?time_continue=203&v=h0edkzrgSg0>

Bei dem Konzept vom Impuls Team baut das Chassis auf einem Modul auf, dass durch 4- Radantrieb gesteuert wird. Zusätzlich ist ein Greifarm angebracht, der gewisse Tätigkeiten ausführen kann. Diese Zusatzfunktion ist für die MarsRover Challenge irrelevant und wird daher nicht weiter berücksichtigt. Zur Verbreiterung und Sicherung der Stabilität werden pro Radaufhängung 3 Räder verwendet. Für den Einsatz bei der MarsRover Challenge muss das Konzept des Fahrzeugs verkleinert werden, da sondern die Restriktionen für die Abmessungen nicht eingehalten werden.

Material und Methoden

Um eine möglichst genaue Abbildung der Produktentwicklung eines realen Projektes in der Wirtschaft nachzubilden, waren die Studierenden gezwungen mit beschränkten Ressourcen zu arbeiten. Es waren sowohl die physikalischen Materialien als auch die immateriellen Ressourcen wie Zeit und Arbeitskraft reduziert. Zur Fertigung des Marsrovers wurden uns folgende Materialien zur Verfügung gestellt:

* 4 x Karton (60x80 cm)
* 2 x Sperrholzplatte (60x80 cm)
* 1 x Plexiglas (60x40 cm)

Den Teams wurde es erlaubt, die Materialien untereinander auszutauschen um z.B. Sperrholzplatten in der Stärke von 4 mm, 6 mm sowie 8 mm zu verwenden.

Weiters wurden Arduino Uno Entwicklerboards, Raspberry Pi Microcomputer, Sensoren des Sensorkits sowie Verbindungselemente wie Jumper-Wire, Widerstände etc. in jeder Menge zur Verfügung gestellt.

Als Arbeitswerkzeug wurde in der Werkstätte Wattens zwei CO2-Lasercutter, vier 3D-Drucker und eine Drehbank bereitgestellt.



## Fertigungstechnologien

Mit reduzierten Ressourcen zu arbeiten wurde von vornherein in unserem Projekt beachtet. Aus diesem Grund wurde bereits zu Beginn festgelegt, dass der Großteil der Komponenten aus Sperrholzplatten bestehen muss und lediglich wenige Einzelteile mithilfe eines 3D-Druckers erzeugt werden. Die Herstellung der Bauteile mit Hilfe eines Lasercutters ist besonders schnell und einfach. Die Bauteile wurden mit der CAD Software SolidWorks dreidimensional entwickelt und als zweidimensionale Ableitung an die CAD Software Inkscape übergeben. Inkscape ermöglicht es, die Bauteile so vorzubereiten, dass sie problemlos von dem CAD-Lasercutter bearbeitet werden können. Dazu müssen diverse Einstellungen vorgenommen werden:

* rote Konturen --> Gravieren; schwarze Konturen --> Schneiden
* Linienstärke auf 0,1 mm stellen
* Lasercutter als Drucker auswählen
* Grundmaterial zum Schneiden auswählen
* Materialstärke auswählen
* Laser-Startpunkt bestimmen

Bevor der Laservorgang gestartet werden konnte, musste noch die Linse des Laserkopfes gereinigt werden. Anschließend wurde das entsprechende Material in die vorgesehene Position gebracht und der Startpunkt festgelegt. Sobald alle Vorkehrungen getroffen wurden, konnte der Schneidevorgang durchgeführt werden.

Das Konzept des Marsrover wurde so erstellt, dass keinerlei Klebe oder Schraubverbindungen (lediglich für die Befestigung der Lichtschranke) notwendig waren. Alle Verbindungen wurden als Presspassung ausgeführt. Die Bauteile bleiben an ihrer Position entweder durch reibschlüssige Verbindung oder formschlüssige Verbindung. Bei der reibschlüssigen Verbindung ist darauf zu achten, dass der Laserschneider direkt auf der Konturlinie schneidet und eine Schnittbreite besitzt. Dadurch weichen die Schnittmaße von der Vorgabe ab. Zu beginn des Projektes wurde eine Schablone erstellt, um den Korrekturfaktor zu ermitteln. Es zeigte sich, dass die Aussparung 0,5 mm kleiner sein soll als der Zapfen um eine selbsthemmende Presspassung sicherzustellen.

3D-Druckteile wurden vermieden, da deren Herstellung zu viel Zeit in Anspruch nehmen würde. Aus diesem Grund wurden die Radachsen aus Kunststoff mit der Drehbank in der Werkstätte Wattens hergestellt. Als Grundwerkstoff wurde ein PVC Kunststoff verwendet.

## CAD-Design

Den Studenten der Fachhochschule Kufstein steht eine kostenfreie Lizenz der CAD Software SolidWorks 2018 zur Verfügung. Diese Lizenz beinhaltet alle Elemente und Zusatzanwendungen einer Vollwertigen Premiumlizenz im Wert von ca. 25.000 €.

Mit der Unterstützung der umfangreichen SolidWorks Funktionen erarbeiteten die Studenten ein dreidimensionales Konzept des Marsrover. Zuvor wurden jedoch drei verschiedene Konzepte von Hand skizziert und bewertet. Aus diesen drei Konzepten wurde das beste gewählt (siehe Kapitel 2.5.2 Entscheidung für das finale Konzept) und mithilfe von SolidWorks detailliert ausgearbeitet.

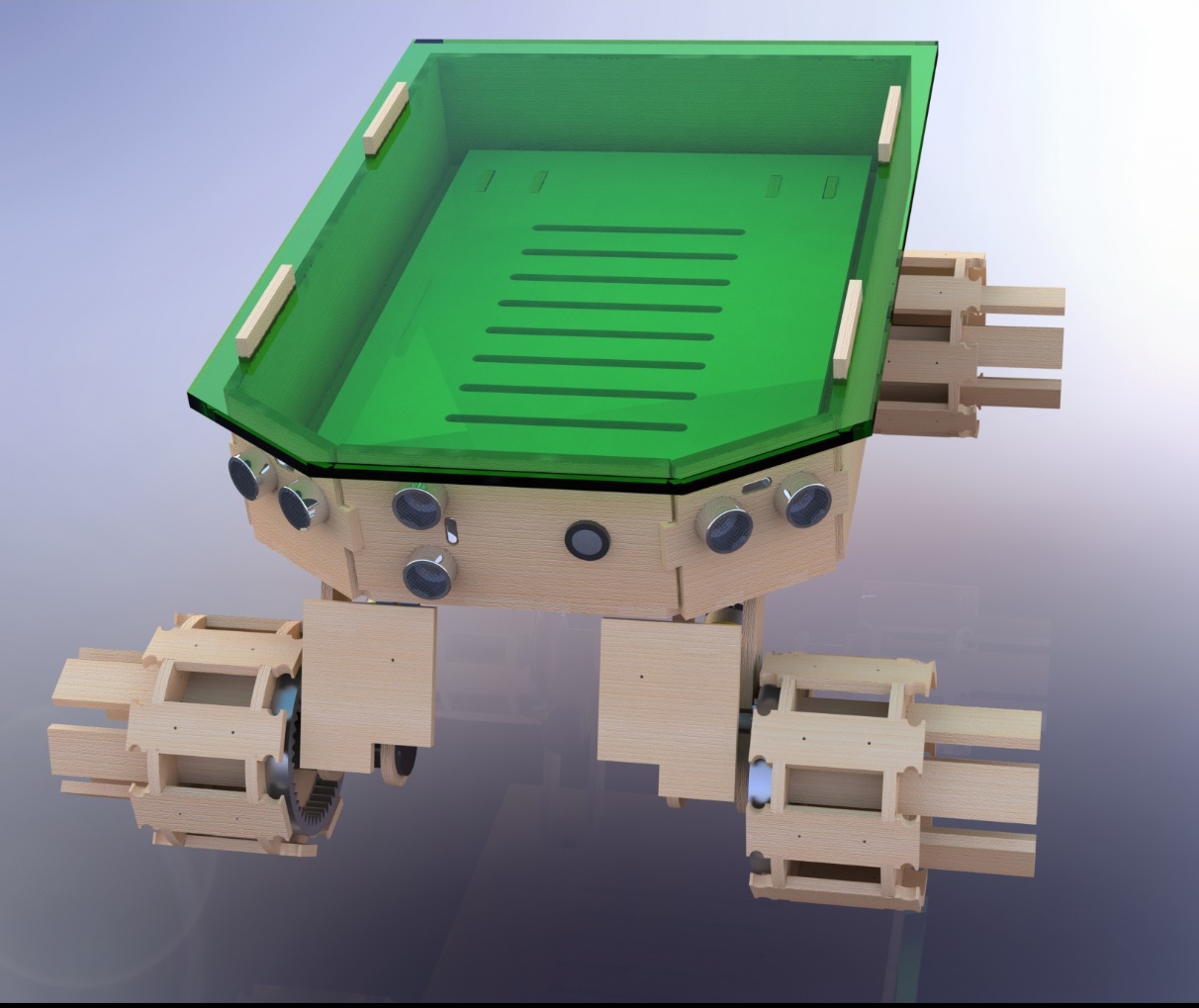


Abbildung 1: Vorderansicht Marsrover

Es wurde eine Baugruppe mit 25 Bauteilen erstellt. Eine Auflistung der Bauteile befindet sich im Anhang A1 Stückliste. Die Bauteile, welche mit dem Lasercutter herstellbar sind, wurden als DXF-Datei abgespeichert um sie mit der Software Inkscape (1991 Free Software Foundation, Inc.; Version 0.92.4) wieder zu öffnen. Mit Hilfe von Inkscape wurden die einzelnen Bauteile zusammengeführt. Ein Beispiel für die zweidimensionale Aufbereitung in Inkscape befindet sich in Anhang

Das PCB Board wurde mit der EAGLE Software 7.1 von Autodesk designt. Das Platinen Design wurde anhand der Richtlinien von der Software, Mechanik und Sensorik ausgerichtet. Die hauptsächliche Aufgabe war hierbei, es zu ermöglichen, dass Rasperry Pis mit anderen Sensoren und Motoren kommunizieren und diese mit einer ausreichenden Spannung versorgen können.

