



# Research Lab for Deep Learning Mars Science Laboratory Curiosity Rover

Studienarbeit (T2\_000)  
für die Prüfung zum  
Bachelor of Science

des Studiengangs Angewandte Informatik an der  
Dualen Hochschule Baden-Württemberg Mosbach

von

Niklas Koopmann

Abgabedatum	26. April 2020
Bearbeitungszeitraum	24 Wochen
Matrikelnummer, Kurs	9742503, MOS-TINF17B
Ausbildungsunternehmen	Deutsche Bundesbank
Gutachter der Dualen Hochschule	Prof. Dr. Carsten Müller

# Ehrenwörtliche Erklärung

## Erklärung

Ich versichere hiermit, dass ich meine Bachelorarbeit (bzw. Studien- und Projektarbeit) mit dem Thema „Mars Science Laboratory Curiosity Rover“ selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt. \*

\* falls beide Fassungen gefordert sind

---

Ort, Datum

---

Niklas Koopmann

**Hinweis:** Aus Gründen der besseren Lesbarkeit wird in dieser Arbeit auf die gleichzeitige Verwendung männlicher und weiblicher Sprachform verzichtet (*generisches Maskulinum*). Sämtliche Personenbezeichnungen gelten gleichwohl für beiderlei Geschlecht.

# **Abstract**

# **Zusammenfassung**

# Inhaltsverzeichnis

<b>Abstract</b>	<b>III</b>
<b>Abkürzungsverzeichnis</b>	<b>V</b>
<b>Abbildungsverzeichnis</b>	<b>VI</b>
<b>Tabellenverzeichnis</b>	<b>VII</b>
<b>Quelltextverzeichnis</b>	<b>VIII</b>
<b>1 Spezifikation</b>	<b>1</b>
1.1 Motivation . . . . .	3
1.2 Aufgabenstellung . . . . .	3
<b>2 Modell</b>	<b>4</b>
2.1 Vorderansicht . . . . .	4
2.2 Seitenansicht . . . . .	4
2.3 Ansicht von oben . . . . .	8
<b>3 Implementierung</b>	<b>9</b>
3.1 Installation und Konfiguration Raspberry Pi . . . . .	9
3.2 Installation und Konfiguration Kamera . . . . .	10
<b>4 Nachweis der Leistungsfähigkeit</b>	<b>13</b>
<b>5 Fazit</b>	<b>14</b>
5.1 Ausblick . . . . .	14
<b>Literaturverzeichnis</b>	<b>15</b>
<b>Anhang</b>	<b>17</b>

# **Abkürzungsverzeichnis**

<b>DHBW</b>	Duale Hochschule Baden-Württemberg
<b>GPIO</b>	General Purpose Input/Output
<b>HSV</b>	Hue, Saturation, Value
<b>NASA</b>	National Aeronautics and Space Administration
<b>RGB</b>	Rot, Grün, Blau

# **Abbildungsverzeichnis**

2.1	Gerenderte Frontansicht des digitalen Rover-Modells . . . . .	5
2.2	Gerenderte Seitenansicht des digitalen Rover-Modells . . . . .	6
2.3	Gerenderte Draufsicht des digitalen Rover-Modells . . . . .	7
3.1	Belegung der Steuerungsausgänge zu den einzelnen Motoren an den drei BrickPis des Steuergerätes . . . . .	10
3.2	Belegung der Ein- und Ausgänge eines einzelnen BrickPi 3 . . . . .	11
3.3	Blauer LEGO-Stein 3005 . . . . .	11

# **Tabellenverzeichnis**

# **Quelltextverzeichnis**

# 1 Spezifikation

## Studienarbeit

### Spezifikation

<b>Titel</b>	Research Lab for Deep Learning (Prio 1)
	Mars Science Laboratory Curiosity Rover
<b>Bearbeiter</b>	Niklas Koopmann
<b>Zielsetzung</b>	Konzeptionierung und Implementierung eines Rover für die Erkundung einer (Mars-)Oberfläche.
<b>Wichtige Hinweise</b>	<b>Präsentation ausgewählter Ergebnisse auf der renommierten Konferenz „ICSI 2020“, der „AI Night“ sowie der „Hannover Messe 2020“.</b>

#### Beschreibung:

Mars Science Laboratory ist eine NASA-Mission im Rahmen des Flagship-Programms, die den Mars hinsichtlich seiner aktuellen und vergangenen Eignung als Biosphäre erforscht.

Hierzu wurde auf der Oberfläche ein weitgehend autonomer Rover mit dem Namen Curiosity abgesetzt, der mit Instrumenten zur Untersuchung von Gestein, Atmosphäre und Strahlung ausgerüstet ist.

