

# **MASTERARBEIT**

zur Erlangung des akademischen Grades  
„Master of Science in Engineering“  
im Studiengang Embedded Systems

## **Technische Voraussetzungen für den Einsatz von Unterstützungstechnologien bei Kindern mit Autismus Am praktischen Beispiel der Roboterplattform RoboFriend**

Ausgeführt von: Karima Khlousy-Neirukh, BSc.  
Personenkennzeichen: 1410297009

1. BegutachterIn: Dipl.-Ing. Christoph Veigl
2. BegutachterIn: Ing. Martin Deinhofer, MSc.

Wien, 10.10.2017



## Eidesstattliche Erklärung

„Ich, als Autor / als Autorin und Urheber / Urheberin der vorliegenden Arbeit, bestätige mit meiner Unterschrift die Kenntnisnahme der einschlägigen urheber- und hochschulrechtlichen Bestimmungen (vgl. Urheberrechtsgesetz idgF sowie Satzungsteil Studienrechtliche Bestimmungen / Prüfungsordnung der FH Technikum Wien idgF).

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig angefertigt und Gedankengut jeglicher Art aus fremden sowie selbst verfassten Quellen zur Gänze zitiert habe. Ich bin mir bei Nachweis fehlender Eigen- und Selbstständigkeit sowie dem Nachweis eines Vorsatzes zur Erschleichung einer positiven Beurteilung dieser Arbeit der Konsequenzen bewusst, die von der Studiengangsleitung ausgesprochen werden können (vgl. Satzungsteil Studienrechtliche Bestimmungen / Prüfungsordnung der FH Technikum Wien idgF).

Weiters bestätige ich, dass ich die vorliegende Arbeit bis dato nicht veröffentlicht und weder in gleicher noch in ähnlicher Form einer anderen Prüfungsbehörde vorgelegt habe. Ich versichere, dass die abgegebene Version jener im Uploadtool entspricht.“

Wien, 10.10.17

Ort, Datum

Wolfgang Wenzel  
Kärner

Unterschrift

## Kurzfassung

Diese Arbeit befasst sich mit den technischen Möglichkeiten autistische Kinder in unterschiedlichen Bereichen zu fördern. Um im Alltag zurechtzukommen, müssen Menschen mit Autismus, bzw. Asperger-Syndrom, oft viele verschiedene Hürden überwinden. Je nach Stärke der Ausprägung dieser Entwicklungsstörung können Erwachsene ihren Alltag gar nicht alleine bewältigen. Da Autismus bereits im Kindesalter eine große Rolle spielt und möglicherweise zu Beeinträchtigungen führt, gibt es viele Forschungen, die sich damit beschäftigen, diesen Menschen zu helfen und sie zu unterstützen. Bisher gibt es keine Heilung für diese Entwicklungsstörung, weshalb zusätzliche Hilfe, sei es durch technische Möglichkeiten oder anders, von großer Wichtigkeit ist. Kinder die unter Autismus leiden, zeigen oft Schwierigkeiten beim Erkennen von Gefühlen und anderen sozialen Interaktionen. Aus diesem Grund ist der Umgang mit Maschinen, bzw. Robotern, einfacher für diese betroffene Gruppe als mit anderen Menschen. Therapieroboter sind mittlerweile keine Neuheit mehr und so kommen sie in immer mehr Bereichen zum Einsatz. Ihre Entwicklung ist durchaus ein langer Prozess und nicht immer von Erfolg gekrönt. Diese Arbeit beschreibt Voraussetzungen für eine erfolgreiche Unterstützungstechnologie für Menschen mit Autismus. Dafür werden vorhandene Forschungsprojekte und Prototypen, sowie deren Ergebnisse präsentiert und analysiert. Der zweite Teil der Arbeit befasst sich mit der Roboterplattform *RoboFriend*. Diese wurde im Rahmen der Arbeit um Funktionalitäten für autistische Kinder erweitert. Neben dem Austausch von Hardwarekomponenten, wie dem Mikrocontroller und der Webcam, wurde auch die gesamte Softwareseite entworfen. Implementiert wurden vier verschiedene Spiele, die Steuerung des Roboters, beides aufrufbar über ein Tablet, und ein Webserver der zur Telepräsenz und Robotersteuerung dient. Abschließend werden die Evaluationsergebnisse mit acht Kindern der Schule in der Paulusgasse präsentiert, die Einblick in die weiteren Verbesserungsmöglichkeiten des *RoboFriends* bieten.

**Schlagwörter:** Therapieroboter, Unterstützungstechnologie, Autismus, Neuroentwicklungsstörung, Kinder

# Abstract

This work is about the technical opportunities to support children with autism in different topics. People with autism or Asperger syndrome often have many different difficulties to get through their everyday life. Depending on the severity of this developmental disorder adults may not even be able live on their own. Since autism often already plays an important role in childhood and possibly results in deficits, there are many researches, which are focused on helping and supporting those people. There is no cure for autism yet, which is why additional help, may it be through technical possibilities or otherwise, is of great importance. A lot of children with autism are having troubles with the recognition of emotions or other social interactions. This is one of the reasons why this group of people can deal with machines, or more specific: robots, more easily than with other people. Therapy robots are not a novelty anymore and are used more and more often. Nevertheless, it takes long processes to develop them and there is no guarantee for them being a success. In this thesis, requirements that must be reached for a successful tool, which supports the given group of people, are discussed. Therefore, existing research projects and prototypes as well as their results are being presented and analysed. The second part of this work is about the robot platform *RoboFriend*. Within this project, it got updated with functionalities designed for autistic children. In addition to the upgrade of hardware components, such as microcontroller and webcam, the entire software was designed. Four different games, the control of the robot, both accessible through a tablet as well as a webserver for telepresence and robot control were implemented. To get an overview over the further possibilities of the *RoboFriend* platform, finally, the results of the evaluations with eight children from the school in Paulusgasse are presented.

**Keywords:** Therapy robots, Assistive technology, Autism, Neurodevelopment disorder, Children

# Danksagung

Allen meiner Mitmenschen, die auch nur einen kleinen Beitrag dazu geleistet haben, dass ich so weit gekommen bin, gehört mein Dank.

Mein größter Dank richtet sich an Dipl.-Ing. Christoph Veigl, der mich mit seinem Engagement und seiner Geduld während dieses Projektes begleitet hat. Zusätzlich möchte ich mich auch bei meinem Kollegen, Gregor Sindl, bedanken, der mir oft eine zusätzliche helfende Hand gereicht hat.

Bei meinem Mann, Ijad Neirukh, möchte ich mich für die Motivation, die Unterstützung und vor allem für sein Verständnis bedanken, das er mir während dieser Arbeit entgegenbrachte.

Bei meiner Familie, sowohl bei meinen geliebten Eltern, als auch bei meinen beiden Schwestern, Amira und Mona, möchte ich mich ebenfalls besonders dafür bedanken, dass sie mich in meinem gesamten Leben unterstützen und mir immer wieder zeigen und sagen, dass alles möglich ist.

Ein zusätzliches Dankeschön richtet sich auch an meine beste Freundin, Mariam Al-Rawi, die sich immer Zeit genommen hat, wenn ich sie gebraucht habe und mich mit Rat und Tat unterstützt hat.

Zum Schluss bleibt mir nur das Dankeschön an Gott. Ich danke Gott sowohl für die Möglichkeit, an diesem Projekt zu arbeiten, als auch für meinen verständnisvollen Betreuer. Danke auch dafür, die beste unterstützende Familie zu haben, die ich mir vorstellen kann und dafür, dass Du mir immer wieder zeigst, was für fantastische Freunde und Menschen ich in meinem Leben habe.

# Inhaltsverzeichnis

1	Einleitung.....	8
2	Problemstellung und Motivation.....	10
2.1	Autismus Spektrum Störung.....	10
2.2	Therapierobotik.....	13
2.2.1	Roboterrobbe PARO.....	14
2.2.2	Kaspar.....	15
2.2.3	Teo.....	18
2.2.4	Puffy.....	20
2.2.5	Softwarebasierte Therapie.....	21
3	Implementierung.....	24
3.1	Hardware.....	25
3.1.1	Hardware Requirements und Aufbau.....	25
3.1.2	Teensy++ 2.0.....	27
3.1.3	Raspberry Pi 2B.....	30
3.1.4	IO-Warrior.....	31
3.1.5	DC-DC Converter - Gleichspannungswandler.....	32
3.1.6	Sensoren.....	32
3.2	Software.....	35
3.2.1	Software Requirements und Aufbau.....	35
3.2.2	Kommunikationsprotokoll.....	37
3.2.3	Mikrocontroller Firmware.....	41
3.2.4	Robot Middleware – Python Hauptprogramm.....	43
3.2.5	Webserver / Web Frontend.....	46
3.2.6	Tablet-GUI User Applikationen.....	48
3.3	Implementierte Spielfunktionen.....	49
3.3.1	Motorik.....	50
3.3.2	Emotionen erkennen.....	50
3.3.3	Naturbezug.....	51
3.3.4	Roboter Steuerung.....	53
4	Evaluierung und Ergebnisse.....	55

4.1	Räumliche Umgebung .....	55
4.2	Datenerfassung .....	55
4.3	Ablauf der Testdurchläufe.....	55
4.4	Erster Testdurchlauf .....	56
4.4.1	Zwischenergebnis.....	56
4.5	Zweiter Testdurchlauf .....	60
4.5.1	Zwischenergebnis.....	61
4.6	Ergebnisse der Evaluierung mit Proband/innen .....	63
5	Zusammenfassung und Ausblick .....	66
	Literaturverzeichnis .....	68
	Abbildungsverzeichnis .....	73
	Tabellenverzeichnis .....	75
	Abkürzungsverzeichnis .....	76
	Anhang A – Antragsformular Stadtschulrat Wien .....	77

# 1 Einleitung

In den letzten Jahrzehnten hat sich die Robotik sehr stark weiterentwickelt und moderne Technologien bieten täglich neue Möglichkeiten für die Umsetzung von innovativen Ideen. Besonders interaktive Roboter werden beliebter und häufiger und sind nicht mehr fremd. Sowohl humanoide Roboter mit einer KI (Künstliche Intelligenz) als auch Roboter wie ASIMO (Advanced Step in Innovative Mobility) sind bekannt (Dautenhahn et al., 2009) (MIT, 2017) (American Honda Motor Co. Inc, 2017). Mit jedem technischen Fortschritt werden interaktive Roboter billiger und werden mehr in den menschlichen Alltag integriert. Dies bietet zusätzliche neue Möglichkeiten für die Entwicklung und Anwendung im Bereich von Unterstützungstechnologien.

Da Kinder mit Autismus Spektrum Störung (ASS), bzw. Asperger-Syndrom, im Alltag vielen Hürden gegenüberstehen, welche kaum bis gar nicht alleine zu bewältigen sind, sind diese auf Unterstützung angewiesen. Neben der Hilfe von anderen Menschen können und sollen technische Tools zum Einsatz kommen, um den Alltag und das Leben von Kindern mit Autismus zu erleichtern. Da menschliches Pflegepersonal teuer und meist nicht rund um die Uhr verfügbar ist, bietet der Einsatz von Robotern zusätzliche Unterstützung und Erleichterung für Pflege und Therapie und könnte in Zukunft das Personal erheblich entlasten.

Der Zustand, in dem sich Menschen mit ASS befinden, variiert sehr nach Stärke der Ausprägung der Entwicklungsstörung. Da Autismus bereits im Kindesalter oft eine große Rolle spielt und möglicherweise zu Beeinträchtigungen führt, gibt es viele Forschungen, die sich damit beschäftigen, diesen Menschen zu helfen und sie zu unterstützen. Bisher gibt es keine Heilung für ASS, weshalb zusätzliche Hilfe, sei es durch technische Möglichkeiten oder anders, von großer Wichtigkeit ist. Kinder, die unter Autismus leiden, zeigen oft Schwierigkeiten mit dem Erkennen von Gefühlen und anderen sozialen Interaktionen. Selbst zeitlose, klassische Kinderspiele, wie „Fangen“ oder „blinde Kuh“, sind für betroffene Kinder eine Herausforderung, da sie bei hierbei zu viele Reize verarbeiten müssten. Diese wären beispielsweise: plötzliche laute Geräusche, Strukturlosigkeit, Veränderung der Lichtverhältnisse oder auch das Erkennen von und das Reagieren auf soziale/n Reize/n. Aus diesem Grund ist der Umgang mit Robotern einfacher als mit anderen Menschen, da diese so programmiert werden können, dass sie sowohl Struktur als auch einen absehbaren Ablauf bieten und keine mit Menschen vergleichbaren Gefühle zeigen. Ein großes und potenziell wachsendes Anwendungsgebiet ist daher die Therapierobotik. Es ist in diesem Bereich möglich verschiedene Krankheiten oder Entwicklungsstörungen zu therapieren (Bonarini, et al., 2016) (Burgar et al., 2000), oder auch einfach nur Menschen, die unter diesen leiden, in ihrem Leben zu unterstützen und ihnen hiermit eine Hilfe zu geben. Es gibt bereits einige erfolgreiche Projekte in der Therapierobotik. Diese Arbeit beschäftigt sich mit der Analyse und den Möglichkeiten, bzw. Grenzen, der interaktiven Roboterplattform *RoboFriend*, die im Zuge dieser Arbeit erweitert (siehe Kapitel „Implementierung“) und getestet (siehe Kapitel „Evaluierung und Ergebnisse“) wurde. Des Weiteren werden



bestehende Forschungsprojekte und Beispiele für Unterstützungstechnologien für Kinder mit ASS gegenübergestellt und evaluiert. Dazu wird im nächsten Kapitel näher auf ASS und Therapierobotik eingegangen (siehe Kapitel „Problemstellung und Motivation“).

## 2 Problemstellung und Motivation

Ziel dieser Arbeit ist es, zukünftig Kindern, die unter ASS leiden, eine größere Bandbreite an Möglichkeiten zur Unterstützung anbieten zu können. Dazu muss die Entwicklungsstörung zunächst genauer untersucht werden. Erst danach kann nachvollzogen werden, was die betroffene Zielgruppe für Anforderungen an eine Unterstützungstechnologie hat. Bestehende und durchgeführte Forschungsprojekte in diesem Bereich zeigen auf, welche Methoden besseren oder vielleicht gar keinen Einfluss auf die Versuchspersonen haben und sollten daher ebenfalls im Fokus liegen. Anschließend folgt die Auseinandersetzung mit der Roboterplattform *RoboFriend*. Die über zehn Jahre alte Plattform soll zukünftig einen Therapieansatz und Spielgefährten für Kinder mit ASS bieten. Hierzu war die Durchführung von Verbesserungen an der Hardware (HW) und die Entwicklung einer geeigneten Software (SW) notwendig, die Kinder mit ASS eventuell fordert, vor allem aber unterstützt und dabei hilft, gegen ihre Schwächen anzukommen.

Die Ziele dieser Adaption beinhalten unter anderem:

- Austausch des Mikrocontrollers
- Austausch der Webcam
- Ansteuerung von Motoren (zur Steuerung des Roboters)
- Verarbeitung von Sensorwerten zur Kollisionsvermeidung
- Fernsteuerung des Roboters über ein Web-Interface
- Entwicklung von Spielen

Da es sich hierbei um ein Forschungsprojekt handelt, dessen Ziel es ist, regelmäßig Praxistests durchzuführen, ist es notwendig, die Plattform so portabel wie möglich zu halten, um sichere Transporte möglich zu machen.

In den nachfolgenden Unterkapiteln werden daher die Autismus Spektrum Störung und der Stand der Technik, im spezielleren Bezug auf Therapierobotik behandelt.

### 2.1 Autismus Spektrum Störung

Eine tiefgreifende Entwicklungsstörung, die bis zu 1% der Bevölkerung (Baird et.al. 2006) betrifft ist Autismus, diese ist auch unter dem Oberbegriff ASS (Autismus Spektrum Störung) zusammengefasst. Zusätzlich zu dieser Diagnose unterscheidet man aktuell noch laut ICD10 (Internationale statistische Klassifikation der Krankheiten und verwandter Gesundheitsprobleme) (ICD10 Index, 2017) zwischen frühkindlichem Autismus, Asperger-Syndrom oder auch atypischem Autismus.

Merkmale von ASS inkludieren auffällige soziale Interaktionen, ein begrenztes Interessengebiet, dem meist obsessiv nachgegangen wird, eingeschränkte Kommunikation und auch problematische Wahrnehmungsverarbeitung - seien es Geräusche, bestimmte Textilien oder auch Geschmäcker. Kognitive Funktionen, die beeinträchtigt sind, beinhalten

zum Beispiel den exekutiven Bereich wie die Unfähigkeit voranzuplanen oder Interaktionen selbst zu initiieren, sowie Inflexibilität. (Bölte, 2009)

Ein weiteres Defizit bei der zentralen Kohärenz zeigt sich mit Kontexterfassungsschwäche und bruchstückhafter Informationsverarbeitung. Ein gutes Beispiel hierfür ist folgende mögliche Aussage von betroffenen Menschen: „Ich sehe hunderte einzelner Bäume, aber einen Wald erkenne ich nicht.“ (Roy et al., 2009) Dafür zeigen Autisten oft eine immense Detailorientiertheit - vor allem was das Objekt ihres Interesses betrifft - und verblüffen oft mit Wissen und Fakten.