## Projektmanagement

### Projektteam „Mars Raiders – Not fast, but furious“

Projektstart ist am 18. Januar 2019 in der ersten Systems Engineering Vorlesung. Die Auftraggeber des Projekts sind hierbei die Aufgabensteller im Rahmen des Studiengangs aus den drei kombinierten Fächern Systems Engineering, Embedded Systems und Projektmanagement.

Die gesamte Klasse bespricht zunächst die Einteilung der Gruppen selbstständig. Nach kurzem Abwägen und Diskussion der Argumente wird demokratisch entschieden, die Klasse in zwei Gruppen mit je 7 – 8 Mitgliedern zu teilen. Es wird weiterhin beschlossen, dass die Gruppen mit gleich starken Kompetenzen und Talenten ausgestattet werden, um so den bestmöglichen Output für beide Teams zu gewährleisten und das Erreichen des Projektziels sicherzustellen. Jeder stellt seine Stärken vor und so werden die einzelnen Mitglieder in die vier Hauptbereiche der Aufgaben verteilt – Mechanik, Elektronik, Software und Management. Das Team „Mars Raiders – Not fast, but furious“ ist geboren. Anschließend werden die Teammitglieder nach ihren Kompetenzen in Arbeitsgruppen eingeteilt. Diese sehen wie folgt aus:

Mechanik: Stefan Sulzenbacher, Nicolas Roth

Elektronik: Alexander Mairhofer, Kaj Nocker

Software: Marco Gökmen, Stefan Sulzenbacher (Doppelrolle)

Management: Bettina Oberloher (PM), Martin Schafferer, Philip Reitstätter

Der Projektmanager (PM) läutet direkt nach der Teamauswahl die erste Teambesprechung ein und es wird gemeinsam definiert, wie das Projekt angegangen und das Projektziel gesteckt wird, um die wie angekündigt, sehr gering bemessene Zeit effizient zu nutzen. Danach folgt direkt das erste Brainstorming zu möglichen Konzepten für die Umsetzung des Mars Rovers. Es werden drei Konzepte erdacht (wie oben beschrieben) und für den darauffolgenden Tag in Wattens zur weiteren Auswahl vorbereitet.

### Projektdefinition und -planung

Ziel des Projekts ist es, wie oben ausführlich beschrieben, einen Mars Rover zu bauen und vier vorgegebene Tracks erfolgreich zu durchlaufen. Es gilt also mit 8 Teammitgliedern einen voll funktionsfähigen, autonom fahrenden Roboter zu erfinden und zu bauen.

Um in der kurzen vorgegebenen Zeit das Ziel effizient zu erreichen und erfolgreich zu beenden, wird entschieden, das traditionelle Projektmanagement (TPM) zu verwenden. Es gibt ein klares Ziel sowie eine Idee der möglichen Umsetzung, deshalb kann ein vorläufiger Projektplan erstellt werden. Dieser unterteilt sich in drei Projektphasen: Konzeptfindung (ThinkTank), mechanische Umsetzung und elektronische Planung (HandsOn) sowie Softwarefinalisierung (HeartRate).

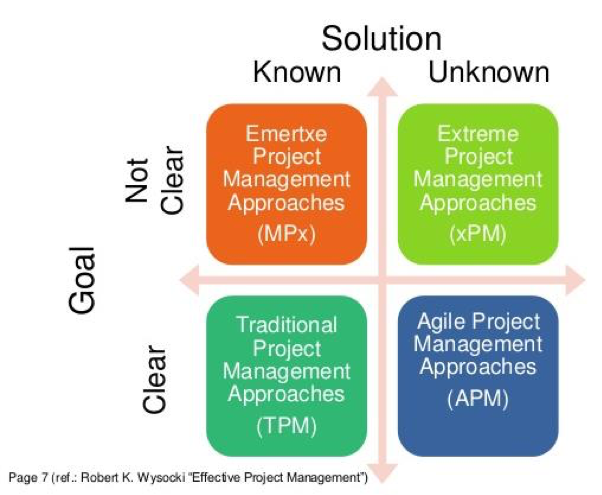


Abbildung 2: Auswahl Projektmanagement Methode

Zum Start an jedem gemeinsamen Arbeitstag wird als erstes im Team besprochen, welches Tagesziel erreicht werden muss. Dementsprechend werden Arbeitspakete geschnürt und Ressourcen verteilt. Am Nachmittag bzw. nach Ablauf der Hälfte der zur Verfügung stehenden Zeit sammelt der Projektmanager den aktuellen Status der jeweiligen Aufgaben aller Arbeitsbereiche bei den Teammitgliedern ein und informiert über die verbleibende Zeit. Um das Tagesziel zu erreichen, wird dann entschieden ob Ressourcen neu verteilt werden. Am Ende des Tages trifft sich die Gruppe, um den finalen Stand des Projekts zu bestimmen und das Tagesziel für den nächsten Tag festzulegen.

Der Projektmanager fungiert vorrangig als Zeitmanager und behält den Überblick über den Fortschritt der einzelnen Arbeitsbereiche und den Einsatz der Teammitglieder. Vor allem die Tage in Wattens müssen so effizient wie möglich gestaltet werden, damit die Geräte und Materialien vor Ort optimal genutzt werden können. Das Managementteam unterstützt den Projektmanager und sorgt für einen reibungslosen Ablauf während der Projektphasen. Wird eine Projektphase zu spät gestartet, kann das Erreichen des Projektziels in Gefahr geraten.

**Etappenziel 1:** Wattens, 19.01.2019 – Festlegen des Mars Rover Konzepts und Erstellen des CAD-Entwurfs, Festlegen der Anforderungen an die Platine.

**Etappenziel 2:** Wattens, 26.01.2019 – Fertiger Prototyp und Schaltplan.

**Etappenziel 3:** Wattens, 01.02.2019 – Erfolgreiches Testing und Zusammenführung der Softwarepakete.

Auszug Projektplan (Projektplan Mars Rover Challenge.pdf):

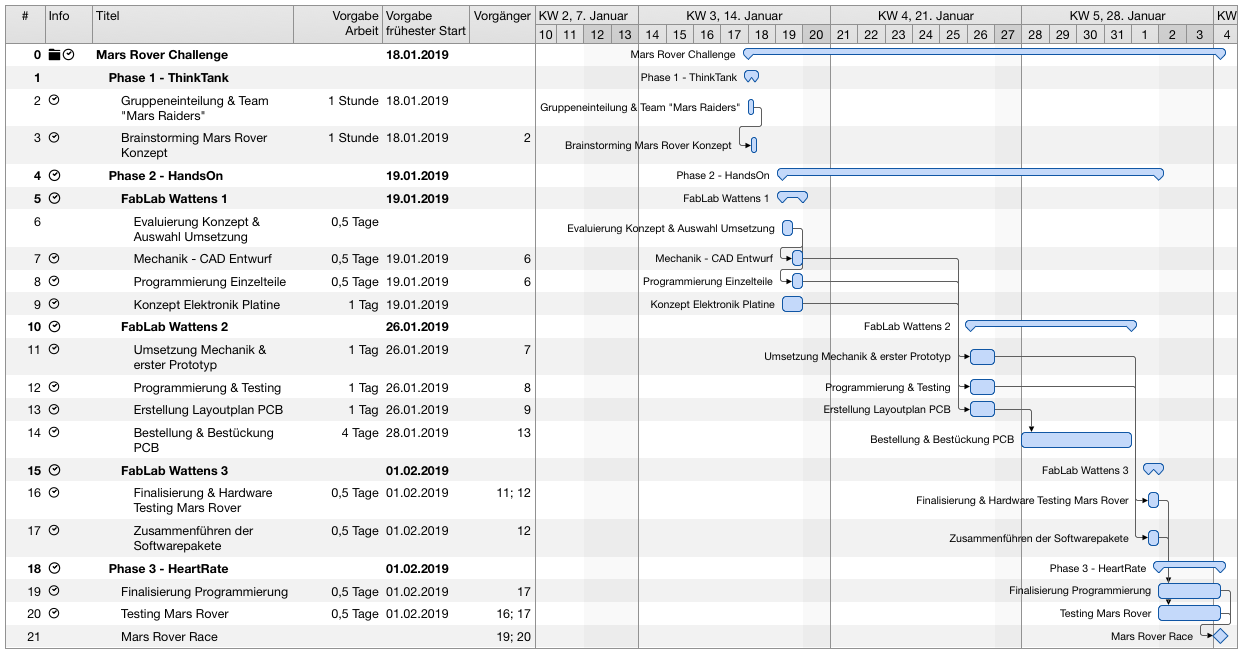


Abbildung 3: Projektplan

### Krisenmanagement

Am letzten Tag, kurz vor Fertigstellung der Programmierung und 45 Minuten vor Start des Rennens ist das Team „Mars Raider“ vor einem unvorhersehbaren Ereignis nicht gefeit. Das Krisenmanagement kommt zum Einsatz. Beim Test des LineTrackers wird mit einem 5V-Kabel versehentlich ein Kurzschluss verursacht. Dies hat zur Folge, dass die verbaute SD-Karte funktionsunfähig ist und das geschriebene Programm somit nicht mehr ausgeführt werden kann. Ohne aktuellem Update des Backlogs ist eine Teilnahme am Rennen nicht möglich und das Projektziel wird nicht erreicht.

Das Team bleibt gelassen und sucht nach einer Lösung, ohne das Projektziel aus den Augen zu verlieren. Das Krisenmanagement des Teams macht es möglich schnell und pragmatisch auf eine Lösung des Problems zu kommen und die Auftraggeber nicht zu verunsichern. Mit dem Backlog aus einer sehr frühen Phase des Programmstands und viel Einsatz des Softwareteams, ist es möglich den Mars Rover an den Start des Rennens gehen zu lassen.