#### Animation

<https://www.youtube.com/watch?v=P4boyXQuUIw>



Eine LKW-Plane wird mit Mustern – zwecks Klassifizierung von Gesteinstypen – bedruckt. Unter der LKW-Plane wird Sand aufgeschüttet, um ein Gelände mit Höhen und Tiefen zu schaffen.

Der Rover ist durch eine geeignete Fahrwerkskonstruktion weitestgehend gegen Kippen zu sichern.

Über die Sprachkommandos *Start*, *Move Left*, *Move Right* und *Stop* wird der Rover gesteuert.

Zusätzlich werden rechteckige kleine Objekte (Wasser) an beliebigen Stellen auf der LKW-Plane platziert. Der Rover bewegt sich innerhalb definierter Grenzen auf der LKW-Plane. Hat der Rover ein (trainiertes) bekanntes Objekt gefunden, wird die Klassifizierung zu Simulationszwecken als Sprachausgabe kommuniziert, wie z.B. Water found. Bei dem Fund von Wasser wird zusätzlich das Objekt als Probe aufgenommen und in einem Behälter gesammelt.

<b>Organisation</b>	5. Semester: Konzeptionierung, Prototypische Implementierung; 6. Semester: Produktive Implementierung, Test, Dokumentation.
<b>Rahmenbed.</b>	Raspberry Pi, Pixy-Kamera, Programmiersprache: Python; ggf. zusätzliche Teile sind im 3D-Druck zu erstellen  Das Lego-Modell ist formschön für die Messe-Präsentation zu realisieren.  <b>Raspberry Pi und Lego-Teile für die prototypische Implementierung werden leihweise von der DHBW (Labor Digitale Fabrik) zur Verfügung gestellt.</b>
Status	Final

Dr. Carsten Müller // 12.10.2019

## Projektorganisation

### Meilensteine 5. Semester (1. Abschnitt)

Zielsetzung ist eine qualitativ hochwertige Studienarbeit mit der Bewertung 1.0.

5. Semester: Konzeptionierung und Prototypische Implementierung kritischer Komponenten		
#	Meilenstein	Termin
M01	Basis-Installation und Konfiguration Raspberry Pi	20.10.2019
M02	Steuerung Rover über Sprachkommandos	03.11.2019
M03	Basis-Installation und Konfiguration Pixy-Kamera	10.11.2019
M04	Erkennung von Wasserobjekten und Sprachausgabe	24.11.2019
M05	Sechsrädriges Antriebssystem für unebenes Gelände	01.12.2019
M06	Dokumentation mit BrickLink Studio (Version 2.0.10) <a href="https://studio.bricklink.com/v3/studio/download.page">https://studio.bricklink.com/v3/studio/download.page</a>	08.12.2019

Die Termine zu den Meilensteinen M01-M06 sind verbindlich.

Zu jedem Meilenstein findet eine Skype Session zwecks Qualitätssicherung und Abnahme statt.  
Bitte vor der Skype Session die Dropbox mit qualitativ hochwertigen Arbeitsergebnissen aktualisieren.

## 1.1 Motivation

Am 6. August 2012 landete der Rover *Curiosity* als Teil der Mission *Mars Science Laboratory* der US-amerikanischen Raumfahrtbehörde National Aeronautics and Space Administration (NASA) im Gale-Krater auf dem Mars [17]. Seither hat er rund 21,93 Kilometer zurückgelegt (Stand: Sol 2695) [13] und die Landschaft entlang der Fahrtstrecke auf vielfältige Weise im Hinblick auf ihre frühere und aktuelle Eignung als Lebensraum für Organismen untersucht. Ein wichtiger Aspekt für den Nachweis der Existenz von Lebewesen ist das Vorhandensein flüssigen Wassers [14], dessen Rückstände der Rover mithilfe eines aktiven Neutronenspektrometers in Mineralien nachzuweisen versucht. [17]

Um weite Strecken im schroffen, unebenen Gelände der Marsoberfläche zurücklegen zu können, ist *Curiosity* mit einem komplexen, sechsrädrigen Antriebs- und Federungssystem ausgestattet. Das Fahrwerk ist nach dem Rocker-Bogie-System der NASA aufgebaut, um Hindernisse bis zu einer Höhe des Raddurchmessers von 0,5 m überwinden zu können [1]. Eine solche Aufhängung ermöglicht dem Rover eine hohe Geländegängigkeit ohne den Einsatz von Achsen und erlaubt der Verbindung der hinteren Räder beider Seiten eine freie Drehung um den Verbindungspunkt zur Hauptstrebe, an der auch das jeweilige Vorderrad montiert ist [2].