Spiele, mit denen sich neurotypische (Ausdruck der Autismus Bewegung zur Bezeichnung von neurologisch „normal“ entwickelten Menschen bzw. Menschen die nicht von Autismus betroffen sind) (The National Autistic Society 2017, 2017) Kinder gerne unterhalten und durch welche sie auch sozialen und emotionalen Umgang mit anderen Menschen lernen, stellen meist eine weitere große Hürde für autistische Menschen dar. Nicht nur ihr Wunsch nach Struktur und Stille erschwert es ihnen, an solchen Spielen teilzuhaben, sondern auch ihre Schwäche, Emotionen zu erkennen, mitzufühlen und sich in eine andere Person hineinzuversetzen. Dies verkompliziert spielende Interaktionen und raubt den Kindern gleichzeitig auch Lernmöglichkeiten, um gerade jene Defizite auszugleichen.

Der Fokus in der Diagnose liegt beim Wort „Spektrum“. Autisten sind nicht alle gleich und zeigen nicht dieselben Symptome. Es kann durchaus passieren auf der „leichten“ Seite des Spektrums zu liegen und dadurch vielleicht sogar Jahre auf eine Diagnose zu warten - oder die Symptome sind stärker ausgeprägt und liegen auf der „schwereren“ Seite. Unabhängig davon, wo man auf diesem Spektrum eingeordnet wird – Einschränkungen im Leben und Alltag gehören dazu. Einen Überblick über mögliche Symptome bietet Abbildung 1.



Abbildung 1: ASS Symptome (Quelle: (Autismus-Kultur, 2006))

Dieses Verständnis, dass kein Mensch mit Autismus einem anderen gleicht, ist eine wichtige Komponente, wenn man eine Therapiehilfe entwickeln will. Doch auch mit dieser Diversität gibt es einige Symptome, die häufiger vorkommen als andere, auf die man sich bei der Gestaltung eines Therapieroboters konzentrieren kann. Diese Symptome inkludieren zum Beispiel eine Tendenz, großen Menschenmengen auszuweichen, beziehungsweise durch „zu viel Input“ aus der Umwelt überfordert zu sein, und, damit verbunden, auch überhaupt kein Problem zu haben, sich für lange Zeit alleine zu beschäftigen. Andere Symptome wären auch das Bestehen auf Routine und repetitives Verhalten. Gerade wenn Autisten in eine Situation gebracht werden, die neu für sie ist, können Routinen oder Verhaltensweisen, wie zum Beispiel bestimmte wiederholte Körperbewegungen oder Wörter, ein Gefühl von Sicherheit geben.

All diese Faktoren führen zu dem Schluss, dass sich Roboter als Therapiehilfen oder Spielgefährten gut eignen, da sie ein kontrolliertes Umfeld bieten und Lernmöglichkeiten schaffen ohne dabei zu überfordern. Der in dieser Arbeit entwickelte Roboter soll Kindern helfen, Gefühle richtig zuzuordnen, da sie beispielsweise aufgrund oben erwähnter Schwierigkeit, Emotionen richtig zu deuten, in vielen Situationen unangemessen reagieren, sodass sie beispielsweise lachen, wenn sich jemand ernsthaft verletzt und weint. Mittels der in dieser Arbeit vorgestellten Lernplattform wird dies durch Spiele, wie „Emotionen erkennen“ (siehe Kapitel „Emotionen erkennen“) gelöst.

Die wichtigsten Ansätze bei der Therapie beinhalten:

- *Verhaltenstherapie und Kommunikation:*  
Es muss festgestellt werden, welche Fähigkeiten das Kind schon besitzt und welche es noch erlernen muss. Je nach Einordnung entsprechend des Spektrums kann dies von fehlenden sozialen Kompetenzen bis zu einem komplett nonverbalen Kind reichen. Hierbei können verhaltenstherapeutische Techniken wie Kontakt zu neurotypischen Kindern oder Rollenspielen das Erlernen neuer Kompetenzen fördern (Häußler, et. al., 2003) (Bach, 2006).
- *Selbstständigkeit:*  
Allein der Alltag bietet Menschen auf dem ASS Spektrum einige Hürden und Probleme. Um besser hiermit zurechtzukommen, ist es wichtig, dass diagnostizierte Menschen lernen, wichtige Informationen wahrzunehmen und zu verarbeiten, um ihre Umwelt besser zu verstehen (Häußler, 2008).
- *Mindreading und Theory of Mind:*  
Hier wird darauf Wert gelegt, dass autistischen Kindern Empathie nähergebracht wird. Ein Perspektivenwechsel und Verstehen der Gefühle anderer Menschen ist oft eine große Herausforderung und es wird mit dieser Therapie versucht, die Unterschiede zwischen eigenen Gedanken und dem Umfeld aufzuzeigen (Howlin, Baron-Cohen & Hadwin, 1999).
- *Sprachfähigkeit:*  
Eine eingeschränkte oder sogar fehlende Sprachentwicklung ist meist Teil der ASS Symptomatik. Dies zeigt sich durch eine monotone Sprachmelodie, Schwierigkeiten von Verständnis oder auch durch eine komplett fehlende Sprachentwicklung (Volkmar et al., 2005). Durch geeignetes Sprachtraining kann das aktive Sprechen, sprachliche Elemente und Phrasen sowie das Verstehen deren sozialen Bedeutungen gefördert werden.

## 2.2 Therapierobotik

Roboter für Therapiezwecke zu verwenden ist keine neue Idee. Mit jeder neuen wissenschaftlichen Entdeckung erweitert sich das Wissen nicht nur was die Technik, vor allem Roboter, angeht, sondern auch das Verständnis für die menschliche Psyche und den Körper. Die Umsetzung der Ideen für entsprechende Technologien erweist sich jedoch nach wie vor als schwieriger als erhofft. Auf der einen Seite fehlt EntwicklerInnen der entsprechende Hintergrund in Themengebieten wie Medizin oder Erfahrung mit gewissen Krankheiten, um eine geeignete Plattform zu gestalten. Auf der anderen Seite fehlt es TherapeutInnen, MedizinerInnen und PsychologInnen am technischen Wissen und/oder an Ressourcen. Um erfolgreiche Produkte entwickeln zu können, ist eine enge Zusammenarbeit zwischen EntwicklerInnen und AnwenderInnen notwendig, an der es oft scheitert.

Trotz der großen Defizite in der Vernetzung von wissenschaftlichem und therapeutischem Wissen konnte diesen durch gelungene Zusammenarbeit bereits entgegengewirkt werden. Mittlerweile ist man meistens in der Lage, Krankheiten, seien sie psychisch oder physisch, schneller zu diagnostizieren und diese zu behandeln. Dank des wachsenden Wissens kann festgestellt werden, welche technischen Komponenten in der Therapie sinnvoll sind, bzw. was für Hürden Menschen mit spezifischen Syndromen, wie in diesem Fall ASS, zu überwinden haben. Daraus ergeben sich in der Robotik vollkommen neue Möglichkeiten. Das Wissen und Verständnis, über Menschen und vor allem Kinder mit Autismus, wie sie die Welt erleben, und was für Hürden sie täglich überwinden müssen, ist unumgänglich, wenn man ein Therapiemittel, wie zum Beispiel einen Roboter, erschaffen will, weshalb alle in der Folge vorgestellten Forschungsprojekte von Teams aus EntwicklerInnen und TherapeutInnen erarbeitet wurden.

Ein Großteil der veröffentlichten Ergebnisse verschiedener Therapieroboter konzentriert sich vor allem auf den technischen Aspekt der Forschung (weniger auf den klinischen). Deshalb ist es wichtig, die klinischen Anwendungen solcher Studien im Auge zu behalten. Ein weiteres wichtiges Detail sind genaue Diagnosen und Schweregrad der Probanden bei den Versuchen. Gerade dies kann oft ein wichtiger Teil bei der Auswertung der Ergebnisse sein und vor allem für die Weiterarbeit mit jenem gewonnenen Wissen.

Generell kann man veröffentlichte Studien zu Therapierobotern für Autisten in vier Hauptkomponenten teilen: (Diehl et al., 2012)

1. Studien, welche den Umgang von Autisten mit Robotern bzw. Menschen vergleichen
2. Studien, die Roboter verwenden, um eine Reaktion bei Probanden auf dem ASS Spektrum hervorzurufen
3. Studien mit Robotern, die verwendet werden um eine Fähigkeit beizubringen oder zu üben
4. Studien mit Robotern, die verwendet werden, um Reaktion und Feedback auf Leistungen zu bieten

Die nun vorgestellten Projekte lassen sich vor allem in Kategorie zwei und drei einordnen.

## **2.2.1 Roboterrobbe PARO**

PARO ist zwar kein Roboter für Menschen mit ASS, jedoch ein wichtiges Beispiel für erfolgreiche Therapierobotik. PARO hat die Form eines Seehundes (siehe Abbildung 2). Dieser Therapieroboter wurde mit der Idee kreiert, die Vorteile von Therapietieren nachzuahmen, vor allem für Orte, die nicht zugänglich für eine Therapie mit Tieren sind, wie Spitäler oder Pflegeheime (PARO Robots U.S., Inc., 2017).

Die Zielgruppe für PARO sind ältere Menschen und jene, die unter Demenz leiden. Versuche mit diesem Roboter zeigen, dass sich die Interaktionen der Patienten untereinander und zu ihren Pflegern verbessern und zu ihrer Entspannung und Motivation beitragen. (Burton,

2013). Weiter konnte festgestellt werden, dass Patienten durch Umgang mit dem künstlichen Therapietier mehr emotionale Reaktionen zeigten, aktiver waren und mehr kommunizierten. Diese Ergebnisse zeigen das Potential, welches Roboter in der Therapie haben, und dass dieser Ansatz einen großen Vorteil gegenüber Menschen mit emotionalen Einschränkungen aufgrund diverser Syndrome bietet.



Abbildung 2: PARO (Quelle: (Shopify, 2017))

### 2.2.2 Kaspar

Kaspar ist wohl das bekannteste Projekt im Bereich der Therapierobotik für autistische Kinder. Entwickelt wurde es 2005, mittlerweile gibt es knapp 30 Kaspar-Prototypen. (University of Hertfordshire, 2017a)

Die Forschung und Entwicklung von Kaspar wird von der „University of Hertfordshire’s Adaptive Systems Research Group“ durchgeführt. Dieses Team führte bereits verschiedene Forschungsarbeiten im Bereich „KI und Robotik“ durch. Die Arbeiten bauen auf Entwicklungen bereits aus dem Jahre 1998 auf, die zu einem ersten funktionierenden Prototyp von Kaspar führten. Die ersten beiden EntwicklerInnen waren hier Kerstin Dautenhahn und Dr. Ben Robins. Kerstin Dautenhahn ist eine Pionierin in der Forschung von sozialer Robotik und Mensch-Roboter-Interaktion und hat das Feld der roboterunterstützten Therapie für Kinder mit Autismus gegründet. Im Jahr 1999 veröffentlichte sie den ersten Artikel über die Vorteile von interaktiven Robotern für die Therapie und Entwicklung von Kindern mit Autismus. (Dautenhahn, 2017) (University of Hertfordshire, 2017b)

Kaspar ist ein menschenähnlicher Roboter in Kindesgröße, jedoch reduziert auf ein stark vereinfachtes Aussehen (siehe Abbildung 3). Kaspar besteht zwar aus einem fleischfarbenen Gesicht, besitzt jedoch weder Gesichtshaar, noch andere menschlichen Details (wie z.B. Falten). Der Gesichtsausdruck ist neutral und ohne die Darstellung eines bestimmten Geschlechts oder Alters, was Kindern die Möglichkeit bietet, in Kaspar das hineinzuiinterpretieren, was sie sich wünschen. Auf Erwachsene macht Kaspar zwar keinen anziehenden Eindruck, diese realistische, vereinfachte menschenähnliche Darstellung wirkt jedoch sehr stark ansprechend auf Kinder mit ASS.



Abbildung 3: Kaspar (Quelle: (University of Hertfordshire, 2016))

Zwischen Aussehen und Akzeptanz von Robotern gegenüber Menschen ist ein ungleichmäßiger Verlauf festzustellen, man spricht vom „Uncanny Valley“ (Mori, 1970) (siehe Abbildung 4). 1970 unterbreitete Masahiro Mori die Theorie, dass die Akzeptanz eines humanoiden Roboters steigt, je realistischer dieser aussieht. Gelingt es, dass sich der Roboter der Perfektion nähert, schlägt diese Akzeptanz in die entgegengesetzte Richtung und wird als verstörend empfunden. Der Grund wird damit erklärt, dass der Roboter nicht komplett menschlich oder sogar wie eine sich bewegende Leiche wirkt. Eine Studie, die Evaluierungen mit unterschiedlichen Robotern entlang der Menschenähnlichkeits-Koordinate präsentiert, gibt es bisher nicht. Studien, die Bilder von Robotern, sowie einen einzelnen androiden Roboter als Interviewer beinhalten, zeigen zwar Ergebnisse, die die „Uncanny Valley“ Theorie unterstützen, jedoch nicht vollständig beweisen (MacDorman, 2005a) (MacDorman et al., 2005b) (MacDorman & Ishiguro, 2006). Unter Berücksichtigung dieser Studien, sowie der bekannten Informationen über Kinder, die auf dem Autismus Spektrum liegen, wurde Kaspars Design gewählt (Dautenhahn et al., 2009).



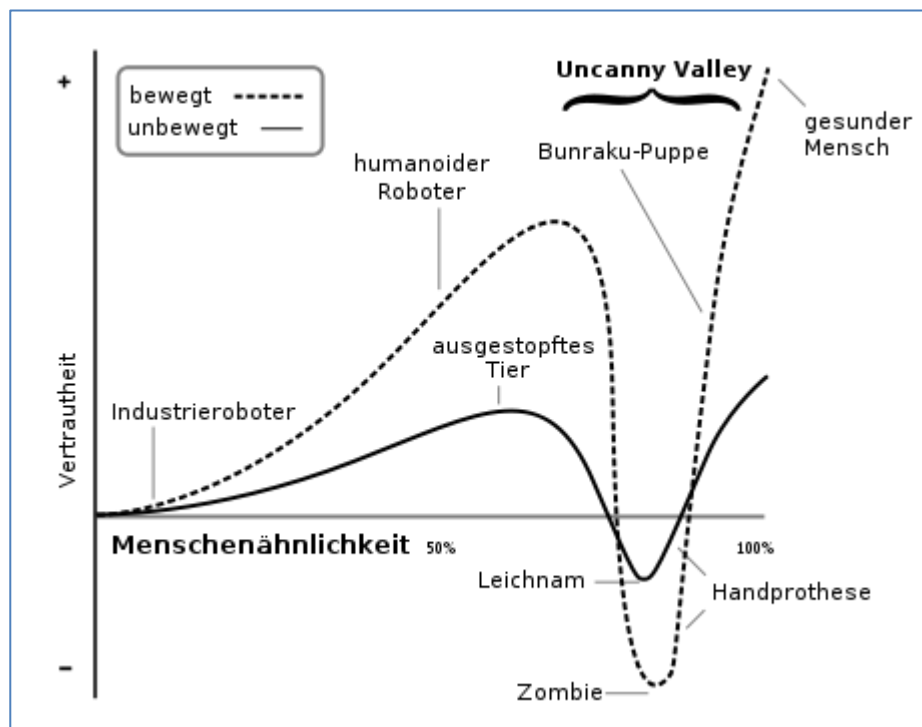


Abbildung 4: deutschsprachige Version des „Uncanny Valley“ (Quelle: (Eggshell Robotics, 2009))

### 2.2.2.1 Kaspar in der Praxis

Die meiste Forschung mit Kaspar wird in Schulen und speziellen Einrichtungen in Großbritannien durchgeführt. Mittlerweile haben ungefähr 170 Kinder weltweit mit Kaspar interagiert. In einer unabhängigen Studie wurde Kaspar von 54 ASS-ExpertInnen untersucht, die überzeugt werden konnten, dass der Roboter in vielen Bereichen sowohl der Therapie, als auch Bildung eine große Unterstützung für Kinder mit ASS bietet. Die Hauptaufgabe und das Ziel jener ExpertInnen ist es, die betroffenen Kinder in ihrem Alltag zu unterstützen und sie, im besten Fall, in Richtung eines eigenständig geführten Lebens zu führen. Um dies zu erreichen gibt es unterschiedliche Zielsetzungen. KASPAR wurde als Therapieroboter erwähnt, der bis zu zwölf verschiedener dieser Objektivie anspricht (Huijnen, Lexis & de Witte, 2016).

PsychologInnen haben bereits den Vorschlag geäußert, Kaspar auch als Unterstützung für Kinder mit anderen sozialen Interaktionsproblemen, oder spezifischen Sprachbeeinträchtigungen, einzusetzen. Kaspar hilft den Kindern, in einem entspannten Umfeld Kommunikation und Interaktion zu entdecken. Die Ausweitung der Zielgruppe ermöglicht einen größeren Nutzfaktor und würde sich nicht mehr nur auf autistische Kinder beziehen.

Auch in dieser Arbeit ist aus ähnlichen Gründen die Zielgruppe zu einem späteren Zeitpunkt erweitert worden (siehe dazu Seite 24 im Kapitel „Implementierung“).



Abbildung 5: Kaspar imitiert Kind (Quelle: (Daynurseries, 2016))

„TRACKS Autism“ ist ein Lernzentrum spezialisiert auf Kinder mit ASS und macht seit sechs Jahren von der Kaspar-Plattform Gebrauch (TRACKS autism, 2017). Sowohl Lehrende, als auch Eltern von betroffenen Kindern bemerken den Einfluss des Roboters: Kinder erkennen beispielsweise Gefühle oder interagieren zum ersten Mal in der Klasse, nachdem sie mit Kaspar gespielt haben (siehe Abbildung 5). „Wir haben viele Wow-Momente gehabt, seit dem Kaspar ein fixer Teil unserer Schule geworden ist.“ (Nan Cannon Jones, Gründer von TRACKS)

Eine Studie, die in einer speziellen Schule in Athen durchgeführt wurde, bezog sich auf 15 Spielsitzungen mit sieben Kindern im Alter von sieben bis elf Jahren statt. Als Ergebnis fassten die LehrerInnen zusammen, dass Kaspar auf lange Sicht gesehen einen positiven Einfluss auf Kinder mit Autismus haben kann.

