

# 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	22. 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 . . . . .	7
<b>3 Implementierung</b>	<b>8</b>
3.1 Installation und Konfiguration Raspberry Pi . . . . .	8
3.2 Installation und Konfiguration Kamera . . . . .	9
<b>4 Nachweis der Leistungsfähigkeit</b>	<b>11</b>
<b>5 Fazit</b>	<b>12</b>
5.1 Ausblick . . . . .	12
<b>Literaturverzeichnis</b>	<b>13</b>
<b>Anhang</b>	<b>15</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	Zusammenfassung der Aufgaben eines Datenbankkontextes im Entity Framework . . . . .	5
2.2	Zusammenfassung der Aufgaben eines Datenbankkontextes im Entity Framework . . . . .	6
3.1	Blauer LEGO-Stein 3005 . . . . .	9

# Tabellenverzeichnis

# Quelltextverzeichnis



# 1 Spezifikation

## Studienarbeit

### Spezifikation

<b>Titel</b>	<b>Research Lab for Deep Learning (Prio 1)</b> 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>
<b>Beschreibung:</b>	
<p>Mars Science Laboratory ist eine NASA-Mission im Rahmen des Flagship-Programms, die den Mars hinsichtlich seiner aktuellen und vergangenen Eignung als Biosphäre erforscht.</p> <p>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.</p> <p><b>Animation</b>  <a href="https://www.youtube.com/watch?v=P4boyXQuUlw">https://www.youtube.com/watch?v=P4boyXQuUlw</a></p>	
	
<p>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.</p> <p>Der Rover ist durch eine geeignete Fahrwerkskonstruktion weitestgehend gegen Kippen zu sichern.</p> <p>Über die Sprachkommandos <i>Start</i>, <i>Move Left</i>, <i>Move Right</i> und <i>Stop</i> wird der Rover gesteuert.</p> <p>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.</p>	
<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>
<b>Status</b>	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 [16]. Seither hat er rund 21,93 Kilometer zurückgelegt (Stand: Sol 2695) [12] 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 [13], dessen Rückstände der Rover mithilfe eines aktiven Neutronenspektrometers in Mineralien nachzuweisen versucht. [16]

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“ [10]. 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: [17] [7] [11] [5] [6] [9] [4] [15]

## 2 Modell

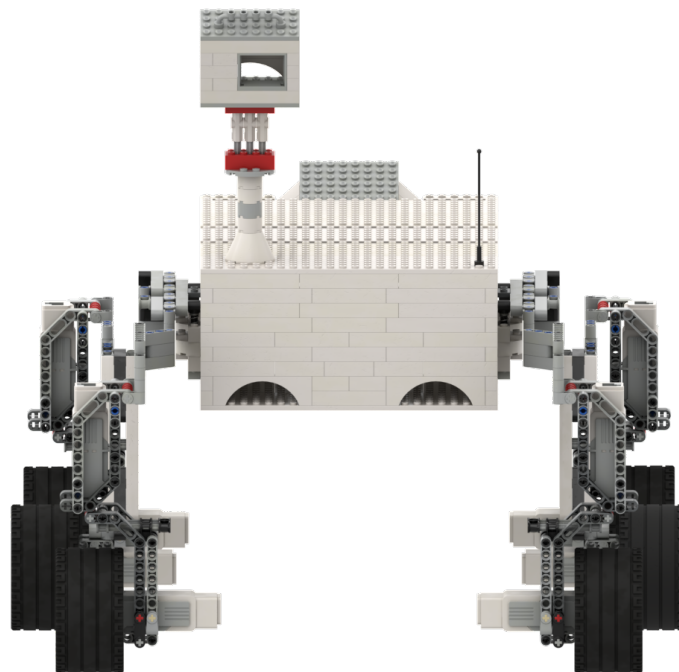
Im ersten Schritt soll die digitale Modellierung des zu erschaffenden Mars Rovers erfolgen. Dazu wird gemäß der Spezifikation die Anwendung BrickLink-Studio (in der Version 2.0.10) genutzt. Diese ermöglicht die umfangreiche und präzise Modellierung mit LEGO-Elementen sowie beispielsweise Funktionen zur 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  $22,5^\circ$  nach unten geneigt, sodass die Erkennung von Wasserobjekten auch auf kurze Distanz möglich bleibt.