Dieses Projekt (Mars Science Laboratory Curiosity Rover) wird im Rahmen des Research Lab for Deep Learning der Dualen Hochschule Baden-Württemberg (DHBW) zeitgleich mit der Erforschung schwarmbasierter Logistikabläufe durchgeführt. Langfristig soll das Ergebnis unter anderem zum Zwecke einer schwarmbasierten Erkundung marsähnlicher Landschaften weiterentwickelt werden können.

## 1.2 Aufgabenstellung

Ziel dieser Arbeit ist die „Konzeptionierung und Implementierung eines Rover für die Erkundung einer (Mars-)Oberfläche“ [11]. Die Suche nach Wasser wird dabei ebenfalls abgebildet und soll mithilfe der visuellen Erkennung blauer LEGO-Steine simuliert werden. Zusätzlich ist das Antriebskonzept des *Curiosity*-Rovers originalgetreu nachzubilden: Es sollen sechs angetriebene Räder installiert sein, von denen vier gelenkt werden. Eine grundlegende mechanische Federung soll die Geländetauglichkeit steigern.

Literatur-Test: [18] [8] [12] [6] [7] [10] [4] [16]

# 2 Modell

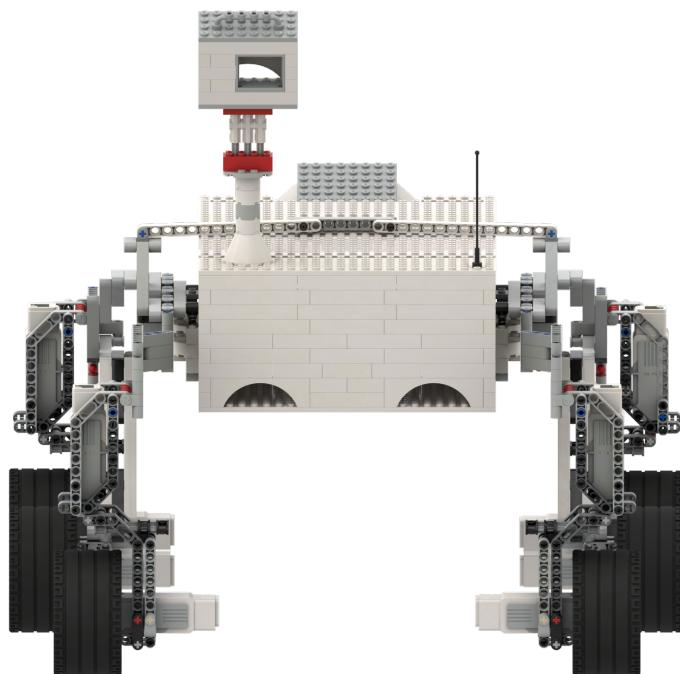
Im ersten Schritt soll die digitale Modellierung des zu erschaffenden Mars Rovers erfolgen. Dazu wird gemäß der Spezifikation die Anwendung BrickLink-Studio (in der Version 2.0.10) genutzt. Diese ermöglicht die umfangreiche und präzise Modellierung mit LEGO-Elementen sowie beispielsweise Funktionen zur Stabilitätsanalyse und Generierung schrittweiser Bauanleitungen. In den folgenden drei Abschnitten sind jeweils eine gerenderte Front- und Seitenansicht und eine Draufsicht des Mars-Rover-Modells dargestellt. Zusätzlich ist zum Vergleich in jedem Abschnitt eine Fotografie des realen Modells aus der entsprechenden Perspektive abgebildet.

## 2.1 Vorderansicht

In der Frontalansicht erkennt man gut die zwei breiten Vorderräder, die über jeweils einen Motor zum Antrieb und einen für die Lenkung des Rades verfügen. Über eine Aufhängung sind sie mit dem Korpus des Rovers verbunden und etwas von diesem abgesetzt, um die Stabilität zu erhöhen. Weiterhin ist in der Realaufnahme die genutzte Kamera (siehe dazu Abschnitt 3.2), platziert im charakteristischen Kopf des Rovers, erkennbar. Der Kopf ist um 22,5° nach unten geneigt, sodass die Erkennung von Wasserobjekten auch auf kurze Distanz möglich bleibt.

