Bachelorarbeit

zur Erlangung des akademischen Grades
Bachelor of Engineering

Technische Hochschule Wildau
Fachbereich Ingenieur- und Naturwissenschaften
Studiengang Telematik (B. Eng.)

Thema (deutsch): Prototypischer Bau eines Tischroboters als Plattform für zukünftige Entwicklungs- und Forschungsprojekte im Bereich des Ambient Assisted Livings und als Alternative zum NAO-Roboter

Thema (englisch): Prototypal construction of a table robot as a future development and research platform in the area of Ambient Assisted Living and as an alternative for the NAO-Robot

vorgelegt von: Valentin Schröter
Seminargruppe: T/16

Betreuerin: Prof. Dr. rer. nat. Janett Mohnke
Zweitgutachterin: Master of Engineering Janine Breßler

Bearbeitungszeitraum: 22.04.2019 – 05.09.2019

Themenstellender Betrieb: Technische Hochschule Wildau

Verlängerung des Bearbeitungszeitraumes

Auf Grund der Antragstellung des Studierenden wurde der Bearbeitungszeitraum für die Erstellung der Arbeit um 4 Wochen verlängert.

Das Abgabedatum nach Antragstellung ist der 05.09.2019.

Bachelorarbeit

Antrag vom: 14.04.2019

Name:	Valentin Schröter	Matrikel-Nr.:	50032251
Studiengang:	Telematik	Seminargruppe:	T/16
Beginn der Arbeit:	22.04.2019		
Abgabetermin:	08.08.2019		
Betreuende Hochschuldozentin:		Prof. Dr. rer. nat. Janett Mohnke	
Zweitgutachterin:		Master of Engineering Janine Breßler	
Fachliche Betreuungsperson des Themenstellers:		Prof. Dr. rer. nat. Janett Mohnke	
Themenstellender Betrieb:		Technische Hochschule Wildau	
Straße:		Hochschulring 1	
PLZ, Ort:		15745, Wildau	

Kurzthema (Deutsch)

Prototypischer Bau eines Tischroboters als Plattform für zukünftige Entwicklungs- und Forschungsprojekte im Bereich des Ambient Assisted Livings und als Alternative zum NAO-Roboter

Kurzthema (Englisch)

Prototypal construction of a table robot as a future development and research platform in the area of Ambient Assisted Living and as an alternative for the NAO-Robot

Zielstellung

Prototypischer Bau eines Tischroboters für den Einsatz im AAL

Inhaltliche Anforderungen / Teilaufgabe

- Analyse der Anforderungen an einen Tischroboter
- Auswahl der notwendigen Hardware für einen Tischroboter
- Verknüpfung der notwendigen Hardware zu einem Gesamtsystem
- Implementation der grundlegenden Schnittstellensoftware und verschiedener Hello World Programme für die zukünftige Weiterentwicklung
- Erstellung eines ansprechenden Roboter-Designs

Sprache der Arbeit

Deutsch

Konsultationen erfolgen nach Vereinbarung mit dem betreuenden Hochschullehrer.

Prof. Dr. rer. nat. Janett Mohnke	Master of Engineering Janine Breßler	Valentin Schröter	<i>genehmigt</i>
(Hochschuldozentin)	(Zweitgutachterin)	(Student)	(Prüfungsausschuss)

Eingang der Abschlussarbeit _____

Sperrvermerk ☐ Ja ☒ Nein

Bibliografische Beschreibung und Referat

Schröter, Valentin

Prototypischer Bau eines Tischroboters als Plattform für zukünftige Entwicklungs- und Forschungsprojekte im Bereich des Ambient Assisted Livings und als Alternative zum NAO-Roboter.

Bachelorarbeit, Technische Hochschule Wildau 2019, 139 Seiten, 55 Abbildungen, 28 Tabellen
1 beigelegte CD-ROM

Ziel

Ziel des Projekts ist die Entwicklung eines Tischroboters, welcher für den Einsatz im Ambient Assisted Living geeignet ist. Als Ergebnis der Bachelorarbeit soll evaluiert werden, inwiefern die Umsetzung eines Tischroboters als Alternative zum NAO möglich und praktisch umsetzbar ist. Nach der Konzeptionierung einer hardware- und softwaretechnischen Systemarchitektur soll am Ende der Arbeit ein funktionsfähiger Prototyp entstehen.

Inhalt

Die Bachelorarbeit beschreibt die Konzeptionierung und prototypische Umsetzung eines Tischroboters, welcher als Alternative zum NAO-Roboter im Bereich des Ambient Assisted Livings eingesetzt werden kann.

Dafür wird zunächst die Ausgangssituation des Projekts anhand der an der Technischen Hochschule Wildau vorhandenen und im Studiengang Telematik genutzten Roboter dargestellt. Daran anschließend werden die Anforderungen analysiert, die an einen Tischroboter für den Einsatz im Ambient Assisted Living gestellt werden. Auf Grundlage der Anforderungen werden Komponenten evaluiert, die für eine Umsetzung des Systems infrage kommen. Mithilfe dieser Auswahl wird im Anschluss ein Konzept für ein Gesamtsystem ausgearbeitet. Dieses beinhaltet sowohl eine Architektur der Hardware als auch eine Architektur der Software und beschreibt für die Teilaspekte des Tischroboters Lösungs- und Realisierungsvorschläge. Im Anschluss an die Konzeptionierung werden verschiedene Punkte der prototypischen Umsetzung des Systems dargestellt.

Dieser Bachelorarbeit ist eine CD-ROM beigelegt, die den Quellcode der entwickelten Programme enthält, welche die Ansteuerung der verwendeten Komponenten des Systems ermöglichen und erläutern.

Abstract

In dieser Bachelorarbeit wurde ein prototypischer Tischroboter für den Einsatz im Bereich des Ambient Assisted Livings konzeptioniert. Innerhalb eines festgelegten Budgets wurden die technische Umsetzbarkeit und die Eignung des Systems als Alternative zum im Telematik-Studiengang der Technische Hochschule Wildau verwendeten NAO-Roboter untersucht. Dabei wurde bewusst auf die Möglichkeit einer humanoiden Fortbewegung verzichtet.

Das entwickelte Konzept basiert auf den in dieser Arbeit festgelegten Anforderungen an einen Tischroboter. Es umfasst die benötigten Hardwarekomponenten, wobei als Grundlage des Systems ein Jetson Nano Einplatinencomputer von Nvidia verwendet wird. Weiterhin wurde eine Softwarearchitektur erstellt, welche Front-End, Back-End und angeschlossene Hardwarekomponenten miteinander verknüpft. Im Front-End wird das Framework Electron.js verwendet, im Back-End kommt das Robot Operating System (ROS) in Verbindung mit der Key-Value-Datenbank Redis zum Einsatz. Für die Steuerung weiterer angeschlossener Hardwarekomponenten und der Energieversorgung werden Arduino Nano Boards eingesetzt.

Die konzipierte Architektur des Gesamtsystems ist auf eine funktionale Erweiterbarkeit ausgelegt, um für die Erforschung und Realisierung von vielfältigen Anwendungsfällen in Folgeprojekten geeignet zu sein.

In einer anschließenden Testphase wurde ein Prototyp entwickelt, mit dessen Hilfe die konzipierte Systemarchitektur umgesetzt und erfolgreich getestet wurde. Die Ergebnisse dieser Bachelorarbeit können somit als prototypische Grundlage für eine Alternative zum NAO-Roboter, aber auch als technische Basis für die Entwicklung anderer Roboter verwendet werden.

This thesis presents the concept for a prototypical table robot for use in the field of Ambient Assisted Living. The technical feasibility, within a given budget, and the suitability of this concept as an alternative to the robot NAO, which is used in the Telematics course of the Technical University of Applied Science Wildau, has been investigated. The aspect of humanoid locomotion was deliberately excluded from the scope of this thesis.

The developed concept is based on a set of requirements for a table robot that are defined as part of this thesis. The concept includes the required hardware components and uses a Jetson Nano single-board computer from Nvidia as core of the system. In addition, a software architecture has been created in order to connect front-end, back-end and hardware components. The front-end uses the Electron.js framework, in the back-end the Robot Operating System (ROS) is used, as well as the key-value-database Redis. To control further hardware and the energy supplement, Arduino Nano Boards are used.

The resulting architecture of the entire system is designed to be functionally extensible. Therefore, it is suitable for the investigation and realization of versatile use cases in future follow-up projects.

As part of the following testing phase, a prototype of the system has been developed. The prototype allowed successful realization and testing of the conceptualized architecture. Thus, the results of this thesis can serve both as a prototypical basis for an alternative to the robot NAO, and more broadly also as a technical foundation for the development of other robots.

Inhaltsverzeichnis

Danksagung	4
Hinweise zum Lesen der Arbeit	5
1. Einleitung.....	6
1.1 Motivation.....	6
1.2 Zielsetzung.....	8
1.3 Abgrenzung.....	8
1.4 Gliederung der Arbeit.....	9
2. Analyse des aktuellen Zustands	10
2.1 Roboter an der TH Wildau	10
2.1.1 NIBO	11
2.1.2 NAO	12
2.1.3 Pepper	15
2.2 RoboticLab der TH Wildau.....	17
3. Analyse der Anforderungen an den Tischroboter.....	19
3.1 Personae.....	19
3.1.1 Persona: Karin Schröter – Seniorin	20
3.1.2 Persona: Mario Freud – Entwickler	23
3.2 User Stories	25
3.2.1 Anwendungsfälle für die Unterstützung im Alltag.....	25
3.2.2 Anwendungsfälle in der Gebäudeautomation	26
3.2.3 Anwendungsfälle für die Freizeitgestaltung.....	27
3.2.4 Allgemeine Anwendungsfälle.....	27
3.2.5 Anwendungsfälle in der Entwicklung	28
3.3 Definition der Anforderungen.....	29
3.3.1 Anforderungen an die Hardware des Roboters	29
3.3.2 Anforderungen an die Software des Roboters	31
3.3.3 Anforderungen an das Design des Roboters	32
3.3.4 Tabellarische Übersicht der Anforderungen	33

4. Evaluation der Komponenten für das Gesamtsystem	36
4.1 Auswahl der Hardware	36
4.1.1 Allgemeines Vorgehen zur Auswahl der Hardware.....	38
4.1.2 Zentrale Recheneinheit	39
4.1.3 Bildschirm.....	46
4.1.4 Geräuschwahrnehmung.....	48
4.1.5 Audioausgabe.....	49
4.1.6 Aufnahme von Bildern.....	51
4.1.7 Temperaturregulierung	52
4.1.8 Energieversorgung.....	53
4.1.9 Einschalter	54
4.1.10 Augen	54
4.1.11 Bewegungsmöglichkeiten	56
4.1.12 Elektronische Schaltungen.....	57
4.2 Auswahl der Software	58
4.3 Finanzieller Zwischenstand	59
5. Konzept des Gesamtsystems	60
5.1 Hardwaretechnische Systemaspekte.....	60
5.1.1 Konnektivität und Anschlüsse des Roboters.....	62
5.1.2 Einschalter	64
5.1.3 Berührungssensoren.....	66
5.1.4 Temperaturregulierung und -messung.....	67
5.1.5 Bewegungsmöglichkeiten	69
5.1.6 Augen.....	70
5.1.7 Lautsprechersystem	72
5.1.8 Energieversorgung.....	74
5.1.9 Architektur des Systems	81
5.2 Softwaretechnische Systemaspekte.....	82
5.2.1 Back-End – Roboter Operating System	83
5.2.2 Front-End – Grafische Benutzeroberfläche mit Electron.....	86
5.2.3 Hardware – Kommunikation mit den Arduinos	88
5.2.4 HelloWorld Programme	89
5.2.5 Architektur der Software des Systems	94

6. Aktueller Stand der prototypischen Umsetzung.....	95
6.1 Energieversorgung.....	95
6.2 Kommunikation zwischen Jetson Nano und Arduino Nano.....	97
6.3 Hello World Programme.....	98
6.4 Grafische Anwendung und ROS Back-End.....	98
6.5 Lautsprecher und Audioausgabe über das ReSpeaker-Modul	101
6.6 Kapazitive Sensoren.....	102
6.7 Finanzieller Stand der prototypischen Umsetzung.....	104
6.8 Abschlussbetrachtung der definierten Anforderungen.....	105
7. Abschlussbetrachtung.....	106
7.1 Fazit	106
7.2 Ausblick.....	109
7.2.1 Gehäuse des Tischroboters.....	109
7.2.2 Basis für die Weiterentwicklung.....	110
7.2.3 Entwicklung eines Akkubetrieb.....	110
7.2.4 Hybride Rechnerarchitektur	111
8. Anhang	113
8.1 Beigelegte CD-ROM	113
8.2 Deutsche Distributoren.....	114
8.3 Details zu den evaluierten Einplatinencomputern.....	115
8.3.1 Allgemeine Merkmale	115
8.3.2 Leistungsspezifikationen.....	115
8.3.3 Ergebnisse von Benchmark-Tests	116
8.3.4 Anschlüsse und Konnektivität	117
8.4 Technische Merkmale des NAO-Roboters	118
8.5 Energieversorgung des Tischroboters mit Netzteil	119
Abkürzungsverzeichnis	120
Glossar	122
Quellenverzeichnis.....	126
Abbildungsverzeichnis.....	131
Tabellenverzeichnis	133
Selbstständigkeitserklärung.....	134

Danksagung

Nachdem die Anfertigung dieser Bachelorarbeit mich nun die letzten vier Monate begleitet hat, möchte ich an dieser Stelle kurz meinen Dank an all diejenigen aussprechen, die mich während dieser Zeit unterstützt und motiviert haben.

Zunächst möchte ich mich bei meiner Betreuerin Prof. Dr. Janett Mohnke bedanken, die mir die Bearbeitung dieses Themas nicht nur zugetraut hat, sondern es quasi für mich freigehalten hat. Sie hat richtungsweisende Ideen in das Projekt eingebracht und frühzeitig dafür gesorgt, dass ich mich auf das Wesentliche konzentriere und nicht das Ziel aus den Augen verliere.

Auch meiner Zweitgutachterin Janine Breßler möchte ich danken. Mit den Erfahrungen, die sie selbst während des Bachelor- und Masterstudiums gemacht hat, konnte sie mir wertvolle Hinweise für das Schreiben der Arbeit geben.

Ein besonderer Dank gilt Vincent Rehlinger, der bereits seit der fünften Klasse einer der wichtigsten Freunde für mich ist. Er hat wesentliche Teile der Arbeit kritisch hinterfragt und durch seine Anmerkungen Ordnung in meine Gedanken gebracht. Auch möchte ich Vincents Mutter Ramona Helbing danken. Sie hat die Arbeit in Rekordtempo gelesen und auch die letzten eingeschlichenen Fehler herausgefiltert.

Ich bedanke mich auch bei meinen Großeltern Karin und Peter Schröter, die nicht nur für einen Abschnitt der Arbeit als Vorlage dienten, sondern ebenfalls aufschlussreiche Gedanken aus einem anderen Blickwinkel eingebracht haben.

Abschließend möchte ich mich herzlichst bei meinen Eltern Sabine Schröter und Steffen Deutscher bedanken. Vor allem in den letzten Monaten haben sie mir ein extrem unkompliziertes Leben ermöglicht, damit ich mich auf die wichtigen Dinge konzentrieren konnte. Sie haben von Anfang an viele Einfälle eingebracht und etliche Blicke auf die Arbeit geworfen. Ich konnte mit ihnen stundenlang über die Umsetzung bestimmter Aspekte des Projekts diskutieren, um anschließend den richtigen Weg zu wählen.

Dankeschön!

Berlin, den 27.08.2019

Valentin Schröter

Hinweise zum Lesen der Arbeit

Im Folgenden werden Hinweise zu den in der Arbeit verwendeten Konventionen genannt.

Die Angabe von Quellen erfolgt in der Form von zwei eckigen Klammern, die mit Hilfe einer Nummer auf die zugehörige Quelle im Quellenverzeichnis verweisen.

Quellen, die sich auf einen ganzen Textabschnitt beziehen, werden jeweils am Ende des Abschnitts in genannter Form angegeben. [0]

Quellen, die sich auf spezifische Informationen innerhalb eines Textabschnitts beziehen, werden direkt hinter dem zugehörigen Satz in gleicher Form angegeben [0].

Bei Abbildungen wird die Quelle direkt im Anschluss an die Bildunterschrift eingefügt. Bei selbsterstellten Abbildungen wird keine Quelle im Beschreibungstext angegeben.

Wörtliche Zitate werden durch „*Anführungszeichen und kursive Schrift*“ gekennzeichnet. Die Quellenangabe erfolgt direkt nach dem Zitat.

Innerhalb von Zitaten werden Auslassungen durch drei Punkte in eckigen Klammern [...] und grammatikalische Änderungen durch eckige Klammern kenntlich gemacht.

Weiterführende Informationen zu bestimmten Textpassagen werden mit einer Fußnote direkt erwähnt, gegebenenfalls erfolgt ein Verweis auf eine Online-Quelle.

Firmennamen und Eigennamen werden bei erster Erwähnung innerhalb eines Abschnitts *kursiv* geschrieben. Anschließend werden sie normal im Text verwendet.

Für eine bessere Übersichtlichkeit werden bei längeren Textpassagen innerhalb eines zusammenhängenden Abschnitts wichtige Schlagworte eines Textabschnitts **fett** markiert. Die Kürzel von Anforderungen (**R000**) werden ebenfalls einheitlich **fett** geschrieben.

Abkürzungen, die laut Duden offiziell anerkannt sind, werden nicht ausgeschrieben und ebenfalls nicht im Abkürzungsverzeichnis aufgeführt.

Abkürzungen, die in der Informatik bzw. in der Telematik bekannt sind, werden nicht ausgeschrieben, sind aber im Abkürzungsverzeichnis erläutert.

Andere, spezifische Abkürzungen werden bei erstmaliger Nennung ausgeschrieben, dahinter wird in Klammern die zugehörige Abkürzung genannt. Im weiteren Verlauf der Arbeit wird die genannte Abkürzung weiterverwendet. Eine Erklärung zu jeder Abkürzung befindet sich im Abkürzungsverzeichnis.

Zusätzlich zum Abkürzungsverzeichnis befindet sich im Anhang ein Glossar, in welchem erklärungsbedürftige Wörter erläutert werden. Wörter, die im Glossar zu finden sind, werden bei erster Nennung innerhalb eines Kapitels mit einem hochgestellten Punkt* direkt nach dem Wort gekennzeichnet.

1. Einleitung

1.1 Motivation

„We have succeeded in making people live longer.

Now we need to make them live better.” – Sebastian Thrun [1]

Der demografische Wandel in Deutschland ist bereits heutzutage eine gesellschaftsverändernde Gegebenheit, die sich in Zukunft weiter verstärken wird. Die Zahl der Menschen jüngeren Alters sinkt, die Zahl der älteren Menschen nimmt stetig zu. Besonders die nächsten 20 Jahre werden dieses Verhältnis deutlich beeinflussen, da die geburtenstarken Jahrgänge des zwanzigsten Jahrhunderts in das rentenfähige Alter kommen. [2]

Während 1970 etwa 25 Menschen im Rentenalter 100 Menschen im erwerbstätigen Alter gegenüberstanden, liegt dieser Altersquotient* heutzutage (2019) bei ca. 37 und wird sich bis zum Jahr 2040 voraussichtlich auf ca. 58 erhöhen. Diese Entwicklung impliziert, dass die Belastung der jüngeren Generationen signifikant ansteigen wird, sowohl finanziell durch Renten- und Pflegeversicherungsbeiträge als auch aufwandstechnisch durch Pflegeleistungen oder unterstützende Dienstleistungen im Haushalt. Durchgeführte Studien zeigen, dass neben den wachsenden Kosten im Pflegebereich vor allem die Sorge vor einem stetig steigenden Fachkräftemangel besteht. [3, 4]

Ein Ansatz, den Personalmangel in der Pflege zu kompensieren, besteht in der Nutzung von innovativen, technischen Systemen. Das sogenannte Ambient Assisted Living (kurz AAL, dt. Übersetzung: Alltagstaugliche Assistenzlösungen für ein selbstbestimmtes Leben) ist ein Teilgebiet des SmartHomes bzw. der Gebäudeautomation und steht für den Einsatz von Konzepten, Produkten und Dienstleistungen, um Alltagstätigkeiten zu erleichtern.

Der höhere Lebensstandard und die Entwicklungen im Gesundheitswesen haben in den letzten Jahrzehnten erreicht, dass Menschen immer länger leben. Eine essenzielle Aufgabe unserer Gesellschaft besteht nun darin, Möglichkeiten zu erforschen, welche die Lebensqualität im Alter steigern können. Als Telematiker sucht man diese Möglichkeiten genau in diesen innovativen Technologien, speziell in den Bereichen der Robotik, des SmartHomes und insbesondere des Ambient Assisted Livings. Denn: *„Ältere Menschen sind offen gegenüber Digitalisierung; sie sind neugierig und wollen zu aktiven und kompetenten Akteuren der digitalen Gesellschaft werden. Die Gestaltung der Digitalisierung muss diese Neugier fördern [...] [und] souveräne digitale Akteure hervorbringen, die kompetent die Vielfalt digitaler Technologien nutzen, um selbstbestimmt am gesellschaftlichen Leben teilzunehmen.“* [5, S. 7]

Im sechsten Bachelor-Semester des Telematik-Studiengangs an der Technischen Hochschule Wildau erhalten die Studierenden die Gelegenheit, im Rahmen des Moduls *Eingebettet Systeme und Robotik* Projekte mit einem NAO-Roboter umzusetzen. Dieser humanoide Roboter der Firma *SoftBank Robotics* bietet einige Möglichkeiten: Er besitzt Arme und Beine, kann aufrecht laufen, sehen, sprechen und gesprochene Sprache verstehen. Gleichzeitig ist der NAO mit seinem Aussehen ein Publikumsmagnet bei Veranstaltungen wie dem Hochschulinformationstag und den Schülerlaboren des RoboticLabs der TH Wildau.

Im vergangenen Sommersemester 2019 lag der Fokus der studentischen Projekte auf der Umsetzung von Anwendungsfällen, in welchen die Roboter den Alltag bereichern und speziell ältere Menschen unterstützen können. Zum Abschluss des praktischen Moduls konnten die NAOs unter anderem empfangene Chatnachrichten auf Wunsch laut vorlesen, Menschen über Denkspiele und Quizfragen geistig und als Fitnesstrainer auch körperlich aktivieren oder über Sprachsteuerung das aktuelle Fernsehprogramm vermitteln.

Im Anschluss an die Projekte mit dem Roboter folgten in den letzten Jahren einige Abschlussarbeiten, die den NAO als zentrale Komponente behandelt haben. In vielen dieser Projekte wird der NAO als Bindeglied zum Menschen verwendet. Trotz seiner theoretisch umfangreichen Möglichkeiten wurden dabei seine Schwächen offensichtlich: Seine Fortbewegung ist instabil, die Spracheingabe funktioniert nur begrenzt, die zuverlässigen Interaktionsmöglichkeiten beschränken sich auf das Betätigen von Drucksensoren an den Gliedmaßen und am Kopf. Zusätzlich dazu ist der Preis von mindestens 5000 € ein Ausschlusskriterium für einen realen und breiteren Einsatz in verschiedenen Anwendungsszenarien im Privatbereich.

Ungeachtet der technischen Schwächen besitzt der NAO Roboter eine beeindruckende Wirkung auf den Menschen. Er löst vor allem bei Kindern, genauso aber auch bei Erwachsenen und älteren Menschen spielerische Neugier und Freude bei Interaktionen mit dem Roboter aus. Dieselbe Wirkung ist beim Pepper-Roboter, dem großen Bruder des NAOs, zu beobachten.

Die Anwendungsfälle, die bisher mit einem NAO umgesetzt wurden, die emotionale Wirkung auf den Menschen und die Schwächen des Roboters sind zusammen mit den Ansätzen und Zielen des AAL Anreiz für diese Bachelorarbeit. Als zentrale Frage soll untersucht werden, ob es möglich ist, einen sozialen Roboter zu konzipieren, der für die Unterstützung des Alltags geeignet ist und gleichzeitig in einem preislich erschwinglichen Rahmen liegt. Damit soll eine Alternative zum NAO-Roboter geschaffen werden, welche zum einen qualitativer und flexibler und zum anderen günstiger ist, um für einen realen Einsatz im AAL geeignet zu sein.

1.2 Zielsetzung

Im Telematik-Studiengang der TH Wildau existieren zahlreiche Ideen für die Umsetzung von Anwendungsfällen aus der Robotik und des AAL. Doch die Möglichkeiten des NAOs wurden über die letzten Jahre größtenteils ausgeschöpft, sodass mit der vorliegenden Arbeit eine neue Basis für weitere Forschungsprojekte geschaffen werden soll.

Ziel des Projekts ist die Entwicklung eines Roboters, welcher für den Einsatz im Ambient Assisted Living geeignet ist. Für diesen Zweck wird ein Tischroboter konzipiert, der den Fokus auf Interaktionsmöglichkeiten mit Menschen legt. Aufgrund der damit verbundenen Komplexität und Instabilität wird dafür bewusst auf eine humanoide Fortbewegung, wie sie beim NAO vorhanden ist, verzichtet.

Als Ergebnis der Bachelorarbeit soll zunächst evaluiert werden, inwiefern die Umsetzung eines solchen Tischroboters als Alternative zum NAO möglich und praktisch umsetzbar ist. Dabei liegt der Fokus auf der Konzeptionierung der hardware- und softwaretechnischen Systemarchitektur, welche die analysierten Anforderungen berücksichtigt. Am Ende der Arbeit soll ein funktionsfähiger Prototyp entstehen, welcher alle notwendigen Hardwarekomponenten miteinander verknüpft und mit passender Schnittstellensoftware zugänglich macht.

Kurzfristig soll als Resultat der Bachelorarbeit die Grundlage einer Forschungs- und Entwicklungsplattform entstehen, die in Folgeprojekten erweitert werden kann, und mit der Studierende in Praktika und Abschlussarbeiten die Möglichkeiten eines Tischroboters evaluieren und konkrete Anwendungsfälle umsetzen können. Langfristig gesehen, soll der Roboter als interaktive Schnittstelle zum Menschen bzw. zwischen Menschen und SmartHomes eingesetzt werden und somit Anwendungsmöglichkeiten für Senioren, aber auch für Kinder und Familien bieten können. Gerade für erstere besteht die Vision, dass der Roboter durch verschiedene Dienste den Alltag im Alter unterstützt, bereichert und somit ein längeres, selbstständiges Leben ermöglicht. Für Familien soll der Roboter das Potenzial bieten, ein soziales Mitglied im Haushalt zu werden und den Alltag zu verbessern.

1.3 Abgrenzung

Die Bachelorarbeit befasst sich mit der Konzeptionierung und dem prototypischen Bau eines Tischroboters, der im Bereich des AAL eingesetzt werden kann. Für die Analyse der technischen Anforderungen werden mögliche Anwendungsfälle des Roboters genauer betrachtet, jedoch ist die Umsetzung konkreterer Anwendungsfälle für spätere Einsatzszenarien kein Bestandteil dieser Bachelorarbeit.

Der prototypische Bau des Roboters umfasst die Auswahl und Verknüpfung der notwendigen Hardware und die Entwicklung passender Schnittstellensoftware, sodass ein funktionsfähiges System entsteht. Die Anforderungen an ein Gehäusedesign werden in dieser Arbeit zwar bedacht, jedoch wird das konkrete Aussehen des Roboters nicht entworfen.

1.4 Gliederung der Arbeit

Im Folgenden wird der inhaltliche Aufbau der Arbeit erläutert, welcher gleichzeitig die Herangehensweise an das Projekt widerspiegelt.

Der erste Teil der Arbeit beschreibt die Ausgangssituation des Projekts anhand der Roboter, die an der TH Wildau vorhanden sind und im Studiengang Telematik genutzt werden. Relevant ist dabei vor allem der NAO-Roboter, zu dem mit diesem Projekt die Basis für eine Alternative geschaffen werden soll. Außerdem wird das RoboticLab als Arbeitsgruppe der TH Wildau vorgestellt, welche Themen aus der Robotik erforscht und sie in Projekten umsetzt.

Als Grundlage für den weiteren Verlauf der Arbeit folgt dann die Analyse der Anforderungen, die an einen Tischroboter gestellt werden. Diese erfolgt zunächst durch die Beschreibung von zwei exemplarischen Szenarien, die den potenziellen Einsatz eines solchen Roboters darstellen. Anschließend werden daraus die möglichen Anwendungsfälle zusammengefasst, woraufhin im Anschluss die konkreten Anforderungen für diese Arbeit abgeleitet werden.

Vor der Konzepterstellung werden im nächsten Schritt auf Basis der aufgestellten Anforderungen Komponenten für das Gesamtsystem evaluiert, die für eine Verwendung infrage kommen. Diese Evaluation wird sowohl für die notwendige Hardware als auch für die verwendete Software durchgeführt. Die Evaluation soll in einer Vorauswahl an Hauptkomponenten resultieren.

Daraufhin kann das Konzept für ein Gesamtsystem erstellt werden, bei dem ebenfalls nach Hardware und Software unterteilt wird. Basierend auf den zuvor evaluierten Komponenten werden während der Konzeptionierung für jeden Teilaspekt des Systems Lösungsvorschläge für eine praktische Umsetzung ausgearbeitet. Als Ergebnis aller Teilaspekte soll eine vollständige Architektur des Systems entstehen.

Im Anschluss an die Konzepterstellung erfolgt die prototypische Umsetzung des Systems. Dabei wird der aktuelle Stand beschrieben und auf verschiedene Aspekte der Realisierung eingegangen.

Im letzten Kapitel erfolgt eine Abschlussbetrachtung, in welcher die Ergebnisse der Arbeit zusammengefasst werden und außerdem ein Ausblick auf die zukünftige Entwicklung und offene Themen gegeben wird.

2. Analyse des aktuellen Zustands

Dieses Kapitel beschreibt den aktuellen Zustand und damit die Ausgangssituation für diese Arbeit.

Zunächst werden im ersten Abschnitt die an der TH Wildau vorhandenen und im Studiengang Telematik verwendeten Roboter dargestellt. Dazu gehören der NIBO-, der NAO- und der Pepper-Roboter. Vor allem die Betrachtung des NAOs ist relevant, da der in dieser Arbeit konzipierte Tischroboter eine potenzielle Alternative zum NAO-Roboter darstellen soll.

Darauffolgend wird das RoboticLab der TH Wildau als Auftraggeber für diese Arbeit vorgestellt.

2.1 Roboter an der TH Wildau

Dieser Abschnitt stellt die drei im Studiengang Telematik verwendeten Roboter vor. Dabei werden ihre Stärken und Schwächen in Bezug auf die Eignung als interaktive Schnittstelle zum Menschen dargestellt.

Während der Lehre wird im fünften und sechsten Bachelor-Semester der Telematik das Modul *Eingebettete Systeme & Robotik* angeboten, in welchem die Studierenden zunächst Projekte mit einem NIBO und anschließend mit einem NAO umsetzen können.

Neben diesen beiden Robotern wird in der Hochschulbibliothek ein Pepper Roboter als autonomes Assistenzsystem eingesetzt.

2.1.1 NIBO

Der NIBO ist ein Tischroboter der Firma *nicaii systems*¹. Im Telematik-Studiengang der TH Wildau wird der NIBO im fünften Semester eingesetzt, um den Studierenden das hardwarenahe Programmieren von eingebetteten Systemen* in der Programmiersprache C zu vermitteln. In Abbildung 1 werden die vorhandenen Sensoren und Aktoren des Roboters dargestellt. Diese können miteinander verknüpft werden, um dem NIBO ein autonomes Fahren mit Wahrnehmung der aktuellen Umgebung zu ermöglichen und dabei Hindernissen auszuweichen. [6]

NIBO 2

Länge: 136 mm
Breite: 131 mm
Gewicht: 544 g

Abbildung 1: Übersicht eines NIBO-Roboters [6]

Die Stärken des NIBOs liegen in seiner Simplizität: Die Möglichkeiten des Roboters sind durch die vorhandene Hardware, die Qualität der Sensorwerte und die gegebene Rechenleistung der Mikrocontroller beschränkt. Die Aufgabe der Studierenden besteht darin, mit den vorgegebenen Mitteln Projekte umzusetzen, um dabei die Begrenzung eingebetteter Systeme auf ihren Einsatzzweck nachvollziehen zu können.

Der NIBO wird der Vollständigkeit halber aufgeführt, da er als Tischroboter Teil des Telematik-Studiums ist. Jedoch ist er durch die begrenzte Rechenleistung und seine eingeschränkten Interaktionsmöglichkeiten mit Menschen für den Einsatz im AAL ungeeignet.

¹ Offizielle Website: <http://www.nicaii-systems.com/de/> (abgerufen am 01.08.2019)

2.1.2 NAO

Im sechsten Semester des Telematik-Studiengangs erhalten die Studierenden die Möglichkeit, selbstgewählte Projekte mit einem NAO-Roboter der Firma *SoftBank Robotics*² umzusetzen. Anhand des NAOs werden Kenntnisse in der Programmiersprache C++ vermittelt und gleichzeitig Möglichkeiten und Herausforderungen bei der Arbeit mit humanoiden Robotern verdeutlicht.

Der NAO (Abbildung 2) ist ein ca. 58cm großer, humanoider Roboter, welcher mit Armen und Beinen über 25 Freiheitsgrade der Bewegung verfügt. Daraus resultiert ein breites Spektrum an **Bewegungsmöglichkeiten**, sodass der NAO verschiedene Gesten darstellen, aufrecht laufen und auch selbstständig aufstehen kann. Ebenso sind komplexere Bewegungsabläufe wie tanzähnliche Choreographien umsetzbar. [7]

Zur Wahrnehmung der Umgebung besitzt der NAO zwei **Kameras**, welche die Erkennung von Objekten und Gesichtern ermöglichen. Außerdem befinden sich im Kopf des Roboters vier **Mikrofone**, über die Geräusche und Sprache erfasst und auch deren Ursprungsort bestimmt werden kann. Über zwei im Kopf integrierte **Lautsprecher** kann der NAO ebenfalls Töne und Sprache ausgeben. [8]

Für eine taktile Interaktion mit dem Menschen existieren am Kopf und an den Händen des Roboters insgesamt neun kapazitive **Berührungssensoren**. Zusätzlich ist an den Füßen jeweils ein mechanischer **Drucktaster** vorhanden. Um den Roboter ein- und auszuschalten, gibt es einen weiteren Knopf im Brustbereich. Für eine visuelle Rückmeldung besitzt der NAO mehrere LEDs in den Augen, in den Ohren und auf dem Kopf. [8]

Abbildung 2: NAO Roboter [9]

² Offizielle Seite von SoftBank Robotics (ehemals Aldebaran Robotics):
<https://www.softbankrobotics.com/emea/en> (abgerufen am 01.08.2019)

Vom NAO gibt es verschiedene **Versionen**. In den Projektarbeiten des Telematik-Studiums wird die Version NAO V4 verwendet. Dieser besitzt einen Ein-Kern Prozessor und 1 GByte Arbeitsspeicher. In der aktuell neusten Version NAO V6 ist ein performanterer Prozessor und ein größerer Arbeitsspeicher vorhanden, außerdem bietet die Kamera eine höhere Auflösung. Eine vollständige Übersicht der technischen Merkmale ist in Tabelle 28 im Anhang auf Seite 118 zu finden.

Für die Programmierung des NAOs stehen unterschiedliche Methoden zur Verfügung. Das von SoftBank Robotics bereitgestellte Programm *Choregraphe* ermöglicht zunächst einen einfachen Einstieg in die Programmierung des NAOs, indem die grundlegenden Funktionen des Roboters über eine **grafische Oberfläche** miteinander verknüpft werden können. Diese Art der Programmierung nach dem Baukastenprinzip ist auch für Jugendliche und Personen mit wenig Programmiererfahrung geeignet. Choregraphe kann auch für die Erstellung und Simulation von Bewegungsabläufen und die Überwachung eines verbundenen NAO-Roboters verwendet werden. [10]

Um komplexere Projekte umzusetzen, können Programme in verschiedenen **Programmiersprachen** wie C++, Python oder Java entwickelt werden. Die Programme werden anschließend als lokales Modul direkt auf dem NAO oder als Remote-Modul von einem externen PC ausgeführt. Für den Zugriff auf die Hardware des NAOs stehen dafür in den jeweiligen Sprachen Schnittstellen-Pakete zur Verfügung. [11]

Seit der Anschaffung sind die NAOs eine **Besucher-Attraktion** in den Schülerlaboren des RoboticLabs und auf Veranstaltungen wie dem Hochschulinformationstag und der Kinderuniversität. Das kindliche Aussehen sowie die nette Stimme sorgen sowohl bei Kindern als auch bei Erwachsenen für Spaß bei der Interaktion mit dem Roboter.

Die vorhandenen Möglichkeiten des NAOs, die freie Programmierbarkeit und die Wirkung auf Personen machen den Roboter grundsätzlich als **interaktive Schnittstelle zum Menschen** nutzbar. Im nächsten Abschnitt 2.2 dieses Kapitels werden einige Projekte genannt, in denen ein NAO für eine solche Schnittstelle genutzt wurde. Jedoch wird der NAO aus folgenden Gründen für einen großflächigen Einsatz im AAL als ungeeignet bewertet³:

Die lokale **Spracherkennung** funktioniert nur zuverlässig, wenn zwischen wenigen, vordefinierten Wörtern unterschieden werden muss und der Mensch laut und deutlich spricht⁴. Eine flüssige Erkennung von gesprochener Sprache ist ohne weiteres nicht möglich. Dadurch beschränkt sich die zuverlässige Interaktion mit dem NAO auf das Betätigen der Berührungssensoren.

³ Die Bewertung basiert auf den technischen Merkmalen, aber auch auf persönlichen und allgemeinen Erfahrungen, die im Umgang mit den NAO an der TH Wildau gemacht wurden.