### 2.2.3 Teo

Teo ist ein mobiler, „umarm-barer“ Roboter, welcher für Kinder mit Störungen der Entwicklung des Nervensystems (= NDD – NeuroDevelopmental Disorder) gefertigt worden ist. In die Klassifizierungsgruppe von NDDs fallen diverse Syndrome, unter anderem auch ASS (Bonarini et al, 2016).

Teo hat einen weichen Korpus (siehe Abbildung 6), welcher auf verschiedene Arten der Berührung reagiert und sich frei am Boden bewegen kann. Seine „emotionalen Reaktionen“ manifestieren sich in Licht-, Sound- und Bewegungseffekten. Das Einsatzgebiet des Teo-Roboters beinhaltet sowohl die spiel-basierte Therapierobotik, als auch „freies Spiel“ (spontane Interaktionen mit Teo, welche soziales Verhalten, positive Emotionen und Ausdruck fördern und vorantreiben).



Abbildung 6: Therapieroboter Teo (Quelle: (Bonarini et al, 2016))

Teo unterstützt drei Formen der Ganzkörper-Interaktion:

- Roboter Manipulation
- Interaktion zwischen Roboter und Kind auf Distanz
- Gemeinsame Interaktion in der virtuellen Welt

Die Roboter-Manipulation beinhaltet taktilen Kontakt mit Teo, in welchem Teos Körpersensoren unterscheiden, was gerade mit ihm geschieht (Streicheln, Umarmung, sowie zwei verschiedene Stärken von Schlägen können gemessen werden). Abhängig von der Intensität des physischen Kontaktes werden entsprechende Stimuli erzeugt, die Teos emotionalen Zustand repräsentieren. Teo ist glücklich, wenn sein Körper gestreichelt oder sanft berührt wird, und reagiert mit Vibration, heiteren Drehungen, sowie einem langsam blinkenden, grünen LED-Licht. Bei leichten Schlägen wird Teo wütend und fährt schnell auf das Kind zu, die LEDs blinken dann rot. Bei brutalen Schlägen wird Teo ängstlich, was sich in langsamen Rückzugsfahrten sowie pulsierend gelb leuchtenden LEDs zeigt.

Bei der Interaktion zwischen Roboter und Kind bewegen sich Roboter und Kind gemeinsam durch den Raum. Die Bewegungen, bzw. die Position des Kindes, lösen bestimmte emotionale Effekte von Teo aus, ähnlich den bereits genannten. Diese hängen sowohl von der Distanz zueinander, als auch der Geschwindigkeit, Richtung und relativen Bewegung ab. Da Ultraschall-Distanzsensoren hierfür relativ schlechte (unpräzise) Daten liefern, kommt die Microsoft Kinect 3D-Kamera<sup>1</sup> als externer Sensor zum Messen der Präsenz und Bewegungen des Kindes zum Einsatz. Die Kinect ist eine von Microsoft im Jahr 2010 veröffentlichte und kostengünstige Kamera, die mit einem RGB-D Sensor (Red Green Blue – Depth) arbeitet. Sie ist erstmals als Zubehör für die Spielkonsole Xbox 360 erschienen.

---

<sup>1</sup> <https://developer.microsoft.com/en-us/windows/kinect>

Da es der eingebaute Sensor möglich macht, ein Farbbild inklusive einer Tiefenkarte darzustellen, wurde sie schnell für andere Zwecke (als für die Spielkonsole) eingesetzt. Sie ermöglicht es Gesten zu erkennen und kann auch für die Pfadplanung von Robotern im Raum eingesetzt werden. Viele derartige Projekte sind im Internet oder in Fachartikeln und Büchern zu finden. (Kramer et al, 2012)

Eine gemeinsame virtuelle Interaktion findet statt, wenn sich der Roboter und das Kind in der Nähe einer Digitalanzeige oder Projektion befinden, die Multimedia-Inhalte (z.B. Geschichten oder Charaktere) zeigt. Die Kinect-Kamera kann die Position und die Bewegungen des Roboters und die Gesten des Kindes erkennen. Das integrierte System löst Effekte in der virtuellen Welt entsprechend den Bewegungen und Positionen des Kindes und des Roboters aus.

Teos Verhalten kann in drei Modi ausgeführt werden: ferngesteuert, autonom reaktiv und programmiert. Im Fernbedienungsmodus löst der Betreuer die gewünschten Stimuli auf Teos Körper aus und gibt die Roboterbewegungen und Verhaltensweisen mittels Fernbedienung vor (diese Form findet meist bei Robotern in therapeutischen Kontexten Anwendung). Die Fernbedienung kann ein Joystick, ein Tablet oder ein Bluetooth-Stift sein. Mit einem Tablet haben die Betreuer die zusätzliche Möglichkeit, die verbale Interaktion zu fördern, indem sie einen Ausspruch wählen, welcher vom Roboter gesprochen werden soll (z. B. "Bravo!"). Wenn ein Spiel noch nicht gestartet hat, arbeitet Teo im autonomen, reaktiven Modus. Der Roboter befindet sich in einem Warte-Zustand, indem er auf die Anwesenheit von Menschen reagiert, die sich in der Nähe befinden. Hierbei wendet sich der Roboter den Personen zu und lädt sie auch verbal zum Spielen ein. Befindet sich niemand in der Nähe, dreht sich Teo nach rechts und links, als ob er sich suchend umsieht. Im programmierten Modus fungiert Teo autonom als Spielkamerad im Rahmen von zielorientierten, strukturierten Aktivitäten und verhält sich entsprechend seiner Rolle, wie sie im aktuellen Spiel definiert ist.

## **2.2.4 Puffy**

Ein Roboter, der von EntwicklerInnen kreiert wurde, die ebenfalls bei vorher erwähntem Therapieroboter Teo mitgewirkt haben, ist Puffy. Der Roboter wurde gemeinsam mit SpezialistInnen für Entwicklungsstörungen des Nervensystems entwickelt (Gelsomini et al., 2017).

Das Ziel von Puffy ist die therapeutische und pädagogische Unterstützung von autistischen Kindern mit einem speziellen Fokus auf Wahrnehmungs- und Sinnesstörungen sowie kognitiven und sozialen Defiziten. Puffy ist mobil, der Korpus ist oval, weich, aufpumpbar und hat ähnliche Bewegungsmöglichkeiten wie Teo. Der Roboter ist in der Lage, Gesten, Mimik, Bewegungen und Emotionen von Kindern zu verarbeiten. Dazu nutzt Puffy, wie auch der Therapieroboter Teo die Microsoft Kinect 3D-Kamera. Durch Bewegung, Musik, Lichter und Projektionen auf dem Korpus kann Puffy mit dem Kind kommunizieren (siehe Abbildung 7a). Durch diese Vielzahl an Möglichkeiten bietet der Roboter eine Reihe unterschiedlicher

Stimuli, die auf verschiedene Sinne der Kinder abzielen und so als therapeutische Unterstützung dienen können. Die soziale Komponente wird zum Beispiel durch die Augen des Roboters unterstützt, um Kindern ein Verständnis dafür zu geben, wohin der Roboter schaut und vor allem auch Augenkontakt zu üben. Auch Puffy ist in der Lage, zwischen weichen (positiven) Berührungen und festen Schlägen zu unterscheiden - und entsprechend darauf zu reagieren. Reaktionen beinhalten eine Änderung des inneren Lichtes (siehe Abbildung 7b), der Geräusche und der Wörter.



Abbildung 7: Puffy: (a) mit Projektion auf dem Korpus (b) Veränderung des inneren Lichts (Quelle: (Gelsomini et al., 2017))

Im Gegensatz zu Teo ist Puffy durch seine veränderbare Größe auch in der Lage, mit Körpersprache emotional zu reagieren. Er schrumpft beispielsweise, wenn er Angst hat. Somit können Kindern lernen von Körperhaltungen und Gesten auf das emotionale Wohlbefinden des Gegenübers zu schließen.

## 2.2.5 Softwarebasierte Therapie

Andere Therapieansätze für Autismus, die keine Roboter inkludieren, sind digitale Spiele. Ein gutes Beispiel hierfür ist „autism games“, eine australische Website<sup>2</sup>. Die Seite ist eine Zusammenarbeit von einem auf Multimedia Design spezialisierten Studiengang, einer Sonderschule, einer Autismus Initiative und einem Therapiecenter.

Das Ziel dieser Website ist es, Kinder mit mittlerem bis schwerem Autismus spielerisch beim Erlernen von alltäglichen Fähigkeiten zu helfen. Hierbei ist das Projekt in zwei Teile getrennt: Der erste Teil ist die offizielle (oben genannte) Website. Diese kann von Lehrern und Eltern verwendet werden, um sich zu informieren und die verschiedenen Spiele und deren Ziele kennenzulernen. Der zweite Teil basiert auf der „Whiz Kid Games“ Website<sup>3</sup>. Durch diese

<sup>2</sup> <http://www.autismgames.com.au>

<sup>3</sup> <http://www.whizkidgames.com/>

Seite können die Kinder auf die Spiele zugreifen, was den Vorteil hat, dass Eltern oder Aufsichtspersonen andere Computer und Internetfunktionen limitieren können.

Die Spiele sind gegliedert in Kategorien, die der Förderung von unterschiedlichen Fähigkeiten entsprechen. Diese beinhalten den Umgang mit Veränderungen, nonverbale Gesten, Emotionen und Empathie zu erlernen und zu verstehen, oder auch um Augenkontakt zu üben. Das Spiel „Robbie the Robot“ soll Kindern mit Autismus helfen, Gefühle richtig zuzuordnen (Swinburne University, 2017a). Das Gesicht von „Robbie“ wird durch eine reale Person repräsentiert, um Gefühle richtig zu kommunizieren. Im Spiel wird „Robbie“ auf seiner Reise auf der Suche nach seinem verlorenen Hut begleitet. Neben verschiedenen anderen Aktivitäten, wird das Kind aufgefordert eine Aufgabe zu erfüllen, die eine Zuordnung von Gefühlen beinhaltet. Ein Beispiel ist in Abbildung 8 zu sehen. Dieses Spiel ist die Inspiration für das in dieser Arbeit implementierte Spiel „Emotionen erkennen“ (siehe Kapitel „Emotionen erkennen“).



Abbildung 8: „Robbie the Robot“, erste Aufgabe (Swinburne University, 2017b)

Ebenfalls spezialisiert im Bereich der softwarebasierten Therapie ist die österreichische Firma Platus. Diese legt ihren Schwerpunkt auf die Entwicklung von Hilfsmitteln für Menschen mit Beeinträchtigungen. Diese Hilfsmittel inkludieren sowohl Lern- und Therapiesysteme als auch Produkte, die körper- und mehrfachbehinderte Menschen in der Kommunikation unterstützen (Platus Learning Systems GmbH, 2017a).

Eines dieser Lernsysteme ist TOM:

TOM ist auf Kinder mit Wahrnehmungsstörungen und Entwicklungsverzögerungen abgestimmt (Platus Learning Systems GmbH, 2017b). Genau wie „autism games“ wurde dieses softwarebasierte Lernsystem von einem Team, sowohl aus Techniker/innen als auch spezialisierten Pädagogen/innen bestehend, kreiert. Ähnlich wie beim ersten in dieser Arbeit implementierten Spiel „Objekte finden“ (Kapitel „Motorik“) sind Objekte mit Computerchips ausgestattet (siehe Abbildung 9). Im Gegensatz zu „autism games“ wird hier die Motorik (das Lernen durch Greifen und Tasten mit den verfügbaren Objekten) der Kinder gefördert.



Abbildung 9: TOM Systeminhalte (Quelle: (Platus Learning Systems GmbH, 2017b))

Platus nennt folgende „erreichbare Lernziele“ (Platus Learning Systems GmbH, 2017b):

- Verbesserung der taktil-haptischen, akustischen und der visuellen Wahrnehmungsorganisation
- Verbesserung des Gedächtnisses
- Anbahnung der Kommunikation bei nichtsprechenden Kindern
- Erreichen eines passiven Wortverständnisses
- Erlangen eines Symbolverständnisses
- Zusammenlauten und Erkennen von Silben und Wörtern
- Erlangen des Mengenverständnisses



### 3 Implementierung

Die Roboterplattform *RoboFriend* (siehe Abbildung 10) ist eine ca. 70cm hohe, mobile, Roboterplattform, die von Thomas Heike bereits 2004 entworfen und produziert wurde. Die Plattform wurde in der Masterarbeit von Florian Hackl (Titel: „RoboFriend – An interactive multi-purpose robotic platform“) analysiert und für eine weitere Verwendung als Therapieplattform aufbereitet (Hackl, 2016).

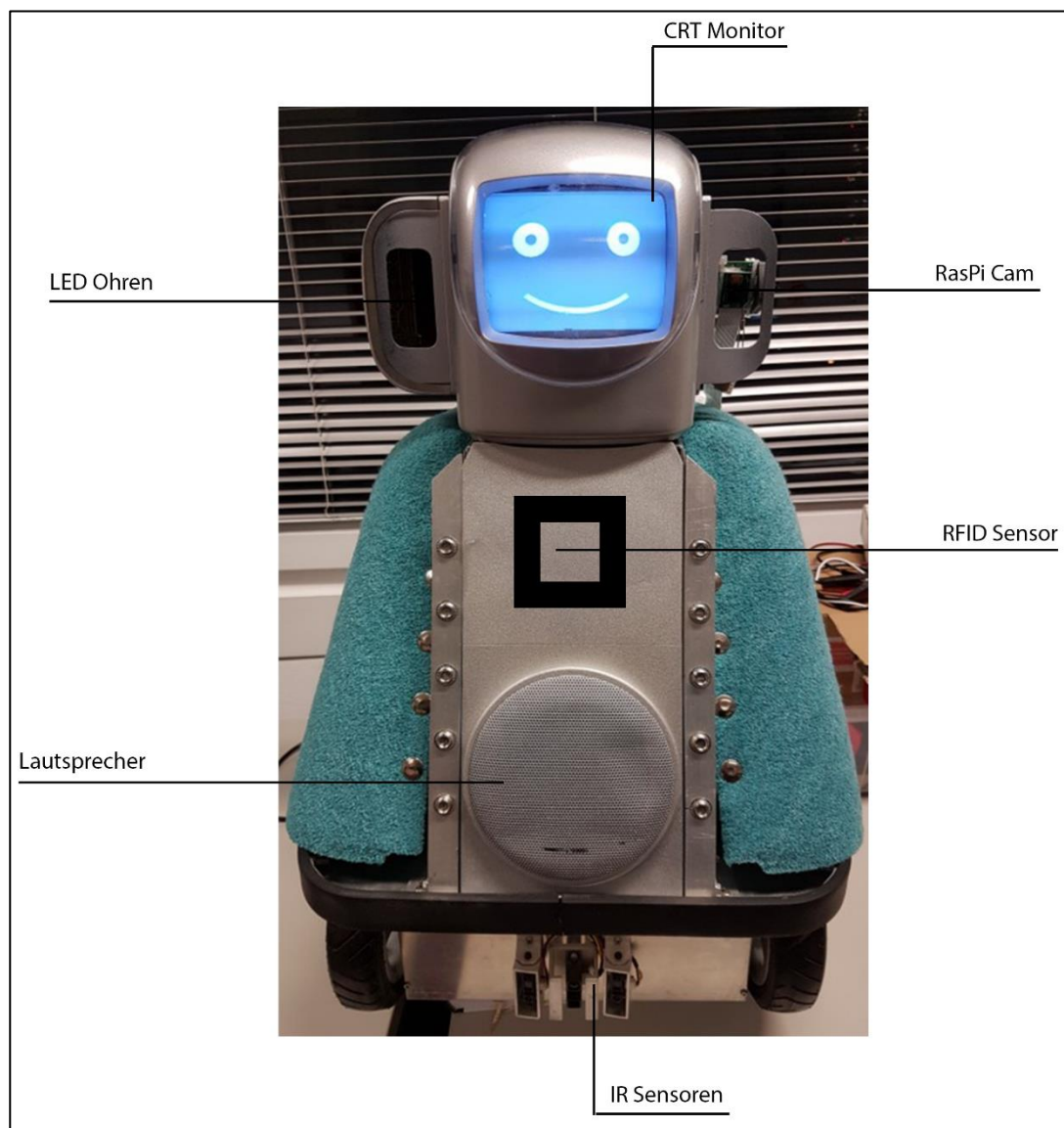


Abbildung 10: RoboFriend mit Beschriftung

Im Rahmen dieser Masterarbeit soll diese bestehende Plattform erweitert werden, um möglichst den Anforderungen einer Unterstützungstechnologie für Kinder mit Autismus gerecht zu werden. Im Verlauf der Arbeit wurde der Fokus von Kindern mit ASS auf eine größere Zielgruppe (Kinder mit erhöhtem Förderbedarf) erweitert. Dies hat den Grund, dass



einerseits die Palette an unterschiedlich starken Ausprägungen von ASS so enorm ist, dass bei einer Spezialisierung die tatsächlich angesprochene Zielgruppe zahlenmäßig so gering ist, dass es kaum möglich wäre, eine entsprechende Plattform sinnvoll zu testen. Andererseits ist im Zuge dieser Arbeit eine Kooperation zwischen der FH Technikum Wien<sup>4</sup> und dem Lehrer Thomas Bettinger<sup>5</sup> der Schule Paulusgasse<sup>6</sup> entstanden, dessen professionelle Meinung und Feedback ein wichtiger Teil dieser Arbeit sind. Bettinger ermöglichte es auch, im Zuge dieser Arbeit die Roboterplattform mit der neuen erweiterten Zielgruppe zu testen.