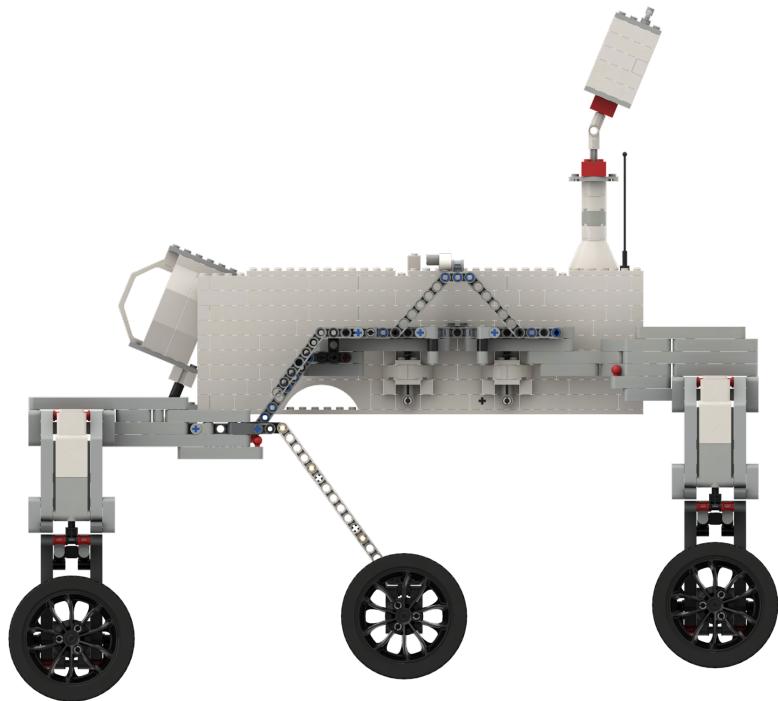
## 2.2 Seitenansicht

Der im vorherigen Abschnitt beschriebene Neigungswinkel des Kopfes wird in der Seitenansicht deutlich. Weiterhin sind die mittleren und die Hinterräder des Rovers zu sehen. Auf beiden Seiten sind diese starr miteinander verbunden und gemeinsam an einem einzigen Punkt mit einer vom Korpus ausgehenden, gefederten Strebe verbunden. Dies entspricht dem Aufbau des Rocker-Bogie-Systems, welches bei *Curiosity* zum Einsatz gekommen ist. So ist sichergestellt, dass die mittleren Räder, die den Rover seitlich stabilisieren und antreiben, sowie die hinteren Räder, die ihn antreiben und gegebenenfalls lenken, in jedem Fall den Bodenkontakt beibehalten. Bezüglich der Verbindung



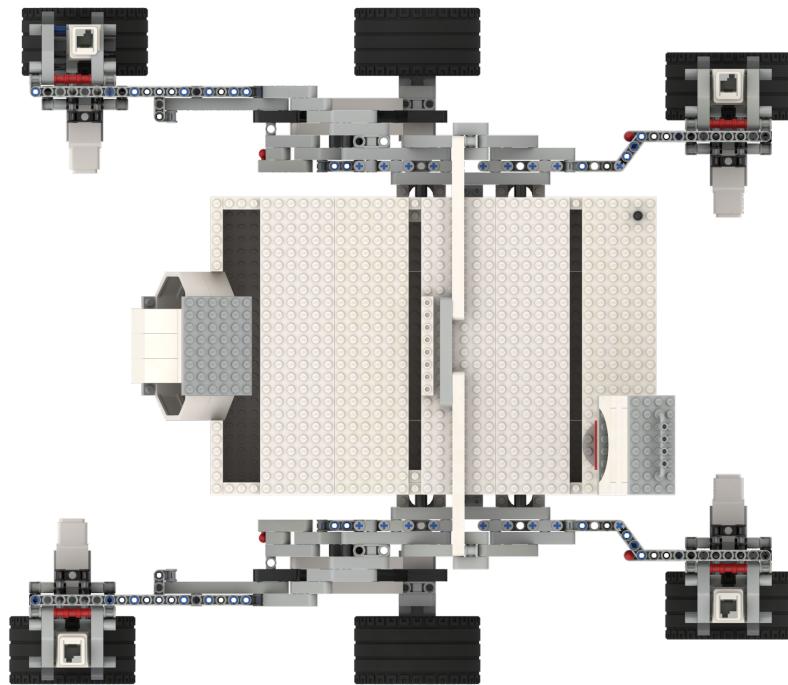
Quelle: eigene Darstellung (Render-Export aus BrickLink Studio 2.0.10)

Abbildung 2.1: Gerenderte Frontansicht des digitalen Rover-Modells



Quelle: eigene Darstellung (Render-Export aus BrickLink Studio 2.0.10)