⁴ Während einer Projektarbeit mit dem NAO wurde für das Testen der Spracherkennung eine Aufnahme angefertigt, welche gezeigt hat, dass die Mikrofone im Kopf durch das Rauschen eines Lüfters signifikant beeinträchtigt werden.

Die zur Verfügung stehende Schnittstellensoftware für den NAO erlaubt das Verknüpfen von existierenden Möglichkeiten. Eine **Erweiterung** mit neuen Softwaremodulen ist aufgrund des abgeschlossenen Betriebssystems und fehlender Rechte nicht bzw. nur schwierig möglich. Teilweise ist daher die Benutzung von modernerer Software nicht vollständig möglich.

Der NAO besitzt die Fähigkeit zu laufen. Die **Fortbewegung** ist jedoch nicht sehr stabil, sodass bei schnellerer Bewegung das Risiko eines Umkippens besteht. Trotz der ebenfalls vorhandenen Fähigkeit, selbstständig wieder aufzustehen, bringt die langsame und unsichere Bewegung des NAOs keinen Mehrwert für den Einsatz im AAL.

Die **Akkulaufzeit** des NAOs beträgt laut Herstellerangaben ca. 90 Minuten, im aktiven Betrieb lediglich 60 Minuten. Zum Aufladen muss der Benutzer des NAOs selbst das vorgesehene Kabel anstecken. Insofern wäre bei einem dauerhaften Betrieb als interaktive Schnittstelle entweder ein ständiges Aufladen notwendig, oder aber eine permanente Verbindung des NAOs mit einem Netzteil, was wiederum die Möglichkeit der gegebenen Bewegungsfreiheit einschränkt.

Die **Kosten** des NAOs liegen zurzeit bei ca. 5000 €⁵. Einrichtungen wie Hochschulen und Universitäten sind zwecks Forschung und Ausbildung bereit, diesen Preis zu zahlen. Für Privatpersonen sind die Anschaffungskosten jedoch deutlich zu hoch.

Für die Lehre bietet der NAO dennoch einen sehr guten Einstieg, indem er den Studierenden Möglichkeiten und Schwierigkeiten der humanoiden Robotik vermittelt.

⁵ Laut Wikipedia: https://de.wikipedia.org/wiki/Nao_%28Roboter%29 (abgerufen am 01.08.2019), genaue Preise werden nach der Kontaktaufnahme mit SoftBank Robotics bzw. mit Distributoren des NAOs definiert.

2.1.3 Pepper

Pepper (Abbildung 3) stammt ebenfalls von der Firma SoftBank Robotics und kann als großer Bruder des NAOs gesehen werden. An der TH Wildau wird ein Pepper Roboter in der Hochschulbibliothek als autonomer Assistenzroboter eingesetzt, wie im nächsten Abschnitt 2.2 genauer erläutert wird.

Pepper ist ca. 120cm groß und bewegt sich anders als der NAO nicht auf Beinen, sondern über ein omnidirektionales* Rollensystem fort. Die autonome Bewegung wird durch Infrarot- und Ultraschallsensoren, 2D-Kameras und Tiefenkameras unterstützt. Mit 20 Freiheitsgraden in Armen, Kopf und Rumpf werden Pepper flexible Bewegungen ermöglicht. [12]

Ähnlich wie der NAO verfügt der Roboter über Mikrofone und Lautsprecher für Spracherkennung und -ausgabe. Für die Interaktion mit Menschen besitzt der Roboter zusätzlich zu Berührungssensoren an Kopf, Armen und dem Bewegungsapparat ein integriertes Tablet, welches neben der vielfältigen Anzeige von Informationen eine bekannte Interaktionsmöglichkeit erlaubt, ohne auf eine Verständigung über Sprachsteuerung oder die Berührungssensoren zurückgreifen zu müssen. [12]

Abbildung 3: Pepper Roboter [12]

Pepper ist auf **humanoide Interaktion** optimiert. Seine Größe und das freundliche Gesicht sind einer kindlichen Gestaltung nachempfunden, welche eine hohe Akzeptanz bei Nutzern auslösen soll. Das Design erlaubt fließende und natürlich aussehende Bewegungen der Arme und des Kopfes, wodurch ein ansprechender Ausdruck von Gestik ermöglicht wird. Laut SoftBank Robotics ist er der erste Roboter, der nicht nur Gesichter, sondern auch grundlegende Emotionen der Gesprächspartner erkennen und unterscheiden kann. Im autonomen Modus ist Pepper in der Lage, Gesichter im Raum mit seinem Blick zu verfolgen,

wodurch das Gefühl erzeugt wird, dass der Roboter die Präsenz von Personen aktiv wahrnimmt und darauf reagiert. [12]

Pepper und NAO basieren auf dem **gleichen Betriebssystem** und werden dementsprechend ähnlich programmiert. Auch Pepper erlaubt einen einfachen Einstieg über die Möglichkeit der grafischen Programmierung mithilfe des Programms Choregraphe. Für umfangreichere Projekte finden vor allem die Sprachen Python und C++ Verwendung.

Im Roboter ist ein **Akku** enthalten, der einen autonomen Betrieb für über 10 Stunden⁶ ermöglicht. Zum Aufladen gibt es einerseits die Möglichkeit, ein Ladekabel manuell anzustecken. Darüber hinaus gibt es auch eine Ladeschale, die Pepper zum Laden selbstständig finden und befahren kann.

Dass Pepper als **interaktive Schnittstelle** zum Menschen geeignet ist, zeigt der erfolgreiche Einsatz des Roboters in der Hochschulbibliothek der TH Wildau⁷. Jedoch sprechen folgende Aspekte gegen einen großflächigen Einsatz im AAL:

Die lokale **Spracherkennung** zeigt dieselben Probleme wie beim NAO und funktioniert nur zuverlässig, wenn zwischen wenigen, vordefinierten Wörtern unterschieden werden muss und der Mensch laut und deutlich spricht⁸. Als Ausweg bietet Pepper allerdings den Vorteil, dass eine zuverlässige Interaktion über das Tablet möglich ist.

Pepper ist durch seine Größe und die Fortbewegung über Rollen für einen Einsatz in großflächigen, ebenen Räumen, wie beispielsweise auf Messen, in Bibliotheken, in Verkaufsläden oder in Foyers geeignet. Beim Einsatz in Eigenheimen wäre die **Bewegungsfreiheit** durch Hindernisse, Türen, Schwellen und Etagen stark **eingeschränkt**, wodurch das Potenzial des Roboters unzureichend genutzt werden würde.

Zur Veröffentlichung von Pepper wurde der Roboter in der ersten Verkaufsphase für unter 2.000 € angeboten [13]. Dieser niedrige Preis diente einer schnellen Verbreitung und dem Sammeln von praktischen Erfahrungen. Mit dem weltweiten Verkauf liegen die **Anschaffungskosten** des Roboters mittlerweile bei ca. 20.000 €⁹. Aufgrund des hohen Preises ist Pepper vor allem ein Produkt für Unternehmen, Forschungseinrichtungen und andere größere Institutionen. Für eine großflächige Verbreitung im Privatbereich wird der Preis jedoch als deutlich zu hoch eingestuft.

Für die **Reparatur** eines defekten Roboters muss dieser mit einem Spezialtransport nach Frankreich versendet werden. Diese Prozedur ist nicht nur ein finanzieller, sondern auch ein zeitlicher Aufwand und dementsprechend für Privatpersonen unzumutbar.

⁶ Dabei handelt es sich um einen Erfahrungswert aus dem Einsatz in der Bibliothek der TH Wildau. Die genaue Akkulaufzeit schwankt je nach Aktivität des Roboters.

⁷ Siehe nächsten Abschnitt 2.2 für Details zum Einsatz des Pepper Roboters in der Bibliothek der TH Wildau.

⁸ Pepper besitzt das gleiche Problem wie der NAO, dass die Mikrofone im Kopf durch das Rauschen von Lüftern beeinträchtigt werden.

⁹ Preis von <https://www.generationrobots.com/de/402422-humanoider-roboter-pepper.html> (abgerufen am 01.08.2019).

2.2 RoboticLab der TH Wildau

„Das RoboticLab der Telematik ist Teil der Forschungsgruppe für mobile Informations- und Assistenzsysteme an der TH Wildau. Die Vision der Arbeitsgruppe ist die Entwicklung und Erprobung von Szenarien, in denen humanoide Roboter in verschiedenen Bereichen des Alltags zum Einsatz kommen - und zwar nicht, um Bewährtes zu verdrängen, sondern um Lücken zu schließen und zu Lösungsansätzen für aktuelle Probleme der Gesellschaft zu inspirieren. Dabei wird bewusst die Neugier und das Interesse der Menschen an diesen technischen Gefährten ausgenutzt.“ [14]

Unter der Leitung von Prof. Dr. Janett Mohnke wurden im RoboticLab bereits vielfältige Projekte im Bereich der Robotik erarbeitet. Für Mitarbeiter und Studierende der TH Wildau ist das bekannteste Projekt des RoboticLabs der Einsatz eines **Pepper-Roboters namens Wilma** in der Hochschulbibliothek. Der Roboter wird als autonomer Assistenzroboter genutzt, um Besuchern der Bibliothek zur Seite zu stehen und in verschiedenen Situationen Hilfe anbieten zu können. Besonders seitdem die Bibliothek für Angehörige der TH Wildau einen durchgängigen Zugang auch außerhalb der regulären Öffnungszeiten ermöglicht, soll Wilma als Ansprechpartnerin aushelfen, wenn kein Bibliotheksangestellter in der Nähe ist. Zum Angebot für Gäste und neue Studierende gehört beispielsweise eine autonome Führung durch das Bibliotheksgebäude. Auch ist Wilma in der Lage, die Funktionsweise des Druckers zu erläutern oder bei der Suche nach einem Buch zu helfen. [15, 16]

In den letzten Jahren wurden ebenfalls einige Abschlussarbeiten angefertigt, in denen humanoide Roboter als interaktive Schnittstelle zum Menschen fungierten. Ein umgesetztes Szenario ist die Verwendung eines **NAO-Roboters als Lesepate** für Grundschulkinder. Dabei soll der Roboter Kinder zum Vorlesen motivieren und beim Lesenlernen unterstützen [17, 18].

Im Modul *Robotik und eingebettete Systeme II* haben Studierende der Telematik ebenfalls bereits diverse **Projekte mit NAO-Robotern** umgesetzt. Im Sommersemester 2019 lag der Fokus dieser Projekte auf Szenarien, in welchen die Roboter den Alltag bereichern und speziell ältere Menschen unterstützen können. Als Ergebnis des Moduls konnte der NAO beispielsweise empfangene Chatnachrichten auf Wunsch laut vorlesen, Menschen über Denkspiele und Quizfragen geistig und als Fitnesstrainer auch körperlich aktivieren oder auch über Sprachsteuerung das aktuelle Fernsehprogramm vermitteln.

In vielen Forschungsprojekten, welche Themen im Bereich der Robotik evaluieren und umsetzen, werden **fertige Komplettlösungen** verwendet. So auch beim Großteil der bisherigen Projekte des RoboticLabs, in denen die vorhandenen NAO- und Pepper-Roboter der TH Wildau als Basis dienen. Im letzten Kapitel 2.1 wurden jedoch die Schwächen dieser Roboter dargestellt, die gegen den Einsatz im Bereich des AAL und als interaktive Schnittstelle zum Menschen sprechen. Vor allem der hohe Preis verhindert eine großflächige Verbreitung der Roboter im Privatbereich.

Darüber hinaus kommt der NAO in vielen Projekten lediglich sitzend zum Einsatz, da die Fähigkeit der Fortbewegung nicht benötigt wird. **Sprachausgabe**, **Interaktion** mit dem Nutzer über Spracheingabe oder Berührungssensoren und die Wahrnehmung der Umgebung über die **Aufnahme** von **Bildern** – diese Eigenschaften bilden im Wesentlichen die Grundlage für die meisten umgesetzten Projekte mit den NAO-Robotern.

Eine aktuelle Projektidee des RoboticLabs ist die Entwicklung und Erprobung von Szenarien, in denen **Service-Roboter** mit Technologien der **Ambient Assisted Livings** verknüpft werden, um ältere Menschen und den Alltag im Eigenheim zu unterstützen. Eine wichtige Grundlage für dieses Projekt ist die Masterarbeit von Janine Breßler, welche die Einbindung von Künstlicher Intelligenz* in das Zusammenspiel aus Robotik und SmartHome-Technologien thematisiert [19]. Ein weiterer Aspekt des Projekts ist die Thematik *Natural Language Processing*, also die Möglichkeit der Interaktion zwischen Mensch und Maschine in Form von natürlich gesprochener Sprache [20].

Die vorliegende Bachelorarbeit ist ebenfalls Teil der Projektidee und **Anstoß für eine Eigenentwicklung** von Service-Robotern, um unabhängig von fertigen Herstellerlösungen zu werden und die existierenden Schwächen und Nachteile der erwerbzbaren Produkte auszugleichen. Es soll somit die Basis geschaffen werden, einen Roboter für die Umsetzung der zahlreichen Ideen des RoboticLab-Teams nutzen, diesen flexibel erweitern und eine hohe Qualität bezüglich der jeweiligen Anwendungsfälle bieten zu können.

3. Analyse der Anforderungen an den Tischroboter

Dieses Kapitel dient der Analyse und der Definition der Anforderungen an das Projekt. Die Anforderungen sind die Grundlage für die Konzepterstellung und die anschließende Umsetzung.

Der Roboter soll für die Umsetzung bestehender Anwendungsfälle des RoboticLabs genutzt werden können. Darüber hinaus ist die funktionale Erweiterbarkeit des Systems eine der wichtigsten Anforderung an das Ergebnis dieses Projekts. Dementsprechend soll eine offene Plattform entstehen, die in den nächsten Semestern für Forschung und Projektarbeiten eingesetzt und weiterentwickelt werden kann.

Für die Analyse der Anforderungen werden in Abschnitt 3.1 zunächst beispielhafte Einsatzszenarien des Roboters anhand von zwei Personae erläutert. Im folgenden Abschnitt 3.2 werden mögliche Anwendungsfälle des Roboters vorgestellt. Diese Anwendungsfälle dienen als Basis für die Ableitung der Anforderungen, werden jedoch im Rahmen dieser Arbeit nicht im Einzelnen umgesetzt. Im letzten Abschnitt 3.3 werden die konkreten Anforderungen an den Tischroboter tabellarisch definiert.

3.1 Personae

Im Folgenden wird der Einsatz des Roboters anhand von zwei Personae erläutert. Die erste Persona betrachtet das Szenario aus Sicht einer Nutzerin, während die zweite Persona den Roboter aus dem Blickwinkel eines Entwicklers betrachtet.

Neben der Nutzerin im Seniorenalter wäre es möglich, weitere Personae zu kreieren, um potenzielle Möglichkeiten für Nutzer beispielsweise aus der Sicht von technikaffinen Familien, Kindern oder auch technikkritischen Personen zu erläutern. Das Ziel dieser Arbeit liegt jedoch nicht in der Umsetzung konkreter Anwendungsfälle, sondern in der Entwicklung eines offenen Systems, das für die Umsetzung verschiedenster Anwendungsfälle geeignet ist. Zusätzliche Szenarien würden keinen weiteren Erkenntnisgewinn bezüglich der Anforderungen an den Tischroboter mit sich bringen. Insofern wird in der Anforderungsanalyse auf eine differenzierte Darstellung aller möglichen Anwendungsfälle und anderer Personae verzichtet.

3.1.1 Persona: Karin Schröter – Seniorin

Abbildung 4: Persona- Karin Schröter¹⁰

Karin Schröter wohnt mit ihrem Mann Peter in einem größeren Haus in einem kleinen Dorf in Sachsen-Anhalt, in der Nähe von Dessau. Sie ist 75 Jahre alt und war vor dem Ruhestand Sekretärin in einem mittelgroßen Unternehmen. Da das Ehepaar für das eigene Haus keine Miete zahlen muss, findet Karin die durchschnittliche Rente ausreichend für ihre alltäglichen Bedürfnisse.

Karin und Peter haben zwei Kinder, die zwar einige Stunden Autofahrt entfernt wohnen, aber trotzdem mehrere Male im Jahr zu Besuch kommen. Karin ist zudem stolz, nicht nur zweifache Oma, sondern mittlerweile auch zweifache Uroma zu sein. Karin ist eine leidenschaftliche und begabte Köchin und Bäckerin, vor allem bei den Besuchen der Familie lässt sie sich stets etwas Besonders einfallen. Außerdem bastelt und näht sie in ihrer Freizeit gerne nützliche und dekorative Dinge für Familie und Freunde. Abgesehen davon kümmert sie sich um Ordnung und Sauberkeit im Haushalt, sie liest sehr viel, ab und zu besucht sie auch gerne Theatervorstellungen.

Karin ist in den letzten Jahren etwas schwerhörig geworden und trägt seit einiger Zeit ein Hörgerät. Genauso wie Peter muss sie täglich mehrere Tabletten nehmen, die größtenteils gegen übliche Zivilisationsbeschwerden einzunehmen sind. Dazu gehören Blutverdünner, Tabletten für die Zucker- und Fettwerte, und Nahrungsergänzungsmittel. Abgesehen davon

¹⁰ Quelle: <https://stock.adobe.com/de/images/just-smile-enthusiastic-senior-couple-sitting-at-cafe-and-using-phone/215988335> (Abgerufen am 15.08.2019)

ist Karin dem Alter entsprechend gesundheitlich fit. Um beweglich zu bleiben, erledigt sie kürzere Wege mit dem Fahrrad und geht einmal wöchentlich zum Sport, der eine Verbindung aus Tanz und koordinativer Bewegung für Senioren ist.

Vor gut 2 Jahren hat sich Karin ein Smartphone gekauft und damit die Kinder und die Enkel überrascht. Hauptsächlich, um über WhatsApp mit Familie und Freunden in Kontakt zu bleiben, nutzt sie es mittlerweile auch für andere Dinge. Beispielsweise guckt sie seit kurzer Zeit vermehrt Videos auf YouTube zum Thema Kochen, Backen und Nähen. Karin steht der neuen Technologie offen gegenüber und freut sich sehr, wenn sie neue Dinge und Funktionen von den Enkeln erklärt bekommt oder sogar von selbst herausfindet und anschließend selbstständig benutzen kann. Dabei ist sie immer wieder beeindruckt, welche Möglichkeiten das Internet mit sich bringt. Trotzdem ist sie grundsätzlich vorsichtig und hat Respekt vor potenziellen Gefahren der Online-Welt, die aufgrund ihrer Unerfahrenheit entstehen könnten. Insofern fragt sie lieber einmal öfter nach, wenn sie etwas nicht versteht oder sie etwas nicht kennt.

Obwohl sich Karin nicht nur körperlich, sondern auch geistig noch recht fit fühlt, vergisst sie ab und zu ein paar Kleinigkeiten. So hat Karin vor kurzem erst eine Wäscheladung einen ganzen Tag lang in der Waschmaschine vergessen und durfte diese anschließend erneut waschen. Auch hat das ältere Ehepaar neulich nach dem Wocheneinkauf die Haustür die ganze Nacht lang offenstehen gelassen, ohne es zu bemerken. Sehr häufig weiß sie auch nicht mehr genau, ob sie bestimmte Tabletten bereits genommen hat oder noch nehmen muss.

Vor kurzem hat Karin im Fernsehen einen Beitrag über *Amazon Echo* gesehen und findet den Gedanken reizvoll, sich durch Technik auf unkomplizierte Art und Weise im Alltag unterstützen zu lassen. Seit der Anschaffung ihres Smartphones ist sie innovativer Technik aufgeschlossener gegenüber und erkennt das darin auch für sie vorhandene Potential. Einen großen Mehrwert würde sie vor allem darin erkennen, wenn ein Amazon Echo in verschiedenen Situationen Bescheid geben und warnen könnte, beispielsweise wenn wieder vergessen wird, die Haustür richtig zu schließen.

Trotzdem steht Karin dem Sprachassistenten kritisch gegenüber. Nicht nur hat sie keine wirkliche Lust, einen schwarzen Zylinder anzusprechen und auf mysteriöse Weise eine Antwort zu erhalten. In der Fernsehsendung wurde auch erklärt, wie das Gerät funktioniert, und dass der Datenschutz dabei kritisch ist. So irrelevant die Gespräche im Haushalt auch sind, hätte sie Angst davor, dass das Gerät ständig zuhört und private Daten an Server übermittelt, die nicht europäischen Datenschutzstandards unterliegen.

In einem Gespräch mit ihrem Enkel hat Karin von einem Tischroboter erfahren, welchen sie sehr interessant findet. Im Vergleich zum Produkt von Amazon erweitert der Roboter die Funktionalität des interaktiven Lautsprechers um ein integriertes Tablet und einen Roboterkopf. Ein paar der beworbenen Funktionen des Roboters adressieren Alltagsprobleme, die Karin gut kennt. So soll der Roboter eine interaktive Erinnerungsfunktion für Aufgaben, Termine und Tabletteneinnahmen bieten, die über eine Personenerkennung automatisch zwischen den verschiedenen Bewohnern des Hauses differenziert. Es wird ebenfalls angeboten, den Roboter als Zentrale für Haustechnik und -Sensorik zu verwenden, wodurch eine Warnung bei offengelassenen Türen und Fenstern durch den Roboter möglich wäre, so

wie Karin es sich vorstellt. Karin würde sich auch freuen, über das größere Display des Roboters YouTube Videos gucken zu können und hofft auch, dass sie den Ton aufgrund ihrer Schwerhörigkeit lauter einstellen könnte, als es bei ihrem Smartphone möglich ist. Für ihre Näh-Tutorials würde sie den Roboter dabei gerne unkompliziert mit ins Bastelzimmer nehmen und akkubetrieben benutzen.

Karin findet es schön zu wissen, dass sich ein deutsches Team hinter der Entwicklung des Tischroboters verbirgt und mögliche anfallende Daten auf deutschen Servern gemäß dem geltenden Datenschutz verarbeitet werden. Darüber hinaus hat Karin erfahren, dass sowohl die Spracherkennung als auch die Personenerkennung des Roboters temporär abgeschaltet werden kann, wenn es gewünscht wird. Für die gebotene Funktionalität ist Karin auch gewillt, etwas mehr Geld als für ein Amazon Echo auszugeben, solange das Preis-Leistungs-Verhältnis stimmt. Ihre Enkel fragen mittlerweile täglich nach, wann sie sich den Tischroboter anschaffen möchte und sie zu Besuch kommen können, um die Möglichkeiten der Technik selbst zu erleben und beim Einrichten zu helfen. Trotzdem möchte sie aufgrund ihrer nicht übermäßigen Rente, dass die Anschaffungskosten im dreistelligen Bereich bleiben.

3.1.2 Persona: Mario Freud – Entwickler

Abbildung 5: Persona - Mario Freud¹¹

Mario Freud ist ein Student der Telematik an der Technischen Hochschule Wildau. Mit seinen 23 Jahren ist er Digital Native[•] und stark an neuen und innovativen Technologien interessiert. Bereits auf dem Gymnasium hat Mario erste, eigene Programme entwickelt und Erfahrungen im Gebiet der Informatik gesammelt.

Bevor Mario mit dem Studium angefangen hat, absolvierte er ein dreimonatiges Pflegepraktikum in einem Krankenhaus in Berlin. Er hatte auch mit dem Gedanken gespielt, ein medizinisches Studium anzufangen, hat sich aber letztendlich aufgrund seiner Informatik-Affinität doch für das Telematik-Studium entschieden. Trotzdem sieht er die Zeit im Krankenhaus als wertvolle Erfahrung in seinem Leben an.

Gerade befindet sich Mario im sechsten Bachelor-Semester und damit kurz vor der Anfertigung seiner Bachelorarbeit. In den letzten Semestern hat er mehrere Pflichtpraktika absolviert, die thematisch größtenteils mit Sensorik und der Programmierung eingebetteter Systeme[•] verbunden waren. Da er sich dementsprechend mit hardwarenaher Programmierung auskennt, wurde er von seiner Professorin auf ein Thema für seine Bachelorarbeit aufmerksam gemacht, welches sich mit der Erweiterung eines Tischroboters beschäftigt. Jener Tischroboter ist das Ergebnis einer Bachelorarbeit des vergangenen Jahres. Der Roboter befindet sich zurzeit im Prototyp-Status und verknüpft grundlegende

¹¹ Quelle: <https://stock.adobe.com/de/images/portrait-of-young-handsome-hipster-man-with-beard-smiling-laughing-looking-at-camera-over-white-background/163639232> (Abgerufen am 15.08.2019)

Hardwarekomponenten miteinander in einem angefertigten Gehäuse. In näherer Zukunft sollen sich jetzt mehrere Abschlussarbeiten mit dem Roboter und der Ergänzung durch weitere Softwaremodule beziehungsweise mit der Umsetzung konkreter Anwendungsfälle beschäftigen.

Da er im aktuellen Semester das Modul *Robotik und eingebettete Systeme* belegt hat, bearbeitet Mario gerade ein Projekt mit einem NAO-Roboter. Er findet die Wirkung der NAOs beeindruckend, da selbst Erwachsene mitgerissen werden und Spaß an der Interaktion mit dem Roboter haben. Trotzdem findet er die Projektarbeit teilweise beschwerlich. Die Kombination von den angebotenen und vorgesehenen Standardfunktionen des Roboters gelingt zwar sehr gut. Aber es kommt mitunter zu undefinierten Abstürzen, für den Test eines Programmes muss das System neugestartet werden und das Hochfahren dauert allgemein sehr lange. Die Spracherkennung funktioniert sehr sporadisch und die Erweiterung um neue, nicht vorgesehene Funktionen ist aufgrund fehlender Rechte und der mittlerweile veralteten Softwareversionen des Roboters ebenfalls nicht trivial. Zusätzlich ist die Bewegungsfähigkeit als eigentliche Besonderheit des NAOs sehr instabil und nicht sinnvoll zu gebrauchen, weswegen auch das Projekt von Mario mit einem stillsitzenden NAO umgesetzt wird.

Mario hat mit dem Tischroboter für seine Bachelorarbeit noch nicht direkt gearbeitet. Bei einer kurzen Vorstellung hat er jedoch bereits gemerkt, wie der Roboter trotz fehlender Gliedmaßen und Fortbewegungsmöglichkeiten eine Freude bei der Interaktion auslöst, was Mario unter anderem in dem sich bewegenden Kopf begründet sieht. Da Mario weiß, dass Spracherkennung und -ausgabe heutzutage noch nicht wirklich perfekt sind, findet er außerdem das integrierte Touch-Display als Anzeige von Informationen und als sichere Interaktionsmöglichkeit wichtig, ähnlich wie es von einem Pepper-Roboter bekannt ist.

Für den Einstieg wünscht er sich eine gute Dokumentation der Benutzung des Roboters, am besten mit HelloWorld* Programmen, die grundlegende Möglichkeiten und den Zugriff auf die existierende Hardware erläutern. Für die Entwicklung und Umsetzung neuer Anwendungsfälle findet Mario nicht nur den uneingeschränkten Zugriff auf das bestehende System wichtig, sondern ebenso die notwendigen Rechte, neue Software zu installieren. Da er sich durch die vergangenen Praktika mit Sensorik auskennt, würde er auch gerne bei Bedarf die Möglichkeit haben, weitere Hardware in das System zu integrieren.

Durch seine Einblicke in die Pflege von älteren und kranken Menschen während des Pflegepraktikums sieht Mario viele Anwendungsfälle für einen Tischroboter vor allem in dieser Zielgruppe. Er freut sich auf die Umsetzung seiner Bachelorarbeit und hofft auf ein gutes und einsetzbares Ergebnis.

3.2 User Stories

Die Anwendungsfälle (engl. **User Stories**) des Roboters lassen sich in verschiedene, funktionale Kategorien unterteilen und werden in den folgenden Punkten tabellarisch zusammengefasst. Dabei sind nicht nur die in den Personae formulierten Anwendungsfälle aufgeführt, sondern auch andere, denkbare Möglichkeiten. Dennoch ist die Auswahl klar begrenzt, da das Ergebnis dieser Arbeit nicht die Umsetzung konkreter Anwendungsfälle, sondern die Entwicklung eines offenen Systems sein soll, welches in näherer Zukunft um vielfältige Anwendungsfälle erweitert werden kann. Die genannten User Stories sind die Grundlage für die abgeleiteten Anforderungen an den Roboter, die im Kapitel 3.3 definiert werden.

3.2.1 Anwendungsfälle für die Unterstützung im Alltag

Primär soll der Tischroboter seinen Nutzer im Alltag unterstützen können. Dazu gehören Erinnerungsfunktionen für Termine, die Möglichkeit, den Kontakt zu Familie und Freunden über Chat-Nachrichten oder Video-Anrufe zu vereinfachen, und andere Aufgaben. Diese Funktionen bieten sich vor allem für ältere Personen an, die zu Vergesslichkeit neigen. Dennoch kann die Unterstützung im Alltag genauso auf andere Zielgruppen wie Familien übertragen werden. Tabelle 1 führt mögliche Anwendungsfälle zur Unterstützung im Alltag auf.

Tabelle 1: Übersicht der Anwendungsfälle für die Unterstützung im Alltag

Als (Akteur)	möchte ich (Beschreibung)	um (Ergebnis)
Benutzer	meinen Kalender / Termine über den Roboter verwalten können	
Benutzer	Erinnerungen einrichten können	
Benutzer	eine Checkliste für den Alltag einrichten können	bestimmte Dinge nicht zu vergessen.
Benutzer	vom Roboter automatisch an Termine und andere Dinge, wie die Einnahme von Tabletten, erinnert werden	
Benutzer	meine Kontakte verwalten können	die Telefonnummern, Adressen, etc. im Blick zu haben.
Benutzer	Videoanrufe über den Roboter erledigen	mit meiner Familie oder auch mit meinem Arzt nicht nur sprechen, sondern den Gesprächspartner auch sehen zu können.

Als (Akteur)	möchte ich (Beschreibung)	um (Ergebnis)
Benutzer	über den Roboter Chatnachrichten verfassen und versenden können	einfacher als am Smartphone mit meiner Familie und meinen Bekannten Nachrichten auszutauschen.
Benutzer	Informationen über das Wetter erhalten	mich den Witterungsbedingungen entsprechend zu kleiden.

3.2.2 Anwendungsfälle in der Gebäudeautomation

Neben der grundsätzlichen Alltagsunterstützung kann die Funktionalität des Roboters mit SmartHome Technologie erweitert werden. In dem Fall soll der Roboter als Zentrale für Information und Bedienung genutzt werden können. Die möglichen Anwendungsfälle erstrecken sich dabei über das gesamte Spektrum der SmartHome Technologien. In der folgenden Tabelle 2 sind drei Beispiele aufgeführt.

Tabelle 2: Übersicht der Anwendungsfälle in der Gebäudeautomation

Als (Akteur)	möchte ich (Beschreibung)	um (Ergebnis)
Benutzer	vom Roboter den Status der Fenster und Türen in meinem Haus vermittelt bekommen	zu bemerken, wenn etwas unplanmäßig geöffnet ist.
Benutzer	vom Roboter den Status meiner Waschmaschine vermittelt bekommen	die aktuelle Waschladung nicht zu vergessen.
Benutzer	über den Roboter Licht und Temperatur in meinem Haus steuern	die Umgebung meinen Bedürfnissen anzupassen.

3.2.3 Anwendungsfälle für die Freizeitgestaltung

Zusätzlich zu den technischen und alltagsgebundenen Anwendungsfällen ist der Unterhaltungsaspekt des Roboters ein nicht zu vernachlässigender Punkt. Zur Unterhaltung gehören nicht nur diverse Spieloptionen, sondern auch die Möglichkeiten, Videos und Musik über den Roboter zu gucken bzw. zu hören. Tabelle 3 führt grundlegende Anwendungsfälle der Unterhaltung auf.

Tabelle 3: Übersicht der Anwendungsfälle für die Freizeitgestaltung

Als (Akteur)	möchte ich (Beschreibung)	um (Ergebnis)
Benutzer	Internetvideos angucken	mein Wissen zu erweitern und um unterhalten zu werden.
Benutzer	Musik hören	unterhalten zu werden.
Benutzer	Spiele mit dem Roboter spielen	unterhalten und beschäftigt zu werden, sowie geistig fit zu bleiben.

3.2.4 Allgemeine Anwendungsfälle

Zusätzlich zu den spezifischen User Stories gibt es für Nutzer allgemeine Anwendungsfälle, die die Benutzung des Roboters betreffen. Tabelle 4 führt diese Anwendungsfälle auf.

Tabelle 4: Übersicht der allgemeinen Anwendungsfälle

Als (Akteur)	möchte ich (Beschreibung)	um (Ergebnis)
Benutzer	Funktionen wie die Aufnahme von Bildern und Sprache deaktivieren können	mich aktiv gegen die Erfassung sensibler Daten zu wehren.
Entwickler / Benutzer	den Roboter transportieren können	ihn an verschiedenen Orten im Haus benutzen zu können.
Entwickler / Benutzer	den Roboter einschalten können	ihn zu verwenden.
Entwickler / Benutzer	den Roboter ausschalten können	Energie zu sparen.
Benutzer / Entwickler	die Lautstärke einstellen	sie der Umgebung anzupassen.

3.2.5 Anwendungsfälle in der Entwicklung

Für Entwickler existieren weitere Anwendungsfälle, die unabhängig von späteren, für den Benutzer relevanten Einsatzgebieten sind. Diese spezifischen Anwendungsfälle werden in folgender Tabelle 5 aufgeführt.

Tabelle 5: Anwendungsfälle in der Entwicklung

Als (Akteur)	möchte ich (Beschreibung)	um (Ergebnis)
Entwickler	mich über SSH mit dem Roboter verbinden können	u.a. Programme zu installieren, auszuführen und zu überwachen und Daten zu übertragen.
Entwickler	HelloWorld Programme zur Verfügung gestellt bekommen	die Funktionen des Roboters kennenzulernen.
Entwickler	Zugriff auf eine ausführliche Dokumentation haben	die Möglichkeiten bzw. die Verwendung des Roboters erklärt zu bekommen.
Entwickler	Schnittstellen-Bibliotheken zur Verfügung gestellt bekommen	auf die Hardware des Roboters in eigenen Programmen zugreifen zu können.
Entwickler	neue Software entwickeln und installieren können	die Funktionalität des Roboters zu erweitern bzw. zu verbessern.
Entwickler	neue Hardware anschließen können bzw. bestehende Hardware austauschen können	die Funktionalität des Roboters zu erweitern bzw. zu verbessern.

3.3 Definition der Anforderungen

Aus den in den vorherigen Abschnitt dargestellten Personae und den dazugehörigen User Stories lassen sich **Anforderungen an den Tischroboter** ableiten. Diese werden in diesem Abschnitt nach Hardware, Software und Design getrennt erläutert und begründet. Abschließend werden alle Anforderungen in Tabelle 6 aufgeführt.

Die Personae und die User Stories beziehen sich auf die Ebene der Anwendungsfälle und stehen im direkten Bezug zum Benutzer des Roboters. Darüber hinaus existieren durchaus Anforderungen, welche sich nicht direkt aus den Anwendungsfällen ableiten lassen. Dabei handelt es sich vor allem um **technische Anforderungen**, die für einen erfolgreichen Betrieb des Systems notwendig sind. Diese Anforderungen werden in den folgenden Punkten ebenfalls aufgeführt.

In der Anforderungsdefinition wird zwischen **Muss**- und **Soll**-Kriterien unterschieden. Die Erfüllung von Muss-Anforderungen ist für eine erfolgreiche Projektumsetzung obligatorisch. Soll-Anforderungen sind wünschenswerte Projektbestandteile, deren Erfüllung zwar geplant ist, ohne die ein grundlegender Betrieb des Systems jedoch trotzdem möglich sein wird.

Zusätzlich werden in der tabellarischen Aufführung funktionale und qualitative Anforderungen unterschieden. Funktionale Anforderungen werden in der letzten Spalte mit einem **F**, qualitative Anforderungen mit einem **Q** markiert.

3.3.1 Anforderungen an die Hardware des Roboters

Der Tischroboter soll als eine Alternative zum NAO genutzt werden können, um vor allem bestehende Anwendungsfälle des RoboticLabs der TH Wildau umzusetzen, aber auch zukünftig die Umsetzung weiterer Anwendungsideen zu ermöglichen. Damit das System für ein breites Einsatzspektrum infrage kommt, wird der **Preis** eines kompletten Roboters auf einen Bruchteil der Kosten für die Anschaffung eines NAOs beschränkt und darf den dreistelligen Bereich nicht überschreiten. Als Zielpreis werden 600 €, vergleichbar mit den Kosten eines hochwertigen Smartphones, angesetzt.

Die **zentrale Recheneinheit** des Roboters ist das Kernstück des Systems, welches alle Komponenten miteinander verknüpft. Um ein Nutzererlebnis ohne lange Wartezeiten zu gewährleisten sowie die lokale Ausführung komplexer Programme zu ermöglichen, muss die zentrale Recheneinheit über eine ausreichende **Rechenleistung** verfügen. Da die Anforderungen an die Performance nur schwierig quantitativ festlegbar sind, wird als Ziel angesetzt, dass Benchmark-Tests der Recheneinheit nicht schlechter abschneiden dürfen, als jene des Einplatinencomputers Raspberry Pi 3¹². Dieser ist aus Telematik-Lehrveranstaltungen der TH Wildau bekannt und wird als Referenz genutzt.

¹² Siehe Abschnitt 4.1.2.1 für Details zum Raspberry 3 Model B+

Für eine allgemeine **Internetanbindung**, aber auch für Anwendungsfälle, in denen der Roboter als Zentrale von anbindbarer SmartHome-Technik fungiert, muss das System eine Netzwerkverbindung über WLAN ermöglichen. Diese **Konnektivität** wird ebenfalls während des Entwicklungsbetriebs benötigt und soll auch kabelgebunden über LAN ermöglicht werden. Für ein breites Spektrum an Anwendungsfällen und für die flexible Anbindung von Geräten soll der Roboter außerdem Bluetooth unterstützen.