Dieses Kapitel beschäftigt sich mit dem Hardware- und Software-Aufbau des Roboters, sowie mit den dafür festgelegten Requirements.

## 3.1 Hardware

Dieses Kapitel setzt sich näher mit dem Hardware-Aufbau des Roboters sowie dessen Aktoren und Sensoren auseinander.

### 3.1.1 Hardware Requirements und Aufbau

Im Zuge dieser Arbeit mussten Hardware Komponenten ausgetauscht werden. Diese Bauteile sollten möglichst kostengünstig ersetzt werden.

Die Hardware Anforderungen beinhalteten:

- *Ersetzen des alten „C-Control“ Mikrocontroller:*  
Der Austausch des C-Control Mikrocontroller war notwendig, da die UART-Schnittstelle (Universal Asynchronous Receiver Transmitter) beschädigt worden war, und dadurch keine neue Software in den Flash-Speicher des Mikrocontrollers programmiert werden konnte. Dazu wurde ein gängiger Mikrocontroller, ein Teensy++ 2.0 (siehe „Teensy++ 2.0“), eingebaut, der die Programmierung über USB (Universal Serial Bus) möglich machte.
- *Ersetzen der alten Webcam:*  
Für die bisher verbaute Webcam waren keine aktuellen Treiber mehr vorhanden und wurde deshalb durch die RasPi Cam ersetzt.
- *Anbindung der Motoren und Sensoren der Roboterplattform über den neuen Mikrocontroller.*  
Auf diesem läuft eine Firmware, die sowohl für die Motoransteuerung, die Sensorverarbeitung, als auch für das Auslesen der Akkuspannung, um die

---

<sup>4</sup> <https://www.technikum-wien.at/>

<sup>5</sup> <https://www.bettinger.at/bettinger>

<sup>6</sup> <http://paulusgasse.at/>

verbleibende Batteriespannung zu berechnen, zuständig ist (siehe Softwarekapitel: „Mikrocontroller Firmware“).

Andere im *RoboFriend* bereits vorhandene Bauteile sind:

- Zwei Getriebemotoren (SWF 403.033, 24V, 13 U/min, 3 Nm)
- Drei 12V Akkus (7,2Ah)
- Drei Infrarot (IR) Sensoren (siehe „IR Sensoren“)
- Raspberry Pi 2B (siehe „Raspberry Pi 2B“)
- CRT (Cathode Ray Tube) Monitor und Lautsprecher
- RFID (**R**adio **F**requency **I**dentification) Sensor (siehe „RFID Sensor“)
- Schnittstelle zu den LED Ohren und dem Kamera Servomotor (siehe „IO-Warrior“)

Zur Kommunikation zwischen Roboter und Kind, sowie der Navigation zwischen den Funktionen (Übungen) soll zusätzlich ein Windows Tablet verwendet werden. In Abbildung 11 ist eine Übersicht der verwendeten Komponenten zu sehen: Die drei 12V Akkus sind im Betrieb seriell geschaltet und mithilfe eines DC (Direct Current – Gleichstrom) Converters (siehe „DC-DC Converter“) dienen sie der Spannungsversorgung der weiteren elektronischen Komponenten (Teensy++ 2.0, Raspberry Pi und CRT Monitor). Die drei IR Sensoren, befinden sich ebenfalls im unteren Teil des *RoboFriends* (siehe Abbildung 10) und sind über ADC (Analog Digital Converter) Pins direkt am Mikrocontroller angeschlossen. Von diesem werden sie zur Abstandsmessung und Kollisionsvermeidung verwendet. Der RFID Sensor befindet sich mittig (siehe Abbildung 10) und ist über USB am Raspberry Pi angeschlossen. Er findet in den erstellten Spielen Anwendung, indem er als zusätzliches Kommunikationsmedium zwischen Kind und Roboter dient. Ebenfalls mit dem Raspberry Pi verbunden ist ein CRT Monitor, der das Gesicht des *RoboFriends* als Composite Videosignal empfängt (Abbildung 10). Über die CSI (Camera Serial Interface) Schnittstelle ist zusätzlich noch die RasPi Cam angeschlossen, die eine Videoübertragung aus Sicht des Roboters (da im Ohr platziert) ermöglicht. Über einen IO-Warrior, welcher über USB mit dem Raspberry Pi verbunden ist, ist außerdem die Ansteuerung des vorhandenen Kamera Servomotors sowie der LED Ohren möglich.

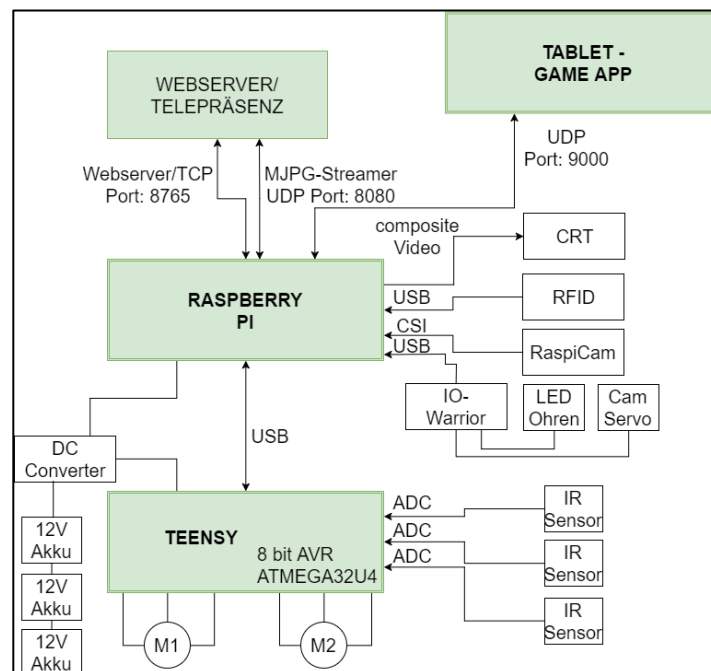


Abbildung 11: Hardware Übersicht

### 3.1.2 Teensy++ 2.0

Der Teensy++ 2.0 ist ein Mikrocontroller, der einen 8bit AVR AT90USB1286 Prozessor verwendet (Atmel Corporation, 2009), der von der Arduino Entwicklungsumgebung unterstützt wird (siehe Tabelle 1).

<b>Prozessor</b>	AT90USB1286 8 bit AVR 16 MHz
<b>Flash Speicher</b>	130048 kB
<b>RAM Speicher</b>	8192 kB
<b>Spannungsversorgung</b>	5V
<b>ADC Pins</b>	8
<b>PWM Pins</b>	9

Tabelle 1: Spezifikationen des Teensy++ 2.0 Mikrocontrollers

Am Teensy++ 2.0 angeschlossen sind, wie bereits erwähnt, zwei Motoren, ein OPV (**O**perations**v**erstärker) zur Ablesung der Batteriespannung an einem ADC Pin, sowie drei IR Sensoren, die zur Abstandsmessung dienen, um gewährleisten zu können, dass genug Platz für die gewünschten Fahrbefehle vorhanden ist. Mit dem Raspberry Pi ist er über USB verbunden und wird gegenüber dem Betriebssystem des Raspberry Pi Computers als virtueller COM-Port repräsentiert.

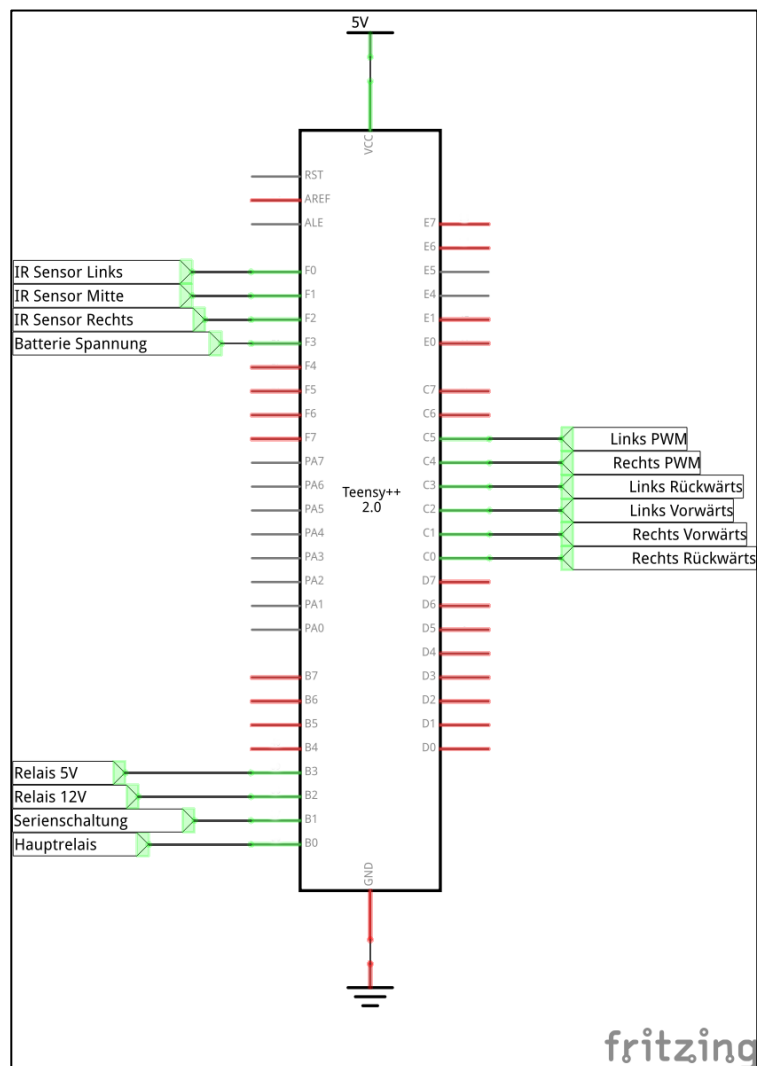


Abbildung 12: Teensy++ 2.0 Schaltung

### 3.1.2.1 Schaltplan Lochrasterplatine

Um die bereits vorhandenen Verbindungen, nach dem Ersetzen des C-Control Mikrocontrollers durch den Teensy++ 2.0, nutzen zu können, musste dieser auf eine Lochrasterplatine gelötet und mit den verfügbaren Schnittstellen verbunden werden. In Abbildung 12 ist ersichtlich, an welchen Pins die entsprechenden Hardwarekomponenten angeschlossen sind. Für die Darstellung wurde das Tool „fritzing“ verwendet. Fritzing<sup>7</sup> bietet eine gleichnamige Software an, die Open-Source ist, und mit welcher Schaltpläne in unterschiedlichstem Style ermöglicht werden. Dies soll helfen Produkte und Projekte zu dokumentieren und mit anderen teilen zu können. Ins Leben gerufen wurde „Fritzing“ von der FH Potsdam und wird jetzt von der „Friends-of-Fritzing“ Foundation weiterentwickelt (Fritzing, 2017). In Abbildung 13 ist die fertig gelötete Lochrasterplatine zu sehen.

<sup>7</sup> <http://fritzing.org/home/>

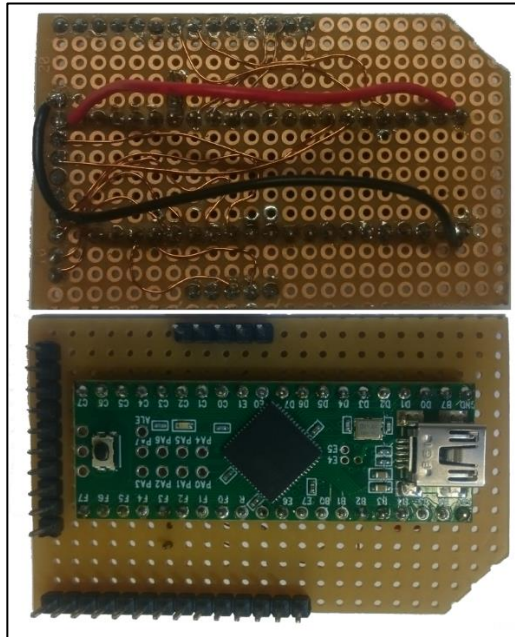


Abbildung 13: gelötete Teensy++ 2.0 Platine

### 3.1.2.2 Messung der Batteriespannung

Da drei 12V Akkus vorhanden sind, kann die Gesamtspannung über 36V betragen. Der Teensy++ 2.0 erlaubt eine Inputspannung von 0 bis 5V. Um den verbleibenden Akkustand zu überprüfen, wurde eine Schaltung mit einem Operationsverstärker (OPV), zwei Metallschicht-Widerständen und einer Schottky-Diode gewählt. Die Verwendung eines einfachen Spannungsteilers könnte zu einer erhöhten Eingangsspannung am Teensy++ 2.0 führen. Der OPV liefert dem Mikrocontroller eine maximale Eingangsspannung von 5V und dient somit als Schutzelement. Zur Berechnung und Auswahl der Komponenten wird die Formel (1) für einen Spannungsteiler verwendet.

$$V_{out} = \frac{R_2}{R_1 + R_2} V_{in} \quad (1)$$

Verwendet wurden hierfür ein OPV des Typs LM358NG<sup>8</sup>, ein 27kΩ und ein 3kΩ Widerstand, sowie eine Schottky-Diode des Typs BAT41<sup>9</sup>. Der entsprechende Schaltplan ist in Abbildung 14 ersichtlich.

<sup>8</sup> [http://www.produktinfo.conrad.com/datenblaetter/150000-174999/151730-da-01-en-LM358N\\_Motorola.pdf](http://www.produktinfo.conrad.com/datenblaetter/150000-174999/151730-da-01-en-LM358N_Motorola.pdf)

<sup>9</sup> [http://www.produktinfo.conrad.com/datenblaetter/150000-174999/153036-da-01-en-BAT\\_41.pdf](http://www.produktinfo.conrad.com/datenblaetter/150000-174999/153036-da-01-en-BAT_41.pdf)

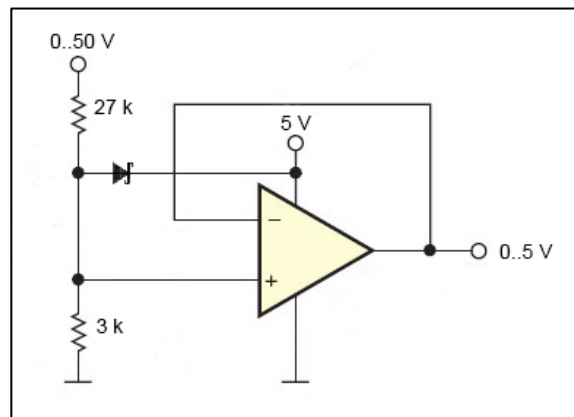


Abbildung 14: OPV Schaltplan (Quelle: (Electrical Engineering, 2012))

Umgesetzt wurde die Schaltung auf einer Lochrasterplatine, wie in Abbildung 15 zu sehen ist.

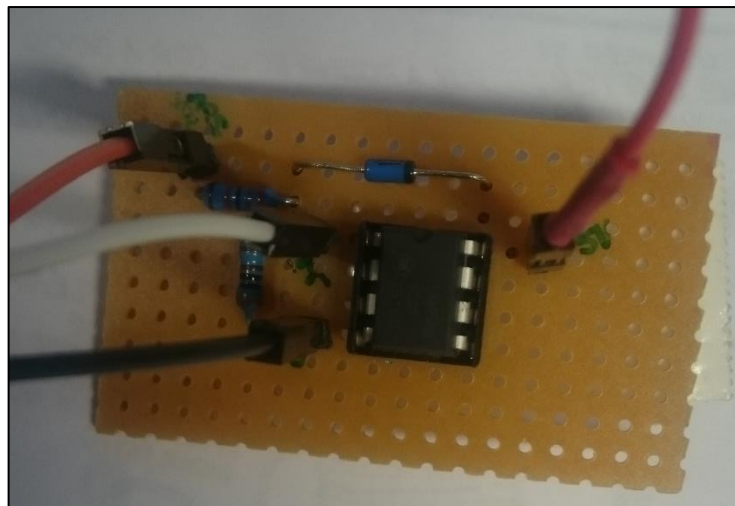


Abbildung 15: gelötete OPV Schaltung

### 3.1.3 Raspberry Pi 2B

Der Raspberry Pi ist ein Einplatinencomputer und hat etwa die Größe einer Kreditkarte. Das Modell 2B (siehe Abbildung 16) ist im Jahr 2015 auf den Markt gekommen und ersetzt die etwas schwächeren Vorgängermodelle. Die Videoausgabe kann über Composite Video oder HDMI erfolgen. Zusätzlich besitzt er einen microSD-Kartenleser, einen CSI und vier USB 2.0 Anschlüsse. (Raspberry Pi Foundation, 2017)

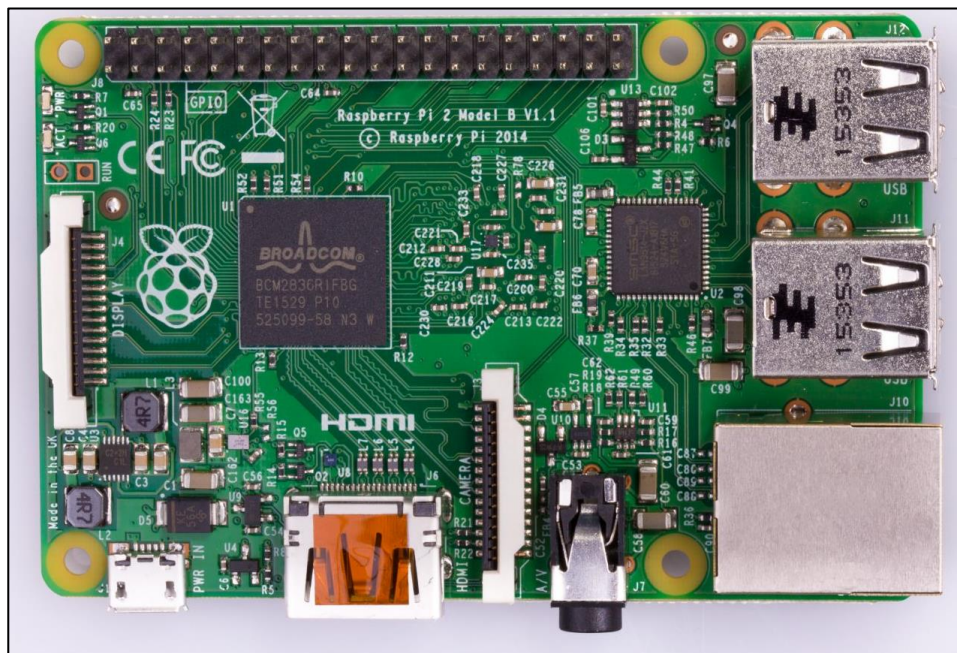


Abbildung 16: Raspberry Pi 2 Model B, Ansicht von oben (Quelle: (Raspberry Pi Foundation, 2017))

Im *RoboFriend* ist der Raspberry Pi Computer seriell über USB mit dem Mikrocontroller verbunden. Der CRT Monitor (inklusive Lautsprecher) ist über Composite Video, ein RFID Sensor über USB, die RasPi Cam über die entsprechende CSI Schnittstelle und LED Treiber und Kamera Servomotor über einen IO-Warrior über USB angeschlossen.