### Resultate und Lessons Learned

Trotz der oben beschriebenen Widrigkeiten ist es dem Team gelungen in letzter Minute den Mars Rover mit Backupdaten zu bespielen und ihn auf die Rennstrecke zu bringen. Trotz großer Enttäuschung nach dem SD-Karten-Crash sind alle Teammitgliedern souverän und lösungsorientiert. Die Gruppe hält zusammen, jeder bleibt ruhig und arbeitet bis zur letzten Sekunde gemeinsam an der Erreichung des gesteckten Ziels – mit dem gebauten Mars Rover die vier Tracks erfolgreich zu meistern.

Das Projekt zieht sich durch den zweiten Teil des Wintersemesters und wird am 9. Februar um 16 Uhr mit dem Rennen beendet.

Das Team „Mars Raiders – Not fast, but furious“ besteht 2 1/3 Tracks und überrascht mit solider Leistung beim zusätzlichen Überraschungstrack „Hold Still“ als letzte Disziplin auf der Rennstrecke.

Im Projektmanagement lässt sich zwar ein guter Projektplan erstellen, aber nicht jedes Detail lässt sich planen. Es geht vor allem darum, auf Unvorhergesehenes flexibel und kreativ einzugehen und als Team zielgerichtete Lösungsansätze der Situation entsprechend umzusetzen. So wird das TPM am letzten Tag kurzerhand in eine Agile Projektmanagement-Methode umgewandelt und die Teammitglieder arbeiten bereichsübergreifend sowie nach freien Kapazitäten. Jeder hilft dort, wo eine freie Hand gebraucht wird, um das Projektziel fristgerecht zu erreichen.

Der Projektplan ist auf dem GitHub Repository zu finden:

<https://github.com/mrc-gkmn/mars_project>

# Resultate und Diskussion



## Mechanisches Konzept

### Konzeptfindung

Zur Wahl des gangbarsten Konzepts wurde von allen Teammitgliedern ein Brainstorming durchgeführt. Die Ergebnisse wurden dem Team kurz erklärt und anschließend gemeinsam durchdacht sowie kritisch betrachtet. Ergebnis dieser Herangehensweise sind drei Versionen mit den Namen „Knicklenker“, „Dreirad“ und „Panzer“. Diese Titel beziehen sich auf die Methodik, mit der später Lenkbefehle umgesetzt werden sollen. Des Weiteren wurde auf die Kompatibilität zu den gegebenen Tracks geachtet.

### Entscheidung für das finale Konzept

In Anbetracht des gegebenen Terminplans ist die Realisierung mehrerer Konzepte parallel nicht sinnvoll zu realisieren, daher wurde anhand der in Tabelle 2 dargestellten Entscheidungsmatrix jenes Konzept gewählt, welches die besten Erfolgschancen bietet. Auf der Y-Achse aufgetragen befinden sich unterschiedliche Themengebiete, zu denen die Konzepte bewertet wurden. Die Bewertung selbst wurde anhand einer Platzierung (1-3) vorgenommen, wobei 1 die beste- und 3 die unvorteilhafteste Variante darstellt. Somit ist jenes Konzept, mit der geringsten Gesamtpunkteanzahl am vorteilhaftesten zu betrachten. Es ist möglich, dass mehrere Konzepte bei einer Eigenschaft mit der gleichen Zahl bewertet werden, wenn diese denselben Nutzen bieten.

Tabelle 1: Entscheidungsmatrix der Konzepte

Die Konzepte „Dreirad“ und „Panzer erreichen eine identische Gesamtwertung, jedoch wird in diese nicht die bisherige Erfahrung der Teammitglieder einbezogen. Da alle Teammitglieder bereits Erfahrung mit dem Konzept „Panzer“ haben, wird dieses im Entscheidungsprozess bevorzugt. So können besondere Eigenschaften des Konzepts bereits während der Planungsphase berücksichtigt werden, ohne diese vorab empirisch zu ermitteln.

Das finale „Panzer“ Konzept besteht auf vier Rädern, welche jeweils unterschiedlich angetrieben werden. Die Vorteile dieses Konzeptes sind hohe Wendigkeit und einfach Ansteuerung. Um die Wendigkeit des Fahrzeuges zu erhöhen wurde in der Planung darauf geachtet, ein möglichst kleines Fahrzeug zu entwickeln. Wobei die Breite durch den Track 2: „Bridge Over Troubled Water“ vorgegeben wurde, um nicht zwischen den Kanthölzern der Brücke nach unten zu fallen.

### Konzeptausführung Räder

Um die Kraftübertragung der Räder auf die Rampe des Track 1: „Climbing Up the Walls“ zu erhöhen, wurden nicht die bereitgestellten Gummiräder verwendet, sondern ein neues Konzept entwickelt.

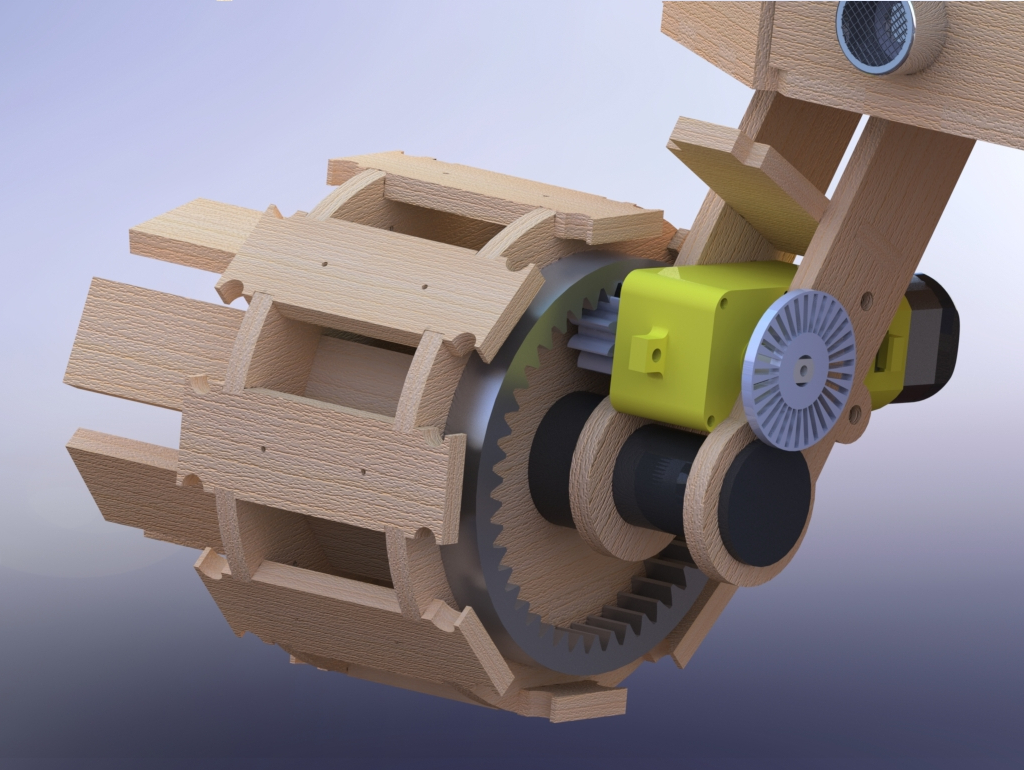


Abbildung 4: Rad mit Antrieb

Die Laufflächen wurden aus Sperrholz hergestellt. Kleine Löcher in den Laufflächen ermöglichten das einfache eindrehen von Holzschauben, die als Spikes dienen. Diese Spikes krallen sich in das Sperrholz der Rampe und maximieren die Kraftübertragung.

Um die Bewegung zu realisieren hat die Fachhochschule Kufstein den Studenten 9 Volt DC-Elektromotoren mit Getriebe bereitgestellt. Jedes Team konnte die Anzahl an verwendeten Motoren frei wählen. Um die benötigte Anzahl bestimmen zu können, wurde eine Leistungsberechnung durchgeführt. Das Ziel war, den Antrieb mit einem Motor pro Rad (somit 4 Motoren gesamt) zu realisieren. Eine Möglichkeit, ein große Getriebeübersetzung zu realisieren ist ein Planetengetriebe, welches jedoch mit den bereitgestellten Fertigungsmethoden nicht realisierbar ist. Die Alternative bietet ein Zahnradgetriebe, welche auch mittels Laserschneider gut realisierbar ist. Um die Räder möglichst klein zu gestalten, wurde das treibende Zahnrad außenverzahnt und das getriebene Zahnrad innenverzahnt. Das treibende Zahnrad sollte klein ausgeführt sein, um eine große Getriebeübersetzung zu ermöglichen jedoch soll die Festigkeit auch gegeben sein. Aus diesem Grund wurde das treibende Zahnrad mit 10 Zähnen und dem Modul 1.25 nach DIN 780 realisiert, was aus Sicht des Konstrukteurs die geforderten Bestimmungen erfüllte. Durch diese Bestimmung ergibt sich ein Teilkreisdurchmesser von 12,5 mm.

Aus dem Datenblatt von JOY-IT[[1]](#footnote-1) kann entnommen werden, dass bei einer Spannungsversorgung von 9 Volt Gleichstrom ein Drehmoment von 1.200 gf\*cm. Diese unübliche Einheitsangabe entspricht 0,117 Nm. Zur weiteren Berechnung wird die Einheit Nm verwendet.



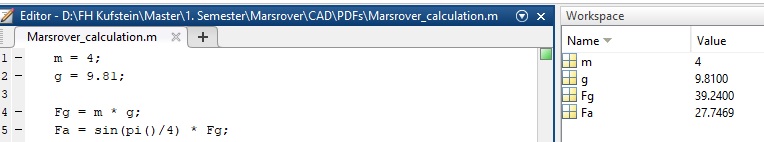
Zur Berechnung der Benötigten Antriebskraft wurde das Fahrzeug mit einer Masse von 4 kg angenommen. Durch aufstellen des Kraftdreieckes ergibt sich eine maximale Antriebskraft von 27,7 N. Dies entspricht einer Antriebskraft pro Rad von ca. 7 N.