Die primäre **Energieversorgung** des Roboters muss über ein handelsübliches Netzteil möglich sein. Als handelsübliche Netzteile für den Roboter gelten beispielsweise USB-Netzteile oder Laptop-Netzteile. Zusätzlich zum normalen Betrieb über das Netzteil soll ein temporärer Akkubetrieb ermöglicht werden. Die Arbeitsdauer im **Akkubetrieb** orientiert sich an den Laufzeiten eines NAOs und soll dementsprechend mindestens zwei Stunden betragen. Über einen **Einschalter** soll der Roboter an- und ausgeschaltet werden können.

Der Roboter muss für die Anzeige und Darstellung von Informationen ein **Display** besitzen. Das Display muss größer als bei einem Smartphone sein, weswegen die Bildschirmdiagonale auf eine Größe von mindestens 7 Zoll festgelegt wird. Um nicht auf Eingaben über Sprache oder andere Sensoren des Roboters begrenzt zu sein, muss das Display ebenfalls als Eingabegerät genutzt werden können.

Für die Ausgabe von Ton, Musik und Sprache muss der Roboter **Lautsprecher** besitzen. Damit die Audioausgabe auch für schwerhörige Menschen geeignet ist, soll die Lautstärke nicht nur regulierbar, sondern auch höher einstellbar sein, als es bei einem Smartphone möglich wäre.

Für die visuelle und akustische Wahrnehmung seiner Umgebung muss der Roboter über eine **Kamera** und eine **Geräuscherfassung** verfügen. Die Auflösung der Kamera ist für die Bildqualität und die Möglichkeiten weiterer Verarbeitung ausschlaggebend. Die minimale Auflösung der Kamera wird auf 2 Megapixel festgelegt, was etwa der Auflösung von FullHD¹³ entspricht.

¹³ FullHD = 1920 x 1080 Pixel

3.3.2 Anforderungen an die Software des Roboters

Das auf dem Roboter verwendete **Betriebssystem** muss als freie Software ohne einmalige oder laufende Lizenzkosten verfügbar sein. Es muss sowohl das parallele Ausführen mehrerer Programme als auch die Ausgabe einer grafischen Oberfläche über das angeschlossene Display ermöglichen. Außerdem muss das Betriebssystem für Entwicklungs- und Wartungsprozesse einen externen Zugriff und eine Steuerung über SSH ermöglichen.

Für die angeschlossene Hardware, insbesondere für die Kamera, die Lautsprecher und die Geräuschwahrnehmung, muss **Schnittstellensoftware** entwickelt werden, die anderen Programmen Zugriff auf notwendige Informationen und Aktionen ermöglicht. Dabei soll sowohl der lesende als auch der schreibende Zugriff von parallel laufenden Programmen koordiniert werden, um Konflikte zu vermeiden und Daten effizient zu verteilen. Für verwendete Fremd-Software bzw. -Softwareframeworks gilt bezüglich der Lizenzkosten dasselbe wie für das Betriebssystem.

Für Folgeprojekte und Weiterentwicklungen müssen die Möglichkeiten des Roboters und der Zugriff auf die Hardware in **HelloWorld Programmen** zur Verfügung gestellt und erläutert werden. Diese Beispielpprogramme müssen mindestens in einer der im Telematik-Studium gelehrtten Programmiersprachen entwickelt werden. Zu den gelehrtten Sprachen zählen Java, C, C++, Python und JavaScript¹⁴.

¹⁴ Vgl. Modulhandbuch der Telematik (Stand 2018): https://www.th-wildau.de/files/Studiengaenge/Telematik/Modulhandbuch_Bachelor_Telematik_26-09-2018.pdf (abgerufen am 01.08.2019)

3.3.3 Anforderungen an das Design des Roboters

Alle Hardwarekomponenten des Roboters müssen in einem **Gehäuse** untergebracht werden. Dabei müssen beim Design Plätze für die einzelnen Komponenten vorgesehen werden, an denen diese sicher befestigt werden können. Die Hardware soll zusammen mit dem Gehäuse ein Gewicht von 5 Kilogramm nicht überschreiten, um eine Transportfähigkeit gewährleisten zu können.

Um einen performanten Betrieb zu garantieren und einer Überhitzung der Komponenten vorzubeugen, muss das Gehäuse eine **Kühlung** beinhalten. Für eine Regelung der Kühlung soll das Gehäuse um eine Temperaturerfassung ergänzt werden. Eine aktive Kühlung soll bei höchster Kühlleistung das Nutzererlebnis nicht negativ beeinträchtigen und daher eine maximale Lautstärke von 30db nicht überschreiten.

Von den Pepper und NAO Robotern ist bekannt, dass die Fähigkeit, Menschen direkt anzugucken, eine große emotionale Akzeptanz hervorruft. Daher soll der Tischroboter ebenfalls einen **Kopf mit Augen** besitzen. Dieser soll beweglich sein und sowohl ein Kippen in der vertikalen Ebene als auch eine Drehung in der horizontalen Ebene ermöglichen. Bei der Bewegung des Kopfes bzw. der Augen soll die Position der Kamera im gleichen Maße verändert werden, um dem Roboter ein größeres und dynamisches Sichtfeld zu ermöglichen.

3.3.4 Tabellarische Übersicht der Anforderungen

Die folgende Tabelle 6 beinhaltet alle definierten Anforderungen an die Hardware, die Software und das Design des Roboters.

Tabelle 6: Tabellarische Übersicht der definierten Anforderungen

ID	Komponente	Kriterium	Anforderung	Art
R001	Der Roboter	muss	insgesamt unter 1000 € kosten.	Q
R002	Der Roboter	soll	insgesamt unter 600 € kosten.	Q
R003	Der Roboter	soll	eine Netzwerkverbindung über LAN ermöglichen.	F
R004	Der Roboter	muss	eine Netzwerkverbindung über WLAN ermöglichen.	F
R005	Der Roboter	soll	eine Verbindung über Bluetooth ermöglichen.	F
R006	Der Roboter	muss	einen Zugriff über SSH erlauben.	F
R007	Die zentrale Recheneinheit	muss	in Performance-Benchmarks mindestens genauso gut abschneiden wie ein Raspberry Pi 3 Model B+.	Q
R008	Der Roboter	muss	die Ausgabe grafischer Darstellungen über ein Display ermöglichen.	F
R009	Das Display	muss	eine Mindestgröße von 7 Zoll Bildschirmdiagonale besitzen.	Q
R010	Der Roboter	muss	Eingaben über ein Display ermöglichen.	F
R011	Der Roboter	muss	die Wahrnehmung von Geräuschen ermöglichen.	F
R012	Der Roboter	muss	die Ausgabe von Ton über Lautsprecher ermöglichen.	F
R013	Die Lautstärke	muss	regulierbar sein.	Q
R014	Die Lautstärke	muss	höher einstellbar sein, als es bei einem Smartphone möglich ist.	Q
R015	Der Roboter	muss	die Aufnahme von Bildern über eine Kamera ermöglichen.	F
R016	Die Kamera	muss	eine Auflösung von mindestens 2 Megapixeln besitzen.	Q
R017	Das System	muss	die Energieversorgung über ein handelsübliches Netzteil ermöglichen.	F

ID	Komponente	Kriterium	Anforderung	Art
R018	Das System	soll	einen Akkubetrieb ermöglichen.	F
R019	Das System	soll	mindestens zwei Stunden im Akkubetrieb arbeiten können.	Q
R020	Der Roboter	soll	das An- und Ausschalten des Systems über einen Einschalter ermöglichen.	F
R021	Das Betriebssystem	muss	als freie Software ohne Lizenzkosten erhältlich sein.	Q
R022	Verwendete Software	muss	als freie Software ohne Lizenzkosten erhältlich sein.	Q
R023	Das Betriebssystem	muss	die parallele Ausführung mehrerer Programme ermöglichen.	F
R024	Das Betriebssystem	muss	die Ausgabe einer grafischen Oberfläche auf einem angeschlossenen Display ermöglichen.	F
R025	Die Software	muss	den Zugriff auf die Kamera des Roboters ermöglichen.	F
R026	Die Software	muss	den Zugriff auf die Geräuschwahrnehmung des Roboters ermöglichen.	F
R027	Die Software	muss	den Zugriff auf die Lautsprecher des Roboters ermöglichen.	F
R028	Die Software	soll	lesende Zugriffe auf die Hardware von parallel laufenden Programmen koordinieren.	F
R029	Die Software	soll	schreibende Zugriffe auf die Hardware von parallel laufenden Programmen koordinieren.	F
R030	Die Komponenten des Roboters	müssen	in Form von HelloWorld Programmen vorgestellt und zugänglich gemacht werden.	Q
R031	Die HelloWorld Programme	müssen	in einer im Telematik-Studium gelehrt Programmiersprache geschrieben werden.	Q
R032	Das Gehäuse	muss	alle Hardwarekomponenten des Roboters beinhalten.	F

ID	Komponente	Kriterium	Anforderung	Art
R033	Das Gehäuse	muss	die feste Anbringung der Hardwarekomponenten an vorgesehenen Plätzen ermöglichen.	Q
R034	Der Roboter	soll	ein Gewicht von 5 kg nicht überschreiten.	Q
R035	Das Gehäuse	muss	eine Kühlung besitzen.	F
R036	Der Roboter	soll	eine Erfassung der Temperatur innerhalb des Gehäuses ermöglichen.	F
R037	Das Gehäuse	soll	beim maximaler Kühlleistung eine Lautstärke von 30 dB nicht überschreiten.	Q
R038	Das Gehäuse	soll	einen Kopf darstellen.	Q
R039	Das Gehäuse	soll	zwei Augen darstellen.	Q
R040	Das Gehäuse	soll	eine Bewegung der Augen bzw. des Kopfes in der horizontalen Ebene ermöglichen (Drehung um die Z-Achse / <i>Gieren</i>).	F
R041	Das Gehäuse	soll	eine Bewegung der Augen bzw. des Kopfes in der vertikalen Ebene ermöglichen (Drehung um die Y-Achse / <i>Nicken</i>).	F
R042	Das Gehäuse	soll	zur Bewegung der Augen bzw. des Kopfes eine gleichzeitige Bewegung der Kamera ermöglichen.	F

4. Evaluation der Komponenten für das Gesamtsystem

Dieses Kapitel dient der Evaluation wesentlicher Komponenten, die für das Gesamtsystem benötigt werden. Zunächst umfasst die Betrachtung grundlegende Hardwaremodule, die den Funktionsumfang des Roboters bestimmen. In Abhängigkeit davon kann anschließend die Festlegung auf ein Betriebssystem sowie eine Vorauswahl der verwendeten Software erfolgen.

Am Ende des Kapitels sollen eine Auswahl an Komponenten bzw. Entscheidungen über Bestandteile des Systems vorliegen, auf deren Grundlage im nächsten Kapitel das Umsetzungskonzept erstellt werden kann.

4.1 Auswahl der Hardware

Aus den definierten Anforderungen lassen sich die benötigten **Hardwarekomponenten** für den Tischroboter ableiten. Dieser Abschnitt soll die Kernkomponenten des Systems evaluieren und auswählen. Zu diesen Komponenten gehören unter anderem die zentrale Recheneinheit, die Geräuschwahrnehmung, der Bildschirm und die Kamera. Andere Aspekte wie die Temperaturregulierung und die Energieversorgung sind Bestandteil des Konzepts, bei denen es teilweise erst während bzw. nach der Erstellung bestimmter Konzeptdetails möglich ist, endgültige Komponenten auszuwählen. In diesen Punkten werden lediglich verschiedene Optionen evaluiert, die für eine Umsetzung infrage kommen.

Die funktional zusammenhängenden Module des Systems sind nachfolgend in Abbildung 6 schematisch dargestellt. Innerhalb der Abbildung sind die Kapitel aufgeführt, in denen die jeweiligen Komponenten des Moduls evaluiert werden.

Abbildung 6: Vorbetrachtung der im System benötigten Module

4.1.1 Allgemeines Vorgehen zur Auswahl der Hardware

Der zu entwickelnde Tischroboter soll eine Alternative zum NAO-Roboter darstellen und für die Umsetzung aktueller Anwendungsfälle geeignet sein. Die benötigten Komponenten für eine Realisierung dieses Systems werden anhand **qualitativer Kriterien** evaluiert, welche eine Eignung bezüglich der gestellten Anforderungen feststellen sollen.

Allen Punkten gemeinsam ist die Beachtung des **Preises** als wichtiges Evaluationskriterium. Laut Anforderungen **R001** und **R002** ist dieser begrenzt und für einen potenziell großflächigeren Einsatz relevant. Die einzelnen Komponenten müssen insofern finanziell aufeinander abgestimmt werden, um das festgelegte Budget einzuhalten. Dennoch liegt der Fokus des Projekts deutlich auf einem qualitativen Ergebnis, sodass zwischen hoher Qualität und angebrachtem Preis Kompromisse eingegangen werden müssen.

Für die Qualitätsbeurteilung der Komponenten wird ebenfalls die **hardware- und softwaretechnische Nutzbarkeit** eingeschätzt. Zu diesem Kriterium zählen die angebotenen Tutorials und Benutzungsanleitungen, verfügbare Programm-Beispiele und die Ausführlichkeit der Dokumentation.

Grundsätzlich werden in dieser Auswahl nur Komponenten betrachtet, die für einen käuflichen Erwerb allgemein **verfügbar** sind¹⁵. Im letzten Unterpunkt der Hardwareevaluation werden die ausgewählten Komponenten in einer preislichen Übersicht zusammengefasst, welche sich auf den Bearbeitungszeitpunkt bezieht und die jeweiligen Distributoren auflistet.

¹⁵ Als allgemein verfügbar gelten Komponenten, die bei Herstellern oder Distributoren käuflich erwerbbar sind. Zu bekannten Distributoren, die Hardware in Deutschland vertreiben, gehören beispielsweise Conrad, Exp-Tech, Mouser, Pollin und Reichelt. Eine detaillierte Übersicht ist in Tabelle 23: Übersicht deutscher Distributoren im Anhang unter Punkt 8.1 zu finden.

4.1.2 Zentrale Recheneinheit

Die **zentrale Recheneinheit** ist das Herzstück des Systems. Es verknüpft alle Komponenten sowohl softwaretechnisch als auch hardwaretechnisch miteinander, verarbeitet eingehende Daten und steuert entsprechende Reaktionen des Systems. Während Desktop-PCs einen Rechner aus dem Zusammenspiel von CPU, Grafikkarte, Arbeitsspeicher, Mainboard und weiteren Komponenten bilden, bieten sich für das Projekt Einplatinencomputer (engl. **SBC** für Single-Board-Computer) an, die alle notwendigen Komponenten auf einem kompakten Board miteinander verknüpfen. Diese bieten zwar im Vergleich zu Desktop-Computern bzw. Servern weniger Leistung, sind jedoch wesentlich kleiner, günstiger und energiesparender.

In den letzten Jahren ist vor allem der Raspberry Pi als kostengünstiger Einplatinencomputer international bekannt geworden. Seit der Einführung des Raspberry Pis im Jahr 2012 sind zahlreiche Konkurrenz-Produkte¹⁶ entstanden, die im Vergleich zum Raspberry Pi andere Anschlüsse, bessere Leistung oder besondere Softwarepakete bieten. Tabelle 7 führt eine Auswahl bekannter Einplatinencomputer auf, die im Rahmen dieser Evaluation vorgestellt und in Punkt 4.1.2.5 abschließend verglichen werden.

Tabelle 7: In Deutschland erhältliche Einplatinencomputer

Serie	Modell	Hersteller
Raspberry Pi	3B+	Raspberry Pi Foundation
Banana Pi	M64	GuangDong BiPai technology
Odroid	N2	Hardkernel
Jetson	Nano	Nvidia

In der Tabelle sind lediglich die **aktuellsten Produkt-Modelle** der Hersteller aufgeführt. So gibt es von Raspberry Pi, Banana Pi und Odroid auch noch weitere, ältere Varianten, die hier nicht betrachtet werden. Ebenfalls liegen alle betrachteten SBCs innerhalb eines angebrachten Preissegments¹⁷.

Ähnlich wie Smartphones, Desktop-Prozessoren und Grafikkarten sind Einplatinencomputer ein sich schnell weiterentwickelndes Technologie-Segment. Die betrachteten Optionen bilden daher einen **aktuellen Stand** von **Juni 2019** ab und können innerhalb kurzer Zeiträume durch moderne Alternativen überholt sein. Die im Folgenden aufgeführten Evaluationskriterien sind jedoch allgemeingültig und auf zukünftige SBCs ebenfalls anwendbar.

¹⁶ Siehe Wikipedia: Vergleich zwischen Einplatinencomputern unter https://en.wikipedia.org/wiki/Comparison_of_single-board_computers (abgerufen am 26.06.2019); die Tabelle führt knapp 200 verschiedene Einplatinencomputer auf

¹⁷ Für dieses Projekt werden Einplatinencomputer unterhalb von 150€ betrachtet, um genügend Freiraum für die anderen Komponenten des Systems zu haben und unterhalb des in **R001** definierten Budgets von maximal 1000€ zu bleiben.

Evaluationskriterien der zentralen Recheneinheit

Neben dem Preis und der Verfügbarkeit sind die im Folgenden erläuterten Evaluationskriterien für die Bewertung der Alternativen von Bedeutung.

Leistung

Je performanter die zentrale Recheneinheit ist, desto mehr lokale Umsetzungsmöglichkeiten ergeben sich beim Einsatz des Roboters. Vor allem Anwendungsfälle, in denen Audio- und Bildverarbeitung zum Einsatz kommt, benötigen viel Rechenleistung für schnelle und qualitative Resultate. Aber auch das allgemeine Benutzererlebnis und eine flüssige Bedienung sind von der Performance der zentralen Recheneinheit abhängig. Die Leistung des Computers lässt sich primär an der CPU festmachen. Als Leistungsmerkmale werden dabei vor allem die Taktrate und die Anzahl der Kerne betrachtet. Neben der CPU ist die Größe und die Geschwindigkeit des Arbeitsspeichers ausschlaggebend für die Performance des Gesamtsystems.

Für die Bewertung der Gesamt-Performance werden oft Benchmark-Tests durchgeführt, die verschiedene Leistungsmerkmale des Single-Board-Computers in Punkten bewerten. Diese können die reale Performance in bestimmten Aufgaben besser widerspiegeln als die reine Leistungsspezifikation. Für die hier vorgestellten Einplatinencomputer sind die Ergebnisse verschiedener Benchmark-Tests in Tabelle 26 (Anhang Punkt 8.3.3 auf S. 116) aufgeführt.

Anschlüsse und Konnektivität

Um andere Hardwarekomponenten an Einplatinencomputer anzuschließen, stellen die Boards diverse Anschlüsse zur Verfügung. Dazu zählen beispielsweise Anschlussmöglichkeiten für Ethernet, USB-Geräte, Displays, Audio, Kameras und weitere Hardwareschnittstellen. Zumindest muss der SBC Anschlüsse bereitstellen, um die funktionalen Anforderungen **R008** bis **R015** zu erfüllen.

Neben den kabelgebundenen Anschlüssen besitzen einige SBCs direkt integrierte Funktechnologien wie WLAN oder Bluetooth, was der Erfüllung von Anforderungen **R004** und **R005** zu Gute kommen würde.

Community

Sowohl für den Entwicklungsprozess als auch für das Beheben von Fehlern ist eine aktive Community wichtig, die Probleme diskutiert und Fragen beantwortet. Je bekannter und verbreiteter ein Board ist, desto wahrscheinlicher ist es, dass ein eigenes Problem bereits aufgetreten ist und eine entsprechende Lösung existiert.

Angebotene Betriebssysteme

Für Einplatinencomputer werden jeweils Betriebssystem-Images bereitgestellt, deren Kompatibilität und Funktionsfähigkeit für das Board gewährleistet werden. Je nach Aktualität der verfügbaren Betriebssysteme werden die Möglichkeiten moderner Anwendungen gewährleistet bzw. eingeschränkt.

4.1.2.1 Raspberry Pi 3 Model B+

Der Raspberry Pi wird von der britischen *Raspberry Pi Foundation*¹⁸ als Open Source Projekt entwickelt und ist einer der bekanntesten Einplatinencomputer. Den aktuellen Raspberry gibt es in unterschiedlichen Ausführungen, wovon hier der aktuelle Raspberry Pi 3 Model B+ betrachtet wird (siehe Abbildung 7; im Folgenden nur noch Raspberry genannt). Der Raspberry ist weltweit verfügbar, aktuell für einen Preis zwischen 30 € und 40 €¹⁹.

Abbildung 7: Raspberry Pi 3 Model B+ [21]

Der Raspberry ist bereits aus Telematik-Lehrveranstaltungen bekannt und dient für die weitere Evaluation als Orientierung. Nach Anforderung **R007** soll der ausgewählte SBC mindestens die Leistung eines Raspberry aufweisen. Mit seinem Quad-Core Prozessor bietet der Raspberry ausreichend **Performance**, um einfache Programme auszuführen und den Ansprüchen vieler Projekte zu genügen. Bei vielen parallel ablaufenden Aufgaben bzw. bei Spezialaufgaben, die einen höheren Rechenaufwand erfordern, sind die Grenzen des Mini-Computers jedoch schnell erreicht. Vor allem der Arbeitsspeicher, von welchem lediglich 1 GByte verfügbar ist, begrenzt die Möglichkeiten eines auf dem Raspberry basierenden Systems. [21]

Von den **Anschlüssen** bietet der Raspberry alle für das Projekt benötigten Möglichkeiten. Außerdem sind auf dem Board die Funktechnologien WLAN und Bluetooth direkt vorhanden, was ein Vorteil bei der Verwendung des Raspberry ist. [21]

Mit über 17 Millionen verkauften Stück (Stand 2017 [22]) ist die sehr große **Community** jedoch der größte Vorzug des Raspberry. Die offizielle Website der Raspberry Pi Foundation stellt nicht nur ein **offizielles Image** bereit, welches fortlaufend Aktualisierungen und Fehlerbehebungen erhält [23]. Zusätzlich werden viele ausführliche Anleitungen und Tutorials angeboten, die den Einstieg in den Raspberry erleichtern [24]. Neben dem **offiziellen Forum**, in dem Fragen gestellt und Probleme diskutiert werden können, gibt es viele weitere Foren, die Themen des Raspberry erläutern. Dadurch ist es sehr wahrscheinlich, für auftretende Probleme schnell Lösungsansätze zu finden.

¹⁸ Offizielle Seite: <https://www.raspberrypi.org/> (abgerufen am 01.08.2019)

¹⁹ Stand Juni 2019

4.1.2.2 Banana Pi M64

Banana Pi ist ein Open Source Projekt der chinesischen Firma *GuangDong BiPai technology co., LTD*²⁰. Das Projekt bietet mehrere SBC-Modelle verschiedener Größen und Leistungen für unterschiedliche Aufgaben, von denen hier der Banana Pi BPI-M64 als einziges Modell, das einen 64-Bit Prozessor besitzt, betrachtet wird [25]. Der Banana Pi BPI-M64 (siehe Abbildung 8; im Folgenden nur noch Banana Pi genannt) ist in Deutschland für gut 60 € erhältlich²¹.

Abbildung 8: Banana Pi BPI-M64 [26]

Die **Anschlüsse** des Banana Pi sind vergleichbar mit denen des Raspberry Pi, genauso wie der Vorteil, dass WLAN und Bluetooth direkt auf dem Board integriert sind. Lediglich bei den USB-Ports bietet der Banana Pi nur zwei Anschlüsse. Bezüglich der **Leistung** unterscheidet sich der Banana Pi vom Raspberry Pi vor allem im Arbeitsspeicher, da mit 2 GByte die doppelte Kapazität vorhanden ist. Die restlichen Leistungsspezifikationen sind nahezu gleichwertig. Dennoch zeigen die Ergebnisse von Benchmark-Tests, dass der Banana Pi trotz vergleichbarer Merkmale eine schlechtere Performance besitzt als der Raspberry Pi²². [26]

Auf der offiziellen Wiki-Seite zum Banana Pi werden über 20 verschiedene **Images** für das Board bereitgestellt. Welches davon genutzt werden sollte, wird trotz einer vorhandenen Anleitung für den Schnelleinstieg nicht deutlich. Darüber hinaus existieren wenig detaillierte Tutorials für die Arbeit mit dem Banana Pi. [27]

Es gibt ein offizielles Banana Pi Forum, auf dem die **Community** Probleme diskutieren kann und Neuigkeiten vorgestellt werden. Generell zeigt das Forum jedoch eine geringe Aktivität, viele der gestellten Fragen aus diesem Jahr (2019) wurden teils gar nicht oder oft ohne Problemlösung beantwortet. Der Hersteller selbst trägt ebenfalls wenig zu den Forumsbeiträgen bei [28]. Es gibt ebenso ein deutsches Forum für den Banana Pi. Dieses zeigt jedoch ein ähnliches Problem, dass zu vielen der in jüngerer Zeit gestellten Fragen keine Lösungen gefunden wurden [29].

²⁰ Ehemals *Sinovoip*, offizielle Seite: <http://www.banana-pi.org/> (abgerufen am 01.08.2019)

²¹ Stand Juni 2019

²² Siehe Abschnitt 8.3.3 im Anhang für die Ergebnisse verschiedener Benchmark-Tests

4.1.2.3 Odroid N2

Odroid ist ein Open Source Projekt der südkoreanischen Firma *Hardkernel*²³. Odroid bietet verschiedene Modellreihen an, von denen in dieser Evaluation der Odroid N2 (siehe Abbildung 9; im Folgenden nur Odroid genannt) betrachtet wird, welcher als neuester Einplatinencomputer von Hardkernel als am leistungsstärksten angegeben wird [30]. Das Board gibt es mit 2 GByte oder 4 GByte Arbeitsspeicher in zwei verschiedenen Varianten und wird in Deutschland für einen Preis zwischen 80 € und 100 € verkauft²⁴.

Abbildung 9: Odroid N2 [30]

Neben dem größeren Arbeitsspeicher ist auch der Prozessor des Odroid mit sechs Kernen **leistungsstärker** als bei den ersten beiden vorgestellten Boards. Die Ergebnisse von Benchmark-Tests bestätigen die höhere CPU-Performance im Vergleich zu den anderen Boards, wodurch der Odroid besonders für den Einsatz bei rechenintensiven Aufgaben geeignet ist.

Ein negativer Aspekt bei den **Anschlüssen** des Odroid ist eine fehlende, dedizierte Kameraschnittstelle, sodass für die in Anforderung **R015** geforderte Aufnahme von Bildern nur eine USB-Kamera infrage kommen würde. Sowohl WLAN als auch Bluetooth sind auf dem Board nicht vorhanden, genauso wenig wie eine entsprechende Erweiterungsmöglichkeit. Diese Funktechnologien müssten dementsprechend ebenfalls über USB nachgerüstet werden. [31]

Hardkernel stellt für den Odroid sowohl ein aktuelles Linux **Image** als auch ein Android Image bereit. Jedoch werden auf dem offiziellen Wiki keine Anleitungen zur Installation und zum Einstieg in die Arbeit mit dem Board angeboten [31]. Die **Community** bzw. die Anzahl der aktiven Mitglieder im offiziellen Odroid-Forum ist zwar wesentlich kleiner als beim Raspberry, trotzdem werden Probleme und Fragen aktiv miteinander diskutiert, auch in direkter Kommunikation mit dem Hersteller. Im Forum werden ebenfalls Anleitungen für bestimmte Aspekte des Boards und Projekte mit dem Odroid vorgestellt [33].

²³ Offizielle Seite: <https://www.hardkernel.com/> (abgerufen am 01.08.2019)

²⁴ Stand Juni 2019

4.1.2.4 Nvidia Jetson Nano

Der Jetson Nano ist ein Computermodule des US-amerikanischen Grafikprozessor-Herstellers *Nvidia*²⁵. Es wurde Anfang dieses Jahres (2019) vorgestellt und ist damit das jüngste SBC in dieser Evaluation. Offiziell ist der Jetson Nano ab August 2019 weltweit im Handel erhältlich. Einige Monate zuvor konnten jedoch bereits Jetson Nano Entwicklerkits direkt bei Nvidia erworben werden, um das Board und dessen Möglichkeiten zu testen. [32][32]

Das Jetson Nano Entwicklerkit (siehe Abbildung 10; im Folgenden nur noch kurz Jetson Nano genannt) beinhaltet das eigentliche Jetson Nano Modul und verbindet dieses mit Anschlussmöglichkeiten für Peripherie. Bei Nvidia beträgt der offizielle Preis in Deutschland 109 €²⁶.

Abbildung 10: Nvidia Jetson Nano Entwicklerkit [32]

Die Besonderheit des Jetson Nano wird im integrierten, **leistungsfähigen Grafikprozessor** angegeben. Während der Raspberry Pi lediglich zwei und der Odroid sechs GPU-Kerne besitzen, verfügt der Jetson Nano über 128 CUDA*-Recheneinheiten. Damit ist es möglich, bestimmte Programmteile auf dem Grafikprozessor statt auf der CPU auszuführen. Gerade für Anwendungsbereiche wie neuronale Netze und Bildverarbeitung, bei denen Algorithmen parallelisiert werden können, besteht die Chance einer deutlichen Performance-Verbesserung. [32][32]

Die **Anschlüsse** des Jetson Nano sind vergleichbar mit denen des Raspberry Pi, lediglich ein analoger Audioausgang fehlt auf dem SBC. Standardmäßig besitzt das Board ebenfalls kein WLAN oder Bluetooth, jedoch gibt es einen vorgesehenen Erweiterungsslot, sodass diese beiden Funktechnologien mit einem dedizierten Modul nachgerüstet werden können. [32]

²⁵ Offizielle Seite: <https://www.nvidia.com/de-de/> (abgerufen am 01.08.2019)

²⁶ Stand Juni 2019

Nvidia stellt für den Jetson Nano ein offizielles Linux **Image** bereit, auf dem bereits grundlegende Frameworks und Bibliotheken installiert sind, vor allem um die Vorteile des Grafikprozessors nutzen zu können [34]. Daneben werden auf der Website von Nvidia Anleitungen zur Inbetriebnahme und viele Tutorials für die Nutzung des Boards angeboten sowie konkrete Projekte und deren Umsetzung vorgestellt [35, 36]. Außerdem können Entwickler das Nvidia-Entwicklerforum nutzen, um Probleme zu besprechen und Fragen zu stellen. Dabei diskutieren **Community**-Mitglieder nicht nur untereinander, sondern auch direkt mit Mitarbeitern von Nvidia²⁷.

4.1.2.5 Auswahl der zentralen Recheneinheit für das Projekt

Für eine **abschließende Auswahl** sind die vier vorgestellten Einplatinencomputer mit ihren Kerndaten in der folgenden Tabelle 8 aufgeführt. Die am Anfang genannten Evaluationskriterien werden in der Tabelle jeweils mit für das Projekt *gut geeignet* (+), *geeignet* (o) oder *ungeeignet* (–) bewertet.

Eine **detaillierte Übersicht** aller Einzelheiten zu den Boards ist im Anhang unter Abschnitt 8.3 ab Seite 115 zu finden.

Tabelle 8: Übersicht und Bewertung der vorgestellten Einplatinencomputer

Kriterium	Raspberry Pi	Banana Pi	Odroid N2	Jetson Nano
CPU Kerne x Takt	4 x 1,4 GHz	4 x 1,2 GHz	4 x 1,8 GHz 2 x 1,9 GHz	4 x 1,4 GHz
Grafikprozessor Kerne	2	2	6	128
Arbeitsspeicher	1 GByte	2 GByte	4 GByte	4 GByte
Bewertung Performance	–	–	+	+
Bewertung Anschlüsse & Konnektivität	+	o	–	+
Bewertung Betriebssystem	+	–	o	+
Bewertung Community	+	–	o	+
Preis (Stand Juni 2019)	ca. 35 €	ca. 60 €	ca. 90 €	ca. 110 €

²⁷ Beispiel für Forums-Diskussion: <https://devtalk.nvidia.com/default/topic/1049782/jetson-nano/jetson-nano-not-booting/> (abgerufen am 26.06.2019)

Der **Raspberry Pi** würde vor allem aufgrund seines geringen Preises, aber auch wegen des hohen Bekanntheitsgrads und der großen Community infrage kommen. So wäre es schnell möglich, alle Komponenten des Systems miteinander zu verknüpfen, in Betrieb zu nehmen und Lösungen bei auftretenden Problemen zu finden. Lediglich die Performance des Raspberry würde das System begrenzen. Der Tischroboter soll in der Lage sein, vielfältige Anwendungsfälle umzusetzen und anfallende Daten dabei größtenteils lokal zu verarbeiten. Vor allem durch den begrenzten und langsamen Arbeitsspeicher würde der Raspberry wahrscheinlich schnell an seine Grenzen kommen.

Der **Banana Pi** wird aufgrund mehrerer Kriterien ausgeschlossen. Nicht nur ist die Performance schwächer als beim Raspberry, vor allem die fehlenden Tutorials und die kleine, inaktive Community würden die Arbeit mit dem Board und die Lösung auftretender Probleme erschweren. Möglicherweise gibt es andere Modelle vom Banana Pi, die für bestimmte Einsatzzwecke besser geeignet sind. Beispielsweise wurden einige Modelle speziell für den Einsatz als Internet-Router entwickelt.

Der **Odroid** ist die leistungstärkste Alternative, was die pure CPU-Performance angeht. Für den Einsatz als Server zur Datenspeicherung und Verarbeitung rechenintensiver Aufgaben wäre er gut geeignet. Die Community hinter den Odroid-Boards ist kleiner als die vom Raspberry, aber trotzdem aktiv. Die negativen Aspekte des Odroid, also die fehlende WiFi-Konnektivität und der fehlende Kameraanschluss, müssten durch die Verwendung entsprechender USB-Komponenten bzw. Adapter ausgeglichen werden.

Für das Projekt wird der **Jetson Nano** von Nvidia ausgewählt. Zwar ist das Board die teuerste Alternative, bietet aber auch großes Potenzial. Vor allem der Grafikprozessor des Jetson Nano birgt interessante Möglichkeiten für die Verwendung in der Robotik, gerade für Aufgaben der Bildverarbeitung. Ebenfalls der mögliche Einsatz künstlicher Intelligenz und die Entwicklung neuronaler Netze heben den Jetson Nano hervor, vor allem für einen späteren Einsatz in Forschungsprojekten. Der Jetson Nano besitzt neben dem Grafikprozessor eine gute CPU-Leistung und im Vergleich zum Raspberry einen größeren und schnelleren Arbeitsspeicher.

4.1.3 Bildschirm

Der Bildschirm des Tischroboters ist eine wichtige Komponente bei vielen der in Abschnitt 3.2 dargestellten Anwendungsfälle. Laut Anforderungen **R008** und **R010** soll dieser sowohl grafische Inhalte ausgeben als auch als Eingabegerät dienen können. Somit ist die Verwendung eines **Touch-Displays** obligatorisch. Diese Bildschirmart ist vor allem von Smartphones und Tablets bekannt und ermöglicht das Betätigen von grafischen Steuerelementen, ohne weitere Hilfsmittel wie eine Maus zu benötigen.

Der Roboter Pepper besitzt ebenfalls einen Touch-Bildschirm, der für viele Anwendungsfälle als Schnittstelle zum Menschen genutzt wird. Dabei handelt es sich um ein vollständiges Tablet, welches auch getrennt vom Roboter genutzt werden könnte. Für dieses Projekt kommt lediglich ein Bildschirm infrage, welcher im preislichen Rahmen von unter 100 € liegt, um das in **R001** und **R002** definierte Budget nicht zu überschreiten. Ein komplettes Tablet kostet meist wesentlich mehr, sodass lediglich reine Bildschirme für die Auswahl infrage kommen.

Der als zentrale Recheneinheit ausgewählte Jetson Nano besitzt für den Anschluss von Bildschirmen sowohl einen **HDMI**- als auch einen **DisplayPort-Anschluss**, von denen beim ausgewählten Display daher ebenfalls geringstenfalls einer dieser Anschlüsse vorhanden sein muss. Zusätzlich soll der Bildschirm laut Anforderung **R009** mindestens eine Bildschirmdiagonale von 7 Zoll, also gut 17 Zentimetern besitzen. Ein solches Display ist größer als ein gängiges Smartphone und bietet dadurch eine größere Darstellung grafischer Elemente. Dieser Vorteil ist besonders unter dem Aspekt sinnvoll, dass der Roboter auf einem Tisch eingesetzt wird und der Betrachtungsabstand daher unter Umständen größer sein kann. Gleichzeitig erlaubt ein **7 Zoll Display** ein kompaktes Design²⁸ des Tischroboters, weswegen für die Umsetzung diese Größe bevorzugt wird.

Das Angebot von Bildschirmen dieser Art und des angegebenen Preissegments ist begrenzt. Die verfügbaren Touch-Displays sind meist direkt für den Einsatz am Raspberry Pi gedacht und sollten dementsprechend auch ohne weiteren Einrichtungsaufwand am Jetson Nano funktionieren. Unterschiede gibt es sowohl in der Bildauflösung²⁹ und in der Art der Befestigung.

Für das Projekt soll ein Bildschirm ausgewählt werden, welcher eine hohe Auflösung von 1024 x 600 Pixeln, einen HDMI-Anschluss und eine ansprechende Verarbeitung inklusive Befestigungslöchern besitzt. Letzteres ist für die Integration des Displays in das Roboter-Gehäuse relevant. Ein möglicher Bildschirm ist in Abbildung 11 zu sehen und für ca. 50 € erhältlich³⁰.

Abbildung 11: 7 Zoll Touch-Display [37]

²⁸ Zum Vergleich: Das Tablet des Pepper Roboters besitzt eine Bildschirmdiagonale von etwa 10 Zoll.

²⁹ Bei 7-Zoll Bildschirmen für den Einsatz an Einplatinencomputern gibt es üblicherweise die Auflösungen 800 x 480 Pixel und 1024 x 600 Pixel, wovon letztere detaillierter ist und damit bevorzugt wird.