Abbildung 2.2: Gerenderte Seitenansicht des digitalen Rover-Modells



Quelle: eigene Darstellung (Render-Export aus BrickLink Studio 2.0.10)

Abbildung 2.3: Gerenderte Draufsicht des digitalen Rover-Modells

zum Korpus wurde vom Rocker-Bogie-System insofern abgewichen, dass dieser nicht frei um einen Punkt rotieren kann, da eine kontrollierte Verlagerung seines Schwerpunkts entlang der Fahrtrichtungsachse aktuell nicht ermöglicht werden kann. Entsprechend bestünde zum Beispiel durch Reibung an der Verbindungsstelle zur Aufhängung oder Windeinwirkung von vorn oder hinten die Gefahr, dass der Korpus kippt und nicht selbstdämmig zurückgedreht werden kann. Es ist jedoch eine minimale Drehung um den zentralen Ankerpunkt zwischen Korpus und Fahrwerk möglich. Durch zwei Federn, die an beiden Seiten auf gleicher Höhe zu diesem Punkt angebracht sind, wird der Korpus soweit nötig zurück in die Waagerechte gefedert.

## 2.3 Ansicht von oben

In der Draufsicht ist ein aus einigen Liftarm-Streben bestehender, brückenartiger Überbau auf dem Rover zu erkennen. Dieser dient einerseits der gleichmäßigeren Lastverteilung und unterstützt andererseits die Ankerpunkte zwischen Korpus und Fahrwerk, da die Teile dauerhaft unter Spannung stehen und dadurch seitlich von außen Druck auf die Ankerpunkte ausüben. Im digitalen Modell ist die Strebe in der Mitte dieses Überbaus asymmetrisch zum Rest, da BrickLink Studio es nicht ermöglicht, die Teile unter Spannung anzubringen. Die Fotografie des realen Modells zeigt, wie die Anbringung konzipiert ist, und dass die diagonal angeordneten Streben an den Ankerpunkten (in Abbildung ?? oben und unten mittig) zur Mitte gezogen werden.

Bei Betrachtung des realen Modells von oben erkennt man das Solarpaneel auf dem Korpus des Rovers. Dieses soll die Energiegewinnung aus Sonnenlicht von Curiosity abbilden, wenngleich der Nennstrom des Moduls mit  $250\text{ mA}$  auf das entwickelte Gesamtsystem keine großen Auswirkungen hätte. Aus einer Messung geht hervor, dass in das Gesamtsystem bereits im Leerlauf (alle BrickPis und Raspberry Pi eingeschaltet, keine Motoren aktiv) ein Strom von rund  $1,3\text{ A}$  fließt.

Darüber hinaus ist in dieser Ansicht ein guter Überblick über die Platzierung der einzelnen Räder und Motoren in Relation zum Korpus gegeben. Um den Rover seitlich zu stabilisieren, sind die beiden mittleren Räder etwas weiter außen platziert. Alle Räder liegen jeweils mehrere Zentimeter vom Korpus entfernt am Boden auf.

# 3 Implementierung

In diesem Kapitel wird die gesamte Systemumgebung um das Steuerelement in Form eines Raspberry Pi beschrieben und erläutert. Die gesamte Implementierung der Steuerungsskripts erfolgt in Python und wird sinnvoll in Module unterteilt. Der Quelltext zum gesamten Projekt wird auf GitHub veröffentlicht.

Insgesamt werden vier Module implementiert, die alle diskrete Funktionalitäten des Modells abdecken: Die Funktionen zur Motoransteuerung sind im Modul `Actuator` implementiert. Für die Bilderkennung der Wasserobjekte auf einem Kamera-Feed der verwendeten Kamera (Details siehe Abschnitt 3.2) ist das Modul `Camera` entwickelt worden. Die programmierten Skripte für Sprachein- und -ausgabe befinden sich respektive in den Modulen `Voice_Recognition` und `Voice_Output`. Koordiniert wird die auf diesen vier Modulen basierende Roversteuerung durch das Skript `rover.py`. In der Konfigurationsdatei `config.py` werden Variablen zur globalen Nutzung zentral deklariert und mit passenden Werten initialisiert.