Abbildung 5: Antriebskraft Berechnung

Der Raddurchmesser wurde mit 80 mm festgelegt. Daraus ergibt sich für das innenverzahnte Zahnrad ein maximaler Außendurchmesser von 70 mm. Das getriebene Zahnrad wird mit 44 Zähnen ausgeführt und ebenfalls mit dem Modul 1.25 nach DIN 780, wodurch sich ein Teilkreisdurchmesser von 55 mm ergibt. Der Raddurchmesser wird durch die Spikes auf 90 mm erweitert wodurch sich eine rechnerische Kraftübertragung pro Rad von 10,3 N ergibt. Da 7 N gefordert werden ist das Getriebe ausreichend. Rechenschritte sind in Anhang A3 Berechnung Getriebe ausgeführt.

Um das Fahrzeug möglichst klein zu halten, wurde das Fahrzeug so dimensioniert, dass es bei Track 2: „Bridge Over Troubled Water“ durch die Kanthölzer hindurchfallen würde. Um die Breite zu erhöhen wurden fünf Verlängerungen an jedes Rad angebracht. Da die Verlängerungen näher am Mittelpunkt liegen, wirkt das Rad wie ein Spurkranz, ähnlich den Rädern moderner Eisenbahnfahrzeugen. Die Verlängerungen werden nicht mit Spikes versehen, da das Reibmoment von Holz-Holz ausreicht, um das Fahrzeug waagerecht nach vorne zu bewegen.

Jedes Rad wird auf einer Achse aus Kunststoff gelagert. Die Lagerung ist eine Gleitlagerung und weißt somit höhere Reibwerte auf als eine Lagerung mittels Kugellager, jedoch wird angenommen, dass die rechnerische Antriebskraftdifferenz die Reibung überwindet.

Die Achse ist an einer Gabelstange befestigt. Diese Gabelstange erfüllt mehrere Funktionen. Zum einen hindert sie die Radachse an der Rotation um die senkrechte Achse. Als nächstes fixiert die Gabelstange den Antriebsmotor in seiner Position und als letztes wird ein Spritzwasserschutz für die Motoren bereitgestellt. Am Spritzwasserschutz ist zudem die Lichtschranke befestigt, die zur Drehzahlermittlung verwendet wird. Die dazugehörige Lochscheibe befindet sich direkt and der Welle des Motors.

Die Gabelstange wird mithilfe eines Keiles aus Sperrholz an seinem Platz gehalten wie in Abbildung 5: Befestigung Gabelstange dargestellt. Unterhalb des Keiles befindet sich eine Aussparung in der Grundplatte, um die außen gelegenen Sensoren und Motoren mit der Platine zu verbinden.

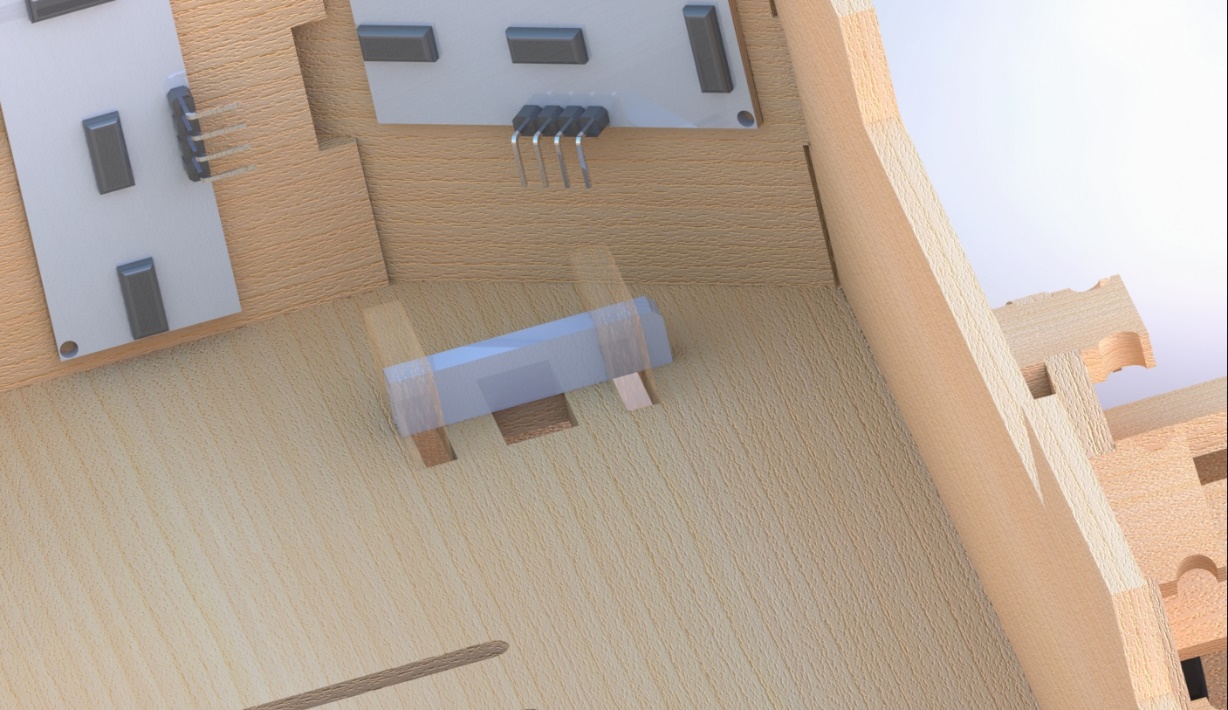


Abbildung 6: Befestigung Gabelstange

### Konzeptausführung Gehäuse

Das Gehäuse wurde ebenfalls So entwickelt, dass keinerlei Klebstoff oder Schraubenverbindungen notwendig sind, lediglich zur Abdichtung der Schräge wurde Schmelzklebestoff verwendet.

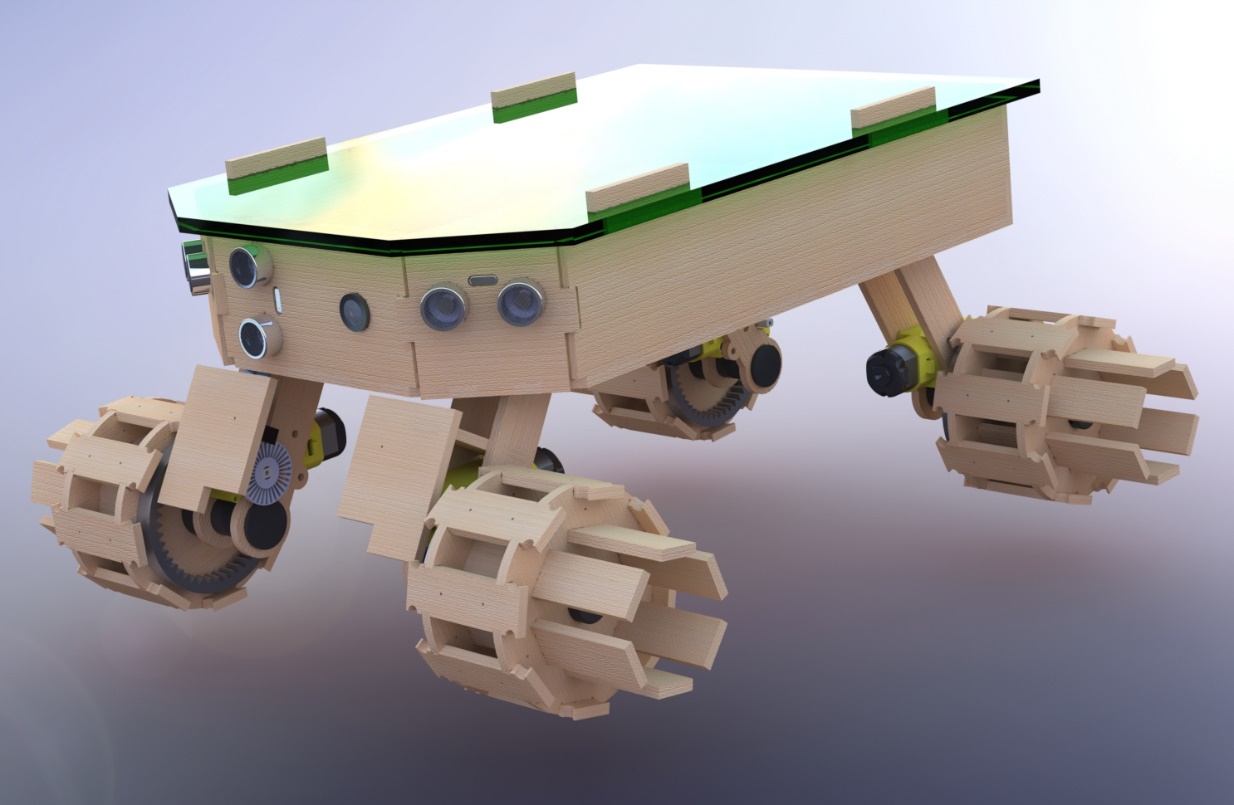


Abbildung 7: Seitenansicht Marsrover

Die schrägen Seitenwände waren notwendig, um die Ultraschallsensoren zu positionieren, sodass seitliche Gegenstände erkannt werden können. Die drei Ultraschallsensoren werden durch Reibschluss in Position gehalten. Ebenso verhält es sich mit der Kamera, die nach vorne ausgerichtet ist.