³⁰ Ein detaillierterer Vergleich verschiedener Bildschirme inklusive der Quellen ist in der Datei *Hardware Komponenten Vergleich.xlsx* zu finden.

4.1.4 Geräuschwahrnehmung

Der Tischroboter soll nach Anforderung **R011** die Möglichkeit bieten, Geräusche wahrzunehmen. Für die Geräuschwahrnehmung wird auf eine hardwaretechnisch selbstentwickelte Umsetzung verzichtet. Stattdessen soll das **ReSpeaker**-Modul der Firma *Seeed Technology*³¹ verwendet werden, da dieses bereits in anderen Projekten des RoboticLabs zum Einsatz kommt³².

Der ReSpeaker (siehe Abbildung 12) ist ein **Mikrofonarray**^{*}, welches vier sensible Mikrofone besitzt, die kreisförmig auf einer Leiterplatte angeordnet sind. Der Vorteil des ReSpeaker besteht darin, dass die notwendige Signalverarbeitung direkt auf dem Board vorgenommen und über eine USB-Schnittstelle für die Weiterverarbeitung bereitgestellt wird. Zu den besonderen Funktionen, die das kompakte Modul für den Einsatz in der Robotik und sprachgesteuerten Systemen hervorheben, gehören eine integrierte Richtungserkennung von Geräuschquellen und eine automatische Rausch- und Echounterdrückung. Zusätzlich sind auf der Oberseite des Boards zwölf RGB-LEDs vorhanden, die frei programmierbar sind und für eine optische Rückmeldung des Roboters genutzt werden können. [38]

Abbildung 12: ReSpeaker Mikrofonarray [38]

³¹ Offizielle Seite: <https://www.seeedstudio.com/> (abgerufen am 01.08.2019)

³² Die Wahl des ReSpeaker-Moduls geschieht vor allem in Absprache mit Tobias Kannenberg, der seine Masterarbeit mit dem Titel „Entwicklung, Evaluation und Integration von Verfahren zur optimierten Nutzung von automatischer Spracherkennung in Robotikanwendungen“ im RoboticLab der TH Wildau schreibt. Das Modul soll sowohl in dieser als auch in seiner Abschlussarbeit verwendet werden.

Es gibt neben dem ReSpeaker noch weitere Mikrofonarray-Module, die ähnliche Funktionen bieten, jedoch nicht bei deutschen Distributoren verfügbar sind. Der ReSpeaker ist für einen Preis von etwa 60 € erhältlich³³.

Gegenüber Konkurrenz-Modulen wie dem *miniDSP UMA-8*³⁴ ist der ReSpeaker mit einem Durchmesser von 7 Zentimetern sehr kompakt. Er bietet außerdem die besondere Möglichkeit, über USB als Ausgabegerät für Audiosignale genutzt werden zu können und an den analogen Audioausgang ein Lautsprechersystem anzuschließen. Dabei kann auf dem Board das ausgegebene Audiosignal direkt aus den Geräuschaufnahmen herausgefiltert werden, wodurch eine zukünftige Spracherkennung bei laufender Musik vereinfacht wird.

4.1.5 Audioausgabe

Damit der Roboter wie in Anforderung R012 gefordert Musik, Töne und Sprache ausgeben kann, wird ein **Lautsprechersystem** benötigt. Zum Lautsprechersystem gehören neben einem oder mehreren **Lautsprechern** ein vorgesehener **Verstärker**, welcher höhere Lautstärken für auszugebende Audiosignale ermöglicht. Laut Anforderung **R014** soll bei dem System eine höhere Lautstärke einstellbar sein als bei einem Smartphone. Diese Anforderung ist besonders in Anwendungsgebieten relevant, in denen der Roboter schwerhörige Personen im Alltag unterstützen soll.

Außer der Lautstärke ist auch der **Klang des Lautsprechersystems** bedeutsam für die Qualität des Roboters. Beim NAO-Roboter, der zwei Lautsprecher im Kopf besitzt, wird deutlich, dass der Klang ab einer bestimmten Lautstärke blechern wirkt und als unangenehm empfunden wird. Die Klangqualität lässt sich schwierig in einer konkreten Anforderung formulieren bzw. theoretisch vorherbestimmen. Dennoch wird ein volleres Klangspektrum angestrebt, indem statt zwei kleinen Lautsprechern, die vor allem hochfrequente Töne wiedergeben können, größere Lautsprecher oder ein Verbund aus Lautsprechern für verschiedene Frequenzbereiche verwendet wird. Im Konzept wird das konkrete Lautsprechersystem geplant, welches vom Gehäuse generell und vom zur Verfügung stehenden Platz abhängt. Daher werden hier lediglich verschiedenartige Lautsprecher für eine flexible Konzepterstellung ausgewählt.

Der deutsche Lautsprecherhersteller *Visaton*³⁵ bietet ein großes Sortiment verschiedener Lautsprecherarten an. Neben den verschiedenen Frequenzbereichen unterscheiden sich angebotene Lautsprecher in der Größe, in der maximalen Eingangsleistung und im Preis. Um im festgelegten Budget dieses Projekts zu bleiben, liegen alle in Tabelle 9 ausgewählten Lautsprecher im mittleren Preissegment und kosten jeweils unter 25 €³⁶.

³³ Stand Juli 2019

³⁴ Siehe <https://www.minidsp.com/products/usb-audio-interface/uma-8-microphone-array> für weitere Informationen (abgerufen am 06.07.2019)

³⁵ Offizielle Seite: <http://www.visaton.de/> (abgerufen am 01.08.2019)

³⁶ Ein detaillierterer Vergleich der verschiedenen Lautsprecher inklusive der Quellen ist in der Datei *Hardware Komponenten Vergleich.xlsx* zu finden.

Tabelle 9: Übersicht verschiedener Lautsprecher

Bezeichnung	Art	Frequenzbereich	Durchmesser	Leistung	Preis ³⁷
FR 58	Breitband	120 – 20.000 Hz	58 mm	10W	6,95 €
R10 S TE	Breitband	100 – 13.000 Hz	102 mm	20W	7,75 €
FX10-4	2 Kanal Breitband	95 – 22.000 Hz	100 mm	40W	16,90 €
DX 4X6X	2 Kanal Breitband	65 – 22.000 Hz	oval 153 x 97 mm	30W	21,75 €
F 8SC-8	Breitband	80 – 15.000 Hz	75 mm	20W	10,25 €
SC 5-8	Hochtöner	1500 – 22.000 Hz	52 mm	60W	9,95 €

Der *FR 58* ist ein kompakter Breitbandlautsprecher, welcher vermutlich am ähnlichsten zu denen des NAO-Roboters ist. Der *R10 S* Lautsprecher ist ein größerer Breitbandlautsprecher, der vor allem mit einer guten Sprachverständlichkeit angegeben ist. Der *FX10-4* und der *DX 4X6X* sind 2-Kanal Breitbandlautsprecher, die bereits aus einem Verbund von Tiefton- und Hochtonlautsprechern bestehen. Letzterer hat die Besonderheit, dass er oval ist und damit gegebenenfalls besser im Gehäuse positioniert werden kann. Die letzten beiden Alternativen gehören zusammen und sind für einen 2-Wege Lautsprecherverbund geeignet.

Um die Lautsprecher verwenden zu können, wird ein vorgesetzter **Verstärker** benötigt, dessen Leistung ausschlaggebend für die maximale Lautstärke des Systems ist. Der Hersteller *Adafruit*³⁸ bietet kompakte Verstärker-Boards in unterschiedlichen Leistungsstufen an. Der Vorteil dieser Boards liegt darin, dass keine eigene Verstärkerschaltungen entworfen werden müssen. Um die Anforderung **R014** zu erfüllen, wird die leistungsstärkste Variante für das Projekt ausgewählt, welche Lautsprecher mit 20 Watt Leistung ansteuern kann. Mit den oben dargestellten Lautsprechern sollte mit diesem Verstärker eine theoretische Lautstärke von ca. 95 Dezibel möglich sein³⁹. Das Board (siehe Abbildung 13) ist für einen Preis von etwa 20 € erhältlich und bietet die Möglichkeit, die Lautstärke digital über I²C einzustellen, wodurch ebenfalls **R013** erfüllt werden kann.

³⁷ Stand Juni 2019. Alle Preise stammen vom Distributor Reichelt, wo die Lautsprecher am günstigsten angeboten wurden.

³⁸ Offizielle Seite: <https://www.adafruit.com/> (abgerufen am 01.08.2019)

³⁹ Die Lautsprecher erzeugen bei 1 Watt Leistung ca. eine Lautstärke zwischen 82 und 90 Dezibel. Je Verdoppelung der Leistung steigt die Lautstärke ca. um 3 Dezibel, wodurch bei 16 Watt Verstärkerleistung eine resultierende Lautstärke zwischen 94 und 102 Dezibel entsteht [39].

Abbildung 13: Stereo 20W Class D Audio Verstärker Board von Adafruit [40]

4.1.6 Aufnahme von Bildern

Der Roboter soll die Aufnahme von Bildern über eine **Kamera** ermöglichen, wie in Anforderung **R015** beschrieben ist. Auf dem Nano Jetson ist ein CSI-Anschluss vorhanden, über welchen entsprechende Kamera-Module direkt angeschlossen werden können. Kameras, die über CSI angeschlossen werden, bieten gegenüber USB-Kameramodulen nicht nur den Vorteil, dass sie meist kompakter sind, sie beanspruchen auch weniger CPU-Leistung bei geringerem Arbeitsspeicherbedarf zum Verarbeiten von Bildern. Bei Anwendungsfällen, in denen eine hohe Bildrate benötigt wird oder wenn parallel zu den Bildaufnahmen weitere, rechenintensive Programme ablaufen, wird daher der performantere CSI-Anschluss bevorzugt. Der einzige Nachteil gegenüber USB-Modulen ist die begrenzte Kabellänge von unter 30 Zentimetern. Diese Einschränkung muss beim Systemdesign und der Positionierung der Komponenten beachtet werden. [41]

Auf der offiziellen Wiki-Seite zum Jetson Nano werden **empfohlene Kamera-Module** für das Board vorgestellt. Neben CSI-Kameras von amerikanischen Distributoren und verschiedenen USB-Kameras wird auch das **Raspberry Pi Kameramodul V2** aufgeführt [34]. Dieses Modul besitzt im Vergleich zu den anderen gelisteten Kameramodulen eine höher auflösende Kamera mit **acht Megapixel** und genügt damit der Anforderung **R016**. Einzelne Bilder können somit in einer Auflösung von 3280 x 2464 Pixel aufgenommen werden. Für Videos mit 30 Bildern pro Sekunde wird eine maximale Auflösung von 1920 x 1080 Pixeln angegeben, was FullHD entspricht [42]. Die Kamera (siehe Abbildung 14) kostet etwa 25 € und kann über die vier Schraubenlöcher am Gehäuse des Roboters befestigt werden.

Abbildung 14: Raspberry Pi Camera Module V2 [42]

4.1.7 Temperaturregulierung

Die Hardwarekomponenten des Roboters produzieren bei laufendem Betrieb **Abwärme**. Vor allem der Jetson Nano kann bei hoher Rechenlast sehr warm werden, weswegen dieser bereits mit einem passiven Kühlelement geliefert wird. Da alle Komponenten in einem Gehäuse untergebracht sind und damit eine freie Luftzirkulation eingeschränkt ist, würde eine rein passive Kühlung bei längerem Betrieb zur Überhitzung und damit zur Beschädigung bzw. zum Abschalten bestimmter Hardwarekomponenten führen. Daher ist eine **aktive Temperaturregulierung** laut Anforderung **R035** obligatorisch für einen sicheren und stabilen Einsatz des Systems.

Für eine aktive Temperaturregulierung werden bei Desktop-Computern **Gehäuselüfter** eingesetzt, welche kalte Luft in das Gehäuse bzw. warme Luft aus dem Gehäuse führen. Die gleiche Art von Lüftern kann für den Tischroboter verwendet werden. Gehäuselüfter besitzen Befestigungslöcher, sodass diese in das Gehäuse des Roboters integriert werden können. Damit die Leistung des Lüfters angepasst werden kann, besitzen viele Gehäuselüfter die Möglichkeit, die gewünschte Drehzahl über **Pulsweitenmodulation**• (PWM) softwaretechnisch zu regulieren. Für diese Steuerung wird gegebenenfalls eine elektronische Schaltung benötigt, die auf einem Arduino basieren kann (siehe Punkt 4.1.12).

Für die Konzepterstellung werden die in Tabelle 10 dargestellten Lüfter in verschiedenen Größen ausgewählt⁴⁰. Alle ausgewählten Lüfter liegen dabei laut Herstellerangaben innerhalb der in Anforderung **R037** definierten Lautstärkegrenze von 30 dB und besitzen die Möglichkeit, über PWM reguliert zu werden. Der erste Lüfter ist ein Produkt der österreichischen Firma *Noctua*⁴¹, die anderen stammen aus dem Sortiment des deutschen Herstellers *be quiet!*⁴².

Tabelle 10: Vergleich von Gehäuselüftern mit PWM-Regulierung

Lüfterserie	Durchmesser	max. Volumenstrom	max. Geräuschpegel	max. Leistungsaufnahme	Preis ⁴³
Noctua NF-A	60mm	29,2 m³/h	19,3 dB	1 Watt	13,90 €
be quiet!	80mm	44,5 m³/h	19,2 dB	1,6 Watt	6,99 €
Pure Wings 2	92mm	56,02 m³/h	19,6 dB	1,8 Watt	7,90 €
be quiet!	120mm	124,58 m³/h	28,6 dB	4,5 Watt	18,95 €
Silent Wings 3	140mm	131,79 m³/h	28,1 dB	6 Watt	19,95 €

⁴⁰ Ein detaillierterer Vergleich der Lüfter inklusive der Quellen ist in der Datei *Hardware Komponenten Vergleich.xlsx* zu finden.

⁴¹ Offizielle Seite: <https://noctua.at/> (abgerufen am 01.08.2019)

⁴² Offizielle Seite: <https://www.bequiet.com/de> (abgerufen am 01.08.2019)

⁴³ Stand Juni 2019. Alle Preise stammen von Reichelt, wo die Lüfter am günstigsten angeboten wurden.

4.1.8 Energieversorgung

Die Energieversorgung soll laut Anforderung **R017** über ein handelsübliches Netzteil erfolgen, welches den benötigten Strom für den Tischroboter bereitstellt. Als handelsüblich werden Netzteile verstanden, die aus dem Alltag bekannt sind. Es gibt USB-Netzteile, welche für das Laden von Smartphones verwendet werden, jedoch für ein größeres System zu wenig Leistung bereitstellen. Daher bieten sich **Tischnetzteile** mit Hohlsteckeranschlüssen an, welche für den Betrieb von Laptops verwendet werden und wesentlich höhere Leistungen abgeben können als USB-Netzteile.

Tischnetzteile gibt es in vielen verschiedenen Ausführungen. Tabelle 11 führt eine Auswahl von Netzteilen mit verschiedenen **Spannungsbereichen** und **Leistungsstufen** auf. Die endgültige Auswahl eines passenden Netzteils ist von allen Komponenten und von der benötigten Gesamtleistung des Systems abhängig und wird daher im Konzept getroffen.

Tabelle 11: Übersicht über die Kennzahlen verschiedener Tischnetzteile

Spannung	max. Leistung	Preis ⁴⁴
5V	30W	18,40 €
	40W	17,95 €
9V	54W	19,95 €
	42W	8,55 €
12V	60W	9,50 €
	78W	11,40 €
	90W	27,10 €

Für einen potenziellen **Akkubetrieb**, welcher in Anforderung **R018** als wünschenswert formuliert wird, bietet sich die Verwendung einer Powerbank an. Powerbanks sind als Stützakkus für Smartphones und Tablets bekannt und bieten den Vorteil, dass die Lade- und Entladeschaltung direkt integriert ist, wodurch der Aufwand für die Entwicklung einer entsprechenden elektronischen Schaltung ausbleibt. Gleichzeitig sind erwerbbarer Powerbanks geprüfte Geräte, die einen gefahrlosen Betrieb gewährleisten, was für die Verwendung im Tischroboter unabdingbar ist. Sie sind zudem einfach austauschbar, was einen weiteren Vorteil darstellt. Bei der Wahl der Powerbank muss die Kapazität beachtet werden, sodass die in **R019** geforderte Laufzeit von zwei Stunden erreicht wird. Außerdem muss die Powerbank ausreichend Leistung bereitstellen können, um alle Komponenten des Systems mit Energie zu versorgen. Die Auswahl einer konkreten Powerbank ist Teil der Konzeptionierung.

⁴⁴ Stand Juni 2019. Alle Preise stammen vom Distributor Reichelt, bei dem ein übersichtliches Angebot unterschiedlicher Netzteile angeboten wurde.

4.1.9 Einschalter

Laut Anforderung **R020** soll das System über einen **Einschalter** sowohl gestartet als auch ausgeschaltet werden können. Für den Einschalter kommen zwei verschiedene Typen infrage: Zum einen gibt es Umschalter, beispielsweise **Kippschalter**, die zwischen zwei Zuständen wechseln können. Zum anderen gibt es **Taster**, welche aktiv betätigt werden können. Ein Taster bietet im Gegensatz zum Umschalter die Möglichkeit, verschiedene Arten von Interaktionen zu unterscheiden. Beispielsweise können kurzes, langes oder doppeltes Betätigen voneinander differenziert werden. Durch die größere Nutzungsvielfalt wird für das Projekt daher ein Taster gewählt, ähnlich wie es beim NAO-Roboter gehandhabt wird. Bei Tastern gibt es ebenfalls unterschiedliche Arten: Einen mechanischen Drucktaster oder einen kapazitiven Sensor als Taster.

Ein **mechanischer Drucktaster** könnte fest in das Gehäuse eingebaut werden und würde den Vorteil bieten, dass der Nutzer direkt ein taktiles Feedback bei der Interaktion spürt. Dieser positive Aspekt der mechanischen Betätigung kann jedoch auch als Nachteil aufgeführt werden. So ist aus Projekterfahrungen bekannt, dass mechanische Taster teilweise nicht reagieren, wenn kein ausreichend großer Druck aufgewendet wurde. Gerade im gedachten Einsatzfeld des Tischroboters – der Unterstützung von Senioren – ist ein robustes und stabiles System besonders wichtig, wodurch solche Ausfälle nicht toleriert werden können. Außerdem benötigen Taster generell etwas Kraft zur Betätigung, was ebenfalls als Nachteil aufgefasst werden kann.

Ein **kapazitiver Taster** würde die Nachteile des Drucktasters nicht besitzen. Er benötigt zur Interaktion nur eine Berührung bzw. eine Annäherung eines Fingers. Über die Einstellung der Sensibilität wird auch eine zuverlässige Interaktion gewährleistet werden können. Jedoch wird hinter einem kapazitiven Sensor eine aufwändigere Elektronik zur Zustands-Auswertung des Tasters benötigt. Außerdem existiert bei dem reinen Taster keine Rückmeldung während der Betätigung.

Für das Projekt wird der **kapazitive Sensor** aufgrund der leichteren Bedienung präferiert und für die Umsetzung ausgewählt. Für die Auswertung des Sensors wird gegebenenfalls eine elektronische Schaltung benötigt, die von einem Arduino gesteuert werden kann (siehe Punkt 4.1.12).

4.1.10 Augen

Die Anforderungen an das Design formulieren in **R039**, dass der Roboter Augen darstellen soll. Für die Darstellung von Augen existieren verschiedene Möglichkeiten, die bei bekannten Robotern eingesetzt werden. Die folgenden drei Augentypen sind in Abbildung 15 dargestellt.

NAO und Pepper besitzen **statische Augen**, die schwarz glänzend sind, sich aber weder bewegen noch verändern können. Um Emotionen anzudeuten, ist bei diesen Robotern um die Augen ein LED-Ring vorhanden, welcher je nach Stimmungslage farblich angepasst wird und auch ein Augenzwinkern andeuten kann. Diese Art der Augen ist sowohl von der Hardware als auch von der Software am einfachsten umzusetzen.

Eine andere Möglichkeit ist eine **mechanische Umsetzung der Augen** und gegebenenfalls auch der Lider, sodass diese real bewegt werden können. Diese Methode kommt beispielsweise beim Roboter *Reeti* zum Einsatz [43]. Die Beweglichkeit dieser Art von Augen ist kompliziert umzusetzen, aber auch am menschenähnlichsten.

Die dritte Möglichkeit besteht darin die Augen **grafisch** auf einem **Display** darzustellen. Diese Methode wird unter anderem beim Roboter *Cozmo* der Firma *Anki*⁴⁵ zum Einsatz gebracht und erlaubt eine flexible Andeutung verschiedener Emotionen [44]. Diese Möglichkeit ist einfach umzusetzen und erlaubt dabei trotzdem eine große, gestische Vielfalt.

Abbildung 15: Verschiedene Umsetzungen von Roboteraugen – Pepper [12], Reeti [43] und Cozmo [44]

Für dieses Projekt kommen die erste und die dritte Umsetzungsmöglichkeit infrage, da die Entwicklung mechanischer Augen mit einem höheren Designaufwand verbunden wäre und daher nicht weiter betrachtet wird. Statische Augen wie beim Pepper-Roboter sind ein Aspekt des Roboterdesigns und benötigen keine weiteren Hardwarekomponenten.

Für Augen, die über ein Display dargestellt werden, würde sich die Verwendung eines **kompakten Bildschirmmoduls** anbieten, beispielsweise das in Abbildung 16 dargestellte OLED-Display der Firma Adafruit. Das Display besitzt eine Auflösung von 128 x 64 Pixel bei einer Bildschirmdiagonale von 3,3 Zentimetern und ist damit vergleichbar mit dem Display des Cozmo-Roboters. Die Auflösung ist ausreichend, um detaillierte Augen, aber auch andere grafische Elemente und Text darzustellen.

Abbildung 16: 1,3 Zoll OLED-Display (128 x 64 Pixel) [45]

Das Display ist für einen Preis von 20 € verfügbar und kann mittels der Schnittstellen I²C oder SPI gesteuert werden, welche beide auf dem Jetson Nano verfügbar sind.

⁴⁵ Offizielle Seite: <https://www.anki.com/> (abgerufen am 01.08.2019; die Firma hat zu diesem Zeitpunkt bereits angekündigt, aufgrund finanzieller Probleme keine weiteren Produkte zu produzieren und zu vertreiben)

4.1.11 Bewegungsmöglichkeiten

Die Anforderungen **R040** und **R041** formulieren die wünschenswerte **Bewegungsfähigkeit des Roboter-Kopfes**. Damit würde es möglich sein, dass der Roboter Menschen direkt angucken und einfache Formen von Gestik zeigen kann. Außerdem wäre der Sichtbereich der Kamera vergrößert, wenn diese wie in **R042** beschrieben, mitbewegt wird.

Bei der Bewegung des Kopfes ist ein genaues Einstellen der **Drehposition** wichtig, damit festgelegte Positionen eingenommen werden können. Dafür bieten sich **Servomotoren** an, bei denen der gewünschte Drehwinkel über PWM-Signale[•] eingestellt werden kann. Andere Motoren wie kontinuierlich drehende Gleichstrommotoren oder Schrittmotoren würden eine zusätzliche Logik zur Feststellung des Drehwinkels benötigen und werden daher für das Projekt nicht in Erwägung gezogen.

Servomotoren gibt es in unterschiedlichen Stärken. Je nach Gewicht und Größe des Kopfes benötigt der Servomotor ein ausreichend großes **Drehmoment**[•], um diesen zuverlässig bewegen zu können. Neben der Stärke unterscheiden sich die Servomotoren in ihrem möglichen **Drehbereich**. Der Drehbereich liegt bei vielen Servos bei einem vollen Kreis von 360°, einige sind jedoch auf 180° beschränkt, was für eine flexible Bewegung des Kopfes ebenfalls ausreichend ist.

Das Angebot von Servomotoren für den Modellbau bzw. für Robotik-Projekte ist relativ umfassend. Folgende Tabelle 12 gibt eine grobe Übersicht von Motoren verschiedener Stärken, die im Preissegment von unter 40 € liegen, um das festgelegte Budget nicht zu überschreiten⁴⁶.

Tabelle 12: Übersicht verschiedener Servomotoren

Art des Motors	Servomotor			
max. Drehmoment	1,3 kg/cm	4,4 kg/cm	6,7 kg/cm	21,5 kg/cm
Preis ⁴⁷	5,95 €	9,95 €	12,95 €	19,30 €

Die endgültige Auswahl eines Servomotors ist vom Design des Roboters und von der Größe des Kopfes abhängig und daher Teil der Umsetzung. Für die Ansteuerung der Motoren wird gegebenenfalls eine elektronische Schaltung benötigt, die auf einem Arduino basieren kann (siehe Punkt 4.1.12).

⁴⁶ Ein detaillierterer Vergleich von Servomotoren inklusive der Quellen ist in der Datei *Hardware Komponenten Vergleich.xlsx* zu finden.

⁴⁷ Stand Juni 2019. Alle Preise stammen von den Distributoren Reichelt und Exp-Tech, die eine übersichtliche Auswahl unterschiedlicher Servomotoren anbieten.

4.1.12 Elektronische Schaltungen

Im System können für die Steuerung bestimmter Komponenten zusätzliche, elektronische Schaltungen eingesetzt werden. Teilweise ist der Jetson Nano für bestimmte, hardwarenahe Aktionen nicht geeignet. Zum einen könnten zeitkritische Aufgaben unzuverlässig ausgeführt werden, was in der quasi-zufälligen Verteilung von Ressourcen durch das Betriebssystem begründet ist. Zum anderen würden bestimmte, unkomplizierte Aufgaben dauerhaft Rechenleistung erfordern. Separate Schaltungen könnten insofern bestimmte Teile des Systems komplett autonom übernehmen oder lediglich über eine Schnittstelle mit der zentralen Recheneinheit verbunden sein, um Steuerbefehle zu erhalten.

Als Basis von separaten, elektronischen Schaltungen bieten sich **Arduino Nano** Boards an. *Arduino*⁴⁸ ist ein quelloffenes Projekt, welches verschiedene Produkte für eine komfortable, hardwarenahe Programmierung anbietet. Der Arduino Nano (siehe Abbildung 17) ist ein kompaktes und energiesparendes Board, welches über USB programmiert werden kann und ca. 20 digitale Ausgänge für die Steuerung von elektronischen Komponenten bietet. Außerdem sind auf dem Board analoge Eingänge für Analog-Digital-Umsetzung und PWM-Ausgänge für die Steuerung von Motoren vorhanden. Der Arduino besitzt lediglich 32 Kilobyte Flash- und 2 Kilobyte Arbeitsspeicher, was allerdings für einfache, hardwarenahe Programme ausreichend ist. Über die Schnittstellen I²C, SPI oder UART kann der Arduino Nano mit dem Jetson Nano verbunden werden. [46]

Abbildung 17: Arduino Nano Board [46]

⁴⁸ Offizielle Seite: <https://www.arduino.cc/> (abgerufen am 01.08.2019)

4.2 Auswahl der Software

Nachdem die Kernkomponenten der Hardware ausgewählt wurden, sollen hier grundlegende Entscheidungen über die verwendete Software getroffen werden. Alle hier und im Laufe des Projekts ausgewählten Softwarekomponenten sind frei von einmaligen und laufenden Lizenzkosten, gemäß den Anforderungen **R021** und **R022**.

Durch die Auswahl des Jetson Nano als zentrale Recheneinheit für den Tischroboter fällt die Wahl des **Betriebssystems** auf das von Nvidia bereitgestellte, offizielle Image. Dieses basiert auf Ubuntu 18.04, das als Long-Term-Support Version bis zum Jahr 2023 unterstützt wird und offizielle Updates erhält [47]. Ubuntu selbst ist als freie Software und somit ohne Lizenzkosten erhältlich [48]. Durch das Betriebssystem werden die Anforderungen **R023**, dass mehrere Programme parallel ausgeführt werden können, und **R024**, dass eine grafische Oberfläche auf einem angeschlossenen Display angezeigt werden kann, implizit erfüllt. Ebenfalls erlaubt es einen Zugriff über SSH, wie in **R006** gefordert wird.

Das bereitgestellte Image von Nvidia erweitert das Betriebssystem direkt um grundlegende **Softwarepakete** und **Frameworks**, welche die Arbeit mit dem Board erleichtern und die Vorzüge des Grafikprozessors zugänglich machen sollen. Als Beispiele sind *JetPack* und *TensorFlow* für künstliche Intelligenz und maschinelles Lernen, *CUDA Toolkit* für die Verwendung der GPU und *OpenCV* für Bildverarbeitung zu nennen. [34]

Als weiteres, freies Framework soll die Verwendung von dem Framework **ROS** (Abkürzung für *Robot Operating System*) in Erwägung gezogen werden, welches speziell für die Entwicklung von Applikationen der Robotik gedacht ist. Dieses bietet unter anderem Möglichkeiten der Hardwareabstraktion und eine Nachrichtenvermittlung im Publisher-Subscriber-Modell. Dadurch soll das Zusammenspiel von Hard- und Software und die Verknüpfung verschiedener Programmkomponenten vereinfacht werden. ROS besitzt den Vorteil, dass es sowohl Python als auch C++ als Programmiersprache unterstützt, und somit HelloWorld*-Programme nach Anforderung **R030** entwickelt werden können. [49]

Für die Verwendung des Displays ist eine **grafische Anwendung** notwendig, welche die Möglichkeiten des Roboters zugänglich machen. Bei der Entwicklung moderner Oberflächen haben sich in den letzten Jahren vor allem Webanwendungen durchgesetzt, welche über HTML, CSS und JavaScript vielfältige Möglichkeiten für zeitgemäße Gestaltung und Nutzerinteraktionen bieten. Gegenüber lokalen Desktop-Anwendungen besitzen Webanwendungen den Nachteil, dass der Zugriff auf lokale Dienste und Ressourcen kompliziert bzw. nicht möglich ist. Um sowohl die Vorteile von Weboberflächen als auch von lokalen Desktop-Anwendungen nutzen zu können, kann das freie **Electron-Framework** für den Roboter verwendet werden, welches bei der Erstellung von Oberflächenanwendungen beide Aspekte miteinander verknüpft. Das Framework wird beispielsweise in bekannten Client-Programmen wie *Visual Studio Code* oder *Slack* benutzt und bietet die Möglichkeit, plattformübergreifende, lokale Anwendungen mit den Sprachen der Webprogrammierung zu entwickeln. [50]

4.3 Finanzieller Zwischenstand

In den vorherigen Abschnitten dieses Kapitels wurden Hauptkomponenten für die Umsetzung des Tischroboters ausgewählt bzw. Optionen für bestimmte Aspekte evaluiert. Dieser Abschnitt soll einen finanziellen Zwischenstand des Projekts anhand der ausgewählten Hardware geben. Tabelle 13 führt die einzelnen Teile mit dem zugehörigen Preis beim angegebenen Distributor auf. Eine detaillierte Übersicht inklusive Quellen ist in der beigefügten Datei *Einkaufsliste.xlsx* zu finden.

Der Zwischenstand bezieht sich auf den aktuellen Recherchestand zum Bearbeitungszeitpunkt der Arbeit⁴⁹. Die Mehrzahl der evaluierten Komponenten wird auch in näherer Zukunft für die Verwendung bei einem Tischroboter geeignet sein. Lediglich die zentrale Recheneinheit ist eine aktuelle Option, welche in den nächsten Jahren durch modernere Alternativen abgelöst werden könnte.

Tabelle 13: Finanzieller Zwischenstand der in der Evaluation ausgewählten Komponenten

Teil des Systems	ausgewählte Komponente	Distributor	Preis ⁵⁰
zentrale Recheneinheit	Jetson Nano	Nvidia	109 €
Bildschirm	7 Zoll Touch Display	Pollin	49,95 €
Geräuschwahrnehmung	ReSpeaker	Reichelt	57,20 €
Lautsprechersystem	Lautsprecher von Visaton	Reichelt	2 x bis 25 €
	20 Watt Audioverstärker	Reichelt	19,40 €
Kamera	Raspberry Pi Kamera V2	Reichelt	24,50 €
Temperaturregulierung	Gehäuselüfter	Reichelt	bis 20 €
Energieversorgung	Netzteil	Reichelt	bis 27 €
Augen	1,3 Zoll OLED Display	Reichelt	19,40 €
Bewegungsmöglichkeiten	2 x Servomotor	Reichelt / Exp-Tech	2 x bis 20 €

Die ausgewählten Hauptkomponenten bzw. Optionen für die spätere Auswahl im Konzept liegen insgesamt bei einem Preis von etwa 420 €. Je nach den endgültig getroffenen Entscheidungen wird dieser Wert noch schwanken. Ebenfalls werden im Konzept durch Zubehör wie Kabel und Adapter, elektronische Bauteile und dem Gehäuse weitere Kosten anfallen.

⁴⁹ Juni / Juli 2019

⁵⁰ Stand Juni 2019

5. Konzept des Gesamtsystems

Nachdem im letzten Kapitel grundlegende Komponenten des Projekts ausgewählt wurden, wird in diesem Kapitel das Umsetzungskonzept für das Gesamtsystem erstellt und beschrieben.

Im ersten Teil 5.1 wird das Konzept der Hardware, anschließend in Abschnitt 5.2 das Konzept der Software beschrieben. Als Ergebnis dieses Kapitels soll die Verknüpfung aller Komponenten und damit der grundlegende Aufbau des Systems feststehen, sodass die notwendigen Funktionalitäten ermöglicht und die gestellten Anforderungen an den Roboter erfüllt werden.

5.1 Hardwaretechnische Systemaspekte

Zunächst wird die Hardware des Systems konzeptionell erläutert. Dazu gehört vor allem die Verknüpfung der ausgewählten Hardwarekomponenten, um eine **Kommunikation** zwischen den verschiedenen Teilen des Systems zu ermöglichen. Zusätzlich muss auch die **Energieversorgung** aller Komponenten für einen reibungslosen Betrieb beachtet werden. Bei einigen Punkten, unter anderem dem Einschalter und der Temperaturregelung, werden neben der Kommunikation zwischen den Komponenten auch logische Abläufe beschrieben, welche die Hardware während des Betriebs betreffen.

Die Konzepterstellung basiert auf den evaluierten Komponenten. Der aktuelle Zwischenstand der Systemarchitektur inklusive der ausgewählten Optionen ist in folgender Abbildung 18 zu sehen. In der Abbildung sind die Abschnittsnummern aufgeführt, in denen die jeweiligen Systemkomponenten genauer erläutert werden.

Die Kamera, das Display und die Geräuschwahrnehmung werden nicht in eigenen Abschnitten erläutert. Sowohl die Kommunikation als auch die Energieversorgung sind bei diesen Komponenten durch die Schnittstelle vorgegeben und bedürfen keiner weiteren konzeptionellen Überlegung. Die Raspberry Pi Kamera wird über die CSI-Schnittstelle und das in Punkt 4.1.4 ausgewählte ReSpeaker-Mikrofonarray* über USB direkt mit der zentralen Recheneinheit verbunden. Das Display erhält die auszugebenden Signale über HDMI und die Energieversorgung sowie die Touch-Eingaben werden über ein USB-Kabel realisiert.

Im letzten Punkt werden die Ergebnisse der hardwaretechnischen Systemarchitektur in einer erweiterten Grafik zusammengefasst.

Abbildung 18: Systemarchitektur mit den in der Evaluation ausgewählten Komponenten

5.1.1 Konnektivität und Anschlüsse des Roboters

Die Konnektivität und die Anschlüsse des Roboters begrenzen bzw. erweitern den Umfang späterer Anwendungsfälle, aber auch die Möglichkeiten während des Entwicklungsprozesses. Die Konnektivität des Roboters umfasst vor allem den **Anschluss** an das **Internet**. Aber auch andere Funktechnologien wie Bluetooth können die Konnektivität des Roboters ausweiten. Besonders der Bereich der Gebäudeautomation, in welchem der Roboter als Steuerzentrale eingesetzt werden könnte, setzt eine drahtlose Konnektivität voraus. Nach Anforderung **R004** ist eine Netzwerkverbindung über WLAN obligatorisch, **R003** und **R005** definieren die Verbindungsmöglichkeiten über LAN und Bluetooth als wünschenswert.

Der Jetson Nano stellt standardmäßig nur einen Gigabit-Ethernet Anschluss zur Verfügung, über den eine kabelgebundene Internetverbindung hergestellt werden kann. Wie bereits beschrieben, besitzt das Board jedoch einen M.2 Key E Erweiterungssteckplatz für ein **WLAN-Modul**. Im Konzept wird diese Erweiterungsmöglichkeit genutzt, um die Netzwerkverbindung über WLAN nach Anforderung **R004** gewährleisten zu können.

Im Community-Forum *JetsonHacks*⁵¹, welches verschiedenste Entwicklungsthemen rund um die Nvidia Jetson Boards diskutiert, ist ein Artikel zur Erweiterung des Nano Jetson mit WLAN und Bluetooth zu finden. Darin wird die Verwendung des 8265NGW WLAN-Moduls von *Intel*⁵² empfohlen. Die Kommentare verschiedener Nutzer zum Artikel zeigen, dass andere Modelle des WLAN-Modules nicht problemlos mit dem Jetson Nano kompatibel sind. Mit dem Intel 8265NGW Modul erhält der Jetson Nano nicht nur WLAN im 802.11ac Standard, sondern auch Bluetooth 4.2, wodurch neben Anforderung **R004** auch **R005** erfüllt ist. [51]

Für eine optimale Funktionsfähigkeit werden zwei Antennen benötigt, die an das Modul angeschlossen werden. Als Anschluss wird dafür ein MHF4-Stecker genutzt, worauf bei der Wahl der Antennen geachtet werden muss. Es können entweder kompakte Folienantennen oder aber größere Dipolantennen verwendet werden, die sensibler sind und eine höhere Empfangsreichweite erlauben. Da der Tischroboter eine gewisse Größe haben wird, können die größeren Antennen voraussichtlich problemlos im Gehäuse untergebracht werden, weswegen diese aufgrund der vorteilhaften Eigenschaften verwendet werden sollen.