## 3.1 Installation und Konfiguration Raspberry Pi

Die Steuerung des Mars-Rover-Modells erfolgt zentral über einen Raspberry Pi 3 Model B Plus (Revision 1.3). Auf diesem ist die Debian-basierte Distribution Raspbian for Robots der US-amerikanischen Firma Dexter Industries installiert worden. Dieses Betriebssystem enthält standardmäßig viele vorinstallierte Anwendungen und Treiber für den Einsatz des Raspberry Pi zur Steuerung von Robotern. Insbesondere sind alle benötigten Bibliotheken zur Nutzung der zusätzlichen Platine BrickPi vorinstalliert und -konfiguriert. Der BrickPi ist ebenfalls ein Produkt von Dexter Industries und stellt eine Schnittstelle zwischen den verschiedenen LEGO-Mindstorms-Sensoren und -Motoren zum Raspberry Pi bereit. Er fungiert dabei auch als Stromquelle für diese sowie für den Raspberry Pi selbst und muss dazu lediglich auf dessen Pins für General Purpose Input/Output (GPIO) aufgesteckt werden. Da jeder BrickPi nur über vier Ausgänge zum Betrieb von EV3-Motoren verfügt, ist der gleichzeitige Einsatz von drei BrickPis nötig, um die zehn Motoren des Modells anzutreiben. Dabei kann jedes der sechs Räder separat durch einen Motor angetrieben sowie die äußeren vier Räder einzeln gelenkt werden. Eine Lenkung

Betrachtungsrichtung ≈ Fahrtrichtung

	A	B	C	D
Antrieb Mitte SN: 45C31FAF514D3937304B2020FF15122B	Mitte links	kein Motor	kein Motor	Mitte rechts
Lenkung SN: 84880DFD514D3937304B2020FF0A172A	hinten links	vorn links	vorn rechts	hinten rechts
Antrieb vorn/hinten SN: 07976FB6515035524E202020FF101B0C				
Raspberry Pi				

Quelle: eigene Darstellung

Abbildung 3.1: Belegung der Steuerungsausgänge zu den einzelnen Motoren an den drei BrickPis des Steuergerätes

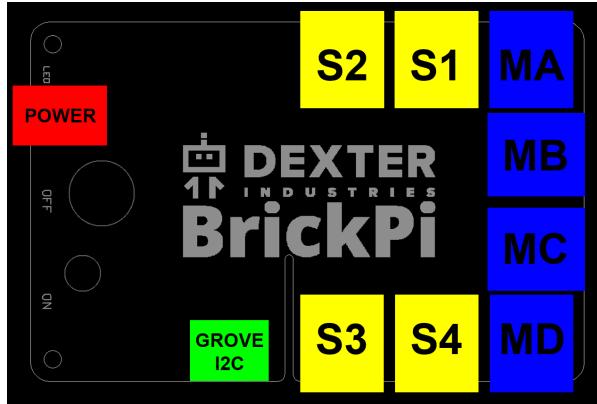
der mittleren Räder ist zum aktuellen Zeitpunkt nicht notwendig, kann aber problemlos über die verbleibenden beiden Ports der BrickPis ergänzt werden, beispielsweise um seitliche Fahrten zu ermöglichen.

In Abbildung 3.1 ist die standardmäßige Belegung der drei BrickPis sowie deren Seriennummern dargestellt. Die Spalten A bis D entsprechen den

Um die Stromversorgung sicherzustellen, werden drei Powerbanks mit einer Gesamtkapazität von  $60,3\text{ Ah}$  und einem Gesamtgewicht von  $1251\text{ g}$  im Korpus des Rovers untergebracht. Der Raspberry Pi selbst wird durch die BrickPis über seine GPIO-Pins mit Energie versorgt [3] und benötigt entsprechend keine eigene Stromquelle.

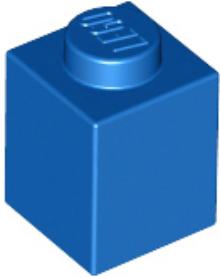
## 3.2 Installation und Konfiguration Kamera

Für die Videoeingabe wird die originale Raspberry Pi Camera V2 genutzt. Diese nutzt den CMOS-Sensor Sony IMX219 mit einer Auflösung von bis zu 8 Megapixeln [15]. Die Aufnahmen erfolgen mit einer reduzierten Auflösung von  $1280 \times 720$  Pixeln, um eine Frequenz von 60 Hz zu ermöglichen. So ist sichergestellt, dass Wasserobjekte auch während der Fahrt gut erkannt werden können.