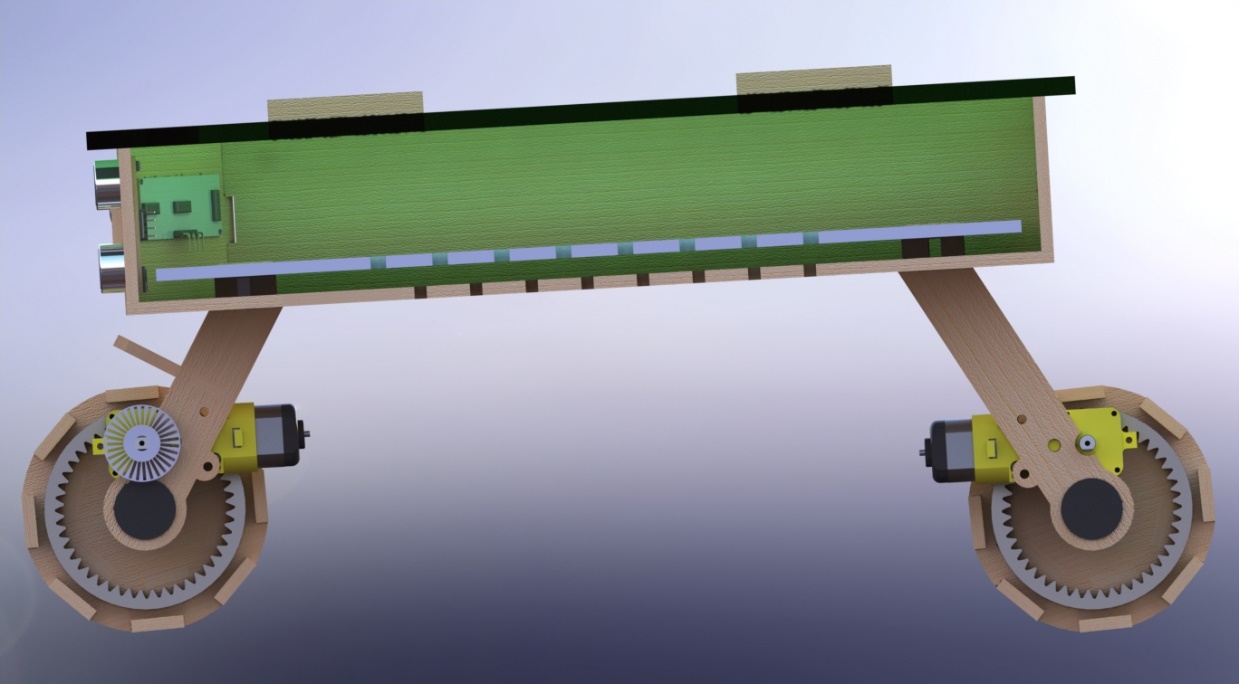


Abbildung 8: Querschnitt Marsrover

Der Marsrover Besitz einen zweiten Boden, auf dem die Steuereinheit mit Platine befestigt wird. Der Zwischenboden kann dann einfach auf die nach obenstehenden Radgabeln gesteckt werden. Der Zwischenboden bietet auch einen Spritzwasserschutz von unten. Sowohl der Zwischenboden als auch die Grundplatte besitzen Luftschlitze, um eventuelle Wärmeentwicklung der Elektronikbauteile entgegenzuwirken. Im späteren Praxistest zeigte sich, dass die Wärmeentwicklung sehr gering ist und keine Luftschlitze notwendig sind.

Der Deckel wurde aus optischen Gründen aus Plexiglas gefertigt und wird in die Laschen der Seitenwand gesteckt. Da der Deckel durchsichtig ist können die Status-LEDs auf der Platine gesehen werden.

## Sensor- und Aktorkonzept

Zur Erfassung der Tracks wird auf mehrere Sensortypen zurückgegriffen, die teils einzeln teils in Kombination der Fahrwegregelung – also dem Lenken des Fahrzeugs dienen:

### Ultraschall Sensoren

An der Front des Fahrzeuges werden 3 Ultraschallsensoren des Typs HC-SR04 eingesetzt. Die Auflösung beträgt 0,3 cm, was für diese Anwendung zufriedenstellend ist. Mit ihnen kann der Abstand zu Gegenständen gemessen werden, die Ultraschallwellen reflektieren. Solche Gegenstände weisen bevorzugt glatte Flächen auf. Der Sensor generiert ein Ultraschallsignal, welches von Objekten reflektiert wird. Um eine ausreichende Reflexion zu erzielen soll der Einfallswinkel der Schallwellen ca. 15 Grad Abweichung zur Senkrechten auf die reflektierenden Flächen nicht überschreiten.

Mit den Sensoren sollen Objekte detektiert werden, die sich vor oder in unmittelbarer Nähe der Fahrzeugfront befinden. Die Sensoren befinden sich in der Mitte sowie an den Ecken der Stirnseite des Fahrzeugs. Der mittlere Sensor ist in Fahrtrichtung ausgerichtet, die zwei seitlichen Sensoren jeweils um 30 Grad von der Fahrtrichtung abgewandt. Die Montage erfolgt durch einstecken in entsprechende Ausschnitte am Fahrzeuggehäuse. Zur winkeligen Anordnung der seitlichen Sensoren wird die Fahrzeugfront in drei Abschnitte unterteilt, welche jeweils mit 30 Grad angestellt sind.

### Linetracking Sensoren

Zwischen den vorderen Rädern werden zwei Linetracking Sensoren angebracht. Sie messen die Reflexion eines selbst generierten Infrarotsignals. Dabei geben sie ein digitales Signal zurück welches proportional der Intensität des reflektierten Signals ist. Ändert sich die Farbe oder Beschaffenheit eines Gegenstandes, der sich im Abstand von 5 bis 15 mm vor dem Sensor befindet, so erhält man eine reproduzierbare Veränderung. Mit dieser Funktion lässt sich das Erreichen einer Linie feststellen.

### Kamera

An der Stirnseite, neben dem Ultraschallsensor wird zur Erfassung von Objekten und des Fahrweges eine Kamera des Typs RB-CAMERA-WW angebracht. Hierbei handelt es sich um eine Weitwinkel – Farb – Kamera mit 5 Megapixel Auflösung. Sie wird zur Objekterkennung eingesetzt. Anhand von Form und Farbe sollen Hindernisse sowie der Fahrweg detektiert werden.

### Drehzahlsensoren

Werden an den Antriebswellen der beiden Vorderräder angebracht. Die eingesetzten Sensormodule basieren auf Infrarottechnologie in Kombination mit einem Low Offset Voltage Dual Comparator Chip. Das Modul bietet einen digitalen sowie einen analogen Ausgang. Ein Infrarotsignal wird über einen mechanischen Zwischenraum überwacht, in diesem läuft eine Lochscheibe. Die Frequenz des detektieren Signals lässt sich anhand der Lochanzahl der Scheibe auf eine Drehzahl umrechnen. Über diese Sensoren wird das Feedback zur Umsetzung des Antriebssignals gegeben.

### Gyro – und Beschleunigungssensor

Wird direkt auf der Fahrzeugplatine integriert, es handelt sich um einen SMD – Bauteil des typs MPU – 6050 V12. Die genaue Funktionsweise des Sensorchips ist den zugehörigen Datenblättern nicht zu entnehmen, lediglich die gelieferten Daten. Der Gyrosensor wird eingesetzt, um die Orientierung des Fahrzeuges zu überwachen, insbesondere beim Überwinden von Obstakeln.

## Softwarekonzept

### Konzeptfindung

Ausgangsbasis für die Grundfunktionalitäten der Software waren die Festlegung der Fortbewegungsart und die Gestaltung der Tracks. Zusätzlich wurden Aspekte zur Überprüfung und Inbetriebnahme des Fahrzeugs berücksichtigt um damit auch die Softwareentwicklung zu unterstützen. Die unterschiedlichen Anforderungen an die Tracks führten dazu, dass entschieden wurde für jeden der Tracks ein eigene Skript-Logik zu entwickeln.

Die Skripte für Track 1 und 2 waren sehr identisch, da hier eine reine vorwärtsfahrt ausreichte, um die Strecken erfolgreich zu bewältigen. Track 3 sah die Verwendung von Linetrackern und anschließender Motoransteuerung vor. Track 4 war softwaretechnisch die anspruchsvollste Strecke, bei der Ultraschallsensoren, Kameradaten mit Hilfe von Bildverarbeitung durch OpenCV, Drehzahlmesser und Motorsteuerung innerhalb eines eigenentwickelten Skriptes umgesetzt werden sollten.

Um den beschriebenen Anforderungen an die Tracks zunächst gerecht zu werden, wurden zwei Softwareprinzipien erstellt. Einerseits wurde die Plattformsoftware für das Marsroverprojekt konzipiert und entwickelt. Als Plattformsoftware bezeichnet man die Umsetzung von Sensorskripten mit den dazugehörigen Übertragungsprotokollen und Treibern. Einfach gesprochen beschäftigt sich die Plattformsoftware mit dem generieren der Sensordaten und die Übertragung an die Steuerungseinheit. Parallel zur Plattformsoftware wurde die Applikationssoftware entwickelt, welche die grundlegenden Fahrlogik des Marsrover beinhaltet. Die Applikationssoftware arbeitet aufbauend auf der Plattformsoftware und muss sich nicht mit der Signalübertragung beschäftigen. Durch die Unterteilung in diese zwei Disziplinen der Softwareentwicklung konnte die Komplexität reduziert und die Lesbarkeit des Codes erhöht werden.

Zunächst haben wir ein Skript zu den Ultraschallsensoren entworfen und anschließend noch Skripte für die Linetracker, Drehzahlmesser und Kamera entwickelt. Basierend auf der Sensorwerte, die wir aus diesen Skripten ziehen konnten, galt es ein Konzept zu entwickeln mit dessen Hilfe speziell die Anforderungen an Track 4 umsetzbar werden. Dabei betrieben wir umfangreiche Tests, durch welche die Detektion von Objekten durch Farberkennung ermöglicht wurden.