Weitere Funktechnologien können mithilfe von externen Boards erweitert und an den I²C-Bus des Jetson Nano angeschlossen werden. So kann beispielsweise ein LoRaWan- oder ZigBee-Modul bei Bedarf hinzugefügt werden und die Funktionalität des Roboters erweitern.

Bei der Erfüllung von Anforderung **R003**, dass die Netzwerkanbindung des Systems über LAN möglich ist, muss die Positionierung des Jetson Nano Moduls beachtet werden. Da die Anschlüsse des Boards im Gehäuse verborgen sein können, muss der Ethernet-Anschluss mit einer Verlängerung aus dem Gehäuse herausgeführt werden.

⁵¹ Siehe <https://www.jetsonhacks.com/> (abgerufen am 01.08.2019)

⁵² Offizielle Seite: <https://www.intel.de/content/www/de/de/homepage.html> (abgerufen am 01.08.2019)

Neben dem Ethernet-Anschluss können auch USB-Ports aus dem Gehäuse herausgeführt werden. Dadurch wäre es für Anwender möglich, Peripheriegeräte wie eine Tastatur oder auch externe Datenträger anzuschließen.

In folgender Abbildung 19 ist das konzeptionelle Schema zu den Anschlüssen und der Konnektivität des Tischroboters dargestellt.

Abbildung 19: Konnektivität und Anschlüsse des Systems

5.1.2 Einschalter

Der Einschalter ist die Interaktionsschnittstelle für den Menschen, um das System starten und ausschalten zu können. Wie bereits in der Evaluation der Komponenten unter 4.1.9 erläutert, wird ein **kapazitiver Sensor** als Einschalter gewählt, da dieser eine vielfältigere und angenehmere Interaktion ermöglicht als ein mechanischer Umschalter oder ein mechanischer Drucktaster. Während das Einschalten über die Aktivierung der Stromzufuhr gesteuert werden könnte, müssen beim Ausschalten des Systems mehr Aspekte beachtet werden als die bloße Unterbrechung der Energie. Beispielsweise muss die zentrale Recheneinheit korrekt heruntergefahren werden, um einen potenziellen Datenverlust zu vermeiden. Insofern wird eine im System integrierte Logik zur Umsetzung von Interaktionen mit dem Einschalter benötigt, welche in diesem Abschnitt erläutert werden soll.

In folgender Abbildung 20 wird zunächst der prinzipielle Ablauf an **Systemzuständen** dargestellt. Initial befindet sich das System im ausgeschalteten Zustand. Über ein kurzes Betätigen und Halten des Tasters für zwei Sekunden kann das System gestartet werden. Die Haltezeit sorgt dafür, dass das System nicht unabsichtlich gestartet wird. Beim Starten des Systems werden zunächst alle Komponenten initialisiert, bevor es in den aktiven, arbeitenden Zustand übergeht. Das Ausschalten des Systems kann über Interaktionen mit der grafischen Oberfläche des Roboters realisiert werden und muss daher nicht über den Einschalter erfolgen. Grundsätzlich soll es jedoch immer möglich sein, über ein längeres Betätigen des Einschalters das System zwanghaft auszuschalten. Diese Option ist beispielsweise für Systemabstürze relevant.

Abbildung 20: Zustandsübergangsdiagramm für den Einschalter des Systems

Zu beachten ist, dass der ausgeschaltete Zustand nicht mit einem komplett von der Energie getrennten System gleichzusetzen ist. In diesem Ausgangszustand muss mindestens die Interaktion mit dem Einschalter wahrgenommen werden. Außerdem müssen auch andere Aspekte beachtet werden, beispielsweise dass während des ausgeschalteten Zustands trotzdem ein Akku geladen werden kann⁵³. Auf der anderen Seite sollen Systemkomponenten wie die zentrale Recheneinheit, der Bildschirm und die Kamera im ausgeschalteten Zustand auch wirklich deaktiviert sein und keine Energie verbrauchen.

⁵³ Siehe Abschnitt 5.1.8 zu den Details der Energieversorgung

Die Auswertung des Tasters sowie die Steuerung des Systemzustands soll von einem separaten **Arduino Nano** übernommen werden. Dieser kann zwischen verschiedenen Interaktionsarten unterscheiden und entsprechende Zustandsübergänge initiieren. Im aktiven Systemzustand besteht auch die Möglichkeit, dass der Arduino einfache oder doppelte Interaktionen mit dem Einschalter erkennt und an die zentrale Recheneinheit signalisiert, um spezielle Aktionen auszuführen.

Grundsätzlich basieren **kapazitive Sensoren** auf der Erfassung von Kapazitätsschwankungen über eine zugehörige Zeitmessung. Eine solche Zeitmessung kann beispielsweise vorgenommen werden, indem ein Ausgangs-Pin und ein Eingangs-Pin eines Arduinos über einen großen Widerstand miteinander verbunden werden und am Eingang zusätzlich eine Metallfolie angebracht wird. Das Prinzip ist in folgender Abbildung 21 dargestellt. Der Eingang besitzt eine feste, leitungsbedingte Grundkapazität, während die Metallfolie eine veränderliche Kapazität darstellt. Schaltet der Ausgang auf einen hohen Spannungspegel, wird durch den Widerstand eine bestimmte Zeit benötigt, bis die Kapazität des Eingangs geladen ist.

Abbildung 21: Prinzip eines kapazitiven Sensors mit einem Arduino

Verändert sich nun die Kapazität der Eingangsseite, indem ein Finger über die Metallfolie gehalten wird, verändert sich ebenfalls die benötigte Zeit zum Laden dieser Kapazität. Die **Zeitschwankung** kann ausgewertet werden und lässt Rückschlüsse über den Status des Sensors zu. Für die beschriebene Zeitmessung ist bereits eine simple Arduino-Bibliothek verfügbar. [52]

Für die Auswertung eines kapazitiven Sensors existieren ebenfalls spezielle Schaltkreise, beispielsweise der **CAP1298** der Firma *Microchip*⁵⁴. Im Vergleich zur Umsetzung mit einem bloßen Arduino bietet der Chip die Möglichkeiten, die Sensitivität softwaretechnisch einzustellen, sich automatisch zu rekalisieren und verschiedene Interaktionsarten zu erkennen. Außerdem kann der Schaltkreis Interrupts* ausgeben, falls sich der Status des Sensors verändert. Dadurch müsste der Arduino nicht dauerhaft aktiv sein, um den Sensor abzufragen. [53]

Sowohl die Realisierung mit dem Arduino als auch die Verwendung des CAP1298 sollen in der praktischen Umsetzung getestet werden.

⁵⁴ Offizielle Seite: <https://www.microchip.com/> (abgerufen am 01.08.2019)

Das Erscheinungsbild des Sensors ist Teil des Gehäusedesigns. So würde es ausreichen, die Position über eine Erhebung oder eine bestimmte grafische Gestaltung anzudeuten. Damit der Anwender ein Feedback bei der Interaktion mit dem Einschalter bekommt, kann der Sensor um **visuelle und taktile Rückmeldung** erweitert werden. Für die visuelle Rückmeldung kann ein LED-Ring eingesetzt werden, der den Sensor umgibt und je nach Interaktion und Systemzustand verschiedene Farben annimmt. Für die taktile Rückmeldung bietet sich ein Vibrationsmotor an, welcher am Gehäuserand angebracht ist und den Nutzer die Interaktion spüren lässt. Folgende Abbildung 22 zeigt den schematischen Aufbau des Einschalters.

Abbildung 22: Schematischer Aufbau des Einschalters

5.1.3 Berührungssensoren

Da bereits für den Einschalter ein kapazitiver Sensor vorgesehen ist, kann das Prinzip für weitere **Berührungssensoren** vorgesehen werden. Diese könnten über den Körper des Roboters verteilt werden und diesem erlauben, auf Berührungen von Menschen reagieren zu können.

Der für den Einschalter zur Auswahl stehende CAP1298 Chip besitzt acht Kanäle für mögliche Sensoren. Bei der Umsetzung mit dem Arduino sind ebenfalls über zehn angeschlossene Sensoren möglich. Die genaue Anbringung weiterer Berührungssensoren ist ein Aspekt des Roboter-Designs.

5.1.4 Temperaturregulierung und -messung

In Abschnitt 4.1.7 wurden Gehäuselüfter für eine aktive **Kühlung** des Tischroboters ausgewählt, die innerhalb des geschlossenen Gehäuses für einen sicheren Betrieb aller Komponenten notwendig ist. Dieser Abschnitt erläutert den Aufbau und die Steuerung der Temperaturregulierung.

Damit die Leistung der Lüfter in Abhängigkeit von der inneren Gehäusetemperatur angepasst werden kann, soll die **Regulierung** um eine **Temperaturerfassung** erweitert werden. Als Basis der Temperaturregulierung bietet sich erneut ein Arduino Nano an, welcher die Sensorwerte ausliest und den Lüfter entsprechend steuert. Für die Temperaturerfassung wird ein digitaler **Si7021**-Sensor von *Silicon Labs*⁵⁵ verwendet, welcher bereits aus anderen Telematik-Projekten bekannt ist und sich vom Arduino oder auch vom Jetson Nano über I²C auslesen lässt. Abhängig von der Temperatur kann die Leistung des Lüfters über ein entsprechendes PWM-Signal* angepasst werden. Folgende Abbildung 23 zeigt den schematischen Aufbau der Temperaturregulierung.

Abbildung 23: Schematischer Aufbau der Temperaturregulierung

In folgender Abbildung 24 sind die **Übergänge der Lüfterzustände** dargestellt. Die Leistung der Lüfter wird in Abhängigkeit von der gemessenen Temperatur in vier möglichen Zuständen angepasst. Dabei ist zu beachten, dass die Grenzwerte der Temperatur bei den Zustandsübergängen davon abhängig sind, ob das Gehäuse gerade wärmer wird oder sich abkühlt. Diese Hysterese bezweckt, dass keine hochfrequenten Wechsel zwischen verschiedenen Leistungsstufen auftreten.

Abbildung 24: Leistungsübergänge der Temperaturregulierung

⁵⁵ Offizielle Seite: <https://www.silabs.com/> (abgerufen am 01.08.2019)

Für den konkreten Lüfter stehen die in Tabelle 10 (Abschnitt 4.1.7) aufgeführten Alternativen zur Auswahl. Am effektivsten ist die Verwendung von **zwei Lüftern**, so wie in Abbildung 25 dargestellt. Dabei würde ein Lüfter an der Unterseite kühle Luft in das Gehäuse führen, der zweite Lüfter an der Oberseite des Roboters würde die erwärmte Luft aus dem Gehäuse herausführen. Bei der Anordnung ist darauf zu achten, dass der Luftstrom alle wärmeproduzierenden Komponenten erfasst und keine Luftstaus entstehen. Außerdem ist die Verwendung von Staubfiltern sinnvoll. Zum einen, um zu verhindern, dass Staub oder andere Fremdkörper in das Gehäuse gelangen, zum anderen, um zu vermeiden, dass die Anwender mit Fingern in die Lüfter gelangen können.

Abbildung 25: Luftstrom im Gehäuse bei zwei Lüftern

Bei der voraussichtlichen Größe des Gehäuses sind zwei 80mm-Lüfter für die Temperaturregelung ausreichend. Diese sind die günstigste Alternative und bieten trotzdem eine gute Leistung. Möglicherweise wäre auch die Verwendung eines einzigen Lüfters bei der Größe des Tischroboters ausreichend. Die genaue Auswahl und Anbringung der Lüfter ist Teil des Roboter-Designs und muss in praktischen Tests evaluiert werden.

5.1.5 Bewegungsmöglichkeiten

Die Bewegungsmöglichkeiten des Roboters fokussieren sich auf die **Bewegung des Kopfes**, welcher laut Anforderungen **R040** und **R041** sowohl gedreht als auch geneigt werden können soll. Über diese zwei Drehachsen wäre der Roboter in der Lage, Menschen direkt angucken zu können. Außerdem kann über Kopfbewegungen einfache Gestik dargestellt werden, beispielsweise ein Kopfnicken oder -schütteln. Für einen Roboter wären auch noch weitere Bewegungsfreiheiten denkbar, beispielsweise die Fähigkeit, den gesamten Körper inklusive des Displays zu drehen. Diese zusätzlichen Möglichkeiten könnten bei Bedarf in Folgeprojekten umgesetzt werden und werden im Rahmen dieser Arbeit nicht betrachtet. Ebenfalls ist eine dritte Drehachse für die Kopfbewegung nicht notwendig.

Je Drehachse wird ein separater **Servomotor** verwendet, welcher ein Einstellen des gewünschten Drehwinkels ermöglicht. Ähnlich wie bei den Gehäuselüftern wird die Position des Servomotors über PWM-Signale eingestellt, weswegen sich erneut die Verwendung eines Arduinos für die Steuerung der Motoren anbietet. Die in Punkt 4.1.11 ausgewählten Servomotoren besitzen einen maximalen Drehradius von 180° , womit eine Kombination zweier solcher Motoren in einem theoretischen Freiheitsbereich von einer Halbkugel resultiert.

Für die Erkennung der aktuellen Neigung des Kopfes wird ein zusätzlicher **Beschleunigungssensor** vorgesehen. Dieser kann dazu verwendet werden, den wirklichen Drehwinkel des Servomotors mit dem gewünschten abzugleichen und so gegebenenfalls auf Blockierungen des Kopfes zu reagieren. Eine Möglichkeit für den Beschleunigungssensor ist der in der Arduino-Community bekannte *MPU6050*, für den bereits passende Bibliotheken existieren und der sowohl einen 3-Achsen Beschleunigungssensor als auch einen 3-Achsen Kreisel sensor besitzt [54].

Folgende Abbildung 26 zeigt den schematischen Aufbau der Kopfsteuerung. Die genaue Anbringung und Verknüpfung der Motoren im Gehäuse ist Teil des Roboter-Designs.

Abbildung 26: Steuerung der Kopfbewegungen

5.1.6 Augen

Wie bereits in Abschnitt 4.1.10 beschrieben, kommen für die Umsetzung der Augen zwei verschiedene Ansätze infrage. Die erste Art, vergleichbar mit der Umsetzung bei den NAO- und Pepper-Robotern, wären statische Augen, welche sich weder bewegen noch aktiv verändern können. Bei der zweiten Variante würden die Augen grafisch über ein Display umgesetzt werden.

Die statischen Augen wären lediglich vom Design des Roboters abhängig. Ähnlich wie beim NAO und beim Pepper könnten die Augen dabei mit LEDs erweitert werden, wodurch eine Veränderung der Augenfarbe und somit unterschiedliche Ausdrücke simpler Gestik möglich wären. Jedoch soll für den Tischroboter ein Display und damit die zweite Art der Augendarstellung verwendet werden. Die Visualisierung über ein Display erlaubt flexible Augenbewegungen unabhängig vom Kopf sowie den Ausdruck verschiedener Emotionen. Die Vielfalt möglicher, emotionaler Ausdrücke wird in folgender Abbildung 27 verdeutlicht. Diese zeigt eine Auswahl an Augendarstellungen des *Cozmo*-Roboters der Firma *Anki*. Werden die verschiedenen Augenarten mit Animationen verknüpft, kann der Roboter einen wesentlich lebendigeren Eindruck vermitteln als mit rein statischen Augen.

Abbildung 27: Grafische Darstellungsmöglichkeiten der Augen über ein Display⁵⁶ – Anki Cozmo Roboter [44]

⁵⁶ Bildquelle: https://static.giga.de/wp-content/uploads/2017/05/Cozmo_faces.jpg (abgerufen am 01.08.2019)

Das in Punkt 4.1.10 vorgestellte OLED-Display von Adafruit kann mittels I²C programmiert werden. Damit ist es möglich, dieses direkt vom Jetson Nano aus zu steuern, wie in Abbildung 28 veranschaulicht ist.

Abbildung 28: Kommunikation zwischen zentraler Recheneinheit und einem Display

Als Alternative zu einem zentralen Display, auf welchem beide Augen nebeneinander dargestellt werden, können **zwei Displays** verwendet werden. Damit würde eine Kombination der Augen vom Pepper- und vom Cozmo-Roboter erreicht werden. Beide Augen könnten separat durch das Roboter-Design dargestellt werden und hätten jeweils die Möglichkeit, über ein eigenes Display verschiedene Arten von Augen darzustellen. Bei der Kommunikation mit zwei Displays muss beachtet werden, dass die Displays nicht über die gleiche I²C-Adresse gesteuert werden können. Das Modul von Adafruit bietet die Möglichkeit, die I²C-Adresse zwischen 3C₁₆ und 3D₁₆ umzustellen [45]. Der schematische Aufbau der Kommunikation mit zwei Displays ist in folgender Abbildung 29 dargestellt.

Abbildung 29: Kommunikation zwischen zentraler Recheneinheit und zwei Displays

Für die Umsetzung wird die Variante mit zwei Displays präferiert, da die Augen räumlich voneinander getrennt werden können. Somit besteht die Möglichkeit, ein ähnliches Design wie beim Pepper-Roboter zu wählen und dieses um vielfältige Emotionen zu ergänzen.

5.1.7 Lautsprechersystem

Ein Lautsprechersystem besteht prinzipiell aus den in Abbildung 30 dargestellten Komponenten. Von den Anwendungen auf der zentralen Recheneinheit können Audiosignale erzeugt werden. Um diese mit Lautsprechern hörbar zu machen, wird ein Verstärker benötigt. Die Leistung des Verstärkers bestimmt die maximale Lautstärke des Lautsprechersystems.

Abbildung 30: Prinzip des Lautsprechersystems

Da der Jetson Nano keinen analogen **Audioausgang** besitzt, kann ein Verstärker nicht direkt angeschlossen werden. Wie in Punkt 4.1.4 bereits erwähnt, bietet jedoch das ReSpeaker-Mikrofonarray den Vorteil, dass es über USB als Ausgabegerät für Audiosignale verwendet werden kann und diese direkt über einen analogen Audioausgang für den Verstärker zugänglich macht. Die Verwendung des ReSpeaker ist besonders unter dem Aspekt sinnvoll, dass die ausgegebenen Audiosignale direkt von den aufgenommenen Signalen herausgefiltert werden können und somit zukünftig eine qualitative Spracherkennung möglich wäre.

Der an den ReSpeaker angeschlossene **Verstärker** ist das in 4.1.5 ausgewählte Modul von Adafruit. Dieses kann Stereo-Audiosignale auf 20 Watt Leistung verstärken und bietet Anschlussmöglichkeiten für einen linken und einen rechten Lautsprecher. Die Verstärkung und damit die ausgegebene Lautstärke kann am Verstärker über I²C-Kommunikation in 64 Stufen reguliert werden, sodass Anforderung **R013** erfüllt wird. Der Aufbau des Lautsprechersystems ist in folgender Abbildung 31 verdeutlicht.

Abbildung 31: Schematischer Aufbau des Lautsprechersystems

Für die Lautsprecher gibt es zwei verschiedene Ansätze. Der erste wäre ein **Zweiwegesystem**, also eine Kombination aus einem größeren Lautsprecher für tiefe Frequenzen und einem Hochtöner. Diese Art würde den Einsatz einer Frequenzweiche voraussetzen. Dabei handelt es sich um eine elektronische Schaltung, welche verschiedene Frequenzanteile des Audiosignals auf den jeweils vorgesehenen Lautsprecher weiterleitet. Ohne eine passende Frequenzweiche für die ausgewählten Lautsprecher würden diese jeweils das komplette Spektrum und somit auch ungeeignete Frequenzen wiedergeben, was eine negative Auswirkung auf die Klangqualität hätte.

Der zweite Ansatz wäre die Verwendung von **Breitbandlautsprechern**, die bereits das gesamte, relevante Frequenzspektrum abdecken. Teilweise haben diese Lautsprecher bereits eine Frequenzweiche integriert⁵⁷. Somit könnte das Audiosignal ohne weitere Verarbeitung vom ReSpeaker über den Verstärker auf die Lautsprecher geleitet werden.

Letztere Art der Umsetzung würde keinen zusätzlichen Aufwand durch die Entwicklung einer Frequenzweiche erfordern und wird daher für die Umsetzung des Tischroboters präferiert. Folgende Abbildung 32 zeigt den schematischen Aufbau des Lautsprechersystems dieser Art.

Abbildung 32: Stereo-Lautsprechersystem mit Breitbandlautsprechern

Statt zwei Lautsprechern für ein Stereo-System kann auch lediglich ein Lautsprecher für ein Mono-System verwendet werden. Für den Klang des Systems ist neben den eigentlichen Lautsprechern auch das Lautsprechergehäuse ausschlaggebend. Größere Lautsprecher benötigen vor allem bei der Wiedergabe von tiefen Frequenzen ein ausreichend großes Gehäusevolumen. Welche konkreten Lautsprecher für den Roboter verwendet werden und die Gestaltung des Lautsprechergehäuses sind Aspekte des Roboter-Designs, die vor allem von der Größe des Roboters und vom verfügbaren Platz abhängig sind. In der prototypischen Umsetzung sollen jedoch bereits beide Arten der Breitbandlautsprecher getestet werden.

⁵⁷ Sowohl der FX10-4 als auch der DX 5X6X aus Tabelle 9 (S. 48) sind 2-Kanal Breitbandlautsprecher mit integrierter Frequenzweiche. Die Lautsprecher FR 58 und R10 S TE sind reine Breitbandlautsprecher.

5.1.8 Energieversorgung

Die Energieversorgung des Roboters ist Grundlage für den zuverlässigen Betrieb aller Komponenten. Dieser Aspekt des Systems umfasst neben der Versorgung mit Energie auch das Steuern des Systemzustands, inklusive des Ein- und Abschaltens bestimmter Komponenten und einen potenziellen Akku-Betrieb.

5.1.8.1 Übersicht der Systemkomponenten

Für die Energieversorgung ist zunächst eine Übersicht aller Systemkomponenten und deren Kennzahlen wichtig, welche in folgender Tabelle 14 aufgeführt sind. Relevant sind dabei die **maximal benötigte Leistung**, der **Arbeitsspannungsbereich** und gegebenenfalls der **Spannungspegel der Kommunikation**. Letztere sind bei einigen Modulen von der Versorgungsspannung abhängig und werden in diesem Fall mit U_v gekennzeichnet. Teilweise besitzen Komponenten einen flexiblen Spannungsbereich. Bei diesen wird die präferierte Spannung in der Tabelle unterstrichen⁵⁸.

In der Tabelle sind nur zwei Arduinos zu finden. Da ein Arduino Nano durchaus mehrere Aufgaben parallel übernehmen kann, sind hier bereits die Steuerung der Servomotoren für die Kopfbewegung und die Steuerung der Temperaturregulierung auf einem Arduino zusammengelegt.

Tabelle 14: Energierelevante Kennzahlen der Systemkomponenten

Komponente	gehört zu	Spannungs- bereich in Volt	Spannung der Kommunikation in Volt	max. Leistungs- bedarf in Watt
Jetson Nano	SBC	5	3,3	10
Touch- Display	Video	vom Jetson	vom Jetson	2,5
Kamera	Video	vom Jetson	vom Jetson	vernachlässigbar
ReSpeaker	Audio	vom Jetson	vom Jetson	1
20W Verstärker	Audio	5 - <u>12</u>	3 oder <u>5</u>	20
Arduino Nano	Einschalter & Berührungssensoren	3 - <u>5</u>	U_v	0,5

⁵⁸ Bei dem Verstärker, den Lüftern und den Servomotoren wird die höchstmögliche Spannung präferiert, da diese Komponenten bei höheren Spannungen auch höhere Leistungen erbringen können. Grundsätzlich wird bei den anderen Komponenten eine Spannung von 5 Volt präferiert, da für den Jetson Nano ein 5-Volt-Netz obligatorisch ist. Siehe folgenden Abschnitt 5.1.8.2 für die Übersicht der verschiedenen Spannungsbereiche.

Komponente	gehört zu	Spannungsbereich in Volt	Spannung der Kommunikation in Volt	max. Leistungsbedarf in Watt
Arduino Nano	Kopfbewegung & Temp.-reg.	3 – <u>5</u>	U_v	0,5
80mm Lüfter	Temp.-reg.	5 – <u>12</u>	$\underline{5} - U_v$	1,6
80mm Lüfter	Temp.-reg.	5 – <u>12</u>	$\underline{5} - U_v$	1,6
Si7021 Sensor	Temp.-reg.	3 – <u>5</u> vom Arduino	U_v	vernachlässigbar
CAP1298 Sensor	Einschalter	3 – <u>5</u> vom Arduino	U_v	vernachlässigbar
LED Ring	Einschalter	5	5	3,6
Vibrationsmotor	Einschalter	3	U_v	0,5
Servo (Drehung)	Kopfbewegung	5 – <u>7,4</u>	$\underline{5} - U_v$	10
Servo (Neigung)	Kopfbewegung	5 – <u>7,4</u>	$\underline{5} - U_v$	10
OLED Display	Augen	3,3 – <u>5</u>	U_v	0,5

Zusammengerechnet benötigen die Komponenten des Systems maximal **63 Watt Leistung**. Dabei ist jedoch zu beachten, dass dieser Wert nur bei einem gleichzeitigen Maximalbedarf aller Komponenten erreicht wird. Durchschnittlich wird der Leistungsbedarf auf ca. 30 Watt geschätzt. Vor allem der Verstärker und die Servomotoren werden die angegebenen Spitzenleistungen nur stoßweise benötigen und im Normalbetrieb mit wesentlich weniger Leistung auskommen.

Im Hinblick auf spätere Hardwareerweiterungen und einen möglichen Akku wird im Rahmen dieser Arbeit das **78 Watt Netzteil** aus Tabelle 11 (siehe S. 53) ausgewählt, welches genügende Reserve bietet und im Vergleich zur nächstleistungsstärkeren Variante preisgünstig ist.

5.1.8.2 Spannungsbereiche der Energieversorgung

In Tabelle 14 fällt auf, dass die verschiedenen Komponenten mit **unterschiedlichen Spannungen** arbeiten, die eingehalten werden müssen. Liegt eine Spannung außerhalb des ausgewiesenen Bereichs, kann es zu Fehlfunktionen oder zur Beschädigung von Komponenten kommen. Folgende Tabelle 15 fasst die verschiedenen Spannungsbereiche des Systems zusammen.

Tabelle 15: Übersicht der verschiedenen Spannungsbereiche des Systems

Spannungsbereich	Komponenten
3V	Vibrationsmotor
5V	Jetson Nano Arduino Nano LEDs OLED Display
7,4V	Servomotoren
12V	Verstärker Gehäuselüfter

Um die 12 Volt Spannung des Netzteils an die von den einzelnen Bauteilen benötigte Spannung anzupassen, können **Step-Down-Spannungsregler**⁵⁹ eingesetzt werden. Dabei handelt es sich um elektronische Bauteile, die eine hohe Eingangsspannung in eine niedrigere Ausgangsspannung umwandeln. Als Spannungsregler kann der *LM2596* des amerikanischen Unternehmens *Texas Instruments*⁵⁹ verwendet werden, welcher hohe Ausgangsströme von bis zu 3 Ampere ermöglicht [55]. Für jeden Spannungsbereich, der unterhalb der 12 Volt des Netzteils liegt, kann dementsprechend ein separater Spannungsregler verwendet werden, um die gewünschte Spannung einzustellen. Folgende Abbildung 33 zeigt den vereinfacht dargestellten Aufbau der Spannungsregulierung beispielhaft für das 5-Volt-Netz.

Abbildung 33: Step-Down Spannungsregulierung mit LM2596

⁵⁹ Offizielle Seite: <http://www.ti.com/> (abgerufen am 01.08.2019)

5.1.8.3 Ein- und Abschalten von Komponenten

Wie im Abschnitt 5.1.2 zum Einschalter erklärt wurde, soll das komplette System ein- und ausgeschaltet werden können. Im ausgeschalteten Zustand kann der Großteil der Komponenten von der Energieversorgung getrennt werden. Vor allem bei Komponenten wie den Servomotoren ist eine solche Trennung notwendig, da diese auch ohne Steuerbefehle dauerhaft Energie zum Halten der Position benötigen. Der Arduino Nano, welcher auf Interaktionen mit dem Einschalter reagiert und den Systemzustand steuert, ist der einzige Systemteil, der immer aktiv ist und nie von der Energieversorgung getrennt wird.

Die Schaltung der Energieversorgung wird mittels einer Kombination aus einem **P-Kanal MOSFET** und einem **NPN Transistor** realisiert. Das eigentliche Schalten übernimmt der MOSFET, welcher eine geringe Verlustleistung aufweist und daher auch hohe Ströme schalten kann. Der Transistor ist für die Verknüpfung verschiedener Spannungsbereiche notwendig. Über diesen kann der Arduino Nano, welcher auf 5 Volt basiert, auch höhere Spannungen schalten.

Folgende Abbildung 34 veranschaulicht das Prinzip einer elektronischen Schaltung, die bestimmte Komponenten von der Energieversorgung trennen kann. Links liegt in der Abbildung die 12-Volt-Spannung des Netzteils an, auf der rechten Seite soll das 12-Volt-Netz für die Lüfter und den Verstärker an- und abgeschaltet werden. Grundsätzlich ist das Prinzip allgemeingültig, sodass beliebige Spannungsbereiche geschaltet werden können. So kann sich auf der linken Seite statt des direkten Netzteilanschlusses auch ein Spannungsregler befinden.

Abbildung 34: Prinzip zum Abschalten von Komponenten

Eine Übersicht des gesamten Schaltplans der Energieversorgung für einen Betrieb mit Netzteil ist in Abbildung 55 im Anhang auf Seite 119 dargestellt.

5.1.8.4 Kommunikation zwischen verschiedenen Spannungsbereichen

Neben den verschiedenen Spannungsbereichen für den Betrieb der Komponenten fällt in Tabelle 14 ebenfalls auf, dass die Spannungspegel der Kommunikation unterschiedlich sind. So arbeiten die Arduinos und der Großteil der anderen Komponenten mit 5 Volt, während die GPIO-Pins des Jetson Nano nur mit einer Spannung von 3,3 Volt funktionieren. Damit dementsprechend der Jetson Nano mit Arduinos oder anderen Bauteilen über digitale Schnittstellen kommunizieren kann, müssen die verschiedenen Spannungspegel zueinander kompatibel umgewandelt werden.

Für einzelne Leitungen kann das sogenannte **Level-Shifting** mithilfe eines **N-Kanal MOSFETs** umgesetzt werden, wie in folgender Abbildung 35 dargestellt ist. Gezeigt ist die bidirektionale Umwandlung von 3,3 Volt in 5 Volt auf einer Leitung. Das abgebildete Prinzip funktioniert jedoch für beliebige Spannungen, solange die Spannung auf der linken Seite geringer ist als jene auf der rechten Seite.

Abbildung 35: Level-Shifting mit einem MOSFET

Da für die verschiedenen Schnittstellen nicht nur eine Leitung benötigt wird, bietet sich der Einsatz eines dedizierten Schaltkreises an, welcher Level-Shifting für mehrere Kanäle übernehmen kann. Eine Möglichkeit ist der *TXS0108E*-Schaltkreis der Firma *Texas Instruments*, welcher 8 Level-Shifting-Kanäle besitzt [56]. Somit kann ein *TXS0108E* sowohl für den I²C-Bus als auch für SPI-Kommunikation zwischen dem Jetson Nano und anderen Komponenten genutzt werden.

5.1.8.5 Akkubetrieb

Um einen Akkubetrieb des Tischroboters zu realisieren, wie in **R018** als Soll-Anforderung formuliert wird, muss der Aufbau der Energieversorgung angepasst werden. Wie in Punkt 4.1.8 beschrieben, soll als Akku eine Powerbank verwendet werden. Unabhängig von der endgültigen Auswahl einer Powerbank sind folgende Punkte bei der Entwicklung der Energieversorgung mit Akkuunterstützung relevant:

1. Wenn das Netzteil angeschlossen ist, sollen alle Komponenten ihre Energie aus diesem Netzteil und nicht aus dem Akku beziehen.
2. Wenn das Netzteil angeschlossen ist, soll der Akku aufgeladen werden.
3. Sobald das Netzteil entfernt wird, soll das System automatisch Energie aus dem Akku beziehen.

Der **erste Punkt** ist mithilfe von Dioden⁶⁰ realisierbar, mit denen sich die Energienetze des Akkus und des Netzteils voneinander trennen lassen. Die Dioden sorgen zum einen dafür, dass der Strom jederzeit nur vom Netzteil zum Verbraucher bzw. vom Akku zum Verbraucher fließen kann. Insbesondere wird damit verhindert, dass Strom vom Netzteil in einen Ausgang des Akkus fließt. Zum anderen wird über die Dioden erreicht, dass immer nur die Energiequelle mit der höheren Spannung aktiv ist. Das Prinzip ist in folgender Abbildung 36 dargestellt, beispielhaft mit einem 5 Volt Netz für den Jetson Nano.

Abbildung 36: Trennung von Netzteil und Akku mit Dioden

Ist in der dargestellten Schaltung die regulierte Spannung des Netzteils höher als die des Akkus, blockiert die untere Diode, sodass keine Energie aus dem Akku entnommen wird. Wird das Netzteil vom Roboter getrennt, ist die Spannung des Akkus höher, sodass die obere Diode sperrt und der Akku Leistung abgibt. Durch diese Schaltung wird der **dritte Punkt** implizit miterfüllt.

Bei dem Einsatz von Dioden muss beachtet werden, dass diese einen Spannungsabfall aufweisen, welcher bei den in Abbildung 36 dargestellten Schottky-Dioden bei etwa 0,5 Volt liegt⁶⁰. Während die vom Netzteil bereitgestellte Spannung mittels des Step-Down-Reglers etwas größer als 5 V eingestellt werden kann, sodass die Durchflussspannung der Diode gerade kompensiert wird, liegt die Spannung des Akkus hingegen fest bei 5 V. Damit stehen

⁶⁰ Der Spannungsabfall einer Diode ist vom spezifischen Produkt abhängig und wird von der Temperatur und dem Stromfluss beeinflusst. Als Referenz wird die Schottky-Diode *SBX2030* der deutschen Firma Diotec (<https://diotec.com/>, 01.08.2019) verwendet, welche bei 25°C und einem Strom von 5A mit einer Flussspannung von 0.45V angegeben ist [57]. Je höher der Stromfluss ist, desto mehr Spannung fällt bei Dioden ab.

jedoch dem Jetson Nano lediglich die verbleibenden 4,5 V zur Verfügung. Diese simple Lösung mittels Dioden ist daher nur möglich, wenn für den Betrieb des Jetson Nano auch 4,5 V ausreichend sind, was in praktischen Tests evaluiert werden muss.

Durch den Spannungsabfall der Dioden geht folglich auch Leistung im Gesamtsystem verloren, wodurch die maximal nutzbare Energie des Akkus reduziert wird. Ein alternativer Ansatz ist die Verwendung einer Regelschaltung, die auf MOSFETs[•] basiert. MOSFETs können ähnliche Effekte erzielen wie Dioden und weisen dabei wesentlich weniger Verlustleistung auf. Eine solche Regelschaltung würde eine höhere Komplexität und mehr Bauteile mit sich bringen, aber insgesamt die Energieeffizienz des Systems optimieren. Ein möglicher Schaltkreis, auf dem die Regelschaltung basieren könnte, ist der *LM5050* von Texas Instruments. Im Datenblatt zu dem Chip ist ein Referenzdesign zur Spannungsversorgung mit zwei getrennten Netzen zu finden [58].

Der **zweite Punkt** lässt sich ebenfalls mit dem Einsatz von Dioden lösen. So kann der Akku aufgeladen werden, indem vom Netzteil eine Verbindung zum entsprechenden Eingang der Powerbank geführt wird, wie in Abbildung 37 dargestellt ist⁶¹.

Abbildung 37: Aufladen des Akkus

Um Anforderung **R019** zu erfüllen und einen Akkubetrieb von mindestens zwei Stunden zu ermöglichen, müsste die Powerbank bei der maximalen Systemleistung gut 120 Wattstunden Energie speichern können. Die größten angebotenen Powerbanks bieten lediglich 100 Wattstunden⁶². Diese Energie reicht jedoch bei der angenommenen Durchschnittsleistung von 30 Watt aus, um den Roboter für etwa drei Stunden mit Energie zu versorgen.

Powerbanks arbeiten standardmäßig mit einer Spannung von 5 Volt. Um die Komponenten zu betreiben, die eine höhere Spannung benötigen, müssen Step-Up-Spannungsregler[•] eingesetzt werden. Diese sind das Pendant zu Step-Down-Reglern und können eine niedrige Eingangsspannung in eine höhere Ausgangsspannung umwandeln.

⁶¹ Auf die mittlere Diode in der Abbildung könnte gegebenenfalls verzichtet werden, wenn der Akku-Eingang nicht gleichzeitig auch als Ausgang genutzt werden kann.

⁶² Die Grenze von 100 Wattstunden hängt mit einer Bestimmung der *International Air Transport Association* (IATA) zusammen. Diese besagt, dass Privatpersonen ohne Genehmigung der Fluggesellschaft keine größeren Lithium-Batterien transportieren dürfen, um größere Gefahren zu vermeiden. Siehe <https://www.iata.org/whatwedo/cargo/dgr/Documents/DGR-60-DE-2.3a.pdf> (abgerufen am 25.07.2019)

5.1.9 Architektur des Systems

Abbildung 38 zeigt eine Gesamtübersicht der konzeptionellen Systemarchitektur. Es sind sowohl die einzelnen, ausgewählten Komponenten sowie deren Verknüpfungen dargestellt.