### 3.1.4 IO-Warrior

Im *RoboFriend* ist der IOWarrior24 im „Kopf-Bereich“ eingebaut. Dieser ist ein Mikrocontroller, der über USB mit dem Raspberry Pi verbunden ist. Er ist ausgestattet mit einer Firmware, die es ermöglicht, dessen I/O Pins über USB-HID (Human Interface Device) Befehle zu steuern (Code Mercenaries Hard- und Software GmbH, 2013). Geräte, mit denen der Benutzer direkt interagiert zählen zur HID Geräteklasse, wie zum Beispiel Tastatur, Maus und Joystick. Die Ansteuerung des IO-Warriors erfolgt über ein Programm in der Programmiersprache C. Mit dem IO-Warrior kann durch das Schalten eines Relais der CRT Monitor Ein- bzw. Ausgeschaltet werden. Zum Ansteuern der LED Ohren ist ein simpler Leistungstransistor am IO-Warrior angeschlossen. Des Weiteren kann mittels PWM der verfügbare Kamera Servomotor kontrolliert werden. Da weder für die LED Ohren, noch für den Kamera Servomotor eine Dokumentation vorhanden war, wurden diese im Zuge dieser Arbeit nicht priorisiert und erst zu einem späteren Zeitpunkt erstmals erfolgreich angesteuert. Aufgrund der knappen Zeitressourcen konnten die Funktionen jedoch nicht mehr für die vorhandenen Anwendungen implementiert werden.

### 3.1.5 DC-DC Converter - Gleichspannungswandler

DC-DC Converter wandeln eine am Eingang zugeführte Gleichspannung in eine Gleichspannung mit höherem, niedrigerem, invertierten oder isolierten Spannungsniveau. Dazu wird die Eingangsspannung zunächst in eine Wechselspannung gewandelt, welche transformiert und abschließend gleichgerichtet wird (IT Wissen, 2016).

Der im *RoboFriend* eingebaute Converter ist der GS-R400VB des SGS-Thomson Herstellers<sup>10</sup> und kann Eingangsspannungen bis 46V in Ausgangsspannungen zwischen 5-40V umwandeln.

### 3.1.6 Sensoren

Für die „Sinne“ des Roboters wurden drei IR Sensoren, ein RFID Reader sowie eine RasPi Cam verwendet. In den folgenden zwei Unterkapiteln wird auf die Funktionsweise der Sensoren eingegangen.

#### 3.1.6.1 IR Sensoren

Zur Abstandsmessung werden drei IR Sensoren verwendet. Diese befinden sich alle an der vorderen Seite des Roboters, je einer mittig, nach rechts und nach links geneigt. IR Sensoren bestehen aus einem Sender und einem Empfänger, diese sind übereinander angebracht. Der Sender, der sich oberhalb befindet, sendet infrarotes Licht, während der darunterliegende Empfänger, der eigentliche Sensor, das reflektierte Licht misst (siehe dazu Abbildung 17). Um den Abstand feststellen zu können, wird die Menge des reflektierten infraroten Lichts gemessen.

---

<sup>10</sup> [http://www.mouser.com/ds/2/389/stmicroelectronics\\_cd00000563-329845.pdf](http://www.mouser.com/ds/2/389/stmicroelectronics_cd00000563-329845.pdf)



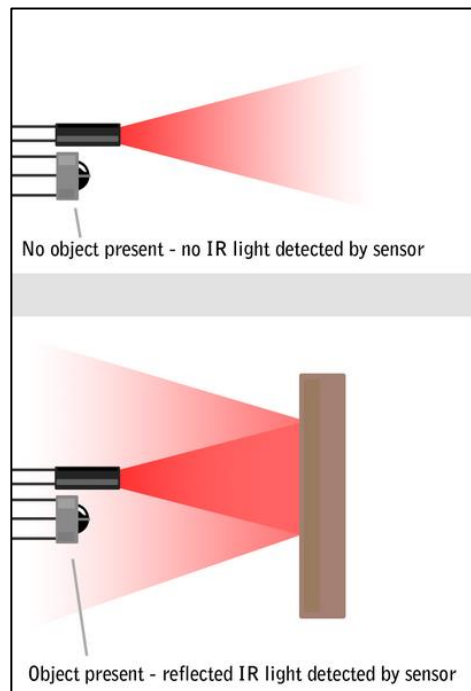


Abbildung 17: Funktionsweise eines IR Sensors (Quelle: (Robotics Academy, 2015))

Im Roboter verbaut sind zwei SHARP IR-Sensoren GP2D12<sup>11</sup> (für den linken und rechten Bereich) sowie ein SHARP IR-Sensor GP2Y0A02YK<sup>12</sup> (für die Mitte). In Abbildung 18 ist das Verhältnis des Abstandes zur Ausgangsspannung des mittleren Sensors zu sehen.

<sup>11</sup> Datenblatt: [https://engineering.purdue.edu/ME588/SpecSheets/sharp\\_gp2d12.pdf](https://engineering.purdue.edu/ME588/SpecSheets/sharp_gp2d12.pdf)

<sup>12</sup> Datenblatt: <https://www.sparkfun.com/datasheets/Components/GP2Y0A21YK.pdf>

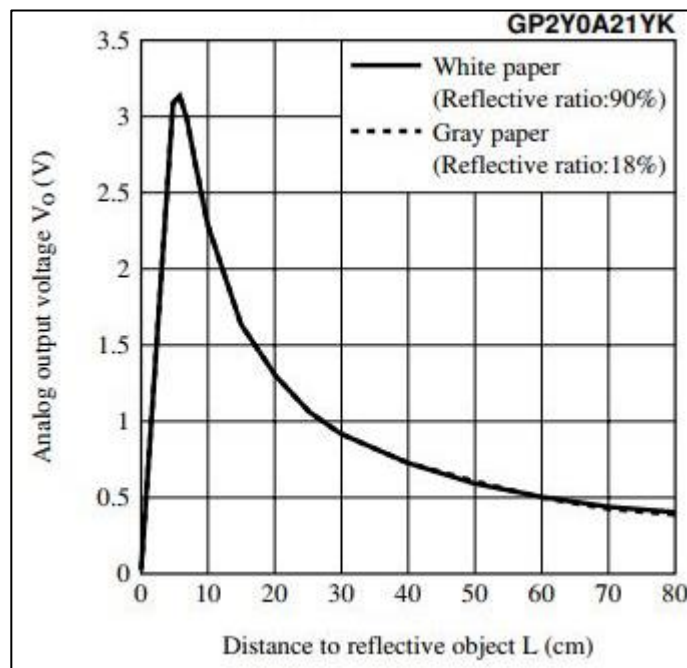


Abbildung 18: Gegenüberstellung von Ausgangsspannung zum Objektabstand (Quelle: (Sparkfun Electronics®, 2017b))

### 3.1.6.2 RFID Sensor

Als Interaktionsmittel zwischen Kind und Roboter wird ein Sparkfun RFID Reader (Sparkfun Electronics®, 2017a) verwendet. RFID verwendet Radiowellen um Informationen zu lesen, die sich auf einem „RFID Tag“ befinden.

Es gibt hierfür verschiedene Frequenzbereiche (Finkenzeller, 2010) (IEE, 2005) (Ward et al., 2006):

- Langwellen (Low Frequency - LF, 30-500kHz): müssen dem Standard (ISO/IEC 18000-2:2009<sup>13</sup>) entsprechen; kurze Reichweite (<0,5m); geringe Datenrate (<1kbit/s). Zugriffssysteme mit Karten befinden sich in diesem Bereich, oft bei 125kHz, aber auch Tier IDs.
- Kurzwellen (High Frequency – HF, 3-30MHz): höhere Reichweite mit bis zu 1,5m; mittlere Übertragungsgeschwindigkeit (25kbit/s). In diesen Bereich fällt Near Field Communication (NFC) (13,56MHz). NFC findet zum Beispiel Anwendung beim kontaktlosen Bezahlen mit der Bankomatkarte.
- Sehr hohe Frequenzen (Ultra High Frequency – UHF, 300MHz-3GHz): Hohe Reichweite (bei 433MHz bis zu 100m, bei 865-956MHz zwischen 0,5 und 5m); Datenrate von 30-100kbit/s. Wird angewendet bei (halb-)automatisierter Warenverteilung, Kontrollen bei Versand (Logistik)

<sup>13</sup> <https://www.iso.org/standard/46146.html>

- Mikrowellen-Frequenz (Super High Frequency – SHF, 2-30GHz): Reichweite bis zu 10m; hohe Lesegeschwindigkeit mit bis zu 100kbit/s. Anwendung bei Fahrzeugen: Parkhäuser, Bahnhöfe, etc.

Wie auch die meisten Zugriffssysteme ist der verwendete RFID Reader<sup>14</sup> für LF entworfen. Die verwendeten RFID Tags (Karten und Sticker) haben die Frequenz 125kHz.

## 3.2 Software

Dieses Kapitel befasst sich mit den implementierten Softwaremodulen (die entwickelten Funktionen, wie sie schließlich auch aus User Ansicht zu sehen sind, werden im nächsten Kapitel „Implementierte Spielfunktionen“ beschrieben).

### 3.2.1 Software Requirements und Aufbau

Im Rahmen dieser Arbeit wurde die komplette Software für den Betrieb des Roboters entworfen und implementiert.

Die Anforderungen hierfür waren:

- Entwicklung der Firmware für den Mikrocontroller
- Entwicklung einer Roboter-Applikation am Raspberry Pi als „Middleware“
- Entwicklung einer Weboberfläche für Telepräsenz und Fernsteuerung
- Entwicklung einer Spiel-GUI
- Verwendung des CRT Monitors und des Lautsprechers
- leichte Erweiterbarkeit der Software
- Anlegung und Wartung eines GitHub Repositories<sup>15</sup>
- Verwendung von Open-Source Ressourcen

Als Open Source wird jene Software bezeichnet, deren Code öffentlich zugänglich und von Dritten sowohl genutzt als auch geändert werden kann. In den meisten Fällen ist die Open Source Software auch kostenlos. (Open Source Initiative, 2017)

Eine Übersicht der implementierten Softwaremodule ist in Abbildung 19 zu sehen. Der Mikrocontroller kümmert sich um die Motoransteuerung, die Sensorwertverarbeitung sowie die Spannungsmessung der Akkus (siehe „Mikrocontroller Firmware“). Der Raspberry Pi Computer dient als „Middleware“ und beinhaltet die Roboter-Applikation (siehe „Robot Middleware – Python Hauptprogramm“), die Fahrbefehle an den Teensy++ 2.0 schickt. Der Webserver läuft ebenfalls am Raspberry Pi. Die Übertragung der Videodaten erfolgt dabei

---

<sup>14</sup> <https://www.sparkfun.com/products/13198>

<sup>15</sup> <https://github.com/ProjectKitchen/RoboFriend>

über den MJPEG Streamer mittels UDP Sockets. Für die Kommunikation zwischen Spiel-GUI und der „Middleware“ wird auch eine UDP Verbindung verwendet (siehe „Kommunikationsprotokoll“). Eine tabellarische Übersicht der SW-Module und ihrer Funktionen ist ebenfalls aus Tabelle 2 ersichtlich.

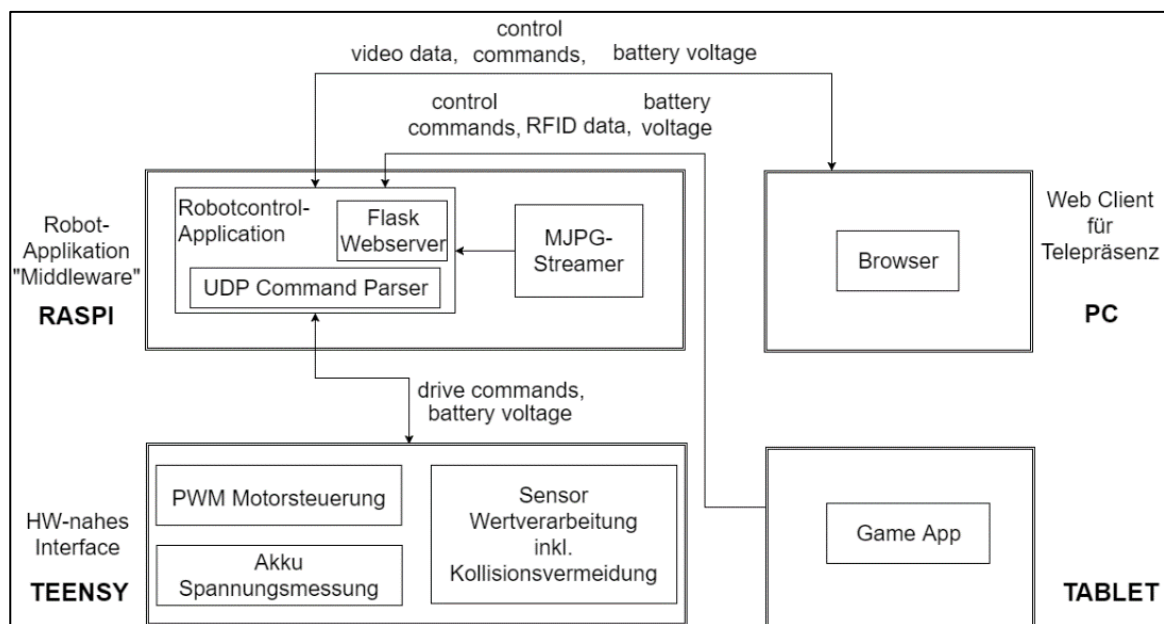


Abbildung 19: Software Übersicht

Für die Softwareentwicklung wurden hauptsächlich zwei Programmiersprachen verwendet: Python (Version 2.7) für die Implementierung unter Linux/Raspberry Pi Computer (siehe dazu „Robot Middleware – Python Hauptprogramm“) und C++ (unter Verwendung der Arduino-IDE, für die Implementierung am Teensy++ 2.0 Mikrocontroller) (siehe dazu „Mikrocontroller Firmware“).

Hardware	Softwaremodul	Funktion	Schnittstellen	Neu entwickelt
Teensy++ 2.0	Firmware	Motoransteuerung; Sensorwertverarbeitung; Kollisionsvermeidung; Akkuspannungsmessung	USB ↔ Raspberry Pi	ja
Raspberry Pi 2B	Python Hauptprogramm	Webserver; Übermittlung von: - Fahrbefehlen an Mikrocontroller; - RFID Daten an Tablet-GUI; - Akkuanzeige an Tablet-GUI & Web Frontend;	UDP ↔ Tablet-GUI; TCP ↔ Web Frontend	ja
	MJPEG-Streamer	Video Übertragung an Web Frontend	UDP → Web Frontend	nein
Tablet	Tablet-GUI	Vier Spiele; Robotersteuerung; Akkuanzeige	UDP ↔ Python Hauptprogramm	ja
PC	Web Frontend	Robotersteuerung; Akkuanzeige; Kamera Video Stream	TCP ↔ Python Hauptprogramm;	ja

Tabelle 2: Übersicht der SW Module mit Beschreibung

### 3.2.2 Kommunikationsprotokoll

In den folgenden zwei Unterkapiteln wird die Kommunikation zwischen den wichtigsten Komponenten beschrieben. In Abbildung 19 sind die zu übertragenden Daten eingezeichnet.

#### 3.2.2.1 Serielle Kommunikation zwischen Raspberry Pi und Teensy++ 2.0

Zwischen Teensy++ 2.0 und Raspberry Pi findet eine serielle Kommunikation über USB statt, bei der sowohl Fahrbefehle als auch Akkuspannungsanfragen von der Robot Middleware an den Mikrocontroller gesendet werden. Der Teensy++ 2.0 liefert die geforderte Akkuspannung zurück, interpretiert wird diese erst im Python Hauptprogramm.