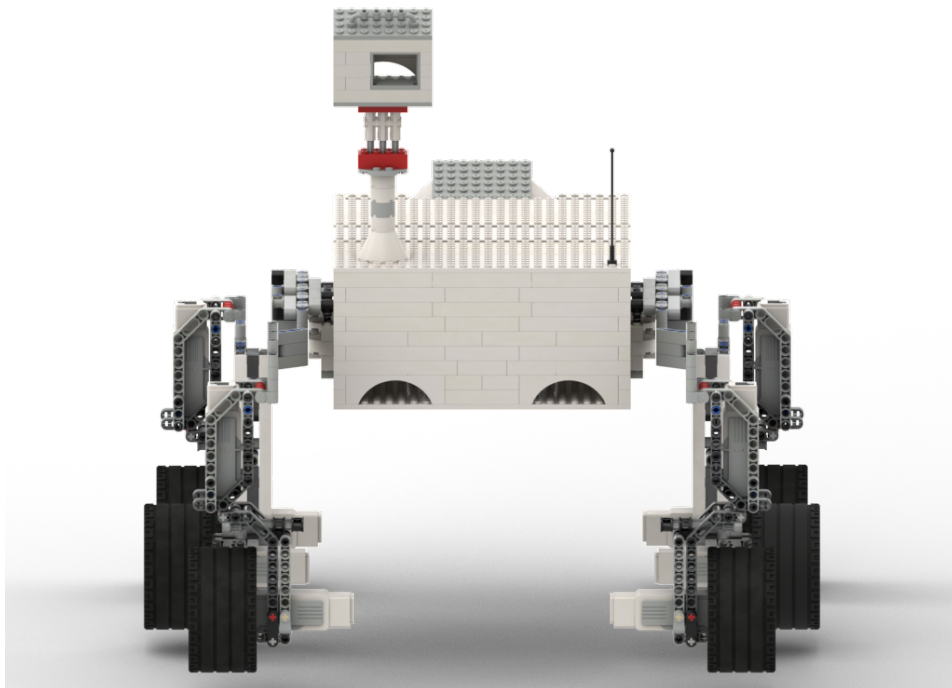
### 2.2 Seitenansicht

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 Stäbe 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

Abbildung 2.1: Zusammenfassung der Aufgaben eines Datenbankkontextes im Entity Framework



Quelle: eigene Darstellung

Abbildung 2.2: Zusammenfassung der Aufgaben eines Datenbankcontextes im Entity Framework

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 besteht zum Beispiel durch Reibung an der Verbindungsstelle zur Aufhängung oder Windeinwirkung von vorn oder hinten die Gefahr, dass der Korpus kippt und nicht selbsttätig zurückgedreht werden kann.

## 2.3 Ansicht von oben

Bei Betrachtung des realen Modells von oben erkennt man das Solarpaneel auf dem Korpus des Rovers. Dieses soll die Energiegewinnung aus Sonnenlicht abbilden, wenngleich der Nennstrom des Moduls mit  $250\text{ mA}$  auf das Gesamtsystem keine großen Auswirkungen hätte.

Darüber hinaus ist hier ein guter Überblick über die Platzierung der einzelnen Räder und Motoren in Relation zum Korpus erkennbar.

## 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



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.

Um die Stromversorgung sicherzustellen, werden drei Powerbanks mit einer Gesamtkapazität von 60,3 Ah und einem Gesamtgewicht von 1251 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 [14]. 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: <https://www.bricklink.com/v2/catalog/catalogitem.page?P=3005&C=7>

Abbildung 3.1: 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“ [8]. 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 3.1 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 (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 ist mit dem folgenden (vereinfachten) Modell zu erbringen:

- 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. 9).
- [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] 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).
- [6] 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).
- [7] B. Horan. *Practical Raspberry Pi*. Kalifornien: Apress Media, 2013. ISBN: 978-1-4302-4972-6 (siehe S. 3).
- [8] 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. 9).
- [9] 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).
- [10] C. Müller. *Studienarbeit: Spezifikation*. Version final. 12. 10. 2019 (siehe S. 3).

- [11] 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).
- [12] 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).
- [13] NASA JPL. *Mars Science Laboratory: Curiosity Rover*. Präsentation. NASA, 2013 (siehe S. 3).
- [14] 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. 9).
- [15] A. Pajankar. *Raspberry Pi Computer Vision Programming*. Birmingham, UK: Packt Publishing, 05/2015. ISBN: 978-1-78439-828-6 (siehe S. 3).
- [16] 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).
- [17] S. Yamanoor und S. Yamanoor. *Python Programming with Raspberry Pi*. Birmingham, UK: Packt Publishing, 04/2017. ISBN: 978-1-78646-757-7 (siehe S. 3).

# Anhang