Abbildung 38: Gesamtübersicht der Systemarchitektur

5.2 Softwaretechnische Systemaspekte

Dieses Kapitel erläutert die konzeptionellen Ansätze der Software des Systems. Die Software beinhaltet verschiedene Aspekte, die nachfolgend in Abbildung 39 dargestellt sind. Kernbestandteil für die Funktionalität des Roboters ist das **Back-End** zusammen mit den Hardwarekomponenten des Systems. Das Back-End verknüpft alle Teile des Systems, verarbeitet eingehende Daten und steuert entsprechende Reaktionen. Im **Front-End** werden die Funktionen des Roboters mithilfe einer grafischen Anwendung über das Touch-Display zur Verfügung gestellt, sodass dies vor allem für den späteren Benutzer relevant ist. Im Rahmen dieser Arbeit sind insbesondere die **HelloWorld** Programme von Bedeutung, welche Entwicklern die Möglichkeiten des Systems zugänglich machen. Diese bilden die Grundlage für Folgeprojekte und die Umsetzung weiterer Anwendungsfälle mit dem Tischroboter.

Abbildung 39: Aspekte der Systemsoftware

Das Ziel dieses Abschnitts liegt in der Entwicklung einer Softwarearchitektur, welche die verschiedenen Bestandteile des Systems miteinander verknüpft. Diese Verknüpfung soll möglichst modular erfolgen, sodass das System einfach wart- und erweiterbar ist und einzelne Komponenten gegebenenfalls ausgetauscht werden können. Zusätzlich soll eine Übersicht gegeben werden, welche Aspekte des Systems mit HelloWorld Programmen erläutert und welche Programmiersprachen für die jeweiligen Bestandteile verwendet werden.

Die Gesamtübersicht der Architektur ist in Abschnitt 5.2.5 zu finden.

5.2.1 Back-End – Roboter Operating System

Das Back-End bildet den Kern der Software, welcher entscheidend die Funktionalität des Tischroboters ausmacht. Ein Aspekt ist dabei die Schnittstellensoftware zum Ansteuern der einzelnen Hardwaremodule. Bei der Konzeptionierung des Back-Ends steht jedoch vor allem die Verknüpfung aller Softwaremodule im Vordergrund. Dieser Abschnitt beschreibt die Architektur des Back-Ends mithilfe des **Robot Operating Systems** (ROS), welches in Punkt 4.2 als freies Framework aufgeführt wurde. Die konkrete Implementation von Schnittstellensoftware und anderen Softwaremodulen ist Teil der Umsetzung.

ROS ist speziell für die Entwicklung von Software für Roboter entwickelt worden und stellt dafür grundlegende Werkzeuge, aber auch Bibliotheken für spezielle Aufgaben zur Verfügung. Das grundlegende Konzept von ROS ist der sogenannte **Computation Graph**. Dabei handelt es sich um ein Peer-to-Peer-Netzwerk von ROS-Knoten, die zusammen Daten verarbeiten bzw. Daten untereinander austauschen. Grundlage des Netzwerks ist der **Master**. Dieser verwaltet den Computation Graph und ist dafür zuständig, den Datenaustausch zwischen Knoten zu ermöglichen. Ein ROS-**Knoten** ist ein eigenständiger Prozess, welcher eine bestimmte Aufgabe besitzt und anderen Knoten zugehörige Dienste oder Daten zur Verfügung stellt. Die Vermittlung von Daten geschieht nach dem Publisher-Subscriber-Entwurfsmuster, bei welchem Publisher-Knoten Informationen unter einem bestimmten **Topic**, also einer bestimmten Bezeichnung, veröffentlichen. Subscriber-Knoten können dieses Topic abonnieren, um die Daten nach der Veröffentlichung automatisch zu erhalten. [59]

Das Prinzip der Datenvermittlung ist in folgender Abbildung 40 veranschaulicht. In dem Beispiel gibt es einen Publisher-Knoten und einen Subscriber-Knoten. Der Publisher kündigt beim Master an, dass er unter dem Topic „topic“ Nachrichten veröffentlichen wird. Sobald ein Subscriber-Knoten ankündigt, dass er dieses Topic abonnieren möchte, leitet der Master diese Information zum Publisher-Knoten weiter. Anschließend kann der Publisher seine Daten über TCP direkt an den oder die jeweiligen Subscriber senden. [60]

Abbildung 40: Publishing und Subscribing von Nachrichten in ROS

Es ist sowohl möglich, dass mehrere Publisher Daten unter dem gleichen Topic veröffentlichen, als auch, dass viele Subscriber Nachrichten des gleichen Topics erhalten. Dadurch wird eine Koordination von lesenden und schreibenden Zugriffen auf Module des Systems ermöglicht, wie sie in den Anforderungen **R028** und **R029** gefordert wird. Neben dem Veröffentlichen und Abonnieren von Daten über Topics können ROS-Knoten auch

Services anbieten. Dabei handelt es sich um Methoden, welche Daten entgegennehmen, ein entsprechendes Ergebnis berechnen und dieses zurückgeben. Somit können bestimmte Funktionen des Roboters auf Knoten ausgelagert werden und bei Bedarf von beliebigen, anderen Knoten im System genutzt werden. [59]

Für den Datenaustausch über Topics und über Services können eigene **Nachrichtenformate** definiert werden, welche aus beliebigen Kombinationen von Variablen bestehen können. Dabei können sowohl Standardtypen wie Ganzzahlen oder Zeichenketten, als auch selbsterstellte Nachrichtenobjekte verwendet werden. Über das Nachrichtensystem mit Topics und die Entwicklung von Services bietet ROS die Entwicklung einer **flexiblen und modularen Architektur**, bei welcher Softwaremodule funktional voneinander getrennt, parallel entwickelt und leicht gewartet, ausgetauscht und erweitert werden können.

ROS-Knoten werden in **Packages** zusammengefasst und organisiert. Ein Package kann mehrere Knoten und zusätzlich weitere Informationen wie Nachrichtenformate, Abhängigkeiten und Konfigurationen enthalten. [59]

Folgende Abbildung 41 zeigt ein Konzept für die Aufteilung der Funktionalitäten in verschiedenen Packages und den potenziellen Datenverkehr über Topics und Services.

Abbildung 41: Konzeptionelle Übersicht möglicher ROS-Packages, -Topics und -Services im System

In Abbildung 41 repräsentieren die dargestellten Packages die verschiedenen Hauptaspekte der Roboterfunktionalität. Die Packages können dabei einen oder mehrere Knoten enthalten, welche folgende Aufgaben erfüllen:

Das *camera*-Package beinhaltet Knoten, welche die Kamera ansteuern und Videos bzw. Bilder aufnehmen können. Das Topic *camera/video* kann einen kontinuierlichen Videostream veröffentlichen, während der Service *camera/takePicture* auf Anfrage ein einzelnes, höher auflösendes Bild bereitstellen kann. Zusätzlich wird das Topic *camera/settings* abonniert, um die Einstellungen der Kamera verändern, bzw. die Kamera auch abschalten zu können.

Das *visual*-Package stellt ROS-Knoten bereit, die das äußere Erscheinungsbild beeinflussen können. Dazu gehören vor allem die Displays, welche die Augen und Emotionen des Roboters darstellen. Aber auch weitere visuelle Aspekte, wie der LED-Ring des Einschalters oder weitere LEDs im Gehäuse, werden von Knoten dieses Packages gesteuert.

Das *head*-Package ist zuständig für die Bewegung des Kopfes. Das Topic *head/currentPosition* stellt die aktuelle Kopfposition bereit, während über das Topic *head/setPosition* die gewünschte Position des Kopfes eingestellt werden kann. Das Topic *head/setStiffness* beeinflusst die Steifheit der Servomotoren des Kopfes⁶³.

Das *sound*-Package beinhaltet sowohl die Wahrnehmung von Geräuschen als auch die Ausgabe von Musik, Sprache und anderen Tönen. Die untergeordneten Topics von *sound/mic/* veröffentlichen Daten des ReSpeaker-Mikrofonarrays. Dazu gehören die Informationen, ob gerade Sprache erkannt wird, aus welcher Richtung die Geräusche wahrgenommen werden und auch die Daten des aufgenommenen Audiosignals für eine spätere Spracherkennung. Die Audioausgabe wird von den Topics unter *sound/out/* gesteuert, wobei eine konkrete Audiodatei abgespielt, Sprache ausgegeben und die Lautstärke eingestellt werden kann.

Das *system*-Package stellt Daten über das gesamte System bereit, beispielsweise die Temperatur des Gehäuses, den aktuellen Ladestand des Akkus oder auch die Information, dass mit dem Einschalter interagiert wurde. Außerdem kann dieses Package ROS-Knoten beinhalten, welche Daten über die Schnittstellen I²C oder SPI senden und empfangen können, und damit eine Kommunikation mit den Arduinos oder anderen Hardwaremodulen erlauben.

Die Packages können in Folgeprojekten, welche die Funktionalität des Roboters ergänzen, mit beliebigen ROS-Knoten, Topics und Services erweitert werden. Ein besonderer Vorteil von ROS besteht darin, dass Knoten sowohl in C++ als auch in Python entwickelt und über den Computation Graph unabhängig von der Sprache miteinander verknüpft werden können.

⁶³ Die Möglichkeit, die Steifheit der Motoren einzustellen, orientiert sich an den NAO-Robotern. Bei diesen ist es ebenfalls möglich, zwischen festgestellten oder frei bewegbaren Gelenken zu wechseln. Letzteres spart Energie und erlaubt außerdem eine Positionierung der Gelenke mit der Hand.

5.2.2 Front-End – Grafische Benutzeroberfläche mit Electron

Das Front-End der Software besteht im Wesentlichen aus der grafischen Oberfläche (engl. **GUI** für Graphical User Interface), welche auf dem Touch-Display des Roboters dargestellt wird. Ähnlich wie beim Pepper-Roboter stellt diese grafische Anwendung eine primäre Interaktionsmöglichkeit für den Benutzer dar und ist daher ein wesentlicher Teil des Systems. Dieser Abschnitt erläutert das Konzept der grafischen Anwendung sowie die Kommunikation mit dem Back-End.

Das genaue **Design der grafischen Oberfläche** ist von den speziellen Anwendungsfällen abhängig und wird daher in dieser Arbeit nicht detaillierter betrachtet. Die Darstellung in folgender Abbildung 42 ist ein Vorschlag für die Hauptansicht der Anwendung. Dabei werden die einzelnen Funktionen des Roboters über große Kacheln zugänglich gemacht, die mit unterschiedlichen Farben und Symbolen gut voneinander unterscheidbar sind. Bei Interaktion mit einer Kachel kann sich anschließend eine neue, übergeordnete Ansicht für die jeweilige Funktionalität öffnen. Die Kacheln könnten modular voneinander getrennt werden, sodass ein flexibles Hinzufügen von Funktionen möglich ist, vergleichbar mit dem Installieren von Apps für mobile Endgeräte.

Abbildung 42: Mockup als Vorschlag für die grafische Anwendung

Im zweiten Semester des Telematik-Studiums wird die Entwicklung grafischer Oberflächen mithilfe von JavaFX^{*} gelehrt. Viele andere Sprachen, darunter auch C++ und Python, bieten über entsprechende Bibliotheken oder Frameworks ebenfalls die Möglichkeit, GUIs zu entwickeln.

In Abschnitt 4.2 wurde bereits das Framework **Electron** als Möglichkeit vorgestellt, Desktopanwendungen mit den Möglichkeiten der Webentwicklung zu verknüpfen. Im Gegensatz zu C++, Python und Java sind die Sprachen **HTML**, **CSS** und **JavaScript** speziell für die Entwicklung von grafischen Oberflächen bzw. von Websites ausgelegt und bieten flexible und moderne Möglichkeiten der Gestaltung. Die Sprachen der Webentwicklung sind ebenfalls Teil des Telematik-Studiums, weswegen auch die Einarbeitungszeit für die Erweiterung der grafischen Oberfläche in Folgeprojekten als geringer eingestuft wird als bei speziellen Frameworks von C++ oder Python.

Die prinzipielle Architektur einer Electron Anwendung wird in folgender Abbildung 43 dargestellt. Die oberste Ebene, welche über HTML, CSS und JavaScript für die Darstellung und die Interaktion mit dem Nutzer zuständig ist, steht in direkter Verbindung mit dem Hauptprozess einer Electron App. Dieser Hauptprozess basiert auf Node.js^{*} und kann direkt mit dem Betriebssystem, mit lokalen Ressourcen oder mit dem Back-End kommunizieren.[61]

Abbildung 43: Prinzipielle Architektur einer Electron App

Da Electron auf JavaScript basiert und ROS lediglich Python und C++ unterstützt, ist eine direkte Verknüpfung der GUI mit dem Back-End und eine Kommunikation über Topics und Services nicht möglich. Stattdessen muss eine andere Möglichkeit der Interprozesskommunikation gefunden werden.

Eine Möglichkeit wäre eine direkte Kommunikation mit den ROS-Knoten über **TCP-Sockets**. TCP-Sockets werden von allen modernen Programmiersprachen unterstützt und bieten hohe Übertragungsraten. Der Nachteil dieser Methode besteht darin, dass die Kommunikationspartner sich kennen, und aktiv Verbindungen zu bestimmten Servern bzw. zu bestimmten Ports aufgebaut werden müssen. Bei ROS wird dieser Verwaltungsaufwand vom Master übernommen, sodass die Infrastruktur sehr flexibel bleibt.

Um eine ähnliche Flexibilität wie bei ROS und einen Datenaustausch über Publisher und Subscriber zu ermöglichen, kann eine Abstraktionsschicht zwischen Front- und Back-End gesetzt werden. Die nichtrelationale Key-Value-Datenbank* **Redis** bietet eine solche Abstraktionsmöglichkeit. Neben der persistenten Speicherung von Key-Value-Paaren beinhaltet Redis einen **Publish-Subscribe-Service**, bei welchem ebenfalls Nachrichten unter beliebigen Topics veröffentlicht und abonniert werden können [62].

Werden bestimmte Daten von ROS-Knoten ebenfalls über Redis veröffentlicht, können diese auf der anderen Seite vom Front-End abonniert werden. Genauso kann die grafische Anwendung Daten über Redis veröffentlichen und somit an das ROS Back-End weiterleiten. Für den Zugriff auf Redis und damit auf den Message-Service gibt es für eine große Anzahl von Sprachen entsprechende Bibliotheken⁶⁴. Somit ist es nicht nur möglich, die Electron App mit dem Back-End zu verknüpfen, sondern auch **Programme anderer Sprachen** in das System zu integrieren.

⁶⁴ Siehe <https://redis.io/clients> (abgerufen am 07.08.2019) für eine Übersicht verfügbarer Bibliotheken und unterstützter Sprachen.

5.2.3 Hardware – Kommunikation mit den Arduinos

Bestimmte Hardwarekomponenten lassen sich direkt mithilfe des Betriebssystems bzw. über Schnittstellensoftware des Back-Ends ansteuern. Dazu gehören Module, die über USB angebunden sind wie der ReSpeaker, aber auch die Kamera, welche über CSI angeschlossen ist. Ebenfalls können die Displays der Augen und das Verstärker-Modul direkt über I²C kontrolliert werden.

Für die Interaktion mit den Komponenten, die auf Arduinos ausgelagert sind, ist eine eigendefinierte **Kommunikation** über die **SPI-Schnittstelle** notwendig. Zu den ausgelagerten Komponenten gehören die Motorsteuerung des Kopfes, die aktuelle Gehäusetemperatur, der Einschalter und die Energieversorgung. Die Steuerung geschieht dabei über die Kombination eines 8-Bit Kommandos und anschließenden Steuerdaten. Gleiches gilt für das Auslesen von Informationen, bei dem ein 8-Bit Kommando die gewünschten Daten definiert und diese anschließend vom Arduino zurückgegeben werden. Die Kommandos und die zugehörigen Daten werden nachfolgend in Tabelle 16 aufgeführt.

Tabelle 16: Übersicht der Kommunikation mit ausgelagerten Hardwarekomponenten

Teil des Systems	Kommando	8-Bit Code ⁶⁵	Modus	Daten
Kopf	Motoren Steifheit	0x10	write	0 → frei beweglich 1 → steif
	Drehung absolut	0x11	write	Gradzahl zwischen -90° und 90° und Geschwindigkeit
	Drehung relativ	0x12		
	Neigung absolut	0x13		
	Neigung relativ	0x14		
	aktuelle Drehung	0x21	read	Gradzahl zwischen -90° und 90°
	aktuelle Neigung	0x22		
Gehäuse	aktuelle Temperatur	0x31	read	Temperatur zwischen -10°C und +85°C
Einschalter	Status des Einschalters	0x01	read	0 → keine Interaktion > 0 → Interaktionsdauer
	LED Ring	0x51	write	RGB Farbwert
	Vibrationsmotor	0x61	write	Dauer des gewünschten, taktilen Feedbacks
Energie	Akkustand	0xE1	read	Wert zwischen 0% und 100%

⁶⁵ In Hexadezimalschreibweise

Die Kommunikation über die SPI-Schnittstelle kann von einem eigenständigen ROS-Knoten übernommen werden. Dieser kann auf konkrete Anfragen anderer Knoten reagieren oder auch bestimmte Daten kontinuierlich abfragen und dem System bereitstellen.

5.2.4 HelloWorld Programme

Nach Anforderung **R030** müssen die Komponenten des Roboters mithilfe von HelloWorld erläutert werden. HelloWorld Programme können einerseits eine gute Übersicht der vorhandenen Möglichkeiten des Roboters geben. Darüber hinaus bilden diese Programme die wesentliche Grundlage einer schnellen Einarbeitung für Entwickler, die in Folgeprojekten die Funktionalität des Systems erweitern werden. Dieser Abschnitt führt in den folgenden Tabellen eine Übersicht auf, welche HelloWorld Programme im Rahmen dieser Arbeit entwickelt werden sollen.

Als Hauptsprache für die Programme wird **Python** ausgewählt. Die Sprache wird im Telematik-Studium im Modul *Internetprogrammierung* gelehrt und genügt damit der Anforderung **R031**. Python belegt auf dem Tiobe-Index*, welcher die Popularität von Programmiersprachen darstellt, derzeit Platz 3 hinter Java und C⁶⁶. Die Beliebtheit von Python lässt sich vor allem damit begründen, dass die Sprache ein sehr schnelles Prototyping und Experimentieren ermöglicht, ohne genauere Details von Implementierungen kennen und verstehen zu müssen. Python Programme sind leicht lesbar und verständlich. Vor allem bietet Python eine extreme Vielfalt an einfach zu installierenden Bibliotheken für viele Anwendungszwecke. So existieren ebenfalls Bibliotheken, welche einen simplen Zugriff auf die Hardwaremodule dieses Projekts ermöglichen.

Als Interpretersprache besitzt Python jedoch den Nachteil, dass die Ausführung im Vergleich zu kompilierten Programmen wesentlich langsamer ist. Dieser Nachteil wird beim Einsatz bestimmter Bibliotheken kompensiert, indem diese im Hintergrund kompilierte Programme verwenden. Jedoch kann es bei zeitkritischen oder rechenintensiven Aufgaben sinnvoll sein, komplett in anderen Sprachen wie C oder C++ zu entwickeln. Für eine flexible Erweiterung des Systems werden daher bestimmte HelloWorld Programme in mehreren Sprachen bereitgestellt.

⁶⁶ Stand August 2019; siehe <https://www.tiobe.com/tiobe-index/> (abgerufen am 08.08.2019)

5.2.4.1 HelloWorld für die Ansteuerung von Hardwaremodulen

Für die direkte Kommunikation mit grundlegenden Hardwaremodule können eigenständige HelloWorld Programme entwickelt werden, die nachfolgend in Tabelle 17 aufgeführt werden. Die Programme erläutern die jeweilige Ansteuerung der Module unabhängig von ROS. Dazu gehört der ReSpeaker, die Kamera und die Displays der Augen. Auch die Ausgabe von Audiodateien und die Regulierung der Lautstärke werden in diesen HelloWorld Programmen vorgestellt.

Tabelle 17: HelloWorld Programme für die separate Ansteuerung von Hardwaremodulen

Bezeichnung	Programmiersprachen	Inhalt
HelloReSpeaker	Python	Konfiguration und Auslesen des ReSpeaker-Mikrofonarrays
HelloAudio	Python	Ausgabe von Audio Dateien Regulierung der Lautstärke
HelloCamera	Python	Ansteuerung der Kamera, um Video und Bilder zu erfassen
HelloEyes	Python	Ansteuerung der OLED-Displays für die Augen Darstellung von grafischen Elementen

5.2.4.2 HelloWorld für den Umgang mit ROS

Für den Umgang mit ROS werden die HelloWorld Programme der folgenden Tabelle 18 entwickelt. Zunächst gehört dazu eine Anleitung für die Arbeit mit ROS. Diese beinhaltet wichtige Befehle zum Aufsetzen der ROS-Umgebung, zum Anlegen von Packages, zum Starten von ROS-Knoten und das Vorgehen zum Definieren von eigenen Nachrichten. Außerdem werden für das Veröffentlichen und für das Abonnieren von Nachrichten jeweils HelloWorld Programme in Python und C++ geschrieben, um die beiden Sprachen von ROS zu unterstützen. Zusätzlich wird die Nutzung des Publish-Subscribe-Service von Redis in den Sprachen Python, C, C++ und JavaScript vorgestellt, um darüber eine flexible Erweiterung des Systems zu ermöglichen.

Tabelle 18: HelloWorld Programme für die Arbeit mit ROS

Bezeichnung	Programmiersprachen	Inhalt
HelloRos	Shell	Anleitungen für die Arbeit mit ROS
HelloRosPublisher	Python C++	Veröffentlichen von Nachrichten und Daten mit ROS
HelloRosSubscriber	Python C++	Abonnieren von Nachrichten mit ROS
HelloRedis	Python C C++ JavaScript (Node.js)	Schreiben und Lesen von persistenten Daten Publishing & Subscribing von Daten

5.2.4.3 HelloWorld für die Ansteuerung von Hardwarechnittstellen

Für eine Ansteuerung von externen Modulen, welche über die verfügbaren Hardwarechnittstellen angeschlossen sind, sollen die HelloWorld Programme aus Tabelle 19 entwickelt werden. Diese erläutern die grundlegende Kommunikation über I²C, SPI, UART und die Nutzung der GPIO-Pins.

Tabelle 19: HelloWorld Programme für die Benutzung von Hardwarechnittstellen

Bezeichnung	Programmiersprachen	Inhalt
HelloI2C	Python	Senden und Empfangen von Daten über den I ² C-Bus
HelloSPI	Python	Senden und Empfangen von Daten über den SPI-Bus
HelloUART	Python	Senden und Empfangen von Daten über die UART-Schnittstelle
HelloGPIO	Python	Steuern von GPIO Pins

5.2.4.4 HelloWorld für Desktopanwendungen mit Electron

Für die grafische Anwendung sollen die beiden HelloWorld Programme aus Tabelle 20 entwickelt werden. Dazu gehört die Installation des Frameworks und das Aufsetzen eines Electron-Projekts. Außerdem wird ein einfaches Beispielprojekt mit mehreren Fenstern vorgestellt.

Tabelle 20: HelloWorld Programme für grafische Anwendungen mit Electron

HelloWorld	Programmiersprachen	Inhalt
HelloElectronSetup	Shell	Installieren und Aufsetzen einer Electron App
HelloElectron	JavaScript HTML CSS	Grundlegender Aufbau einer Electron App

5.2.4.5 HelloWorld für die Arduino Nanos

Die HelloWorld Programme aus Tabelle 21 sollen die Steuerung der einzelnen Hardwarekomponenten erläutern, die in separaten Schaltungen an den Arduino Nanos angeschlossen sind. Da Arduinos mit C++ programmiert werden, verwenden die Hello Worlds dieses Abschnitts ebenfalls einheitlich diese Programmiersprache.

Tabelle 21: HelloWorld Programme für die Arduino Nanos

Bezeichnung	Programmiersprachen	Inhalt
HelloServo	C++	Ansteuerung von Servomotoren
HelloMpu6050	C++	Auslesen des MPU6050 Neigungssensors
HelloSi7021	C++	Auslesen des Si7021 Temperatursensors
HelloFan	C++	Steuerung von Gehäuselüftern
HelloLedRing	C++	Steuerung von LED Ringen
HelloCapacitiveSensor	C++	Auslesen von einfachen, kapazitiven Sensoren mit einem Arduino
HelloCap1298	C++	Auslesen von kapazitiven Sensoren über den Cap1298 Schaltkreis
HelloVibration	C++	Ansteuerung des Vibrationsmotors

5.2.5 Architektur der Software des Systems

Folgende Abbildung 44 zeigt eine vereinfachte Gesamtübersicht der Softwarearchitektur des Systems. In Kombination mit der Hardwarearchitektur unter Abschnitt 5.1.9 (S. 81) entsteht somit eine Gesamtarchitektur des Systems.

Abbildung 44: Gesamtübersicht der Softwarearchitektur

6. Aktueller Stand der prototypischen Umsetzung

In Kapitel 4 und 5 wurden Komponenten ausgewählt und ein Konzept für die Hardware und die Software des Tischroboters ausgearbeitet. Auf Grundlage des erstellten Konzepts konnte daraufhin ein Prototyp angefertigt werden, um die technische Umsetzbarkeit der verschiedenen Systemteile sowie deren Zusammenspiel zu untersuchen und auftretende Probleme zu evaluieren.

Dieses Kapitel beschreibt die wichtigsten Fortschritte und gewonnenen Erkenntnisse bei der prototypischen Umsetzung. Dabei wird insbesondere darauf eingegangen, welche der während der Konzeptionierung erstellten Lösungsansätze zum Erfolg führten und an welchen Stellen vom Konzept abgewichen werden musste.

Grundsätzlich wurde die Verknüpfung der Hardwarekomponenten so umgesetzt, wie es im Konzept in Abbildung 38 (S. 81) dargestellt wurde. Änderungen waren lediglich bei der Kommunikation zwischen der zentralen Recheneinheit und den Arduinos erforderlich. Diese werden in Punkt 6.2 beschrieben.

Punkt 6.4 beschreibt eine erste Erprobung des Gesamtsystems mithilfe der Steuerung von Servomotoren. Dabei werden sowohl das Front-End* und das Back-End* als auch die softwaretechnische Anbindung von Hardware getestet.

In Punkt 6.7 wird der finanzielle Zwischenstand vorgestellt. Dieser enthält die im Konzept ausgewählten Komponenten und beachtet außerdem zusätzliche, während der prototypischen Umsetzung entstandene Kosten.

Im letzten Abschnitt wird eine Abschlussbetrachtung der definierten Anforderungen durchgeführt, in welcher der Erfüllungsgrad zum aktuellen Zeitpunkt beurteilt wird.

6.1 Energieversorgung

Für eine Inbetriebnahme der Komponenten war zunächst die Realisierung einer funktionierenden **Energieversorgung** wichtig. Wie in Punkt 5.1.8 beschrieben, wurde dafür das 12 Volt Netzteil mit entsprechenden **Step-Down-Spannungsreglern*** verknüpft, um alle notwendigen Spannungen des Systems zur Verfügung zu stellen. Damit mehrere Komponenten an die Energieversorgung angeschlossen werden können, wurde zusätzlich ein Energieverteiler hinzugefügt, auf welchem sich für jede Spannung jeweils 10 bis 20 Anschlusskontakte befinden. Der Aufbau der prototypischen Energieversorgung ist in Abbildung 45 zu sehen. Aus Stabilitätsgründen wurden die Komponenten auf einem Holzbrett befestigt.

Abbildung 45: Prototyp der Energieversorgung mit Netzteil

Die Energieversorgung über die **Powerbank** wurde ebenfalls getestet. Ein Betrieb des Jetson Nano Boards konnte darüber problemlos gewährleistet werden. Dabei ist allerdings aufgefallen, dass die Powerbank bei Bedarf erst nach einer kurzen Verzögerung von ca. einer Sekunde Energie liefert. Bei einem alleinigen Akkubetrieb wäre diese Verzögerung nicht problematisch. Bei einem gemischten Betrieb mit Netzteil würde die Zeitspanne jedoch ausreichen, um beim Entfernen des Netzteils einen kurzzeitigen Absturz des gesamten Systems herbeizuführen.

Ein einfaches **Umschalten** der **Energiequellen** über die in 5.1.8.5 beschriebene Dioden-Schaltung ist daher nicht ausreichend, um einen kontinuierlichen Betrieb des Systems zu gewährleisten. Als Lösung könnten zusätzliche Stützkondensatoren eingebaut werden, welche den Jetson solange mit Strom versorgen würden, bis der Akku Energie liefert. Der Aufbau der Schaltung erfordert jedoch eine dauerhafte Überwachung und ein intelligentes Schalten der verschiedenen Energiequellen, damit die Energiezufuhr zu keiner Zeit unterbrochen ist.

Im Rahmen dieser Arbeit wurde der Akkubetrieb aufgrund der gegebenen Komplexität nicht umgesetzt. Jedoch wird die Entwicklung einer stabilen Energieversorgung mit Akkubetrieb für die Zukunft angestrebt. Bei der Konzeptionierung dieser Energieversorgung könnte ebenfalls vorgesehen werden, auf Dioden zu verzichten, da diese durch den Spannungsabfall eine zu hohe Verlustleistung aufweisen. Alternativ kann das Umschalten der Spannungsquellen auf der Grundlage von MOSFETs⁶⁷ entwickelt werden.

Beim Testen des Akkubetriebs wurde außerdem festgestellt, dass das Auslesen des aktuellen Ladezustands bei einer normalen Powerbank nicht möglich ist. Als Lösung könnte der aktuelle Energiefluss dauerhaft erfasst und verfolgt werden, sowohl beim Aufladen als auch beim Entladen des Akkus. Dadurch wäre das System in der Lage, die zur Verfügung stehende Energie einzuschätzen.

⁶⁷ Als Referenz für eine MOSFET-Schaltung zur Umschaltung von Energiequellen kann beispielsweise der LM5050-Schaltkreis von Texas Instruments verwendet werden. Dieser bietet die gleiche Funktionalität wie die entwickelte Dioden-Schaltung, bloß ohne Spannungsabfall. Der interne Aufbau des Schaltkreises ist im Datenblatt beschrieben. Siehe <http://www.ti.com/product/LM5050-1> (abgerufen am 13.08.2019)

6.2 Kommunikation zwischen Jetson Nano und Arduino Nano

Laut Konzept soll die Verbindung zwischen dem Jetson Nano und den angeschlossenen Arduinos über **SPI** erfolgen. Beim Testen wurde festgestellt, dass der Jetson Nano zwar zwei SPI Schnittstellen besitzt, diese aber standardmäßig deaktiviert sind. Für eine Benutzung müssten Teile des Betriebssystems bzw. des Kernels konfiguriert und neu installiert werden. [63]

Die Installation ist mit dem Risiko verbunden, wichtige Systemdaten oder andere Funktionen zu beschädigen. Daher wurde für die prototypische Umsetzung beschlossen, mit allen Komponenten über die **I²C Schnittstelle** zu kommunizieren. Diese ist bereits standardmäßig im Betriebssystem aktiviert und kann mit entsprechenden Python oder C Bibliotheken angesprochen werden. [64, 65]

Wie in Punkt 5.1.8.4 beschrieben wurde, muss für die Kommunikation zwischen verschiedenen Spannungen eine Pegelwandlung vorgenommen werden. Der ausgewählte TXS0108E-Schaltkreis, welcher acht Kanäle für ein solches **Level-Shifting** besitzt, stellte sich jedoch als dafür nicht geeignet heraus. Bei der Untersuchung mit einem Oszilloskop wurde herausgefunden, dass die Datenrate von I²C für den Schaltkreis zu hoch war und die Pegel vom Schaltkreis nicht schnell genug umgewandelt werden konnten.

Daher wurde die in Abbildung 35 (S. 78) beschriebene Pegelwandlung mithilfe von **MOSFETs** umgesetzt. Der prototypische Aufbau auf einer Universalleiterplatte ist in folgender Abbildung 46 zu sehen. Auf der Oberseite gibt es für das Level-Shifting vier Kanäle mit jeweils neun Anschlüssen. Die notwendigen MOSFETs und Widerstände befinden sich auf der Unterseite der Leiterplatte. Diese sind als SMD-Bauteile• sehr klein und daher ohne entsprechende Vergrößerung schwer zu erkennen.

Abbildung 46: Prototypischer Aufbau des Level-Shiftings mit MOSFETs

Mit dem Prototyp war es möglich, über I²C eine erfolgreiche Kommunikation zwischen dem Jetson Nano und einem Arduino Nano herzustellen und Daten auszutauschen.

6.3 Hello World Programme

Im Rahmen der Bearbeitungszeit wurde der Großteil der in 5.2.4 geforderten HelloWorld Programme entwickelt. Da diese eine Grundlage für die Weiterentwicklung des Projekts darstellen, wurde Wert auf eine verständliche Strukturierung der Programme und auf umfassende Kommentierung gelegt. Viele der Komponenten und Frameworks besitzen bereits HelloWorld Programme und Tutorials für den Schnelleinstieg. Diese mussten lediglich auf das Projekt angepasst, miteinander verknüpft und geringfügig erweitert werden.

Lediglich das Programm **HelloSPI**, welches die Verwendung der SPI-Schnittstelle erläutern soll, wurde zum aktuellen Bearbeitungsstand nicht umgesetzt. Grund dafür sind die in Abschnitt 6.2 aufgeführten Probleme bezüglich der Aktivierung der Schnittstelle.

6.4 Grafische Anwendung und ROS Back-End

Ein Bestandteil der prototypischen Umsetzung war die praktische Evaluation des Softwarekonzepts. Dafür wurde exemplarisch ein **Anwendungsfall** umgesetzt, bei welchem angeschlossene Servomotoren über das Display des Roboters gesteuert werden sollen. Dieser Anwendungsfall bezieht alle Aspekte der Softwarearchitektur ein, von der Eingabe auf einer grafischen Oberfläche über die Verarbeitung der Eingabe im Back-End bis hin zur Ansteuerung der Hardware mithilfe eines Arduinos.

Zunächst wurde die Kommunikation zwischen einem **Arduino Nano**, an dem zwei Servomotoren angeschlossen sind, und dem **Jetson Nano** hergestellt, damit dieser über entsprechende I²C Kommandos⁶⁸ die gewünschte Motorposition einstellen kann. Dabei wurde jedoch offensichtlich, dass die genutzte Arduino-Bibliothek für die Steuerung der Servomotoren nicht kompatibel mit den verwendeten Motoren ist, sodass kein exaktes Einstellen des gewünschten Drehwinkels möglich war. Für den Prototypen wurde daher als Datensatz statt der Angabe des Drehwinkels die exakte **Pulsweite für die PWM-Steuerung** des Motors gesendet. Diese wird in Mikrosekunden angegeben und ermöglicht eine vom Typ des Servomotors unabhängige Steuerung der gewünschten Position.

Anschließend wurden im **Back-End** zwei **ROS-Knoten** entwickelt: *motorControl* im Package *head* und *i2cBridge* im Package *system*⁶⁹. Der *i2cBridge*-Knoten implementiert die Abstraktionsschicht zur I²C-Schnittstelle und abonniert verschiedene Topics, über die Daten auf der Hardwareschnittstelle versendet werden können. Der *motorControl*-Knoten abonniert jeweils ein Topic, über welches die gewünschte Pulsweite eines Motors eingestellt werden kann. Sobald eine Nachricht empfangen wird, veröffentlicht der Knoten das passende Kommando und die zugehörigen Daten an den *i2cBridge*-Knoten.

⁶⁸ Siehe Punkt 5.2.3 für die Kommandos zur Kommunikation mit dem Arduino. Siehe außerdem Punkt 6.2 für die Begründung, warum I²C statt SPI für die Kommunikation zwischen Arduino Nano und Jetson Nano verwendet wird.

⁶⁹ Siehe Punkt 5.2.1 für die Übersicht der im Back-End geplanten Packages.

Als dritter Teil des Anwendungsfalls wurde die **grafische Anwendung** mithilfe des **Electron Frameworks** umgesetzt. Das Programm baut dabei auf dem zuvor entwickelten HelloWorld Programm für Electron auf und ergänzt ein separates Fenster für die Motorsteuerung. Dieses Fenster besitzt zwei Schieberegler, welche die Position für zwei angeschlossene Motoren bestimmen sollen. Für ein modernes und responsives Design der Oberfläche wurde das CSS-Framework Bootstrap⁷⁰ eingebunden. Die resultierende Oberfläche ist in folgender Abbildung 47 zu sehen.

The image shows a web application titled "Motor Control". It contains two identical-looking control sections. Each section has a title, a slider, and three input fields. The first section is for "Motor for Head Turn (MicroSeconds)" and the second is for "Motor for Head Pitch (MicroSeconds)". Both sliders have a blue knob. Below each slider are three input fields labeled "Min", "Current", and "Max".

Motor	Min	Current	Max
Motor for Head Turn (MicroSeconds)	400	1435	2500
Motor for Head Pitch (MicroSeconds)	400	2088	2500

Abbildung 47: Grafische Oberfläche für die Steuerung angeschlossener Motoren

Damit die grafische Oberfläche mit dem ROS-Back-End kommunizieren kann, wurde zuletzt ein weiterer **ROS-Knoten** entwickelt, welcher die Verknüpfung über **Redis** ermöglicht⁷¹. Dieser *redisBridge*-Knoten fungiert als Weiterleitung, indem er die notwendigen Topics von Redis abonniert und zugehörige Publisher für ROS implementiert.

Sobald nun ein Wert in der grafischen Oberfläche verändert wurde, kann dieser über Redis veröffentlicht werden. Der *redisBridge*-Knoten leitet die Nachricht an das passenden ROS-Topic weiter, in diesen Fall an den Knoten *motorControl*. Sobald dieser die Nachricht empfängt, versendet er mithilfe des *i2cBridge*-Knotens ein Kommando für den Arduino, um die gewünschte Position des Motors einzustellen. Das Prinzip ist in folgender Abbildung 48 dargestellt.