#### 3.2.2.2 Kommunikationsprotokoll zwischen Raspberry Pi und Tablet

Zwischen Raspberry Pi und Spiel-GUI, die auf dem Windows Tablet läuft, erfolgt die Kommunikation via UDP Sockets. Hierfür wurde ein leicht erweiterbares Kommunikationsprotokoll geschaffen wurde. Die Spiel-GUI schickt verschiedene Befehle an

den Raspberry Pi, und werden dort weiterverarbeitet. Die verfügbaren Kommandos sind in Tabelle 3 zu sehen. In der ersten und fünften Spalte sind das Start- und End-Flag zu finden. Sie dienen lediglich zur Überprüfung des empfangenen Befehls. Folgende Kommandos (zweite Spalte) sind verfügbar: Fahrbefehle („move“), Soundbefehle („sound“), Gesichtsmanipulationen (Veränderung der Augen und des Lächelns) („face“) und Akkustandabfragen („battery“). Das erste Argument (dritte Spalte), spezifiziert den Befehl. Ein optionales zweites Argument (vierte Spalte) gibt weitere Auskunft über das Kommando. Ob ein Rückgabewert erwartet wird, informiert die letzte Spalte. Auch der Webserver macht Gebrauch vom Protokoll, denn über das Frontend der Webpage (siehe „Webserver / Web Frontend“) können ebenfalls Fahr- und Soundbefehle gesendet werden.

Start-Flag	Kommando	Argument 1	(Optionales) Argument 2	End-Flag	Rückgabewert JA/NEIN
:RUN:	move	forward		:EOL:	NEIN
			step		
		backward			
			step		
		left			
			step		
		right			
			step		
		forward_right			
		forward_left			
	sound	backward_right			
		backward_left			
		stop			
		soundpath			
			random		
			mood		
		smile	increase		
			decrease		
	face	eyes	up		
			down		
			right		
			left		
		answer	correct		
			wrong		
	battery	status			JA (integer)

Tabelle 3: Befehlsliste von der GUI zur Robot Middleware

Die Befehle selbst werden als Strings verschickt und beinhalten Semikolons um die einzelnen Teile trennen zu können. Um dem Roboter beispielsweise die Anweisung zu schicken einen Schritt vorwärts zu fahren, wird folgender String von der Spiel-GUI zum Python Programm am Raspberry Pi geschickt: „:RUN:move;forward;step:EOL:“. An den Anfang jedes Befehls wird ein Start-Flag eingefügt: „:RUN:“ sowie an das Ende ein End-Flag „:EOL:“ (End Of Line). Diese beiden Flags gewährleisten, dass auf der Empfängerseite überprüft werden kann, ob ein vollständiger Befehl angekommen ist, oder ob dieser abgebrochen ist. Wenn diese beiden Flags vorhanden sind, wird der String, der sich zwischen diesen Markierungen befindet, weiterverarbeitet. Ein Ausschnitt des Codes, der für diese Weiterverarbeitung zuständig ist, ist in Abbildung 20 zu sehen. Das Kommando und dessen Argumente, die durch die Semikolons getrennt sind, werden anschließend in einem Array gespeichert (Zeile 78) und der Reihe nach überprüft. Im genannten Beispiel sieht dieses nun folgendermaßen aus: [„move“, „forward“, „step“]. Das erste Array-Element ist das Kommando, in diesem Fall „move“ (Zeile 82), entspricht also einem Fahrbefehl. Nach Erkennung des Kommandos wird eine Funktion aufgerufen, die sich um die Fahrbefehle kümmert (Zeile 83). In dieser Funktion entscheidet das nächste Array-Element (erstes Argument) über die Richtung des Fahrbefehls, in diesem Beispiel „vorwärts“. Trifft nun der Fall ein, dass ein weiteres Argument übergeben wurde, wird auch dieses überprüft, und beeinflusst den Fahrbefehl weiterhin. Hier wird nur ein Vorwärts-Schritt ausgeführt. Ist jedoch kein weiteres Argument vorhanden, ist es kein Schritt, sondern eine Schleife, das heißt, der Roboter fährt so lange vorwärts, bis er entweder einen neuen Befehl erhält, oder durch die Erkennung eines Hindernisses selbst zum Stehen kommt.

```

76 def chooseAction(data):
77     global bat_prozent
78     dataArray = data.split(';') # Nachricht wird in ein Array gespeichert
79     print(dataArray)
80     action = dataArray[0] # erstes Argument
81     dataArray = dataArray[1:] # restliche Argumente
82     if action == "move":
83         move(dataArray)
84     elif action == "sound":
85         pygame.mixer.stop()
86         info = dataArray[0]
87         dataArray = dataArray[1:]
88         if info == "play":
89             playsound(dataArray)
90     elif action == "face":
91         faceManipulation(dataArray)
92     elif action == "battery":
93         info = dataArray[0]
94         dataArray = dataArray[1:]
95         if info == "status": #wenn status abgefragt wird
96             ser.write("8")
97             bat = str(ser.readline())
98             bat = bat.strip(' \t\n\r')
99             try:
100                 bat = int(bat)
101             except:
102                 print("no data available, recieved message: " + str(bat))
103             else: #weiterverarbeitung des wertes
104                 bat_prozent = 0
105                 if bat > 242:
106                     bat_prozent = ((bat-242)/4.95)
107                     bat_prozent=int(round(bat_prozent))
108                 sendtogui("battery;" +str(bat_prozent))
109                 print(bat_prozent)
110                 print(bat)

```

Abbildung 20: Code Ausschnitt Robot Middleware – Weiterverarbeitung des erhaltenen Befehls

Auf dieselbe Art und Weise erfolgt die Kommunikation von Raspberry Pi zu GUI. Es wird ebenfalls ein String gesendet, dessen erstes Keyword nach dem Start-Flag das Kommando bestimmt. Ein Beispiel hierfür ist ebenfalls in Abbildung 20 zu sehen (Zeile 108). Neben dem Senden des Batteriestatus, werden auch alle empfangenen RFID Daten an die Spiel-GUI gesendet (siehe Tabelle 4).

Start-Flag	Kommando	Argument 1	End-Flag	Anmerkung
:RUN:	battery	<i>aktueller Akkustand in %</i>	:EOL:	Rückgabewert von „battery;status“
	rfid	<i>empfangene RFID Daten</i>		

Tabelle 4: Befehlsliste vom Python Hauptprogramm zur GUI



### **3.2.2.3 Synchronisierung**

In allen Fällen, bei denen kein Rückgabewert erwartet wird (siehe Tabelle 3), findet eine asynchrone Kommunikation statt. Alle über UDP gesendeten Befehle zwischen GUI und Raspberry Pi werden nach dem „Fire and Forget“ Prinzip verschickt. Das Kommando wird geschickt und das Programm läuft weiter, ohne auf eine Antwort oder Empfangsbestätigung zu warten. In den Kategorien „move“, „sound“ und „face“ ist auch keine Antwort notwendig. Die GUI schickt zu Beginn nur einmal den Befehl für eine „battery;status“ Abfrage um einen definierten Anzeigewert zu bekommen. Der Raspberry Pi Computer sendet nämlich in regelmäßigen Intervallen nämlich unaufgefordert den aktuellen Akkustand.

Das Raspberry Pi erwartet ebenfalls keine Antworten bei der UDP-Socket Kommunikation. Lediglich bei der Akkuspannungsabfrage vom Raspberry Pi an den Teensy++ 2.0 erfolgt eine synchrone Kommunikation, denn hier wird sofort die Antwort abgefragt (siehe Zeile 96f in Abbildung 20). In den anderen Fällen besteht zwischen dem Raspberry Pi und dem Teensy++ 2.0 eine asynchrone Verbindung, da bei diesen keine Rückmeldung erwartet wird.

Am Web Frontend erfolgt die Akkuanzeige ebenfalls asynchron aktualisiert, da der Wert erst bei „Aktualisierung“ der Seite (beim Eintreten eines Events) neu angezeigt wird.

## **3.2.3 Mikrocontroller Firmware**

Wie im Kapitel „Hardware“ bereits erwähnt, werden auf dem Teensy++ 2.0 die Motoren angesteuert und die Sensorwerte verarbeitet.

### **3.2.3.1 C++ (Arduino IDE mit Library Support für HW Abstraktion)**

Programmiert wurde der Mikrocontroller mit der kostenlosen Open-Source Arduino Software. Die dort verwendete Arduino Programmiersprache verwendet eine Mischung aus C und C++ ähnelnden Funktionen.

### **3.2.3.2 Programmcode**

Nach der optionalen Zuordnung von Variablen zu den Pin-Nummern wird eine Setup-Routine ausgeführt. Diese wird einmalig und zu Beginn des Programms aufgerufen. In dieser Funktion müssen zunächst die Pins als Out- oder Input festgelegt werden. In diesem Fall werden hier zusätzlich die entsprechenden Relais angesprochen um die Akkus für die Stromversorgung seriell zu schalten und die Baudrate für den seriellen Port auf 9600 für das Raspberry Pi eingestellt. Danach folgt die „main loop“ in der der Hauptcode zu finden ist, der sich sozusagen in einer Endlosschleife befindet. In dieser Hauptfunktion werden die Sensordaten stets erneuert und es wird durchgehend geprüft, ob Kommandos über die serielle Schnittstelle empfangen werden um diese dann auszuführen. Dazu zählen beispielsweise Fahrbefehle oder aber auch die Abfrage nach dem Akkustand. Wird ein

Befehl nicht erkannt, wird eine Fehlermeldung zurückgeschickt. Soll ein Fahrbefehl ausgeführt werden, werden auch die IR Sensorwerte kontrolliert, um abzufragen, ob das Fahren in der jetzigen Position überhaupt erlaubt werden kann. Sollte ein Sensorwert den Hinweis auf ein Hindernis geben, wird der Fahrbefehl nicht ausgeführt, bzw. sollte während dem Ausführen des Fahrbefehls etwas im Weg sein, wird der Befehl abgebrochen.

Das Ansteuern der Motoren erfolgt jeweils über zwei Pins (Richtung und Geschwindigkeit). Die IR Sensoren und die Batteriespannung werden über ADC Pins eingelesen.

Um Ausreißer bei den Sensorwerten richtig zu behandeln, werden 30 Sensorwerte hintereinander eingelesen und der Durchschnitt davon ermittelt. Um dies auf eine möglichst einfache Weise durchzuführen, wird die Bibliothek „RunningAverage.h“ verwendet. Diese ist Open Source und kann kostenfrei in die Arduino Software importiert und über das GitHub Repository (RobTillaart, 2016) erworben werden.

Für das vereinfachte Ausführen der Fahrbefehle wurde eine Bibliothek geschrieben, die die Deklaration von zwei Enum-Datentypen beinhaltet. Diese sind „Direction“ und „Tire“. Sie werden für das Ausführen von Fahrbefehlen genutzt, um eine einfache Handhabung des Codes zu ermöglichen. Wird beispielsweise die Funktion für das „Vorwärtsfahren“ (siehe Abbildung 21) aufgerufen, werden zunächst die relevanten Abstände kontrolliert (Zeile 187), sollte hierbei eine Möglichkeit der Kollision erkannt werden, so wird die Funktion beendet (Zeile 189). Meldet kein Sensor ein Hindernis, wird der Roboter zunächst zum Stillstand gebracht (Zeile 192), um vorige Fahrbefehle zu unterbrechen. Erst dann wird die Funktion „Move“ zweimal, je einmal für jeden Reifen, ausgeführt. Der Funktionsaufruf enthält die Parameter um die Richtung der einzelnen Reifen zu bestimmen (Zeile 193f).

```
185 void forward_Loop()  
186 {  
187     if (SensorMiddleTriggered || SensorLeftTriggered || SensorRightTriggered)  
188     {  
189         return;  
190     }  
191  
192     Stop();  
193     Move(Forward, Left);  
194     Move(Forward, Right);  
195 }
```

Abbildung 21: Codebeispiel „Vorwärtsfahren“

Die Akkuspannung wird zwar auf dem Teensy++ 2.0 über einen ADC Pin eingelesen, jedoch wird der empfangene Wert direkt über die serielle Schnittstelle an das Raspberry Pi übertragen und erst dort weiterverarbeitet. Anliegen dürfen am Pin zwischen 0 und 5V, worum sich die gefertigte OPV Schaltung kümmert (siehe Kapitel „Messung der Batteriespannung“). Die Auflösung der ADCs beträgt 10bit, das heißt die Werte, die an das Raspberry Pi übertragen werden, befinden sich zwischen 0 und 1023. (siehe Kapitel „Akkustand“)

### 3.2.4 Robot Middleware – Python Hauptprogramm

Das Hauptprogramm für die Steuerung des *RoboFriends*, welches in Python 2.7 programmiert ist, läuft am Raspberry Pi. In diesem werden Lautsprecher und Monitor Ausgaben sowie der Webserver verwaltet. Es beinhaltet das, bzw. macht Gebrauch vom Kommunikationsprotokoll (siehe Kapitel: „Kommunikationsprotokoll“), um die empfangenen Befehle richtig weiterzuverarbeiten und andere Informationen ebenfalls an die Spiel-GUI zurück zu senden. Zusätzlich bekommt der Webserver (siehe Kapitel „Webserver“) vom Raspberry Pi sowohl den Akkustand als auch den Videostream der RasPi Cam.

#### 3.2.4.1 Python

Python ist eine objektorientierte Programmiersprache, die auf den gängigsten Betriebssystemen (OS – Operating System) läuft: Windows, Mac, Linux. Diese Programmiersprache steht unter Entwicklung einer Open Source Lizenz, die es ermöglicht Python uneingeschränkt zu nutzen, zur Entwicklung beizutragen oder sie für kommerzielle Dinge zu nutzen. Administriert ist die Lizenz von der Python Software Foundation (Python Software Foundation, 2001-2017a). Diese ist eine Organisation, die sich um die Weiterentwicklung von Open Source Technologien in Verbindung mit Python bemüht (Python Software Foundation, 2001-2017b). Python lebt unter anderem davon, dass die große Community, die stets wächst, vieles zur Entwicklung beiträgt.

Beispiele für verwendete Python Bibliotheken (Python Module), ihr Verwendungszweck sowie eine Auflistung in welchen SW-Modulen sie benötigt werden, sind ersichtlich in Tabelle 5. Die Verwendung der Python Bibliothek „pygame“ ist für die Umsetzung dieser Arbeit essentiell. Sie beinhaltet alles was für die Darstellung einer graphischen Oberfläche benötigt wird und mehr. „pygame“ wird sowohl in der Roboter Applikation am Raspberry Pi als auch auf dem Windows Tablet für die GUI verwendet. Die Bibliothek wurde, wie ihr Name bereits verrät, speziell für die Spieleprogrammierung entwickelt. Die Bibliothek „socket“, wird für die UDP Socket Verbindung zwischen den beiden Komponenten gebraucht. „sys“ wird benötigt für weitergegebene Argumente beim Programmaufruf, oder für den Aufruf von Pfaden innerhalb des Programms, oder schlicht und einfach für das Beenden eines Programmes. „time“ findet Anwendung in Codebereichen, wo es notwendig ist, für s Sekunden „anzuhalten“ mit dem Funktionsaufruf `time.sleep(s)`. Das „urllib“ Modul wird für das richtige Interpretieren von Befehlen über den Webserver benötigt (siehe „Webserver / Web Frontend“).

Python Modul	Verwendungszweck	Verwendet in:
flask	Flask Webserver	Python Hauptprogramm
pygame	Darstellung der graphischen Oberfläche	Python Hauptprogramm; Tablet-GUI
serial	Serielle Verbindung (USB)	Python Hauptprogramm
socket	UDP Verbindung	Python Hauptprogramm; Tablet-GUI
sys	Argumente bei Programmaufruf Aufruf von Pfaden Beenden des Programms	Python Hauptprogramm; Tablet-GUI
threading	Vier Threads beim Python Hauptprogramm Zwei Threads bei der Tablet-GUI	Python Hauptprogramm; Tablet-GUI
time	„Pausieren“ des Programms (time.sleep(s))	Python Hauptprogramm; Tablet-GUI
urllib	Interpretation von Befehlen des Web Frontends	Python Hauptprogramm

Tabelle 5: verwendete Python Module und ihre Verwendung

### 3.2.4.2 RFID Daten

Die RFID Daten, die über den RFID Sensor empfangen werden, werden verarbeitet und unter Einhaltung des Kommunikationsprotokolls an die Spiel-GUI geschickt. Eine verarbeitete RFID besteht aus 12 Zeichen. Die rohen RFID Daten enthalten vier zusätzliche Zeichen zu der wirklichen ID: Start of Text (STX), End of Text (ETX), New Line/Line Feed (LF) und Carriage Return (CR). Um alle RFID zu empfangen müssen daher zunächst 16 Zeichen über den Reader eingelesen werden (Abbildung 22, Zeile 356). Anschließend wird die Python Methode `string.replace(„old“, „new“)` verwendet. *String* ist die Variable (der String), die verändert werden soll, *old* ist der in *string* vorkommende Teil, der durch *new* ersetzt werden soll. Zeile 357-360 beinhalten die notwendigen Kommandos um die erwähnten zusätzlichen Zeichen zu eliminieren.

```

356 daten = serRFID.read(16)
357 daten = daten.replace("\x02", "" )
358 daten = daten.replace("\x03", "" )
359 daten = daten.replace("\x0a", "" )
360 daten = daten.replace("\x0d", "" )
361 readRFIDnumber=daten

```

Abbildung 22: Lesen und Verarbeitung von RFID Daten

### 3.2.4.3 Akkustand

Über die serielle Schnittstelle werden die Rohdaten der Akkuspannung vom Teensy++ 2.0 Mikrocontroller empfangen. Diese müssen erst weiterverarbeitet werden, damit der Akkustand in Prozent unter Einhaltung des Kommunikationsprotokolls an die Spiel-GUI geschickt werden kann. Wie im Kapitel „Mikrocontroller Firmware“ bereits erwähnt, können über die serielle Schnittstelle Werte zwischen 0 und 1023 empfangen werden. Da bei der OPV Schaltung von einer maximalen Eingangsspannung von 50V ausgegangen wurde, empfängt das Raspberry Pi den Wert 1023, wenn diese anliegt. Da jedoch drei 12V Akkus für die Stromversorgung zuständig sind, kann von einer Eingangsspannung von bis zu 36V ausgegangen werden. Liegen 36V an, sollte der empfangene Wert aufgerundet ca. 737 betragen, wie aus Rechnung (2) vorgeht. Wird also der Wert 737 empfangen, kann von einer Vollladung ausgegangen werden, sprich 100% Akkustand.

$$\frac{1023 * 36}{50} = 736,56 \quad (2)$$

Da es sich hierbei um Bleiakkumulatoren handelt, sollten diese spätestens bei 80% Entladung neu geladen werden. Als unterster Wert (0%) wurde daher 28,8V angenommen, um die Akkus nicht zu stark zu entladen. Ein Beispiel für eine Entladekurve für Blei-Säure-Akkus ist in Abbildung 23 zu sehen. Ab 80%iger Entladung fällt die Kurve stark ab.