Das Softwarekonzept sah vor die Sensor- und Motoransteuerung mit Hilfe eines Bus-Systems zu realisieren, welches im PCB-Design integriert wurde. Erst dann konnte der finale Aufbau der Skripte festgelegt und schlussendlich getestet und angepasst werden. Bedingt durch die Abhängigkeit vom PCB-Design gestaltete sich die Konzeption und anschließende Umsetzung jedoch sehr komplex und in Kürze der Zeit nicht vollständig realisierbar.

Im Folgenden werden die Konzepte und deren Umsetzung für den Kommunikations- und Skriptanteil genauer beleuchtet.

### Plattformsoftware

Wie bereits erwähnt, wurde die Gesamtsoftware in Plattformsoftware und Applikationssoftware unterteil. In diesem Abschnitt wird die Plattformsoftware behandelt. Plattformsoftware befasst sich mit dem grundlegenden Aufbau der Signalerfassung, Betriebssystem der Steuerung und Speicherverwaltung. Der Raspberry Pi verwendet das Betriebssystem Raspbian welches auf LINUX basiert. Raspbian ermöglicht das Betreiben der Anwendersoftware, wie zum Beispiel die Python-Konsole womit der Code für den Marsrover erstellt wurde. Raspbian ist auch für die Integration und starten von Treibern notwendig wie Beispielsweise der Treiber für das Serielle Übertragungsprotokoll I²C. Der Treiber des I²C-Protokolls ist standartgemäß deaktiviert und muss explizit in der Konsole gestartet werden, da der verwendete Portexpander des Marsrover mit Hilfe von I²C angesteuert wird.

Sobald der Treiber aktiviert ist, wird die Verwendung von I²C prinzipiell ermöglicht. Das Übertragen von Nachrichten geschieht durch Übermittlung von „High“ und „Low“ Signalen zu bestimmten Zeitpunkten. Um die Nachrichtenübertragung zu erleichtern, wurde die „Wiring Pi“-Bibliothek eingebunden. Dank dieser Bibliothek kann mit einfachen Befehlen wie „wiringPiI2CWrite()“ ein komplexer Datensatz an ein angeschlossenes Modul übertragen werden. Als Parameter wird die Adresse des Zielmodules sowie der Datensatz angegeben.

Der verwendete Portexpander PCA9555 muss zu Beginn des Codes initialisiert und anschließend als Ausgang definiert werden. Da es sich um zwei Register handelt, müssen Acht Datenbits übertragen werden. Dies geschieht mit dem Befehl: wiringPiI2CWrite(Adresse, 0x02, 0b00000000). Das Initiieren des Bauteiles als Ausgang wird mit dem Befehl 0x02 für das erste Register und mit 0x03 für das zweite Register durchgeführt. Die binären-Nullen stehen für jeweils einen Ausgang wobei die letzte 0 das least-significant-bit darstellt und somit den ersten Port anspricht.

Um den Wert des Ausgangs zu ändern geschieht erneut durch den Befehl: wiringPiI2CWrite(Adresse, 0x07, 0b00000011). Die hexadezimale Zahl 0x07 kennzeichnet, dass das erste Register angesprochen wird. Für das zweite Register würde die hexadezimale Zahl 0x08 verwendet. Die Binärzahl wie oben angeführt setzt die ersten beiden Ausgänge auf „HIGH“ und die weiteren sechs Ausgänge auf „LOW“.

Die Ansteuerung des Motors geschieht mit Hilfe einer H-Brücke der Bauart DRV8814PW. Dieser Bauteil ermöglicht das unabhängige Ansteuern zweier Motoren in zwei verschiedenen Richtungen. Über die Ansteuerung folgender Pins der H-Brücke werden die nebenstehenden Funktionen ausgeführt:



Tabelle 2: PCA9555 Ansteuerung

Die Ansteuerung der H-Brücke erfolgt mit den oben beschriebenen Portexpandern. Lediglich das PWM Signal wird direkt vom Raspberry Pi auf die H-Brücke geschaltet.

Für die Ansteuerung der Motoren wurde ein Skript erstellt, um mit diesem in der Applikationssoftware des Marsrover arbeiten zu können. Der entwickelte Befehl zur Motorsteuerung hatte folgende Struktur: motorA(direction, Speed, duration). Mit diesem Befehl konnten die Fahrmotoren einfach angesteuert, ohne sich jedes Mal mit der Pin-Belegung abzumühen.

### Applikationssoftware

In einem ersten Schritt musste sichergestellt werden, dass beim Start des Raspberry Pi das definierte Python-Skript automatisch aufgerufen wird. Hierfür wurden die Einstellungen des Raspberry konfiguriert. Beim Start des Raspberry Pis wurde das Ansteuerungsskript ausgeführt. Die generelle Idee des Konzepts zur Ansteuerung und zum Ablauf des Skripts wird durch zwei Buttons und einen Switcher gesteuert. Das Skriptkonzept sah vor, dass ein Start- und Stoppbutton definiert werden, welche das Starten und Stoppen des Programmablaufs ermöglichen. Je nach Position des Switchers wird, anschließend bei Drücken des Startbuttons, die Logik des jeweiligen Tracks gestartet. Ist die Position des Switchers beispielsweise auf 2, so wird das Skript mit der Logik für Track 2 gestartet. Diese Logik wird solange geloopt bis das Drücken des Stoppbuttons detektiert wird. Durch das Drücken des Stoppbuttons wird der Loop unterbrochen und der Switcher kann wieder verändert werden. Um die Logik erneuert zu starten wird erneut auf das Drücken des Start-Buttons gewartet. Die folgende Abbildung zeigt den Ablauf des Skriptkonzepts.

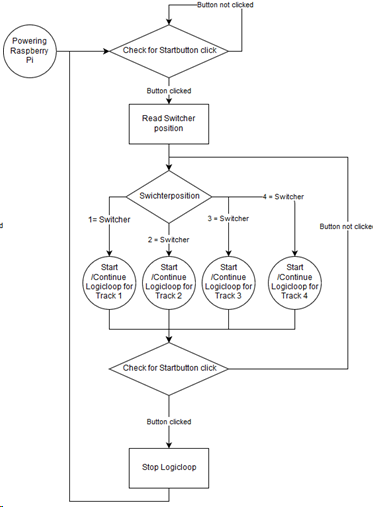


Abbildung 10: Ablauf des Skriptkonzepts

Die Herangehensweise beim Start der Logik für Track 1 ist sehr ähnlich wie die für Logik für Track 2. Beide Strecken konnten durch reine vorwärtsfahrt bewältigt werden. Es wurde lediglich die Geschwindigkeit je nach Track angepasst, für Track 1 wurde letzten Endes der maximale Pulsweitenwert genommen, um mit maximaler Krafteinsatz die Rampe erklimmen zu können. Für Track 2 wurde die maximale Pulsweite halbiert, um den Übergang auf die Brücke möglichst reibungslos zu gestalten. Je nach Logik wurde dieser Wert an die Motorcontrol übermittelt und zur Pulsweitenmodulation (PWM) genutzt. Die Logik wird in folgender Abbildung zur Übersicht visualisiert.

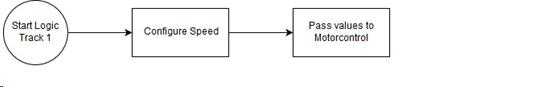


Abbildung 11: Skriptlogik Track 1 und 2

Track 3 konnte nicht mehr durch reine vorwärtsfahrt absolviert werden. Der Einsatz des Linetracking-Sensors war nötig. Mit dessen Hilfe konnte eine Steuerung per Motorcontrol initiiert werden durch welche der Rover entlang der Bodenmarkierungen fahren kann. Wird das Skript für Track 3 gestartet, so werden die beiden rechts (R) und links (L) angebrachten Linetracker überprüft. Wird keine Linie entdeckt wird eine vorwärtsfahrt mit einem Wert von 150 der 255 maximal möglichen PWM-Werte an die Motorcontrol gesendet und schließlich auf beiden Antriebsseiten (links und rechts) durchgeführt. Diese Logik loopt solange bis der linke oder rechte Linetracker eine Linie feststellt und damit die PWM-Werte angepasst werden. Dabei wird je nach erkannter Seite die linke/rechte Seite stark durch die Motrocontrol verlangsamt und die andere Seite entsprechend erhöht (z.B. links 10, rechts 250). Dies hat zur Folge, dass der Rover eine einseitige, starke Kurve beginnt, um der Linie weiter folgen zu können. Wird die Linie dadurch nicht mehr erkannt, so setzt der Rover die vorwärtsfahrt fort. Das Skript konnte leider aus Zeitgründen nicht auf die Detektion von einer Linie durch den Linetracker auf der Rechten und Linken Seite gleichzeitig getestet werden. Die Logik wird in folgender Abbildung zur Übersicht visualisiert.

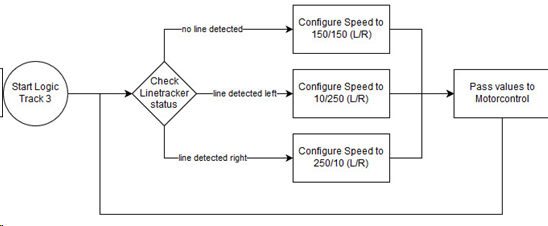


Abbildung 12: Skriptlogik Track 3

rack 4 hatte die mit Abstand komplexesten Anforderungen an das Software- und Skriptkonzept. Da hier Ultraschallsensoren, Kamerabilder und Drehzahlmesser analysiert werden müssen, ist der erforderliche Rechenaufwand so groß, dass der Rover in Schritten fahren muss. Es war kein flüssiges Fahren möglich, da der Rover sonst Teile der Strecke ohne vorherige Analyse durch oben genannte Sensoren befahren hätte. Aus diesem Grund werden bei jedem Schleifendurchlauf der Logik für Track 4 zuerst alle Sensoren ausgewertet.