⁷⁰ Bootstrap ist ein freies CSS-Framework, welches von Twitter entwickelt wird. Es ermöglicht die Gestaltung moderner und responsiver Websites. Offizielle Seite: <https://getbootstrap.com/> (abgerufen am 12.08.2019)

⁷¹ Siehe Punkt 5.2.2 für die Erläuterung der Kommunikation zwischen Front- und Back-End.

Abbildung 48: Architektur der Motorsteuerung

Nach der Verknüpfung von Front-End, Back-End und Hardware konnten die beiden angeschlossenen Motoren über die grafische Anwendung auf dem Touch-Display gesteuert werden.

Die Umsetzung des Anwendungsfalls hat bestätigt, dass die im **Konzept erstellte Architektur der Software realisierbar** ist und zuverlässig funktioniert. Die Kommunikation zwischen den verschiedenen Komponenten der Architektur hat eine sehr geringe Latenz, sodass auch ein Test mit einhundert Steuerbefehlen pro Sekunde verarbeitet werden konnte. Außerdem erwies sich der Einsatz von ROS insofern als sehr positiv, dass Knoten bei laufendem Betrieb flexibel gestartet, beendet und bearbeitet werden können. Im Vergleich zur Arbeit mit einem NAO-Roboter wird somit ein schnelleres Testen ermöglicht.

Folgende Abbildung 49 zeigt ein Foto der realen Umsetzung von der Motorsteuerung. Die angeschlossenen Servomotoren besitzen jeweils einen Arbeitsbereich von etwa 180°. Im linken Teil des Bildes ist bei beiden Motoren der kleinste Winkel, im rechten Teil des Bildes der größte Winkel eingestellt.

Abbildung 49: Foto der praktisch umgesetzten Motorsteuerung

6.5 Lautsprecher und Audioausgabe über das ReSpeaker-Modul

Im Rahmen der Arbeit wurden die Lautsprecher lediglich ohne eine umgebendes Lautsprechergehäuse getestet. In Abbildung 50 ist der Aufbau eines prototypischen **Lautsprechersystems** zu sehen. Dabei ist ein FR 58 Breitbandlautsprecher an den 20 Watt Audioverstärker angeschlossen, welcher mit dem ReSpeaker-Modul verbunden ist.

Abbildung 50: Verknüpfung von Audioverstärker, FR 58 Breitbandlautsprecher und ReSpeaker-Modul (v.l.n.r)

Bereits ohne Gehäuse ist der Klang des Lautsprechers angenehmer als bei einem Smartphone. Vor allem hat der Test ergeben, dass die Lautstärke durch den vorgesetzten Verstärker wesentlich höher einstellbar ist. Jedoch wird die **Qualität des Klangs** im späteren Betrieb erst in Verbindung mit einem Lautsprechergehäuse bzw. des Robotergehäuses beurteilt werden können.

Gleiches gilt für den Test des größeren DX 4X6X Lautsprechers mit integrierter Frequenzweiche. Besonders tiefere Töne sollten bei diesem Lautsprecher besser klingen als beim FR 58, jedoch werden Bassfrequenzen ohne ein passendes Gehäuse nahezu vollständig eliminiert.

Bei der Verwendung des ReSpeaker als Ausgabegerät für Audiosignale ist jedoch ein permanentes, leichtes **Rauschen** aufgefallen. Dieses ist besonders dann zu hören, wenn keine anderen Töne ausgegeben werden. Es muss noch evaluiert werden, ob es sich dabei um einen möglichen Softwarefehler oder ein Problem mit dem ReSpeaker selbst handelt.

6.6 Kapazitive Sensoren

Bei der Konzeptionierung des Systems wurde im Abschnitt 5.1.2 der Einschalter des Roboters und das Prinzip von kapazitiven Sensoren erläutert. Für die Realisierung eines solchen Sensors sollten in der prototypischen Umsetzung sowohl die alleinige Verwendung eines Arduinos als auch die Verwendung des dedizierten Schaltkreises CAP1298 getestet werden.

Folgende Abbildung 51 zeigt den Aufbau eines **kapazitiven Sensors** ohne einen speziellen Schaltkreis⁷². Dabei sind zwei Pins des Arduinos über einen 4 Mega-Ohm Widerstand miteinander verbunden, zusätzlich ist ein Stück Alufolie, welche als Sensor fungiert, über ein Kabel mit dem Eingangs-Pin verbunden. In der Software⁷³ konnte anschließend eine Schwankung der Zeitmessung festgestellt werden, sobald ein Finger über die Alufolie bewegt wird. Im rechten Teil der Abbildung ist eine leuchtende LED zu sehen, die ab einem definierten Schwellwert aktiviert wird. Im Bild befindet sich der Finger etwa einen Zentimeter über der Folie.

Abbildung 51: Umsetzung eines kapazitiven Sensors mit einem Arduino

Der Test des **CAP1298** verlief analog zum ersten Test und ist in Abbildung 52 dargestellt. Der Schaltkreis wurde zunächst auf die im Mittelteil des Bilds zu sehende Adapterplatine gelötet, um ihn prototypisch verwenden zu können. An einem der acht Sensoreingänge des Schaltkreises ist über ein Kabel ein Stück Kupferfolie⁷⁴ angebracht. Über I²C konnte eine Verbindung zwischen dem Arduino und dem Schaltkreis aufgebaut werden, um die Sensitivität einzustellen und den Status des Sensors auszulesen. Erneut ist im rechten Teil der Abbildung zu sehen, wie die LED zu leuchten anfängt, sobald sich der Finger etwa einen Zentimeter über der Folie befindet.

⁷² Siehe Punkt 5.1.2 (S. 64) für die Erläuterung des Prinzips eines kapazitiven Sensors.

⁷³ Siehe das HelloWorld Programm **HelloCapacitiveSensor**.

⁷⁴ Kupferfolie bzw. Kupferklebeband liefert bezüglich des Sensors dieselben Ergebnisse wie Alufolie, lässt sich aber einfacher verwenden, da Kabel direkt angelötet werden können.

Abbildung 52: Umsetzung eines kapazitiven Sensors mit einem CAP1298 Schaltkreis

Sowohl die Umsetzung mit dem Arduino als auch die Verwendung des CAP1298 ermöglichen den Einsatz eines kapazitiven Sensors. Langfristig wird der **dedizierte Schaltkreis präferiert**, da dieser einen stabileren Einsatz und eine simplere Auswertung erlaubt. Vor allem die automatische Rekalibrierung wird bei einem dauerhaften Einsatz als Vorteil gegenüber den selbstfestgelegten Schwellwerten des ersten Tests gesehen.

6.7 Finanzieller Stand der prototypischen Umsetzung

Der **finanzielle Zwischenstand**, welcher im letzten Abschnitt von Kapitel 4 betrachtet wurde, resultierte nach der Auswahl der Kernkomponenten für das System in einem Preis von etwa **420 €**. Während der Konzeptionierung und der prototypischen Umsetzung wurden diese Komponenten um weitere Bauteile ergänzt, die für das System notwendig sind. Dazu zählen beispielsweise das WLAN Modul für den Jetson, ein zweites Display für die Augen, die visuelle und taktile Rückmeldung für den Einschalter und die Powerbank. Für den Zusammenbau des Prototyps wurden außerdem elektronische Kleinbauteile und weiteres Zubehör wie Kabel benötigt.

Insgesamt ergeben alle Teile des Konzepts und der prototypischen Umsetzung einen Preis von knapp **630 €**. Durch zusätzliche Komponenten für praktische Versuche, die letztendlich nicht verwendet werden, sowie durch eingeplante Puffer für den Fall von Ausfällen während der Testphase sind diese Kosten jedoch etwas höher als letztendlich für das Ergebnis benötigt. Eine genaue Übersicht der erworbenen Teile für den Prototypen inklusive der Bezugsquellen ist in der beigefügten Tabelle *Einkaufsliste.xlsx* zu finden. Eine Zusammenfassung der Preise für die verschiedenen Teile des Systems ist in folgender Tabelle 22 zu sehen.

Tabelle 22: Kostenübersicht der prototypischen Umsetzung

Systemteil	Kosten
Zentrale Recheneinheit	ca. 142 €
Geräuschwahrnehmung	ca. 58 €
Audioausgabe	ca. 35 €
Display	ca. 50 €
Kamera	ca. 31 €
Kopf & Augen	ca. 83 €
Einschalter	ca. 22 €
Arduino	ca. 22 €
Energieversorgung	ca. 115 €
Gehäusekühlung	ca. 26 €
Zubehör	ca. 46 €
	ca. 630 €

In dem Preis von 630 € sind lediglich die Kosten für das Gehäuse und den Zusammenbau des Roboters nicht enthalten. Je nach verwendeten Materialien werden für das Gehäuse weitere Kosten von schätzungsweise 50 € bis 150 € entstehen⁷⁵.

⁷⁵ Dieser Schätzwert beruht auf Erfahrungswerten des 3D-Drucks und des Laserschneidens im Vinn:Lab der TH Wildau; offizielle Seite des Vinn:Labs: <https://vinnlab.th-wildau.de/> (abgerufen am 14.08.2019)

6.8 Abschlussbetrachtung der definierten Anforderungen

Im Folgenden soll eine Übersicht gegeben werden, inwieweit die in Abschnitt 3.3 definierten Anforderungen zum aktuellen Bearbeitungsstand⁷⁶ der Arbeit erfüllt wurden. Insgesamt wurden 42 Anforderungen an das System definiert. Die beiden Diagramme der folgenden Abbildung 53 stellen den Anteil der erfüllten Anforderungen grafisch dar.

Abbildung 53: Erfüllung der gestellten Anforderungen

Die **Muss-Anforderungen** wurden im Rahmen dieser Arbeit zu einem Großteil erfüllt. Alle obligatorischen Kriterien bezüglich der Hardwareverknüpfung und der notwendigen Software konnten realisiert werden.

Lediglich das Design und damit das Gehäuse des Roboters wurden zum aktuellen Zeitpunkt noch nicht umgesetzt. Zum einen wurde der Fokus im Verlauf der Arbeit auf die konzeptionelle Entwicklung des Tischroboters gelegt, zum anderen wurde beschlossen, das Aussehen des Roboters gemeinsam im Team des RoboticLabs zu erarbeiten. Das fehlende Gehäuse beinhaltet die **drei** bislang nicht erfüllten **Muss-Anforderungen** und außerdem **acht** der noch unerfüllten **Soll-Anforderungen** (R032 bis R042).

Neben dem Gehäuse wurde außerdem der **Akkubetrieb** nicht realisiert (**R018** und **R019**). Wie in Punkt 6.1 erläutert wurde, ist das zuverlässige Umschalten zwischen der Energieversorgung über das Netzteil und der Versorgung über die Powerbank eine komplexere Aufgabe und erfordert die Entwicklung einer intelligenten, elektronischen Schaltung. Angeknüpft an die Energieversorgung wurde ebenfalls noch kein **Einschalter** in die prototypische Umsetzung integriert, wodurch Anforderung **R020** gleichermaßen als noch nicht erfüllt gilt.

Die **finanzielle Übersicht** des vorangegangenen Abschnitts 6.7 hat aktuelle Kosten von etwa 630 € ergeben. Damit wird die in **R002** definierte Grenze von 600 € leicht überschritten, wodurch diese Anforderung mit den aktuell ausgewählten Bauteilen ebenfalls als nicht erfüllt betrachtet wird.

⁷⁶ Stand August 2019

7. Abschlussbetrachtung

Dieses letzte Kapitel soll eine zusammenfassende Betrachtung dieser Bachelorarbeit in Form eines Fazits geben. Abschließend werden die nächsten Schritte für den weiteren Projektverlauf in Form eines kurzen Ausblicks dargestellt.

7.1 Fazit

Im Rahmen dieser Bachelorarbeit wurde ein prototypischer Tischroboter für den Einsatz im Bereich des AAL konzeptioniert. Das Ziel bestand vor allem darin, die technische Umsetzbarkeit innerhalb des festgelegten Budgets zu untersuchen und damit eine Eignung als Alternative zum NAO festzustellen.

Zumindest bezüglich des Aussehens ist der entstandene Prototyp des Systems noch ein Stück von einem einsetzbaren Tischroboter entfernt, wie in Abbildung 54 zu sehen ist. Da die Komponenten zum endgültigen Stand der Bachelorarbeit noch nicht in einem Gehäuse untergebracht wurden, ist die Entwicklung eines Designs noch eine offene Aufgabe, welche die höchste Priorität für den weiteren Verlauf des Projekts besitzt. Abgesehen davon ist jedoch das Ziel der Arbeit erreicht worden: Es ist möglich, einen Tischroboter zu konzeptionieren, der auf Interaktionsmöglichkeiten mit Menschen spezialisiert ist und der preislich einen dreistelligen Bereich nicht überschreitet. Dabei wurde trotzdem der Fokus auf die Qualität der einzelnen Komponenten gelegt.

Jetson Nano	1	7 LED Ring
Kamera Modul	2	8 Level Shifter für I²C
Energieversorgung	3	9 Servomotoren
ReSpeaker	4	10 Kapazitiver Sensor
Breitbandlautsprecher	5	11 Touch Display
20W Audio Verstärker	6	

Abbildung 54: Prototypische Verknüpfung der Systemkomponenten

Die wichtigste Anforderung des Projekts – die **funktionale Erweiterbarkeit des Systems** – ist durch das Ergebnis der Arbeit klar gegeben. Die modulare Architektur erlaubt ein flexibles Hinzufügen und Austauschen von Hardwarekomponenten. Nicht nur einzelne Teile wie die Kamera, das Display oder die Motoren können ausgetauscht werden. Es ist ebenso möglich, den Roboter auf einer anderen Grundlage wie dem Raspberry Pi aufzubauen oder auch ein hybrides System zu entwickeln, wie im Ausblick unter Punkt 7.2.4 beschrieben wird.

Aber auch die **Softwarearchitektur** bietet ein hohes Maß an Flexibilität. Die Verwendung von **ROS** im Back-End erlaubt den Aufbau eines modularen Systems, in dem einzelne Knoten verschiedene Daten und Dienste bereitstellen, sodass nachträglich weitere Knoten hinzugefügt und damit die Fähigkeiten des Roboters erweitert werden können. Generell hat sich die Verwendung von ROS in den praktischen Tests bewährt. Anders als es beim NAO der Fall ist, können einzelne Knoten im laufenden Betrieb problemlos gestartet, beendet und bearbeitet werden, wodurch ein agiles Entwickeln und Testen möglich ist. Auch das Erstellen von eigenen Nachrichtenformaten ermöglicht einen flexiblen und wohldefinierten Datenaustausch zwischen den Knoten.

Des Weiteren hat sich die Verwendung von **Redis** in den ersten, praktischen Tests als sehr nützlich erwiesen. Der Publish-Subscribe-Service der Datenbank erlaubt einen simpel zu benutzenden Datenaustausch zwischen verschiedenen Programmen, der quasi unabhängig von der Programmiersprache ist. Außerdem kann Redis als Key-Value-Datenbank^{*} auch gleichzeitig dafür genutzt werden, globale Konfigurationsdaten des Systems persistent und für alle Programme zugänglich abzuspeichern.

Neben ROS und Redis hat sich ebenfalls die Verwendung des **Electron-Frameworks** bewährt. Die Möglichkeit, eine Desktopanwendung mit den Elementen der Webprogrammierung zu verknüpfen, erlaubt eine sehr flexible, moderne Gestaltung der grafischen Oberfläche. Außerdem können beliebte Frameworks wie *Bootstrap* oder *Vue.js* in die Anwendung integriert werden, sodass bestehende Kenntnisse angewandt werden können und weitere, vielfältige Möglichkeiten entstehen. Ein weiterer Vorteil ist die Plattformunabhängigkeit von Electron-Apps. Da diese sowohl unter Windows als auch unter iOS und Linux laufen, kann die Entwicklung der Anwendung unabhängig vom Robotersystem stattfinden, sodass neue Funktionen komfortabel erarbeitet und getestet werden können.

Bis auf den Kopf sind für den Tischroboter derzeit keine Bewegungsmöglichkeiten geplant. Jedoch kann die entworfene Architektur gleichermaßen für Erweiterungen oder als Basis für andere Systeme genutzt werden. So ist es durchaus denkbar, dass weitere, bewegliche Service-Roboter auf der Grundlage des entstandenen Konzepts entwickelt werden.

Abgesehen von der fehlenden Fortbewegungsmöglichkeit ist der konzipierte Tischroboter deutlich **flexibler als ein NAO-Roboter**⁷⁷. Vor allem das Touch-Display bietet eine vielfältig nutzbare Möglichkeit für die Darstellung von Informationen und die Interaktion mit Anwendern. Darüber hinaus ermöglicht die Architektur des Systems die Verknüpfung

⁷⁷ Bezüglich der Version NAO V4, die hauptsächlich in der Telematik-Lehre eingesetzt wird, bietet der Jetson Nano zusätzlich eine signifikant höhere Performance (siehe die Leistungsmerkmale des NAO-Roboters unter Punkt 8.4 im Anhang S. 131). Die aktuelle Version NAO V6 besitzt eine vergleichbare CPU zum Jetson Nano.

weiterer Hardwaremodule, um die Funktionalität des Roboters auszubauen. Des Weiteren ist durch den Eigenbau die Unabhängigkeit von einer Komplettlösung gegeben. Entwickler bzw. Studierende haben somit vollen Zugriff auf das System, können neue Softwarepakete hinzufügen und den Roboter auf einem aktuellen Stand halten.

Mit den derzeitigen Kosten von rund 630 € liegt das System auch deutlich unter dem Preis des NAOs. Zusammen mit den fehlenden Kosten des Gehäuses wird die als wünschenswert definierte Grenze von 600 € zwar leicht überschritten, jedoch wird der Preis aufgrund der qualitativen Komponenten als gerechtfertigt bewertet. Auf jeden Fall ist der Bau eines Tischroboters mit Kosten im dreistelligen Bereich möglich.

Die Rechenleistung des Systems wird als ausreichend eingeschätzt, um eine Vielzahl an Anwendungsfällen lokal umzusetzen. Bei Aufgaben, für welche die lokale Rechenleistung nicht ausreichend ist, existiert die Ausweichlösung einer **Cloud-Verarbeitung**. Dabei wird der Hauptanteil von Datenverarbeitung und Auswertung auf leistungsfähige Cloud-Server verlagert, sodass die lokale Recheneinheit lediglich die Eingabedaten bereitstellen und übermitteln muss und anschließend die Ergebnisse weiterverarbeiten kann. Besonders in Szenarien wie der Auswertung gesprochener Sätze kann eine solche Verlagerung der Verarbeitung in die Cloud notwendig sein. Das System wird dabei jedoch abhängig von Back-End-Servern und Übertragungsgeschwindigkeiten bzw. -latenzen. Grundsätzlich wird daher eine lokale Verarbeitung bevorzugt.

Aktuelles Ziel ist es, noch bis zum Ende des Jahres 2019 **einen fertigen Prototyp** des Tischroboters mit Gehäuse zu bauen. Anschließend kann die Umsetzung realer Anwendungsfälle erfolgen. Im ersten Schritt wäre es denkbar, bereits mit dem NAO-Roboter umgesetzte Projekte zu adaptieren und auf dem Tischroboter zu testen. Bei der Realisierung von Anwendungsfällen im Bereich des AAL sollte der Roboter dabei auch direkt im späteren Umfeld und unter Einbeziehung von Personen der Zielgruppe getestet und weiterentwickelt werden.

Damit der Nutzer eine emotionale Beziehung bzw. eine **emotionale Akzeptanz** zum Tischroboter aufbauen kann, ist die Umsetzung weiterer Arbeitspakete erforderlich. So ist es etwa wichtig, dass der Roboter über Gesichtserkennung in der Lage ist, Menschen in der Nähe direkt anzugucken und dass über die Animation von Augenformen auf den zugehörigen Displays verschiedene Emotionen zum Ausdruck gebracht werden können. Generell werden ein autonomes Verhalten und eine Art von individueller Persönlichkeit für die emotionale Akzeptanz eine entscheidende Rolle spielen.

Neben den genannten nächsten Umsetzungsschritten fehlt zum jetzigen Zeitpunkt eine Bezeichnung für das Projekt bzw. ein Name für den Roboter.

7.2 Ausblick

Die Bachelorarbeit hat die konzeptionellen Grundlagen für den Bau eines Tischroboters geschaffen. Dieser Abschnitt führt abschließend offen gebliebene Punkte und Ideen für den weiteren Projektverlauf auf.

7.2.1 Gehäuse des Tischroboters

Zum jetzigen Zeitpunkt⁷⁸ sind sowohl Hardware als auch Software für ein funktionierendes System miteinander verknüpft. Allerdings fehlt noch ein **Gehäuse** für den Roboter. Das Gehäuse dient sowohl einer sinnvollen Anordnung aller Komponenten und der Sensorik als auch dem Schutz der für den Nutzer irrelevanten Technik. Vor allem bestimmt das Gehäuse jedoch maßgeblich das Aussehen des Roboters, welches wiederum ein wichtiges Kriterium für die **emotionale Akzeptanz** beim Nutzer ist. Daher soll in näherer Zukunft großer Wert auf die Entwicklung eines ansprechenden Designs gelegt werden.

Wie bereits erwähnt, soll das Design gemeinsam im RoboticLab erarbeitet werden, um ein qualitatives Ergebnis zu erreichen. Für die prototypische Entwicklung des Gehäuses bietet sich **3D-Druck** als sehr flexible Methode an. Denkbar wäre auch eine Kombination aus 3D-Druck mit einem weiteren Material. So könnte beispielsweise ein formgebendes Skelett ausgedruckt werden, in welchem ebene Flächen mit **Acryl-Glas** oder einem ähnlichen Kunststoff gefüllt werden. Dies lässt sich nicht nur designtechnisch einsetzen, sondern kann auch Kosten und vor allem Zeit gegenüber einem reinen 3D-Druck des kompletten Gehäuses einsparen.

Bevor ein endgültiges Design erstellt wird, müssen ebenfalls die **Lautsprecher** und deren Klang bei der Integration in ein Gehäuse getestet werden.

⁷⁸ Ende August 2019

7.2.2 Basis für die Weiterentwicklung

Der Tischroboter soll als Forschungs- und Entwicklungsplattform genutzt und in Folgeprojekten erweitert werden können. In Abschlussarbeiten, in Praktika oder auch in der Lehre sollen Studierende die Möglichkeiten eines Tischroboters evaluieren und konkrete Anwendungsfälle umsetzen können. Zum aktuellen Bearbeitungsstand der Bachelorarbeit existieren die beschriebenen HelloWorld Programme, in denen die vorhandenen Komponenten des Roboters und deren Verwendung kompakt erläutert werden. Diese werden allerdings zurzeit noch nicht in einen weiteren Zusammenhang miteinander gebracht, um komplexere Abläufe des Roboters exemplarisch darzustellen.

Aufgrund des prototypischen Status fehlt eine umfassende Dokumentation mit weiterführenden Tutorials und Erläuterungen zu bestimmten Aspekten des Roboters. Eine solche wird zu einem späteren Entwicklungsstand notwendig sein. Angelehnt an das Programm Choregraphie für den NAOs wäre es ebenfalls denkbar, zukünftig ein ähnliches Programm für die grafische Verknüpfung und Programmierung der Komponenten des Roboters zu entwickeln, um diese auf eine simple Weise zugänglich zu machen.

7.2.3 Entwicklung eines Akkubetrieb

Für einen flexiblen Einsatz des Roboters und eine temporäre Unabhängigkeit vom Netzteil ist in Zukunft die Möglichkeit eines **Akkubetriebs** erforderlich. Wie in Abschnitt 6.1 erläutert, ist die Entwicklung einer stabilen Energieversorgung eine komplexere Aufgabe. Dazu gehören ein unterbrechungsfreies Umschalten zwischen Netz- und Akkubetrieb und eine Erfassung des aktuellen Energiestands beim Laden und Entladen des Akkus.

Grundsätzlich ist die Verwendung von **Powerbanks** für den Akkubetrieb insofern sinnvoll, dass diese Geräte zertifiziert sind und somit ein sicherer Betrieb gewährleistet wird. Außerdem ist langfristig ein einfacher Austausch der Powerbanks bei Defekten oder Altersschwächen möglich, da diese über USB an das System angeschlossen werden. Möglicherweise ist jedoch die Leistung einer einzelnen Powerbank nicht für den gleichzeitigen Betrieb aller Systemkomponenten ausreichend, weswegen zukünftig auch der parallele Einsatz mehrerer Powerbanks in Betracht gezogen werden kann.

7.2.4 Hybride Rechnerarchitektur

In der Evaluation der Systemkomponenten (Abschnitt 4.1.2) wurde der Raspberry Pi 3 Model B+ als Referenz für die Auswahl der zentralen Recheneinheit betrachtet. Während der Anfertigung dieser Bachelorarbeit ist der **Raspberry Pi 4 Model B** als neueste Version des Einplatinencomputers veröffentlicht worden⁷⁹.

Der Raspberry Pi 4 besitzt im Vergleich zum Vorgänger einen moderneren Prozessor, dessen CPU-Rechenleistung die des Jetson Nano übersteigt. Zusätzlich bietet der neue Raspberry bis zu 4 GByte Arbeitsspeicher. Damit wurden die Kritikpunkte, welche während der Evaluation gegen die Verwendung des Raspberry sprachen, in der neuen Version behoben. Vermutlich ist daher ein Raspberry als Basis für den Roboters ebenfalls geeignet, was in einer genaueren Untersuchung betrachtet werden müsste. Vor allem die große Community des Raspberry, mit dessen Hilfe auftretende Probleme sehr schnell behoben werden können, bleibt der größte Vorteil dieses SBC.

Dennoch bleibt das Potenzial des **Jetson Nano** nach wie vor bestehen. Der leistungsstarke Grafikprozessor bietet in den Bereichen der Künstlichen Intelligenz und der Bildverarbeitung viele Möglichkeiten, die im Rahmen dieser Arbeit nicht erforscht werden konnten. Nvidia stellt zudem ein großes, kostenloses Angebot für Entwickler zur Verfügung, in denen gerade diese Möglichkeiten des Jetson vermittelt werden⁸⁰.

Für eine deutliche Leistungssteigerung des Tischroboters wäre es auch denkbar, das System nicht nur auf einer zentralen Recheneinheit aufzubauen, sondern eine **hybride Architektur** zu verwenden. So könnte der Jetson Nano eingesetzt werden, um rechenintensive Prozesse und Analysen im Hintergrund zu verarbeiten, während ein Raspberry Pi 4 als zweiter Rechner die grafische Anwendung und die Interaktion mit dem Nutzer steuert. Bei der Aufteilung der Systemaufgaben unter zwei performanten Recheneinheiten kann eine deutliche Leistungssteigerung angenommen werden. Besonders das Nutzererlebnis bei der Verwendung der grafischen Oberfläche und beim Anschauen von Videoinhalten würde stabiler und unabhängig von den Hintergrundprozessen werden. Lediglich die Kommunikation zwischen den Systemteilen würde einen Mehraufwand mit sich bringen.

Die konzeptionierte Softwarearchitektur, in welcher Redis als Abstraktionsschicht für die grafische Anwendung genutzt wird, erlaubt jedoch eine problemlose Erweiterung auf ein verteiltes System. Werden die Recheneinheiten über Ethernet miteinander verbunden, können beide Computer auf dieselbe Redis Datenbank zugreifen und somit Daten über den Publish-Subscribe-Service austauschen können. Auch ROS erlaubt eine Verknüpfung mehrerer Knoten, welche sich auf physikalisch getrennten Maschinen befinden⁸¹.

⁷⁹ Siehe <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/> (abgerufen am 18.08.2019) für Informationen zum Raspberry Pi 4 Model B

⁸⁰ Siehe <https://developer.nvidia.com/embedded-computing> (abgerufen am 17.08.2019) für das Angebot für Entwickler von Nvidia.

⁸¹ Siehe <http://wiki.ros.org/ROS/NetworkSetup> (abgerufen am 17.08.2019) für die Konfiguration eines verteilten ROS-Systems.

*„Die beste Möglichkeit, die Zukunft vorherzusagen,
ist, sie selbst zu gestalten.“*

8. Anhang

8.1 Beigelegte CD-ROM

Die beigelegte CD-ROM enthält folgende Inhalte:

- **Doku:**
 - *Bachelorarbeit.docx* → Word Dokument der Bachelorarbeit
 - *Bachelorarbeit.pdf* → PDF Dokument der Bachelorarbeit
- **Tabellen:**
 - *Anforderungsendbetrachtung.xlsx* → Übersicht der Erfüllung der Anforderungen
 - *Einkaufsliste.xlsx* → Übersicht aller im Konzept ausgewählten und der für die Umsetzung bestellten Bauteile
 - *Hardware Komponenten Vergleich.xlsx* → Vergleiche von Hardwarekomponenten, mehrere Tabellen
 - *SBC Vergleich.xlsx* → detaillierter Vergleich verschiedener Einplatinencomputer
- **Bilder:**
 - enthält alle in der Bachelorarbeit verwendeten Bilder
- **Hardware:**
 - enthält das Projekt *EnergieversorgungBA* für das Programm Eagle von Autodesk → in diesem Projekt/Ordner sind Schaltpläne für die Energieversorgung und das Level-Shifting vorhanden
- **Software:**
 - **Arduino:**
 - enthält alle HelloWorld Programme für die Arduino Nanos
 - **Jetson:**
 - **electron:**
 - *helloElectron* → das HelloWorld Projekt für die Benutzung des Electron.js Frameworks
 - *robDesk* → das Electron Projekt für die grafische Anwendung des in Punkt 6.4 umgesetzten Anwendungsfalls
 - **helloWorld:**
 - enthält alle HelloWorld Programme für die direkte Ansteuerung von Hardwarekomponenten
 - **ros:**
 - *HelloRos.md* → Zusammenstellung nützlicher Befehle für den Umgang mit ROS
 - **ros_workspace/src:**
 - enthält alle in Punkt 5.2.1 aufgeführten ROS-Packages
 - in den Packages sind außerdem die Knoten des in Punkt 6.4 umgesetzten Anwendungsfalls enthalten

8.2 Deutsche Distributoren

Folgende Tabelle 23 führt bekannte, deutsche Distributoren auf.

Tabelle 23: Übersicht deutscher Distributoren

Distributor	Offizielle Website	unter anderem bekannt für
Alternate	https://www.alternate.de/	Hardwarekomponenten für PCs und Server
Amazon	https://www.amazon.de/	Produkte aus allen Bereichen, in den letzten Jahren auch immer mehr für elektronische Bauteile aus China
Conrad	https://www.conrad.de/	Breites Elektronik Sortiment
ELV	https://www.elv.de/	Elektronik aus Eigenbau Produkte für Elektronik-Projekte, Techniker und Bastler
Exp-Tech	https://www.exp-tech.de/	Module und Komponenten für SBC-Systeme, eingebettete Systeme und Mikrocontroller
Mouser	https://www.exp-tech.de/	Großhändler für Elektronikbauteile
Nvidia	https://www.nvidia.com/de-de/shop/	Grafikprozessoren
Pollin	https://www.pollin.de/	Restposten und besondere Bauteile
Reichelt	https://www.reichelt.de/	Breites und günstiges Elektronik Sortiment Standard Elektronik Bauteile und Komponenten

8.3 Details zu den evaluierten Einplatinencomputern

Die folgenden Tabellen führen Details zu den in Abschnitt 4.1.2 evaluierten Einplatinencomputern auf.

8.3.1 Allgemeine Merkmale

Tabelle 24: Allgemeiner Merkmale der Einplatinencomputer

	Raspberry Pi 3 Model B+	Banana Pi M64	Odroid N2	Jetson Nano
Quellen ⁸²	[21]	[26, 27]	[30, 31]	[32, 34]
Preis	32,50 € Reichelt	61,49 € Conrad	94,95 € Pollin	109 € Nvidia
Größe	85 x 56 mm ²	92 x 60 mm ²	100 x 91 mm ²	100 x 80 mm ²

8.3.2 Leistungsspezifikationen

Tabelle 25: Leistungsspezifikationen der Einplatinencomputer

	Raspberry Pi 3 Model B+	Banana Pi M64	Odroid N2	Jetson Nano
CPU Typ (ARM Cortex)	A53	A53	A73 + A53	A57
CPU Kerne	4	4	4 + 2	4
CPU Frequenz	1400 MHz	1200 MHz	4 x 1800 MHz 2 x 1900 MHz	1430 MHz
GPU Typ	Broadcom VideoCore IV	Mali-400MP2	Mali-G52	Nvidia Maxwell
GPU Kerne	2	2	6	128
GPU Frequenz	400 MHz	400 MHz	846 MHz	920 MHz
RAM Typ	DDR2	DDR3	DDR4	DDR4
RAM Größe	1 GByte	2 GByte	4 GByte	4 GByte

⁸² Die Quellenverweise, aus denen die Fakten für Tabelle 24, Tabelle 25 und Tabelle 27 stammen.

8.3.3 Ergebnisse von Benchmark-Tests

Um die Performance besser beurteilen zu können, sind in Tabelle 26 die Ergebnisse einiger Benchmark Tests aufgelistet, die von der Fachzeitschrift *Make:* für verschiedene SBC durchgeführt wurden. Die Prozessorleistung wird sowohl beim Einsatz eines CPU-Kerns als auch bei der Verwendung aller Kerne anhand eines im Komprimierprogramms 7Zip enthaltenen Geschwindigkeitstests ermittelt⁸³. Der zweite Test misst die Geschwindigkeit des Arbeitsspeichers mithilfe des Programms *tinymembench* gemessen, wobei das Ergebnis den maximalen Durchsatz beim Lesen und Kopieren von zufälligen Daten aus dem Arbeitsspeicher darstellt⁸⁴ [66].

Tabelle 26: Benchmark Vergleich der Einplatinencomputer [66]

	Raspberry Pi 3 Model B+	Banana Pi M64	Odroid N2 (2 GByte Variante)	Jetson Nano
maximale Bildauflösung	1920 x 1080 @ 60 Hz	1280 x 720 @ 60 Hz	3840 x 2160 @ 60 Hz	3840 x 2160 @ 60 Hz
7zip Compress (Single Core)	674 MIPS	539 MIPS	895 MIPS	1273 MIPS
7zip Decompress (Single Core)	1202 MIPS	984 MIPS	1784 MIPS	1372 MIPS
7zip Compress (Multi Core)	1751 MIPS	1500 MIPS	4936 MIPS	3797 MIPS
7zip Decompress (Multi Core)	4028 MIPS	3480 MIPS	9071 MIPS	5025 MIPS
tinymembench (memcpy)	1327 MB/s	1318 MB/s	1627 MB/s	3589 MB/s
hdparm (cached)	632 MB/s	393 MB/s	406 MB/s	1557 MB/s

Bei den Benchmark-Tests wird die Leistung der GPU nicht miteinbezogen. Somit kann den Ergebnissen keine Aussage über die Performance des Jetson Nano bzw. den potenziell starken Geschwindigkeitsschub bei der Verwendung seines Grafikprozessors entnommen werden.

⁸³ Ergebnis in Million Instructions per Second (MIPS) → höher = besser

⁸⁴ Ergebnis in MByte pro Sekunde (MB/s) → höher = besser

8.3.4 Anschlüsse und Konnektivität

Tabelle 27: Vergleich der Konnektivität und der Anschlüsse der Einplatinencomputern

	Raspberry Pi 3 Model B+	Banana Pi M64	Odroid N2	Jetson Nano
Speicher	microSD	microSD 8 GByte eMMC vorhanden	microSD eMMC Anschluss	microSD
WLAN	802.11.b/g/n/ac	802.11.b/g/n	nicht vorhanden	802.11.ac erweiterbar
Bluetooth	4.2	4.0	nicht vorhanden	4.2 erweiterbar
Ethernet	1 x Gigabit	1 x Gigabit	1 x Gigabit	1 x Gigabit
USB 2	4	2	0	0
USB 3	0	0	4	4
Display	1 x HDMI	1 x HDMI	1 x HDMI	1 x HDMI 1 x DP
Kamera	1 x CSI	1 x CSI	kein Anschluss vorhanden	1 x CSI
Audio	Stereo Klinke	Stereo Klinke	Stereo Klinke	kein analoger Anschluss vorhanden
I²C	1	1	2	2
I²S	0	0	0	1
UART	1	1	1	1
SPI	1	1	1	2

8.4 Technische Merkmale des NAO-Roboters

Die folgende Tabelle 28 fasst alle technischen Merkmale des NAO Roboters zusammen. Dabei werden die Version NAO V4, welche in der Telematik verwendet wird, und die neue Version NAO V6 betrachtet, die einen performanteren Prozessor und höher auflösende Kameras besitzt.

Tabelle 28: Technischen Merkmale des Roboters NAO V4 und des NAO V6 [8, 67 und jeweils die untergeordneten Links]

Merkmale	NAO V4	NAO V6
CPU Typ	Intel Atom Z530	Intel Atom E3845
CPU Kerne	1	4
CPU Frequenz	1,6 GHz	1,91 GHz
integrierte GPU	nicht vorhanden	vorhanden
RAM Größe	1 GByte	4 GByte
Speicher	2 GByte Flash 8 GByte MicroSD	keine Information
Konnektivität	Ethernet WiFi 802.11 b/g/n 1 x USB	
Batterie	ca. 49 Wh	ca. 63 Wh
Lautsprecher	2 Lautsprecher im Kopf	
Geräusch- wahrnehmung	4 Mikrofone im Kopf	
2 Kameras	1288x968 @ 30 FPS	640 x 480 @ 30 FPS 2560 x 1920 @ 1 FPS
Visuelle Effekte	LEDs im Kopf, in den Augen und in den Ohren	
Interaktions- möglichkeiten	3 kapazitive Sensoren auf dem Kopf 1 taktiler Knopf im Brustbereich je Hand 3 kapazitive Sensoren je Fuß zwei taktile Knöpfe	
Sensorik	Drucksensoren in den Füßen Gyroskop (Kreiselsensor) und Beschleunigungssensor Sonar im Brustbereich für die Distanzmessung naher Objekte	
Bewegungs- möglichkeiten	25 Freiheitsgrade Positionsrückmeldung aller Motoren	

8.5 Energieversorgung des Tischroboters mit Netzteil

Die folgende Abbildung 55 zeigt den kompletten Schaltplan der Energieversorgung des Systems mit einem Netzteil. Ein Akkubetrieb ist in dieser Schaltung nicht eingeplant.