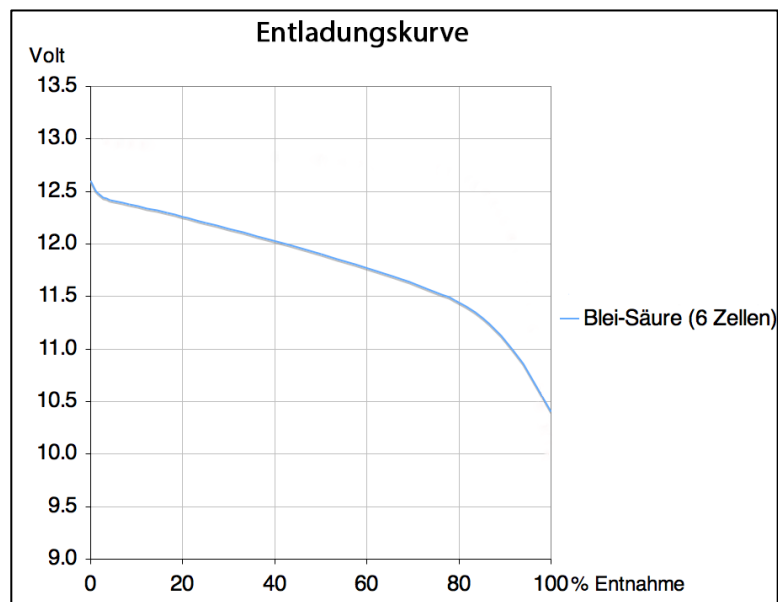


Abbildung 23: Entladungskurve (bearbeitet (Quelle bearbeitet: (Bremach-Reisemobile, 2013))

Zur Vereinfachung der Prozentberechnung wurde die Entladekurve linear angenommen. Berechnet wurde die Funktion für die Gerade mit den Punkten (0|590) als 0% bei einem

ADC – Wert von 590 und (100|737) 100% geladen bei einem ADC – Wert von 737. Die entsprechende Funktion ist in Abbildung 24 zu sehen.

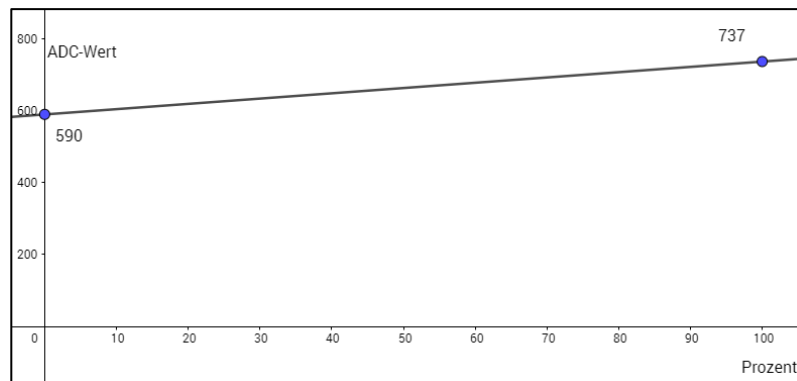


Abbildung 24: ADC-Wert/Prozent Kurve

### 3.2.5 Webserver / Web Frontend

Der Webserver wird durch ein simples HTML File konfiguriert und wird im Hauptprogramm am Raspberry Pi verwaltet. Dies geschieht über das Python Modul „Flask“. Der Flask Server lauscht auf dem Port 8765 für HTTP-Anfragen. Am Frontend des Webserver ist der Output der RasPi Cam zu sehen, der über den MJPG-Streamer live am Port 8080 übertragen wird. Die Auflösung beträgt 320x240 bei 15 FPS.

Damit der Webserver ebenfalls das Kommunikationsprotokoll verwenden kann, ist es notwendig gewesen zusätzlich das Python Modul „urllib“ zu importieren. Die Steuerung über den Webserver ist eventbasiert („POST“ Methode), sprich wenn einer der Buttons gedrückt wird, wird ein entsprechendes Datenpaket an das Python-Modul gesendet. Beispiele für Soundbefehle über das Web Frontend sind in Zeile 33f und Zeile 37f der Abbildung 25 zu sehen. Für den Befehl „Mood Sound“ (Zeile 34) wird der URL aus Zeile 33 („/sound%3Bplay%3Bdata%252F%3Bmood“) an das Python Hauptprogramm gesendet.

```

32 <td>
33 <form action="/sound%3Bplay%3Bdata%252F%3Bmood" method="POST">
34   <button type="submit" name="sound"> Mood Sound </button>
35 </br>
36 </form>
37 <form action="/sound%3Bplay%3Bdata%252F%3Bfabibox%252F%3Brandom" method="POST">
38   <button type="submit" name="sound2"> Random Sound </button>
39 </br>
40 </form>
41 </td>

```

Abbildung 25: Codeausschnitt HTML-File

Um im Python Hauptprogramm mit den empfangenen Daten weiterarbeiten zu können müssen diese erst mithilfe des „urllib“ Moduls decodiert werden. Der hierfür verantwortliche Funktionsaufruf ist in Zeile 270 in Abbildung 26 zu sehen. Die Funktion „unquote“ ersetzt

„%xx“ durch ihr Zeichenäquivalent, was in diesem Fall zu folgendem String führt: „sound;play;data;/mood“. Dieser kann nun mit dem Kommunikationsprotokoll verwendet werden (siehe Tabelle 3 im Kapitel „Kommunikationsprotokoll zwischen Raspberry Pi und Tablet“).

```
266 def reroute(changePin):
267     global ser
268     print(changePin)
269     try:
270         changePin = urllib.unquote(changePin).decode('utf8') #decode changePin to string
271     except:
272         print("changePin is not valid")
273     else:
274         chooseAction(changePin)
275     response = make_response(redirect(url_for('index')))
276     return(response)
```

Abbildung 26: Decodierung von Befehlen des Web Frontends

Der Webserver ermöglicht es den Roboter über einen Browser zu steuern. Ein clientseitiger Screenshot ist in Abbildung 27 zu sehen. Neben dem Kamera-Output ist auch der Akkustand sichtbar. Außerdem ist es möglich Fahrbefehle wie auf dem Touchscreen (siehe Kapitel: „Roboter Steuerung“ unter „Implementierte Spielfunktionen“) an den Roboter zu senden. Soundbefehle sind ebenfalls konfigurierbar und sind derzeit mit zwei Buttons belegt, die den letzten beiden Tasten der FAB-Box entsprechen (ebenfalls zu finden im Kapitel „Roboter Steuerung“ unter „Implementierte Spielfunktionen“).

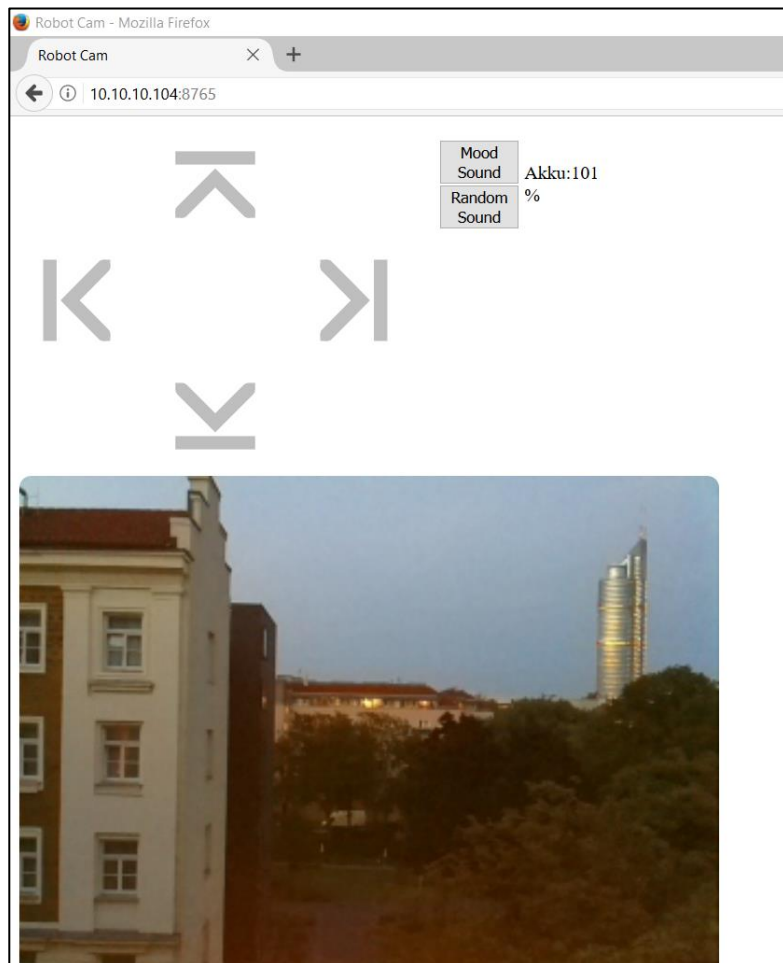


Abbildung 27: Screenshot des Webservers

### 3.2.6 Tablet-GUI User Applikationen

Die Spiel-GUI, die auf dem Tablet läuft, soll als Kommunikationsmittel zwischen Kind und Roboter dienen. Die GUI ist ebenfalls in Python 2.7 programmiert. Sie beinhaltet die Möglichkeiten vier verschiedene Spiele zu spielen, die in unterschiedlichen Formen den Roboter miteinbeziehen. Auch das Fernsteuern des Roboters, sowie das Verändern des Gesichts ist über die Spiel-GUI möglich (für die entsprechende Bedienoberfläche siehe Abbildung 33 im nachfolgenden Kapitel „Implementierte Spielfunktionen“). Beispiele für Gesichtskonfigurationen sind in Abbildung 28 zu sehen.

Um die Akkuanzeige stets zu ermöglichen, wurde ein Thread „BatToScreen“ implementiert. Dieser führt alle drei Sekunden ein Update des Bildschirmbereiches durch, in dem die Akkuanzeige zu sehen ist.

Der zweite Thread „ReadFromPi“ ist ab Anfang des Programmes bereit, Informationen vom Raspberry Pi zu empfangen. In diesem Thread wird das Kommunikationsprotokoll (siehe „Kommunikationsprotokoll“) für den Empfang von Daten verwendet. Wird ein Batteriestatus empfangen, wird die globale Variable „bat\_prozent“ aktualisiert. Neben dem Akkustand



können auch RFID Daten empfangen werden. Empfängt die GUI RFID Daten, ohne dass ein Spiel gestartet wird, werden die empfangenen Daten ignoriert und aus dem Puffer gelöscht. Ist ein entsprechendes Spiel aktiv, werden die RFID Daten in der Spielefunktion überprüft und weiterverarbeitet.

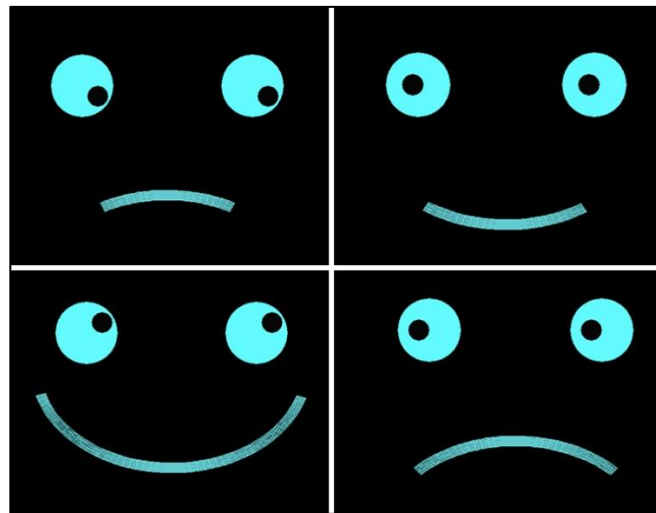


Abbildung 28: verschiedene Gesichtskonfigurationen

### 3.3 Implementierte Spielfunktionen

In diesem Kapitel werden die für diese Arbeit entworfenen Spielfunktionen der User Applikationen hinsichtlich ihrer Schwerpunkte untersucht. Eine Übersicht ist in Tabelle 6 zu sehen.

Funktion / Spiel	Hauptzweck
Spiel 1: „Objekte finden“	Motorik, Zuordnung von Farben und Formen
Spiel 2: „Emotionen erkennen“	Gefühle zu Situationen und Mimik zuordnen
Spiel 3: „Tiergeräusche zuordnen“	Tiergeräusche identifizieren
Spiel 4: „Tiergeräusche machen“	Tiere auswählen und Geräusche ertönen lassen
Roboter Steuerung	Steuern des Roboters durch Fahren und Gesichtsmanipulation
Webserver	Roboter Steuerung und Telepräsenz

Tabelle 6: Übersicht der implementierten (Spiel-) Funktionen

### 3.3.1 Motorik

Für die Unterstützung von Motorik und Unterscheidung von Farben und Formen dient das erste Spiel „Objekte finden“. Bei diesem Spiel wird das Kind über die Ausgabe eines Sounds aufgefordert einen bestimmten Baustein zu finden. Zusätzlich wird der gesuchte Baustein auch optisch am Windows Tablet dargestellt. Die Unterscheidung der Bausteine liegt hierbei bei Form und/oder Farbe. Die Bausteine sind, wie in Abbildung 29 zu sehen, mit RFID Stickern versehen und müssen von dem Kind nahe an den Roboter gehalten werden. Dieser ist entweder „erfreut“ und lobt das Kind, wenn es der richtige Baustein ist, oder ist beim falschen Baustein „traurig“, bzw. schüttelt sich und fordert das Kind auf, es erneut zu probieren. Der Roboter gibt also optisches und akustisches Feedback zum ausgewählten Baustein ab. Erweiterbar wäre dieses Spiel noch mit unterschiedlichen Abbildungen von Dingen (wie Stofftieren oder anderen Figuren). Eine Überlegung war, weitere Holzobjekte zu verwenden, jedoch wurde diese Idee verworfen, da es vorkommen kann, dass die Kinder die Objekte werfen, und durch das Werfen von harten Gegenständen andere Kinder gefährdet werden könnten.



Abbildung 29: Bausteine mit RFID Sticker

Diese Übung findet ebenfalls in der Montessorie Pädagogik Anwendung, welche auch besonders bei Kindern mit Autismus eingesetzt wird (Pichler & Pichler, 2014). Normalerweise wird dieses Spiel ohne Roboter durchgeführt, wobei das Kind von seiner Aufsichtsperson – Elternteil, Pädagogin, o.ä. – aufgefordert wird, einen Baustein mit bestimmter Farbe oder bestimmter Form herauszusuchen. Eine andere (schwierigere) Methodik beinhaltet zwei Eigenschaften, wie beispielsweise die Kombination von Farbe und Form (z.B. „roter Würfel“).

### 3.3.2 Emotionen erkennen

Kinder mit ASS haben große Schwierigkeiten Emotionen zu erkennen, vor allem im Bereich der Mimik und Gestik. Das zweite Spiel „Gefühle erkennen“ soll dabei helfen, dieser Schwäche entgegen zu wirken. Der Roboter erzählt eine Geschichte, bei der die beiden Protagonisten (das Mädchen Anna und der Bursche Elias) einen Tag im Zoo verbringen und drei verschiedene Gefühlszustände durchleben: glücklich, traurig und wütend. Bei dem

Aufkommen eines neuen oder geänderten Gefühlszustandes wird darauf deutlich eingegangen und das Kind wird aufgefordert, das richtige Gesicht, das der Emotion entspricht, auf dem Tablet anzuklicken. Auf dem Tablet selbst sind Bilder von Kindern in eben diesen drei Gefühlslagen zu sehen, aus welchen das Kind dann auswählen soll. Abbildung 30 zeigt einen Beispielbildschirm, wenn gerade die Gefühle von der Protagonistin Anna erkannt werden sollen. Über richtig oder falsch informiert der Roboter ebenfalls, wie beim ersten Spiel, sowohl über Sound als auch Gesicht. Wurde das falsche Gefühl ausgewählt, wiederholt der Roboter den letzten Abschnitt der Geschichte, kann aber unterbrochen werden, durch den Klick auf eine Antwort. Die verwendeten Kindergesichter wurden teilweise übernommen von (KidsSoup Inc. Und Tochtergesellschaften, 2004-2017) sowie von (Pixabay, 2017).