Quelle: [5]

Abbildung 3.2: Belegung der Ein- und Ausgänge eines einzelnen BrickPi 3



Quelle: <https://www.bricklink.com/v2/catalog/catalogitem.page?P=3005&C=7>

Abbildung 3.3: Blauer LEGO-Stein 3005

Die Objekterkennung für Wasserobjekte erfolgt mithilfe der Programmbibliotheken picamera und OpenCV. Erstere stellt eine Schnittstelle zwischen der Raspberry-Pi-Kamera und Python zur Verfügung. Letztere ermöglicht die Echtzeitverarbeitung der Kamerabilder: Die Objekte werden auf Basis ihrer blauen Farbe erkannt, da sich diese deutlich von der charakteristischen Farbe der Vorbildlandschaften des Mars unterscheidet. Letztere sei eine „predominantly yellowish brown color with only subtle variation“ [9]. Die LEGO-Steine, welche als Wasserobjekte fungieren, haben die LEGO-Farb-ID 7. Diese korrespondiert mit dem Farbcode #0057A6 in Hexadezimaldarstellung. Ein solcher LEGO-Stein ist in Abbildung 3.3 dargestellt.

Da eine farbwertbasierte Erkennung der Wasserobjekte genutzt wird, entfällt das Trainieren eines Erkennungsalgorithmus, wie es bei einem neuronalen Netz nötig wäre. Die Farbeigenschaften des LEGO-Steins werden in Komponenten für rote, grüne und blaue Farbanteile aufgeteilt und in der Konfiguration hinterlegt. Das Python-Skript berechnet für die RGB-Farbangaben die entsprechenden HSV-Werte. In letzterer Darstellung ist die Information über tatsächliche Farbe nur noch im Hue-Wert (0 bis 255) hinterlegt, sodass für diesen eine bestimmte Abweichung (Standard:  $\pm 10$ ) definiert werden kann. Die Sättigung (Saturation) und der Hellwert

### 3.2 Installation und Konfiguration Kamera

(Value) (beide ebenfalls von 0 bis 255 definiert) werden in einem breiten Spektrum toleriert: Standardmäßig lässt das Skript hier Werte von mindestens 80 und höchstens 255 zu. Mithilfe dieser Parameter wird eine solide Erkennung der Präsenz der LEGO-Steine gewährleistet.

# 4 Nachweis der Leistungsfähigkeit

Der Nachweis der Leistungsfähigkeit entsprechend der angepassten Anforderungen wird durch die folgenden Dokumentationsteile erbracht:

Das BrickLink-Modell wird möglichst originalgetreu gehalten und dokumentiert die Struktur des Rovers. Durch verschiedene Teile, die unter bestimmten Bedingungen (aufgebockt, stehend, fahrend et cetera) unter Spannung stehen und bewegliche Teile kann das digitale Modell nicht exakt die reale LEGO-Struktur abbilden. An einzelnen Stellen sind derartige Teile (häufig beschränkt auf einzelne Pins) ausgelassen worden, um Überschneidungen im digitalen Modell zu vermeiden. Die Teile sind aber in den verschiedenen Fotografien des Rovers (siehe Kapitel 2) zu erkennen.

Durch Videoaufnahmen verschiedener Einsatzszenarien ist die Leistungsfähigkeit der einzelnen Anforderungen nachgewiesen. In einer ersten Aufnahme klassifiziert der Rover stehend Wasserobjekte, die vor ihm auf dem Boden verteilt werden, und es erfolgt eine Sprachausgabe des spezifizierten Textes „Water found“. Eine weitere Aufnahme zeigt den Rover stehend, während er wiederholt Wasserobjekte in einer Hand erkennt, die abwechselnd verdeckt und präsentiert werden. Hierbei wird besonders die Sprachausgabe hörbar, da der Rover aus der Nähe gefilmt wird. Die Fahrtauglichkeit wird durch eine Aufnahme dokumentiert, in welcher dem Rover ein Sprachkommando „Start“ gegeben wird, woraufhin er aus dem Stand losfährt, bis er nach drei Sekunden wieder zum Stehen kommt. Für die Spracheingabe muss zum aktuellen Zeitpunkt ein kabelgebundenes Mikrofon genutzt werden, da die Bestellung eines neuen Mikrofons aufgrund äußerer Einflüsse nicht möglich war.

- Die Geländetauglichkeit ist obsolet, hingegen ist die Formschönheit und Stabilität zu perfektionieren.
- Auf dem Boden sind kleine blaue Objekte - welche Wasser symbolisieren - verteilt.
- Mars Rover wird über die Sprachkommandos Start, Move Left, Move Right und Stop gesteuert.
- Hat der Mars Rover ein (trainiertes) bekanntes Objekt gefunden, wird die Klassifizierung zu Simulationszwecken als Sprachausgabe kommuniziert.
- Das Sammeln der klassifizierten Objekte in einem Behälter ist obsolet.