Es wird zunächst der Abstand der 3 installierten Ultraschallsensoren ermittelt. Hierbei werden Objekte mit einem Frontalabstand des mittigen Sensors ab 50cm und seitliche Objektedurch den linken und rechten Sensor ab 20cm als kritisch erkannt. Der entwickelte Algorithmus sah vor die PWM-Werte, je nach gemessener Distanz zum Objekt und ab den oben beschriebenen Grenzen, zu beeinflussen. Nährt sich ein Objekt dem Ultraschallsensor von links, so werden ab einer Distanz von 20cm die PMW-Werte beeinflusst. Je näher das Objekt sich dem Ultraschallsensor nähert, desto größer ist der Einfluss auf die PWM-Werte. Wichtig zu berücksichtigen ist hierbei, dass der PWM-Wert dabei noch nicht an die Motorcontrol übertragen wird, da die Analyse der Bildanalyse und Drehzahlmesser noch aussteht.

Bei der Bildanalyse wird durch die installierte Kamera am Raspberry Pi ein Bild gemacht, welches durch ein OpenCV-basiertes Skript analysiert wird. Bei dieser Analyse wird das Bild nach einer definierter rot-Farbrange gefiltert und jede Detektion mit Hilfe eines Rechtecks markiert. Entscheidend war dabei, Fehlmarkierungen durch das Berechnen der Rechteckgrößen zu vermeiden. So wurden zu kleine Erkennungen verworfen und die erkannten, rotegefärbten Objekte weiter analysiert. Es folgt eine Berechnung der Rechteck Position im Bild (x- und y-Wert). So konnte unter Zuhilfenahme der x- und y-Werte, die Position des Objektes im Bild ermittelt werden und daraus Handlungsmaßnahmen durch Manipulation des PWM-Wertes resultieren. Wird ein Objekt beispielsweise mittig in der linken Hälfte, erkannt so soll eine mittelstarke Kurve in die entgegensetzte Seite erfolgen. Ähnlich wie bei den Ultraschallsensoren, sollen die PWM-Werte dabei je nach Objektweite bzw. Objektposition gezielt beeinflusst werden. Durch diese Funktion sollen auch Fehlerkennungen des Weitwinkelobjektives gefiltert werden.

Der letzte Schritt der Sensoranalyse ist die Analyse der Drehzahlsensoren. Hierbei werden die gezählten Umdrehungen der Drehzahlmesser am rechten vorderen und linken vorderen Reifen gemessen. Ziel aus dieser Analyse ist es die Richtung von Start zu Ziel beim Durchfahren von Track 4 beizubehalten. Werden durch vorher genannte Sensoren Kurven bedingt, so sorgt diese Analyse zum langfristigen Ausgleich dieser Kurven. Dies ist technisch durch das Bilden der Differenz aus beiden Umdrehungszähler möglich. So erhält man im Idealfall einen Werte nahe 0. Ist der Wert stark positiv oder negativ, so kann man davon ausgehen, dass Kurven stattgefunden haben, die es gilt auszugleichen. Somit kann je nach Betrag dieser Drehzahldifferenz der PWM-Wert unterschiedliche stark beeinflusst werden.

Nachdem die Sensoranalysen erfolgreich durchlaufen wurden, erfolgt zunächst eine Parametrisierung der Geschwindigkeit mit Standardparameter, welche bei Normalfahrt ohne Detektion eingehalten werden. Anschließend werden die Ergebnisse aus der Sensoranalyse durch den Wert der PWM-Modulation zur Standardgeschwindigkeit addiert bzw. subtrahiert. Das Resultat ist damit ein PWM-Wert für je die linke und rechte Steuerseite des Rovers. Dieses Resultat wird anschließend an die Motorcontrol übertragen. Der Zustand mit den vordefinierten PWM-Werten wird für 1 bis 2 Sekunden gehalten, bevor der Rover wieder gestoppt wird und die Analyse der Sensoren erneut stattfindet. Die Logik zum Track 4 wird in folgender Abbildung zur Übersicht visualisiert.

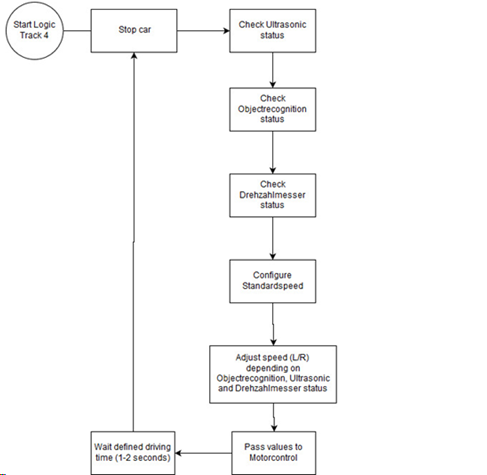


Abbildung 13: Skriptlogik Track 4

Leider konnte diese Logik aus Zeitmangel nicht mehr ausgiebig getestet werden. Lediglich das Konzept und erste Tests bestätigten vielversprechende Resultate.

## Printed Circuit Board PCB - Design

Das Design der Platine ist angepasst an die Vorgaben von Software, Mechanik und Sensorik. Die Idee ist, die Platine so zu gestalten, dass Sensoren, Motoren, ICs (IC - Integrated Circuit) und Raspberry Pis kommunizieren und mit elektrischer Energie versorgt werden können.

Grundbestandteil der Platine sind zwei unabhängige, elektrische Kreise zur Spannungsversorgung, welche über ein gemeinsames Batteriemodul mit Energie versorgt werden. Die Batterien werden über Kabel und Stecker mit der Platine verbunden. Ein Spannungswandler (VDRM – Variable Step Down Regulator Module) auf der Platine versorgt den 5V DC Kreis mit maximal 3A Strom. Das VDRM kann über einen Eingangsspannungsbereich von 6V-42V betrieben werden. Das VDRM soll 2 Raspberry Pis sowie die Sensoren und integrierten Schaltkreise der Platine mit Energie versorgen.

Ein zweiter Kreis versorgt drei Motortreiber vom Typ DRV8814 für 6 Elektromotoren. Dieser Stromkreis ist direkt mit den Batterien verbunden. Die Treiber benötigen eine Eingangsspannung zwischen 8V-45V und liefern maximal 2,5A je Motorkanal / H-Brücke. Jeder Motortreiber ist mit zwei internen H-Brücken ausgestattet. Die Versorgung der Logikschaltkreise der Treiber erfolgt über den 5V Kreis. Die Ansteuerung der Motortreiber erfolgt über zwei I2C Expander, welche direkt auf die Platine gelötet sind. Es ist möglich die Regelung der Motordrehzahl direkt über Hardware-PWM (PWM - Pulsweitenmodulation) Signale zu realisieren. Hardware PWM Signale können vom Raspberry Pi generiert und direkt an die Motortreiber geleitet werden. Eine zweite Möglichkeit ist das PWM Signal über I2C an die Motortreiber zu senden. Letzteres ermöglicht die Nutzung von Software generierten PWM Signalen. Die Verschaltung der PWM Signale erfolgt über ODER Glieder. Somit ist es möglich, ohne Eingreifen PWM Signale entweder direkt vom Raspberry GPIO (GPIO - General Purpose Input/Output) oder über den I2C Expander zu verarbeiten.

Zur Stromversorgung werden 4 Batterien mit der Zellen-Spezifikation Li-ion / 3.7 V / 30 A / 3120 mAh verwendet. Die Batteriezellen werden dabei in Serie geschalten. Das ergibt eine Leerlaufspannung bei voller Zellenladung von 14,8V und eine maximale Betriebszeit bei einer Stromentnahme von 13A von etwa 0,24h bzw.14,4min. Die Stromentnahme des Rover bei maximaler Last ist unbekannt und ist experimentell zu bestimmen, da die Stromaufnahme der Motoren, Sensoren, PCB und der Raspberry Pis wesentlich geringer ausfällt. Eine realistischere Abschätzung bei Annahme Stromverbrauch je Motor (4 Motoren) von 1A, 1x Raspberry Pi 0,5A Sensoren & PCB 0,5A ergibt einen Gesamtstrom von 5A und eine Laufzeit von 37,44min.

Zwei Raspberry Pis können über Bottom Entry Pin Header an die Platine angesteckt werden. Wobei ein Raspberry Pi als Master Gerät die Verwaltung der Platine übernimmt und die Kommunikation zwischen allen Sensoren, Aktoren und ICs sicherstellt. Die Pin-Header der beiden Pis werden zusätzlich auf die Platine herausgeführt, um Signale ggf. zur Fehleranalyse abgreifen zu können oder Sensoren und Aktoren direkt mit GPIOs zu verbinden. Das zweite Pi kann als Slave Gerät Rechenaufgaben vom Pi Master übernehmen. Die Kommunikation zwischen Pi Master und Pi Slave erfolgt über UART (UART - Universal Asynchronous Receiver Transmitter).

16 Sensoranschlüsse werden über Multiplexer angebunden, 2 Sensoren, zwei Taster und ein Programmwahlschalter direkt an GPIOs des Pi Master. Die Spannungssignale der Sensoren werden über Spannungsteiler auf 3.3V eingestellt, da die Spannung an Raspberry Pi Input GPIOs 3.3V nicht überschreiten darf.

Über die I2C Expander werden zusätzlich ein Gyro und Leds, welche sich auf der Platine befinden in das System eingebunden.

Der Schaltplan und das Layout der Platine werden mit dem Programm Eagle der Firma Autodesk erstellt. Die Platine ist in zwei Schichten aufgebaut. Die Bauteile werden per Hand bestückt und werden mit dem Reflow Verfahren und teilweise händisch gelötet. Die Fertigung der Platine sowie die Beschaffung der Komponenten erfolgt extern.

Die Maße der Platine werden so gewählt, dass die Platine, neben den Batterien, Sensoren und Kabeln im Gehäuse des Rover Platz findet.