Abbildung 55: Schaltplan der Energieversorgung mit Netzteil

Abkürzungsverzeichnis

Im Folgenden werden die in dieser Arbeit verwendeten Abkürzungen aufgeführt. Zu den meisten Abkürzungen, welche nicht weiterführend im Glossar erläutert werden, ist im Anschluss eine kurze Erklärung zu finden.

AAL	Ambient Assisted Living (dt.: Altersgerechte Assistenzsysteme) Teilgebiet des SmartHomes bzw. der Gebäudeautomation; steht für den Einsatz von Konzepten, Produkten und Dienstleistungen, um Alltagstätigkeiten zu erleichtern.
CPU	Central Processing Unit (dt.: zentrale Recheneinheit) Bezeichnung des Prozessors eines Computers.
CSI	Camera Serial Interface (dt.: serielle Kameraschnittstelle)
CSS	Cascading Style Sheets (dt.: gestufte Gestaltungsbögen) Sprache der Webprogrammierung zur Erstellung von Gestaltungsanweisungen.
eMMC	embedded Multi Media Card Energie- und platzsparendes Speichermedium.
GPIO	General Purpose Input/Output Kontakte an Schaltkreisen, die für digitale Ein- oder Ausgabe verwendet werden können und dessen Verhalten durch Programmierung frei bestimmbar ist.
GPU	Graphics Processing Unit Bezeichnung des Grafikprozessors eines Computers.
HDMI	High Definition Multimedia Interface Schnittstelle für digitale Bild- und Tonübertragung.
HTML	Hypertext Markup Language Sprache der Webprogrammierung zur Strukturierung elektronischer Dokumente.
I²C	Inter-Integrated Circuit (gesprochen I-Quadrat-C) Serieller Datenbus mit zwei Leitungen. Siehe Glossar für weitere Informationen.
I²S	Inter-IC Sound (gesprochen I-Quadrat-S) Serielle Schnittstelle zur Übertragung von digitalen Audiodaten.
JSON	JavaScript Object Notation Kompaktes Speicherformat in lesbarer Textform zum Austausch von Daten zwischen Anwendungen.

LED	Light Emitting Diode (dt.: Leuchtdiode) Eine Diode, die bei Stromfluss Licht abgibt.
MIPS	Million Instructions per Second Maßeinheit für die Rechenleistung eines Prozessors.
MOSFET	Metall-Oxid-Halbleiter-Feldeffekttransistor Siehe Glossar für weitere Informationen.
PWM	Pulsweitenmodulation Siehe Glossar für weitere Informationen.
RAM	Random-Access Memory (dt.: Direktzugriffsspeicher) Bezeichnung des Arbeitsspeichers eines Computers.
RGB	rot-grün-blau Bezeichnung für einen Farbraum RGB-LED → eine LED, welche alle Farben aus Rot-, Grün- und Blau-Anteilen zusammensetzen kann.
ROS	Robot Operating System Software-Framework für die Entwicklung von Robotern.
SMD	Surface-Mounted Device (dt.: oberflächenmontiertes Bauelement) Siehe Glossar für weitere Informationen.
SPI	Serial Peripheral Interface Serieller Datenbus mit vier Leitungen. Siehe Glossar für weitere Informationen.
SSH	Secure Shell Netzwerkprotokoll für die Herstellung einer verschlüsselten Netzwerkverbindung zu einem entfernten Gerät.
TCP	Transmission Control Protocol (dt.: Übertragungssteuerungsprotokoll) Netzwerkprotokoll, welches die Datenübertragung zwischen Netzwerkkomponenten definiert. Dabei ist eine bidirektionale Datenübertragung möglich, bei welcher Datenverluste erkannt und automatisch behoben werden.
TH Wildau	Technische Hochschule Wildau (engl.: Technical University of Applied Science Wildau)
UART	Universal Asynchronous Receiver Transmitter Eine serielle Schnittstelle zur asynchronen Datenübertragung.

Glossar

Die Erklärungen der im folgenden aufgeführten Begriffe stammen größtenteils von der Online-Enzyklopädie Wikipedia, welche gute Einführungen in bestimmte Themen und Begrifflichkeiten bietet.

Altersquotient	Anzahl von Menschen im Rentenalter pro 100 Menschen im erwerbstätigen Alter. Ein Altersquotient von 40 in einem bestimmten Jahr bedeutet dementsprechend, dass 40 Menschen im Rentenalter 100 Menschen im erwerbstätigen Alter gegenüberstehen.
Back-End	<p>Das Back-End bezeichnet die verarbeitenden Prozesse, welche beispielsweise nach der Eingabe von Daten im Hintergrund ablaufen, um ein Ergebnis zu berechnen.</p> <p>Gegenstück zum Back-End ist das Front-End, welches die Schnittstelle zum Nutzer bildet und Interaktionen entgegennimmt bzw. dem Nutzer Ergebnisse anzeigt.</p> <p>Back-End und Front-End sind über entsprechende Schnittstellen miteinander verknüpft.</p> <p>Siehe: https://de.wikipedia.org/wiki/Front-End_und_Back-End</p>
Beschleunigungs-sensor	<p>Ein Sensor, der seine Beschleunigung misst. In statischen Systemen kann mit einem Beschleunigungssensor eine Neigungsmessung vorgenommen werden. Dabei wird der Einfluss der Erdbeschleunigung auf die verschiedenen Achsen des Beschleunigungssensors erfasst.</p> <p>Siehe: https://de.wikipedia.org/wiki/Beschleunigungssensor</p>
CUDA	<p>(Compute Unified Device Architecture)</p> <p>CUDA® ist eine von Nvidia entwickelte Programmier-Technik, mit der Programmteile durch den Grafikprozessor (GPU) abgearbeitet werden können. Parallelisierbare Programmteile können dadurch signifikant schneller ausgeführt werden als bei einer reinen Verwendung der CPU.</p> <p>Siehe: https://de.wikipedia.org/wiki/CUDA</p> <p>Offizielle Seite: https://developer.nvidia.com/cuda-zone</p>
Digital Native	<p>Als digital native (dt.: „digitaler Eingeborener“) wird eine Person der gesellschaftlichen Generation bezeichnet, die in der digitalen Welt aufgewachsen ist.</p> <p>Siehe: https://de.wikipedia.org/wiki/Digital_Native</p>
Diode	<p>Eine Diode ist ein elektronisches Bauelement, das Strom in einer Richtung passieren lässt und in der anderen Richtung den Stromfluss sperrt. Dioden werden unter anderem zur Gleichrichtung, der Umwandlung von Wechselspannung zu Gleichspannung, eingesetzt.</p> <p>Dioden besitzen eine notwendige Mindestspannung, ab der diese den Strom leiten. Diese sogenannte Schwellspannung ist als Spannungsabfall nach der Diode zu beobachten.</p> <p>Siehe: https://de.wikipedia.org/wiki/Diode</p>

Drehmoment	<p>Das Drehmoment beschreibt die Drehwirkung einer Kraft auf einen Körper. Wirkt eine Kraft rechtwinklig auf einen Hebelarm, so ergibt sich der Betrag des Drehmoments, indem der Betrag der Kraft mit der Länge des Hebelarms multipliziert wird.</p> <p>Siehe: https://de.wikipedia.org/wiki/Drehmoment</p>
Eingebettete Systeme	<p>Ein eingebettetes ist ein elektronischer Rechner oder auch Computer, der in einen technischen Kontext eingebunden (eingebettet) ist. Dabei übernimmt das eingebettete System eine oder mehrere spezielle Aufgaben, für die eine optimierte Hardware-Software-Implementierung gewählt wird.</p> <p>Siehe: https://de.wikipedia.org/wiki/Eingebettetes_System</p>
Front-End	<p>Das Front-End bezeichnet Prozesse, welche die Schnittstelle zum Nutzer bilden. Es verarbeitet Interaktionen und zeigt gegebenenfalls das gewünschte Resultat an.</p> <p>Gegenstück zum Front-End ist das Back-End, welches Daten verarbeitet und Ergebnisse berechnet.</p> <p>Front-End und Back-End sind über entsprechende Schnittstellen miteinander verknüpft.</p> <p>Siehe: https://de.wikipedia.org/wiki/Front-End_und_Back-End</p>
HelloWorld Programm	<p>Ein HelloWorld Programm (dt.: Hallo-Welt-Programm) ist ein kleines Computerprogramm, das auf möglichst einfache Weise zeigen soll, wie bestimmte Aspekte eines Systems mit einer Programmiersprache verwendet werden.</p> <p>Ursprünglich werden mit HelloWorld Programmen Programmiersprachen vorgestellt, um einen ersten Einblick in Syntax zu geben.</p> <p>Siehe: https://de.wikipedia.org/wiki/Hallo-Welt-Programm</p>
I²C	<p>Abkürzung für Inter-Integrated Circuit (gesprochen I-Quadrat-C).</p> <p>Bezeichnet einen seriellen Datenbus mit zwei Leitungen (eine Taktleitung und eine Datenleitung). Der Bus ist für den Austausch von Daten über kurze Entfernungen innerhalb eines technischen Systems gedachte (bspw. zwischen einem Microcontroller und einem digitalen Sensor)</p> <p>Siehe: <a href="https://de.wikipedia.org/wiki/I²C">https://de.wikipedia.org/wiki/I²C</p>
Interrupt	<p>In der Informatik versteht man unter einem Interrupt (dt.: Unterbrechung) eine vorübergehende Unterbrechung eines laufenden Programms, um einen anderen, in der Regel kurzen, aber zeitlich kritischen, Vorgang abzuarbeiten.</p> <p>Interrupts können dazu genutzt werden, um externe Ereignisse (Events) zu erfassen und zu verarbeiten.</p> <p>Siehe: https://de.wikipedia.org/wiki/Interrupt</p>
JavaFX	<p>JavaFX ist ein Framework zur Erstellung plattformübergreifender Java-Applikationen mit grafischen Oberflächen.</p> <p>Siehe: https://de.wikipedia.org/wiki/JavaFX</p>

Key-Value-Datenbank	<p>Eine Key-Value-Datenbank (dt.: Schlüssel-Werte-Datenbank) ist eine Art von NoSQL-Datenbank, welche Key-Value-Paare speichert.</p> <p>Werte (Value) werden in Key-Value-Datenbanken über einen Schlüssel (Key) eindeutig identifiziert. Dabei können die Werte aus Tupeln, Relationen oder Dokumenten bestehen.</p> <p>Redis ist ein Vertreter der Key-Value-Datenbanken.</p> <p>Siehe: https://de.wikipedia.org/wiki/Schlüssel-Werte-Datenbank</p>
Kreiselsensor	<p>Ein Kreisel sensor (auch genannt Gyrosensor) ist ein Sensor, welcher Drehbewegungen erfasst.</p>
Mikrofonarray	<p>Ein Mikrofonarray ist ein Verbund mehrerer Mikrofone. Damit wird beispielsweise eine Richtungserkennung von Geräuschquellen ermöglicht.</p> <p>Siehe: https://de.wikipedia.org/wiki/Beamforming</p>
MOSFET	<p>Abkürzung für Metall-Oxid-Halbleiter-Feldeffekttransistor</p> <p>MOSFETs sind elektronische Bauteile die zum Schalten von Strömen und Spannungen genutzt werden können.</p> <p>Es gibt vier verschiedene Arten von MOSFETs (NPN & PNP, jeweils normal leitend oder normal sperrend).</p> <p>Siehe: https://de.wikipedia.org/wiki/Metall-Oxid-Halbleiter-Feldeffekttransistor</p>
Node.js	<p>Node.js ist eine serverseitige Plattform in der Softwareentwicklung zum Betrieb von Netzwerkanwendungen. Insbesondere lassen sich Webserver damit realisieren.</p> <p>Node.js wird in einer JavaScript-Laufzeitumgebung ausgeführt.</p> <p>Siehe: https://de.wikipedia.org/wiki/Node.js</p>
Omnidirektionaler Antrieb	<p>Bezeichnung für ein Antriebssystem, welches über eine spezielle Radkonstruktion die Bewegung in eine beliebige Richtung ermöglicht.</p> <p>Siehe: https://de.wikipedia.org/wiki/Omnidirektionaler_Antrieb</p>
Peer-to-Peer	<p>Peer-to-Peer bezeichnet im Allgemeinen Rechner-Rechner-Verbindung. In einem Peer-to-Peer-Netzwerk sind Kommunikationspartner gleichberechtigt bzw. im direkten Datenaustausch miteinander ohne die Notwendigkeit eines steuernden Gateways.</p> <p>Ein Peer-to-Peer Netzwerk kann nicht nur die Verbindung zwischen Rechnern, sondern auch die Verbindung verschiedener Kommunikationsknoten innerhalb eines Systems bezeichnen.</p> <p>Siehe: https://de.wikipedia.org/wiki/Peer-to-Peer</p>
Pulsweitenmodulation bzw. PWM-Signale	<p>PWM (Pulsweitenmodulation) ist eine Modulationsart, bei der die Spannung zwischen zwei Werten wechselt (meist 0 und Maximalwert). PWM wird zur Informationsübertragung und zur Steuerung der Energieumwandlung in einem technischen System eingesetzt.</p> <p>Bei einem PWM-Signal ist der Tastgrad eine relevante Größe, welche das Verhältnis zwischen den beiden möglichen Werten der Modulation beschreibt.</p> <p>Siehe: https://de.wikipedia.org/wiki/Pulsdauermodulation</p>

SMD-Bauteile	<p>SMD = Surface-Mounted Device (dt.: oberflächenmontiertes Bauelement)</p> <p>SMD-Bauteile sind kleine, elektronische Bauteile, die auf der Oberfläche einer Leiterplatte montiert werden.</p> <p>Siehe: https://de.wikipedia.org/wiki/Surface-mounted_device</p>
Spannungsregler	<p>Spannungsregler sind elektronische Bauteile bzw. eine Kombination von Bauteilen, um die Spannung von Gleichstrom umzuwandeln.</p> <p>Step-Up-Spannungsregler (dt. Aufwärtsregler) wandeln eine niedrige Eingangsspannung in eine hohe Ausgangsspannung um.</p> <p>Siehe: https://de.wikipedia.org/wiki/Aufwärtswandler</p> <p>Step-Down-Spannungsregler (dt. Abwärtswandler) wandeln eine hohe Eingangsspannung in eine niedrige Ausgangsspannung um.</p> <p>Siehe: https://de.wikipedia.org/wiki/Abwärtswandler</p>
SPI	<p>Abkürzung für Serial Peripheral Interface.</p> <p>Ein synchroner, serieller Datenbus, mit dem digitale Schaltungen nach dem Master-Slave-Prinzip miteinander verbunden werden. SPI benötigt vier Leitungen.</p> <p>Siehe: https://de.wikipedia.org/wiki/Serial_Peripheral_Interface</p>
Telematik	<p>Zusammengesetztes Wort aus Telekommunikation und Informatik. Sowohl Bachelor- als auch Master-Studiengang an der TH Wildau.</p>
Tiobe-Index	<p>Der TIOBE Programming Community Index ist ein seit 2001 publiziertes und monatlich aktualisiertes Ranking von Programmiersprachen nach ihrer Popularität.</p> <p>siehe: https://de.wikipedia.org/wiki/TIOBE-Index für weitere Informationen</p> <p>siehe: https://www.tiobe.com/tiobe-index/ für das aktuelle Ranking</p>
Transistor	<p>Ein Transistor ist ein elektronisches Halbleiter-Bauelement zum Steuern meistens niedriger elektrischer Spannungen und Ströme.</p> <p>Siehe: https://de.wikipedia.org/wiki/Transistor</p>

Quellenverzeichnis

- [1] S. Ghosh, *Google Glass: the scientists behind Google's augmented reality glasses*. [Online] Verfügbar unter: <https://www.telegraph.co.uk/technology/google/9918794/Google-Glass-the-scientists-behind-Google-augmented-reality-glasses.html>. Zugriff am: 20. August 2019.
- [2] Destatis, *Demografischer Wandel*. [Online] Verfügbar unter: https://www.destatis.de/DE/Themen/Querschnitt/Demografischer-Wandel/_inhalt.html. Zugriff am: 12. Mai 2019.
- [3] Destatis, *13. koordinierte Bevölkerungsvorausberechnung für Deutschland*. [Online] Verfügbar unter: <https://service.destatis.de/bevoelkerungspyramide/>. Zugriff am: 12. Mai 2019.
- [4] S. Etgeton, *Perspektive Pflege: Finanzentwicklung der Sozialen Pflegeversicherung*. [Online] Verfügbar unter: <https://www.bertelsmann-stiftung.de/de/unsere-projekte/zukunft-pflege/projektnachrichten/pflegekostenprognose/>. Zugriff am: 13. Mai 2019.
- [5] C. Weiß, J. Stubbe, C. Naujoks und S. Weide, *Digitalisierung für mehr Optionen und Teilhabe im Alter*. [Online] Verfügbar unter: <https://www.bertelsmann-stiftung.de/de/publikationen/publikation/did/digitalisierung-fuer-mehr-optionen-und-teilhabe-im-alter/>. Zugriff am: 13. Mai 2019.
- [6] nicaï systems, *Roboterbausatz NIBO 2*. [Online] Verfügbar unter: <http://www.nicaï-systems.com/de/nibo2>. Zugriff am: 4. Mai 2019.
- [7] SoftBank Robotics, *NAO*. [Online] Verfügbar unter: <https://www.softbankrobotics.com/emea/en/nao>. Zugriff am: 5. Mai 2019.
- [8] SoftBank Robotics, *NAO V4 & V5 - Technical overview*. [Online] Verfügbar unter: http://doc.aldebaran.com/2-4/family/robots/index_robots.html. Zugriff am: 18. August 2019.
- [9] Wikipedia, *NAO Robot*. [Online] Verfügbar unter: https://upload.wikimedia.org/wikipedia/commons/7/79/NAO_Robot_.jpg. Zugriff am: 5. Mai 2019.
- [10] SoftBank Robotics, *What is Choregraphe*. [Online] Verfügbar unter: http://doc.aldebaran.com/2-1/software/choregraphe/choregraphe_overview.html. Zugriff am: 18. August 2019.
- [11] SoftBank Robotics, *NAO Programming Guide*. [Online] Verfügbar unter: <http://doc.aldebaran.com/1-14/dev/index.html>. Zugriff am: 15. August 2019.
- [12] SoftBank Robotics, *Pepper*. [Online] Verfügbar unter: <https://www.softbankrobotics.com/emea/en/pepper>. Zugriff am: 6. Mai 2019.

- [13] C. Drees, *SoftBank "Pepper": 1000 Roboter in Japan in einer Minute ausverkauft*. [Online] Verfügbar unter: <https://www.mobilegeeks.de/news/softbank-pepper-1000-roboter-in-japan-in-einer-minute-ausverkauft/>. Zugriff am: 11. Mai 2019.
- [14] iCampus Wildau, *Einsatzszenarien von humanoiden Robotern*. [Online] Verfügbar unter: <https://icampus.th-wildau.de/icampus/home/de/einsatzszenarien-von-humanoiden-robotern>. Zugriff am: 16. August 2019.
- [15] iCampus Wildau, *Humanoide Roboter in der Bibliothek*. [Online] Verfügbar unter: https://icampus.th-wildau.de/icampus/home/icampusdata/pdf/Humanoide-Roboter-in-der-Bibliothek_RoboticLab.pdf. Zugriff am: 16. August 2019.
- [16] Technische Hochschule Wildau, *HUMANOIDE ROBOTER IM EINSATZ*. [Online] Verfügbar unter: <https://icampus.th-wildau.de/bewerbung-bdj-2017/robotic.html>. Zugriff am: 1. Mai 2019.
- [17] A. Klingner, „Konzept und Umsetzung einer webbasierten Anwendung zur Nutzerverwaltung im Rahmen eines Projektes zur Leseförderung mit einem humanoiden Roboter“. Bachelorarbeit, Technische Hochschule Wildau, 2018.
- [18] T. Lütke, „Konzeption und Umsetzung einer Tablet-Anwendung für den Einsatz zur Leseförderung von Grundschulkindern mit humanoiden Robotern“. Bachelorarbeit, Technische Hochschule Wildau, 2018.
- [19] J. Breßler, „Konzeption und Entwicklung von Lernbausteinen im Themengebiet der Künstlichen Intelligenz mit dem Fokus auf die Anwendungsfelder Servicerobotik und Smart Home“. Masterarbeit, Technische Hochschule Wildau.
- [20] P. Müller, „Mensch-Maschine-Kommunikation basierend auf Natural Language Processing: Evaluierung von Möglichkeiten für den Einsatz eines humanoiden Roboters im Kontext einer Hochschulbibliothek“. Bachelorarbeit, Technische Hochschule Wildau, 2018.
- [21] Raspberry Pi Foundation, *Raspberry Pi 3 Model B+*. [Online] Verfügbar unter: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>. Zugriff am: 17. Juni 2019.
- [22] Raspberry Pi Foundation, *Raspberry Annual Review 2017*. [Online] Verfügbar unter: <https://www.raspberrypi.org/blog/annual-review-2017/>. Zugriff am: 27. Juni 2019.
- [23] Raspberry Pi Foundation, *Raspberry Pi Downloads: Images for Raspberry Pi*. [Online] Verfügbar unter: <https://www.raspberrypi.org/downloads/>. Zugriff am: 26. Juni 2019.
- [24] Raspberry Pi Foundation, *Raspberry Pi Help*. [Online] Verfügbar unter: <https://www.raspberrypi.org/help/>. Zugriff am: 29. Juni 2019.
- [25] Sinovoip und BPI, *Banana Pi Open Source Hardware Products*. [Online] Verfügbar unter: <http://www.banana-pi.org/product.html>. Zugriff am: 28. Juni 2019.
- [26] Sinovoip und BPI, *Banana Pi M64: Official Product Page*. [Online] Verfügbar unter: <http://www.banana-pi.org/m64.html>. Zugriff am: 28. Juni 2019.

- [27] Sinovoip und BPI, *Banana Pi BPI-M64 Wiki*. [Online] Verfügbar unter: http://wiki.banana-pi.org/Banana_Pi_BPI-M64. Zugriff am: 29. Juni 2019.
- [28] Sinovoip und BPI, *Banana Pi Forum for BPI-M64*. [Online] Verfügbar unter: <http://forum.banana-pi.org/c/BPI-M64>. Zugriff am: 29. Juni 2019.
- [29] Banana Pi Forum, *Banana Pi Forum: Inoffizielles deutsches Forum zum Banana Pi*. [Online] Verfügbar unter: <https://forum-bpi.de/>. Zugriff am: 29. Juni 2019.
- [30] Hardkernel, *Odroid N2*. [Online] Verfügbar unter: <https://www.hardkernel.com/shop/odroid-n2-with-4gbyte-ram/>. Zugriff am: 29. Juni 2019.
- [31] Hardkernel, *Odroid N2 Wiki*. [Online] Verfügbar unter: <https://wiki.odroid.com/odroid-n2/>. Zugriff am: 29. Juni 2019.
- [32] Nvidia, *Jetson Nano: Offizielle Produktseite*. [Online] Verfügbar unter: <https://www.nvidia.com/de-de/autonomous-machines/embedded-systems/jetson-nano/>. Zugriff am: 25.06.19.
- [33] Hardkernel, *Odroid N2 Forum*. [Online] Verfügbar unter: <https://forum.odroid.com/viewforum.php?f=175>. Zugriff am: 29. Juni 2019.
- [34] eLinux, *Jetson Nano eLinux Wiki*. [Online] Verfügbar unter: https://elinux.org/Jetson_Nano. Zugriff am: 26. Juni 2019.
- [35] Nvidia, *Jetson Nano Developer Kit: Offizielle Entwicklerseite*. [Online] Verfügbar unter: <https://developer.nvidia.com/embedded/jetson-nano-developer-kit>. Zugriff am: 26. Juni 2019.
- [36] Nvidia, *Jetson Projects*. [Online] Verfügbar unter: <https://developer.nvidia.com/embedded/community/jetson-projects>. Zugriff am: 26. Juni 2019.
- [37] Pollin, *LC-Display 7" (17,8 cm), mit kapazitivem Touchscreen*. [Online] Verfügbar unter: <https://www.pollin.de/p/lc-display-7-17-8-cm-mit-kapazitivem-touchscreen-hdmi-810842>. Zugriff am: 6. Juli 2019.
- [38] ReSpeaker, *ReSpeaker USB 4 Mic Array*. [Online] Verfügbar unter: https://respeaker.io/usb_4_mic_array/. Zugriff am: 6. Juli 2019.
- [39] Burosch, *Audio Verstärker: Wieviel Leistung ist sinnvoll?* [Online] Verfügbar unter: <https://www.burosch.de/audio-technik-blog/260-audio-verstaerker-wieviel-leistung-ist-sinnvoll.html>. Zugriff am: 18. Juli 2019.
- [40] Adafruit, *Stereo 20W Class D Audio Amplifier*. [Online] Verfügbar unter: <https://www.adafruit.com/product/1752>. Zugriff am: 9. Juli 2019.
- [41] P. Moran, *CSI Cameras on the TX2*. [Online] Verfügbar unter: <http://petermoran.org/csi-cameras-on-tx2/#why-csi-cameras-vs-usb>. Zugriff am: 6. Juli 2019.
- [42] Raspberry Pi Foundation, *Raspberry Camera Module V2*. [Online] Verfügbar unter: <https://www.raspberrypi.org/products/camera-module-v2/>. Zugriff am: 7. Juli 2019.

- [43] Reeti, *Reeti: An expressive and communicating robot!* [Online] Verfügbar unter: <http://www.reeti.fr/index.php/en/>. Zugriff am: 9. Juli 2019.
- [44] Anki, *Cozmo: Big brain. Bigger personality.* [Online] Verfügbar unter: <https://www.anki.com/en-us/cozmo.html>. Zugriff am: 9. Juli 2019.
- [45] Adafruit, *Monochrome 1.3" 128x64 OLED graphic display.* [Online] Verfügbar unter: <https://www.adafruit.com/product/938>. Zugriff am: 9. Juli 2019.
- [46] Arduino, *Arduino Nano.* [Online] Verfügbar unter: <https://store.arduino.cc/arduino-nano>. Zugriff am: 11. Juli 2019.
- [47] Ubuntu, *Ubuntu Desktop Downloads: Ubuntu 18.04.2 LTS.* [Online] Verfügbar unter: <https://ubuntu.com/download/desktop>. Zugriff am: 9. Juli 2019.
- [48] Ubuntu, *The story of Ubuntu.* [Online] Verfügbar unter: <https://ubuntu.com/about>. Zugriff am: 9. Juli 2019.
- [49] ROS.org, *ROS Documentation.* [Online] Verfügbar unter: <https://wiki.ros.org/>. Zugriff am: 9. Juli 2019.
- [50] GitHub, *Electron JS.* [Online] Verfügbar unter: <https://electronjs.org/>. Zugriff am: 13. Juli 2019.
- [51] kangalow, *Jetson Nano + Raspberry Pi Camera.* [Online] Verfügbar unter: <https://www.jetsonhacks.com/2019/04/02/jetson-nano-raspberry-pi-camera/>. Zugriff am: 7. Juli 2019.
- [52] P. Badger, *Capacitive Sensing Library.* [Online] Verfügbar unter: <https://playground.arduino.cc/Main/CapacitiveSensor/>. Zugriff am: 11. Juli 2019.
- [53] Microchip, *CAP1298.* [Online] Verfügbar unter: <https://www.microchip.com/wwwproducts/en/CAP1298>. Zugriff am: 11. Juli 2019.
- [54] TDK InvenSense, *MPU-6050: Six-Axis MEMS MotionTracking™ Devices.* [Online] Verfügbar unter: <https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/>. Zugriff am: 13. Juli 2019.
- [55] Texas Instruments, *LM2596: SIMPLE SWITCHER® 4.5V to 40V.* [Online] Verfügbar unter: <http://www.ti.com/product/LM2596>. Zugriff am: 21. Juli 2019.
- [56] Texas Instruments, *TXS0108E: 8-Bit Bidirectional Voltage-Level Shifter.* [Online] Verfügbar unter: <http://www.ti.com/product/TXS0108E>. Zugriff am: 21. Juli 2019.
- [57] Diotec Semiconductor, *SBX2030 Datasheet: Schottky-Gleichrichterdiode.* [Online] Verfügbar unter: https://diotec.com/tl_files/diotec/files/pdf/datasheets/sbx2020.pdf. Zugriff am: 22. Juli 2019.
- [58] Texas Instruments, *LM5050-1/-Q1: High-Side OR-ing FET Controller.* [Online] Verfügbar unter: <http://www.ti.com/lit/ds/symlink/lm5050-1.pdf>. Zugriff am: 25. Juli 2019.
- [59] ROS.org, *ROS Concepts.* [Online] Verfügbar unter: <http://wiki.ros.org/ROS/Concepts>. Zugriff am: 4. August 2019.

- [60] ROS.org, *ROS Master*. [Online] Verfügbar unter: <http://wiki.ros.org/Master>. Zugriff am: 7. August 2019.
- [61] GitHub, *Electron Application Architecture*. [Online] Verfügbar unter: <https://electronjs.org/docs/tutorial/application-architecture>. Zugriff am: 7. August 2019.
- [62] redislabs, *Redis Pub/Sub*. [Online] Verfügbar unter: <https://redis.io/topics/pubsub>. Zugriff am: 7. August 2019.
- [63] Nvidia, *Enabling spidev on the jetson nano is hanging when flashing*. Verfügbar unter: <https://devtalk.nvidia.com/default/topic/1050427/jetson-nano/enabling-spidev-on-the-jetson-nano-is-hanging-when-flashing/1>. Zugriff am: 12. August 2019.
- [64] J. Prof. Plate, *Raspberry Pi: I2C-Konfiguration und -Programmierung*. [Online] Verfügbar unter: http://www.netzmafia.de/skripten/hardware/RasPi/RasPi_I2C.html. Zugriff am: 12. August 2019.
- [65] kangalow, *Jetson Nano – Using I2C*. [Online] Verfügbar unter: <https://www.jetsonhacks.com/2019/07/22/jetson-nano-using-i2c/>. Zugriff am: 12. August 2019.
- [66] H. Behling, „Wenn der Raspberry nicht mehr reicht“, *Make*, Nr. 03, S. 32–45, <https://www.heise.de/make/meldung/Wenn-der-Raspberry-nicht-mehr-reicht-4449940.html>, 2019.
- [67] SoftBank Robotics, *NAO V6 - Technical overview*. [Online] Verfügbar unter: http://doc.aldebaran.com/2-8/family/nao_technical/index_dev_naov6.html. Zugriff am: 19. August 2019.

Abbildungsverzeichnis

Abbildung 1: Übersicht eines NIBO-Roboters [6]	11
Abbildung 2: NAO Roboter [9]	12
Abbildung 3: Pepper Roboter [12]	15
Abbildung 4: Persona- Karin Schröter	20
Abbildung 5: Persona - Mario Freud	23
Abbildung 6: Vorbetrachtung der im System benötigten Module.....	37
Abbildung 7: Raspberry Pi 3 Model B+ [21]	41
Abbildung 8: Banana Pi BPI-M64 [26]	42
Abbildung 9: Odroid N2 [30].....	43
Abbildung 10: Nvidia Jetson Nano Entwicklerkit [32].....	44
Abbildung 11: 7 Zoll Touch-Display [37].....	47
Abbildung 12: ReSpeaker Mikrofonarray [38]	48
Abbildung 13: Stereo 20W Class D Audio Verstärker Board von Adafruit [40]	51
Abbildung 14: Raspberry Pi Camera Module V2 [42]	51
Abbildung 15: Verschiedene Umsetzungen von Roboter Augen – Pepper [12], Reeti [43] und Cozmo [44]	55
Abbildung 16: 1,3 Zoll OLED-Display (128 x 64 Pixel) [45]	55
Abbildung 17: Arduino Nano Board [46]	57
Abbildung 18: Systemarchitektur mit den in der Evaluation ausgewählten Komponenten...	61
Abbildung 19: Konnektivität und Anschlüsse des Systems	63
Abbildung 20: Zustandsübergangsdiagramm für den Einschalter des Systems.....	64
Abbildung 21: Prinzip eines kapazitiven Sensors mit einem Arduino	65
Abbildung 22: Schematischer Aufbau des Einschalters	66
Abbildung 23: Schematischer Aufbau der Temperaturregulierung.....	67
Abbildung 24: Leistungsübergänge der Temperaturregulierung	67
Abbildung 25: Luftstrom im Gehäuse bei zwei Lüftern.....	68
Abbildung 26: Steuerung der Kopfbewegungen.....	69
Abbildung 27: Grafische Darstellungsmöglichkeiten der Augen über ein Display – Anki Cozmo Roboter [44]	70
Abbildung 28: Kommunikation zwischen zentraler Recheneinheit und einem Display	71
Abbildung 29: Kommunikation zwischen zentraler Recheneinheit und zwei Displays.....	71

Abbildung 30: Prinzip des Lautsprechersystems	72
Abbildung 31: Schematischer Aufbau des Lautsprechersystems.....	72
Abbildung 32: Stereo-Lautsprechersystem mit Breitbandlautsprechern.....	73
Abbildung 33: Step-Down Spannungsregulierung mit LM2596	76
Abbildung 34: Prinzip zum Abschalten von Komponenten.....	77
Abbildung 35: Level-Shifting mit einem MOSFET	78
Abbildung 36: Trennung von Netzteil und Akku mit Dioden.....	79
Abbildung 37: Aufladen des Akkus	80
Abbildung 38: Gesamtübersicht der Systemarchitektur	81
Abbildung 39: Aspekte der Systemsoftware.....	82
Abbildung 40: Publishing und Subscribing von Nachrichten in ROS.....	83
Abbildung 41: Konzeptionelle Übersicht möglicher ROS-Packages, -Topics und -Services im System	84
Abbildung 42: Mockup als Vorschlag für die grafischen Anwendung.....	86
Abbildung 43: Prinzipielle Architektur einer Electron App	87
Abbildung 44: Gesamtübersicht der Softwarearchitektur	94
Abbildung 45: Prototyp der Energieversorgung mit Netzteil.....	96
Abbildung 46: Prototypischer Aufbau des Level-Shiftings mit MOSFETs.....	97
Abbildung 47: Grafische Oberfläche für die Steuerung angeschlossener Motoren.....	99
Abbildung 48: Architektur der Motorsteuerung	100
Abbildung 49: Foto der praktisch umgesetzten Motorsteuerung	100
Abbildung 50: Verknüpfung von Audioverstärker, FR 58 Breitbandlautsprecher und ReSpeaker-Modul (v.l.n.r)	101
Abbildung 51: Umsetzung eines kapazitiven Sensors mit einem Arduino.....	102
Abbildung 52: Umsetzung eines kapazitiven Sensors mit einem CAP1298 Schaltkreis.....	103
Abbildung 53: Erfüllung der gestellten Anforderungen	105
Abbildung 54: Prototypische Verknüpfung der Systemkomponenten.....	106
Abbildung 55: Schaltplan der Energieversorgung mit Netzteil.....	119

Tabellenverzeichnis

Tabelle 1: Übersicht der Anwendungsfälle für die Unterstützung im Alltag	25
Tabelle 2: Übersicht der Anwendungsfälle in der Gebäudeautomation	26
Tabelle 3: Übersicht der Anwendungsfälle für die Freizeitgestaltung	27
Tabelle 4: Übersicht der allgemeinen Anwendungsfälle	27
Tabelle 5: Anwendungsfälle in der Entwicklung	28
Tabelle 6: Tabellarische Übersicht der definierten Anforderungen.....	33
Tabelle 7: In Deutschland erhältliche Einplatinencomputer	39
Tabelle 8: Übersicht und Bewertung der vorgestellten Einplatinencomputer	45
Tabelle 9: Übersicht verschiedener Lautsprecher	50
Tabelle 10: Vergleich von Gehäuselüftern mit PWM-Regulierung.....	52
Tabelle 11: Übersicht über die Kennzahlen verschiedener Tischnetzteile	53
Tabelle 12: Übersicht verschiedener Servomotoren	56
Tabelle 13: Finanzieller Zwischenstand der in der Evaluation ausgewählten Komponenten	59
Tabelle 14: Energierelevante Kennzahlen der Systemkomponenten.....	74
Tabelle 15: Übersicht der verschiedenen Spannungsbereiche des Systems	76
Tabelle 16: Übersicht der Kommunikation mit ausgelagerten Hardwarekomponenten	88
Tabelle 17: HelloWorld Programme für die separate Ansteuerung von Hardwaremodulen..	90
Tabelle 18: HelloWorld Programme für die Arbeit mit ROS	91
Tabelle 19: HelloWorld Programme für die Benutzung von Hardwareschnittstellen.....	92
Tabelle 20: HelloWorld Programme für grafische Anwendungen mit Electron.....	92
Tabelle 21: HelloWorld Programme für die Arduino Nanos	93
Tabelle 22: Kostenübersicht der prototypischen Umsetzung	104
Tabelle 23: Übersicht deutscher Distributoren	114
Tabelle 24: Allgemeiner Merkmale der Einplatinencomputer	115
Tabelle 25: Leistungsspezifikationen der Einplatinencomputer	115
Tabelle 26: Benchmark Vergleich der Einplatinencomputer [66]	116
Tabelle 27: Vergleich der Konnektivität und der Anschlüsse der Einplatinencomputern	117
Tabelle 28: Technischen Merkmale des Roboters NAO V4 und des NAO V6 [8, 67 und jeweils die untergeordneten Links]	118

Selbstständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Werken wörtlich oder sinngemäß übernommenen Gedanken sind unter Angabe der Quellen gekennzeichnet.

Wildau, den 02.09.2019

.....

Valentin Schröter