# **5 Fazit**

## **5.1 Ausblick**

# Literaturverzeichnis

- [1] R. Arvidson et al. „Mars Science Laboratory Curiosity Rover Terramechanics Initial Results“. In: *Proceedings of the 44th Lunar and Planetary Science Conference*. The Woodlands, Texas, 03/2013 (siehe S. 3).
- [2] D. Bickler. „Roving on Mars“. In: *Mechanical Engineering* (04/1998), S. 74–77 (siehe S. 3).
- [3] J. Cole. „Building bots with LEGO“. In: *The MagPi* 17 (10/2013), S. 4–7 (siehe S. 10).
- [4] T. Cox. *Raspberry Pi Cookbook for Python Programmers*. Birmingham, UK: Packt Publishing, 04/2014. ISBN: 978-1-84969-662-3 (siehe S. 3).
- [5] Dexter Industries. *BrickPi3 Getting Started Step 3 – Build your BrickPi3 Robot*. 2017. URL: <https://www.dexterindustries.com/BrickPi/brickpi3-getting-started-step-3-build-robot-attach-lego-mindstorms-nxt-ev3-sensors-motors-technic-parts/> (besucht am 25.04.2020) (siehe S. 11).
- [6] W. Donat. *Learn Raspberry Pi Programming with Python. Learn to Program on the World’s Most Popular Tiny Computer*. 2. Aufl. Palmdale, CA: Apress Media, 2018. ISBN: 978-1-4842-3769-4 (siehe S. 3).
- [7] G. Halfacree. *The Official Raspberry Pi Beginner’s Guide. How to use your new computer*. 2. Aufl. Cambridge, UK: Raspberry Pi Press, 2019. ISBN: 978-1-912047-62-8 (siehe S. 3).
- [8] B. Horan. *Practical Raspberry Pi*. Kalifornien: Apress Media, 2013. ISBN: 978-1-4302-4972-6 (siehe S. 3).
- [9] J. N. Maki, J. J. Lorre, P. H. Smith, R. D. Brandt und D. J. Steinwand. „The color of Mars: Spectrophotometric measurements at the Pathfinder landing site“. In: *Journal of Geophysical Research* 104.E4 (25.04.1999), S. 8781–8794 (siehe S. 11).

- [10] S. McManus und M. Cook. *Raspberry Pi for Dummies*. 3. Aufl. Hoboken, NJ: John Wiley & Sons, Inc., 2017. ISBN: 978-1-119-41200-7 (siehe S. 3).
- [11] C. Müller. *Studienarbeit: Spezifikation*. Version final. 12. 10. 2019 (siehe S. 3).
- [12] D. Molloy. *Exploring Raspberry Pi. Interfacing to the Real World with Embedded Linux*. Indianapolis, IN: John Wiley & Sons, Inc., 2016. ISBN: 978-1-119-18868-1 (siehe S. 3).
- [13] NASA. *Where is Curiosity?* 2020. URL: [https://mars.nasa.gov/msl/mission/where-is-the-rover/?page=0&per\\_page=25&order=sol+desc&search=&category=176%3A295&url\\_suffix=%3Fsite%3Dmsl](https://mars.nasa.gov/msl/mission/where-is-the-rover/?page=0&per_page=25&order=sol+desc&search=&category=176%3A295&url_suffix=%3Fsite%3Dmsl) (besucht am 25. 03. 2020) (siehe S. 3).
- [14] NASA JPL. *Mars Science Laboratory: Curiosity Rover*. Präsentation. NASA, 2013 (siehe S. 3).
- [15] M. Pagnutti et al. „Laying the foundation to use Raspberry Pi 3 V2 camera module imagery for scientific and engineering purposes“. In: *Journal of Electronic Imaging* 26.1 (11. 02. 2017). DOI: 10.1117/1.JEI.26.1.013014 (siehe S. 10).
- [16] A. Pajankar. *Raspberry Pi Computer Vision Programming*. Birmingham, UK: Packt Publishing, 05/2015. ISBN: 978-1-78439-828-6 (siehe S. 3).
- [17] A. R. Vasavada et al. „Overview of the Mars Science Laboratory mission: Bradbury Landing to Yellowknife Bay and beyond“. In: *Journal of Geophysical Research: Planets* 119 (05. 06. 2014), S. 1134–1161. DOI: 10.1002/2014JE004622 (siehe S. 3).
- [18] S. Yamanor und S. Yamanor. *Python Programming with Raspberry Pi*. Birmingham, UK: Packt Publishing, 04/2017. ISBN: 978-1-78646-757-7 (siehe S. 3).

# **Anhang**