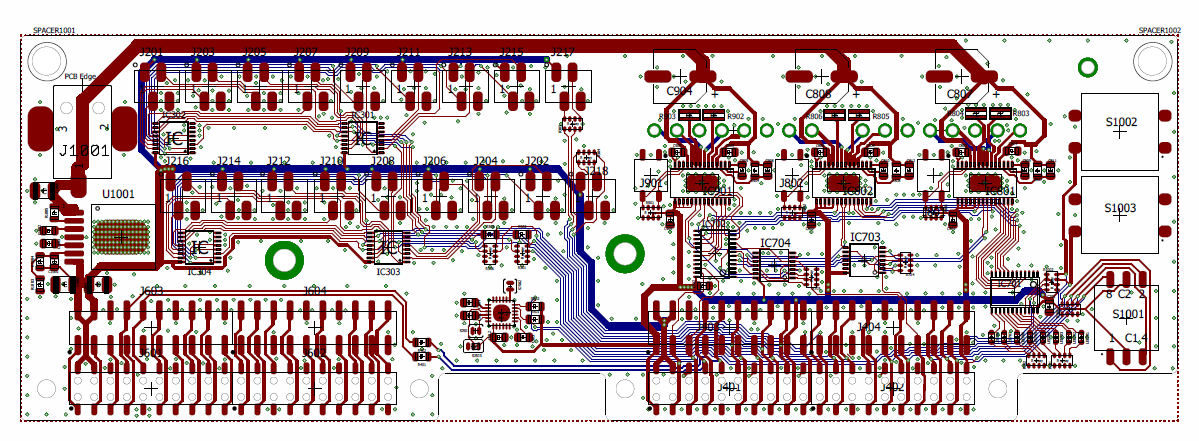


Abbildung 14: Layout der Fertigen Platine1

## Systems Engineering

# Schlussfolgerung

Abschließend ist zu sagen, dass das Team bei diesem Projekt sehr viel gelernt hat. Die Zusammenarbeit in einem Team von 7 Mitgliedern, die sich kaum kennen, ist in jeder Hinsicht eine Herausforderung, doch die einzelnen Persönlichkeiten machten es einfach, auch bei schwierigen Situationen schnell und ruhig einen Konsens zu finden. Leider konnten wir nach vielen investierten Arbeitsstunden und eingesetzter Mühe den Mars Rover nicht so präsentieren, wie ursprünglich geplant. Doch der Zusammenhalt im Team motivierte, um trotzdem an den Start zu gehen und ein mehr als akzeptables Ergebnis zu erzielen.

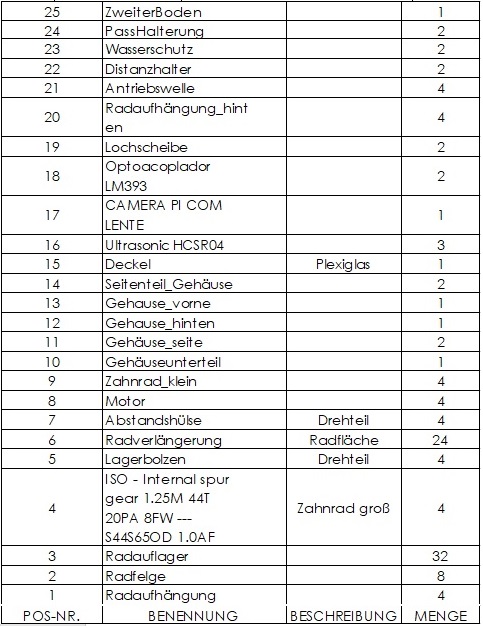
Das Team hatte die Möglichkeit 3D-Drucker und Lasercutter zu nutzen und den effizienten Umgang mit limitiertem Material zu lernen. Wir durften ein autonom fahrendes Fahrzeug von Null auf erdenken und selbst produzieren.

Die kurz bemessene Zeit war eine Anforderung des Projekts, die es zu meistern galt. Für das Team der bedenklichste Faktor zum erfolgreichen Abschluss der Mars Rover Challenge. Rückblickend ist zu sagen, dass die Tage in Wattens die für uns effizientesten waren, da dort auf kurzen Wegen alle Komponenten schnell und betriebsfähig zur Verfügung standen. Außerdem machte es viel Spaß in dem Gründerumfeld mit großem Ideen-Spirit zu arbeiten.

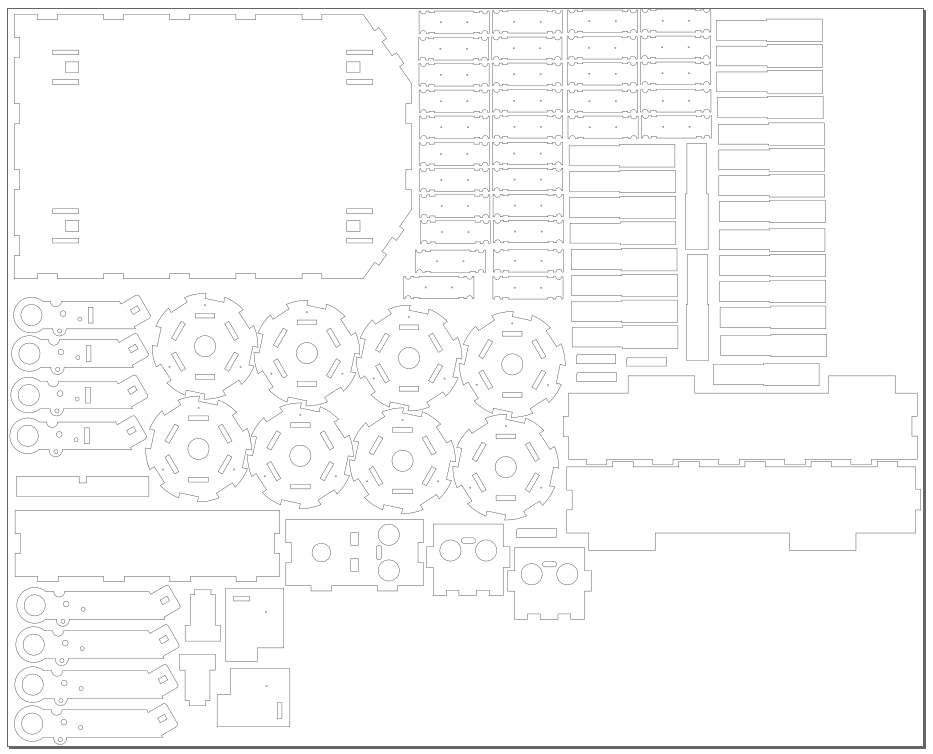
Das Projekt zeigt sehr gut den Begriff der Technischen Interdisziplinarität auf: Diese bedeutet nicht, die Fähigkeit alle Aufgaben selbst perfekt zu beherrschen, sondern viel mehr ein Gefühl und Grundverständnis für die Herausforderungen einzelner Themengebiete mitzubringen und gleichzeitig Probleme und Schwerpunkte des eigenen Spezialgebietes möglichst klar zu kommunizieren.

Anhang

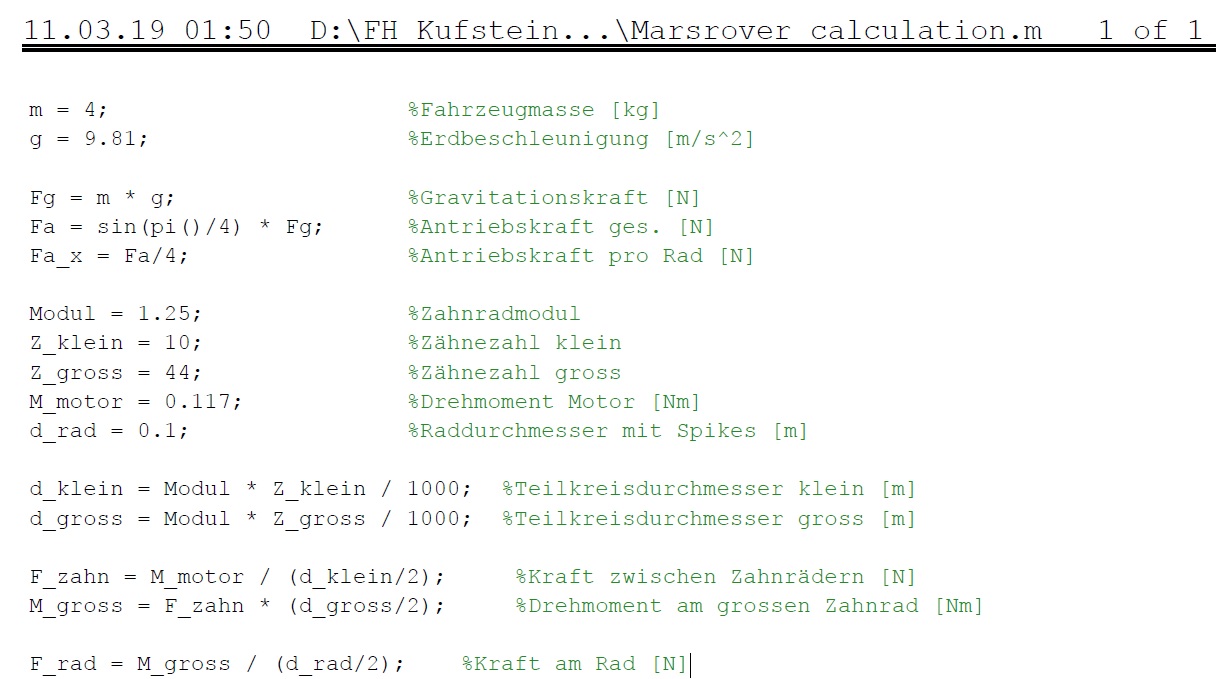
# A1 Stückliste



# A2 Laserzuschnitt Teile



# A3 Berechnung Getriebe



1. Vgl. http://downloads.cdn.re-in.de/1500000-1599999/001573543-da-01-de-JOY\_IT\_GETRIEBEMOTOR\_MIT\_RAD\_COM\_MOTOR01.pdf [↑](#footnote-ref-1)