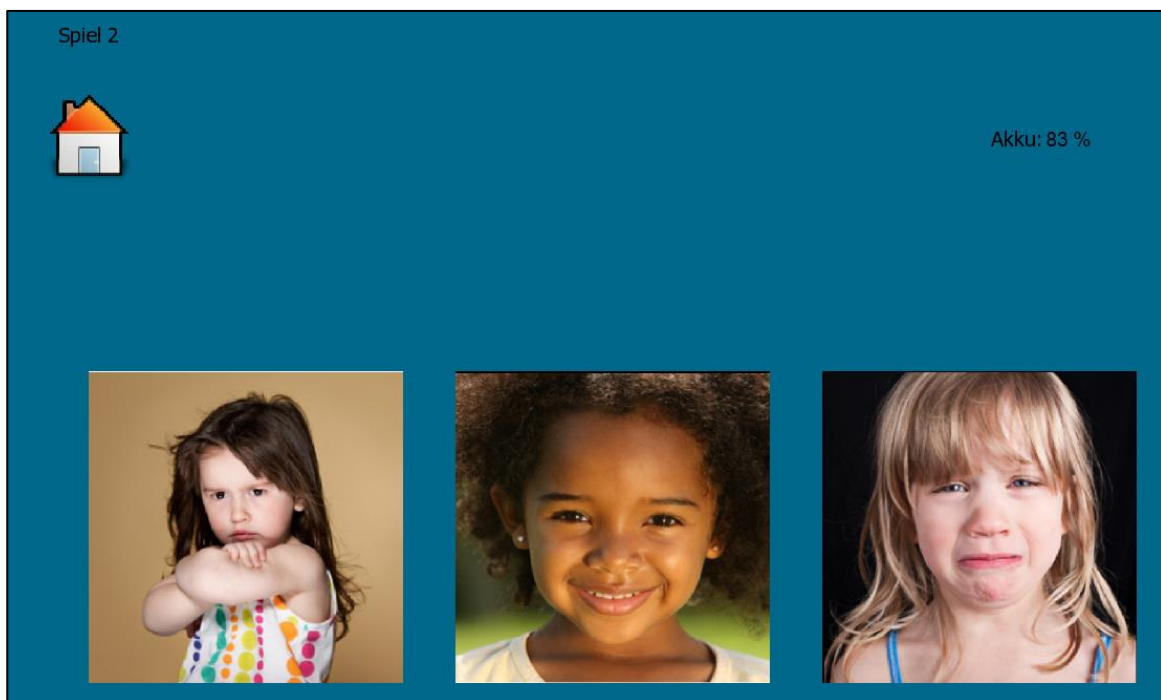


Abbildung 30: Screenshot von Spiel 2 „Gefühle erkennen“

### 3.3.3 Naturbezug

Der Bezug zur Natur, der Umwelt und den Tieren ist bei der kindlichen Entwicklung sehr wichtig. Durch das dritte Spiel „Tiergeräusche zuordnen“ und das vierte Spiel „Tiergeräusche machen“ soll dieser gefördert werden.

Im Spiel „Tiergeräusche zuordnen“ wählt der Roboter zufällig einen Sound zwischen sechs verschiedenen Tiergeräuschen (Hunde, Katze, Elefant, Kuh, Schafe, Vogel) und spielt diesen ab. Auf der Spiel-GUI sind die Tiere in ebenfalls zufälliger Reihenfolge abgebildet, und das Kind soll das richtige Tier zu dem abgespielten Geräusch auswählen. In Abbildung 31 ist zu sehen, wie die Tiere dargestellt werden, zwischen denen gewählt werden soll. Wie

auch bei den beiden vorherigen Spielen gibt der Roboter Feedback zur Auswahl. Auch hier findet eine Wiederholung (des Tiergeräusches) bei inkorrektter Zuordnung statt.

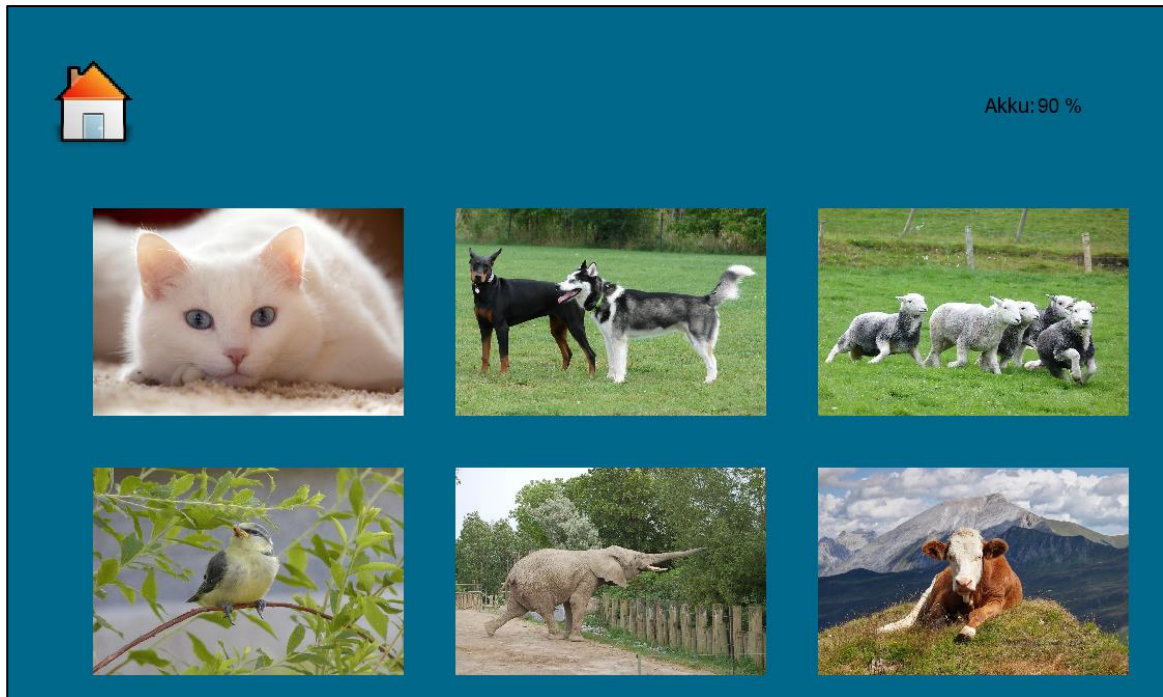


Abbildung 31: Screenshot von Spiel 3 „Tiergeräusche zuordnen“

Das vierte Spiel kann zur Übung für Spiel 3 dienen, eventuell für jüngere Kinder eingesetzt werden oder einfach als „lustige Abwechslung zwischendurch“ behilflich sein. Für dieses Spiel gibt es sechs vorgefertigte RFID Karten (in Form und Größe von Kreditkarten, siehe Abbildung 32), wobei jede Karte ein bestimmtes Tier symbolisiert. Auf der einen Seite ist die fotografische Abbildung des Tieres, auf der anderen Seite die passende Bezeichnung in schriftlicher Form, zu finden. Bei diesem Spiel braucht das Kind lediglich ein Tier auswählen, dessen Geräusch es hören möchte und hält dafür die entsprechende Karte an den Roboter.



Abbildung 32: RFID Karten

### 3.3.4 Roboter Steuerung

Die letzte (über die Spiel-GUI abrufbare) Funktion ist die Robotersteuerung. Hier können über den Touchscreen des Windows Tablets Fahrbefehle in Form von Schritten (Vorwärts, Rückwärts, Links, Rechts) sowie Gesichtsmanipulationen durchgeführt werden. Beim Gesicht kann folgendes bearbeitet werden: Die Position der Pupillen kann nach oben, unten, links, rechts verschoben und das Lächeln verstärkt oder zu einem negativen Ausdruck gebracht werden. Alle Befehle sind über symbolische Buttons aufrufbar, siehe dazu Abbildung 33.

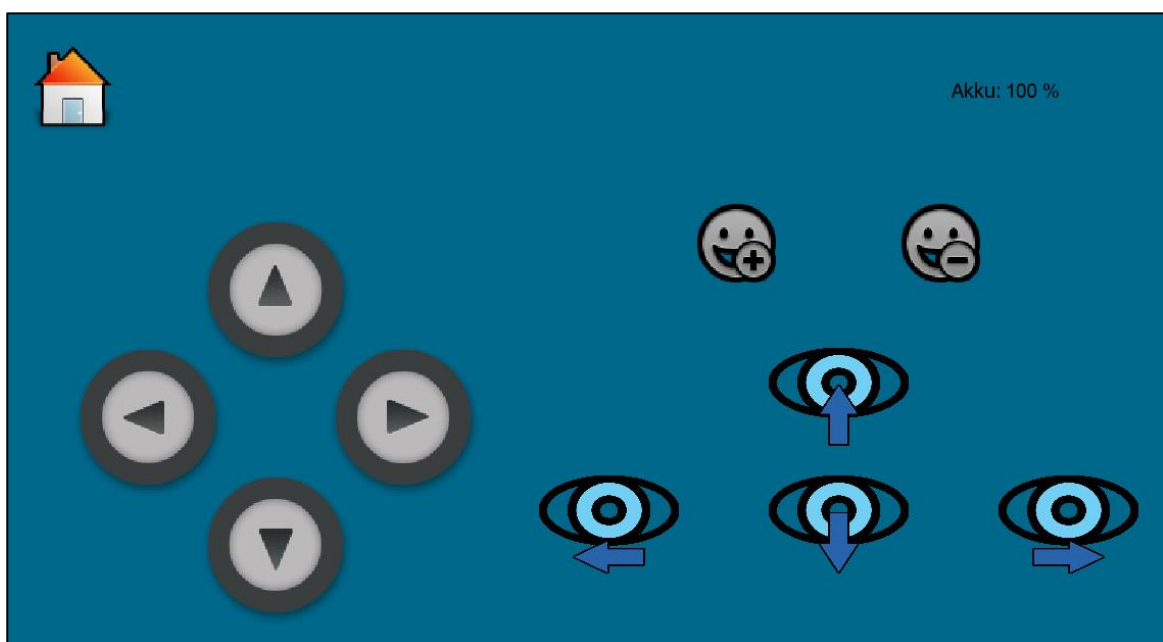


Abbildung 33: Screenshot der Roboter Steuerung am Tablet



Zusätzlich kann über den USB-Port die FABI-Box (Flexible Assistive Button Interface) (siehe Abbildung 34) angesteckt werden. Installations- und Anwendungshinweise sind im GitHub Repository verfügbar (Asterics, 2017). Die vier Tasten (weiß, grün, blau, rot) sowie die Stellungen des Joysticks werden als Tastaturinput interpretiert. Bei richtiger Konfiguration kann der Joystick, der acht verschiedene Positionen einnehmen kann, zur deutlich genaueren Steuerung des Roboters genutzt werden. Die Positionen sind Vorwärts, Rechts-Vorwärts, Rechts, Rechts-Rückwärts, Rückwärts, Links-Rückwärts, Links, Links-Vorwärts und sind anders als über das Tablet, keine Schritte, sondern, wie gewöhnlich, durchgehende Befehle. Solange nach vorne gedrückt wird, fährt der Roboter vorwärts – vorausgesetzt es gibt keine erkannten Hindernisse. Zwei der vier zusätzlichen Tasten der FABI-Box können ebenfalls für die Gesichtsmanipulation genutzt werden – genauer gesagt für das Vergrößern oder Verkleinern des Lächelns (weiß und grün). Einer der beiden übrigen Tasten (blau) sorgt für eine Soundausgabe am Roboter, welche abhängig vom gleichzeitig vorhandenen Gesichtsausdruck ist: Bei einem lächelnden *RoboFriend* ertönt ein helles Kichern, bei einem traurigen *RoboFriend* hört man entsprechend negative Laute. Der letzte Taster (rot) kümmert sich ebenfalls um eine Geräuschausgabe des Roboters. Wird dieser gedrückt, ertönt ein zufälliger Sound aus einem Pool kurzer, lustiger Geräusche, wie Telefonklingeln, Autohupen, Kichern und ähnliches. Diese beiden Soundtasten haben sich als hoher Unterhaltungsfaktor bewiesen.

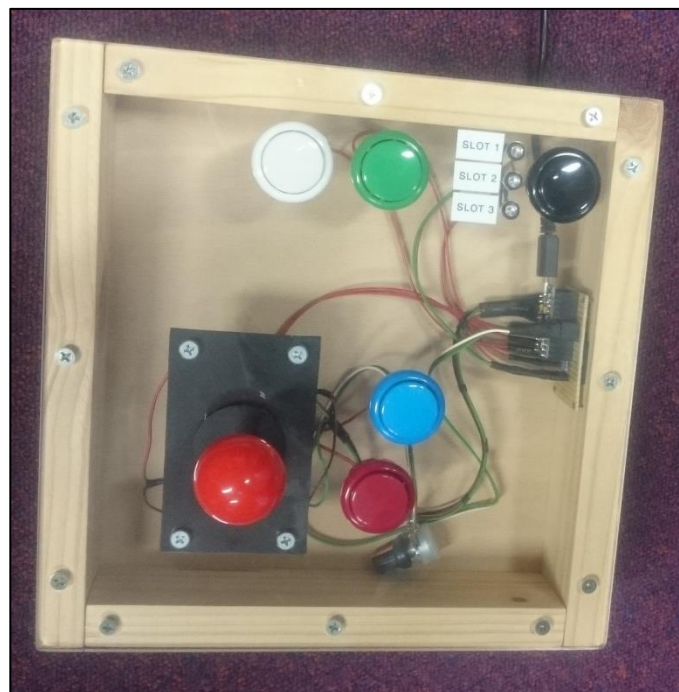


Abbildung 34: Verwendete FABI Box

## **4 Evaluierung und Ergebnisse**

Im Rahmen dieser Arbeit wurde eine Kooperation mit Thomas Bettinger (Bettinger, 2017), einem Lehrer an der Schule Paulusgasse in Wien eingegangen. Es handelt sich hierbei um ein „Zentrum für Inklusiv- und Sonderpädagogik“ (Paulusgasse, 2017) für Kinder mit erhöhtem Förderbedarf. Diese Schule wird von ca. 140 Kindern mit vollkommen unterschiedlichen körperlichen und geistigen Einschränkungen besucht. Im Zuge dessen, wurde der Fokus auf „Kinder mit erhöhtem Förderbedarf“ erweitert, wodurch die Entwicklung der Software beeinflusst wurde.

Insgesamt fanden zwei Testdurchläufe statt, wobei der erste Testdurchlauf lediglich mit einem Probanden durchgeführt wurde und der zweite Testdurchlauf auf zwei aufeinanderfolgende Tage stattgefunden hat, bei dem zusammengefasst sieben ProbandInnen teilgenommen haben.

### **4.1 Räumliche Umgebung**

Beide Testdurchläufe fanden an der Schule statt. Für diese standen zwei unterschiedliche Klassenzimmer zur Verfügung. ProbandInnen 1-4 testeten in Raum A, ProbandInnen 5-8 in Raum B. Um für genügend Freiraum für die Versuche und eine räumliche Abgrenzung zum restlichen Zimmer zu sorgen, wurden die Tische und Sessel verstellt.

Bei den in Raum A durchgeführten Tests stand das gesamte Klassenzimmer zur Verfügung und es waren lediglich für den Testdurchlauf relevante Personen anwesend. In Raum B waren ca. fünf weitere Personen (Lehrerin und SchülerInnen) präsent, die sich jedoch außerhalb des abgegrenzten Bereichs aufhielten.

### **4.2 Datenerfassung**

Für eine Genehmigung der Durchführung der Testdurchläufe an der Schule Paulusgasse mussten entsprechende Anträge des Wiener Stadtschulrates ausgefüllt und genehmigt werden (siehe Anhang A). Um die Anonymität der ProbandInnen zu bewahren wurden keine Namen erwähnt und die Gesichter auf den Fotos unkenntlich gemacht. Lediglich das Alter und die Diagnose wurden aus den persönlichen Daten gesammelt.

Zur Erfassung von Feedbackrelevanten Daten wurden in erster Linie Beobachtungen durchgeführt. In weiterer Folge wurden die ProbandInnen nach Möglichkeit nach ihren Eindrücken bzw. ihrer Meinung befragt (ProbandInnen 1-4).

### **4.3 Ablauf der Testdurchläufe**

Die ProbandInnen wurden vom Lehrpersonal der Schule ausgewählt und nacheinander einzeln mit der Roboterplattform vertraut gemacht, um diese anschließend ausführlich testen zu können. Lediglich Probandinnen 3 & 4 führten den Robotertest „gemeinsam“ durch, da sich Probandin 4 nicht alleine getraut hätte. Während der gesamten Testzeit werden die

Kinder von (mindestens) einer Person begleitet. Diese ist verantwortlich die entsprechenden nachstehenden Punkte zu erläutern und für allfällige Unklarheiten, Problemen oder ähnlichem zur Verfügung zu stehen.

Testablauf der/des Probandin/Probanden:

- Vorstellung der Roboterplattform und der grundlegenden Bedienung über das Windows Tablet
- ProbandIn anleiten Spiel 1 „Objekte finden“ zu starten, ihnen die Bausteine zur Verfügung stellen und erklären, wie die Bausteine an den Roboter gehalten werden müssen
- ProbandIn Spiel 1 üben lassen, bis jeder Baustein mindestens einmal richtig erkannt wurde
- ProbandIn anleiten Spiel 2 „Emotionen erkennen“ auszuprobieren und die Geschichte durchspielen lassen
- ProbandIn anleiten Spiel 3 „Tiergeräusche zuordnen“ auszuwählen und üben zu lassen, bis jedes Tiergeräusch mindestens einmal richtig zugeordnet wurde
- ProbandIn anleiten Spiel 4 „Tiergeräusche machen“ zu starten, die Tierkarten präsentieren und beschreiben, wie die RFID Karte an den Roboter gehalten werden muss
- ProbandIn jedes Tiergeräusch mindestens einmal auswählen lassen
- ProbandIn die Funktion „Roboter Steuerung“ zeigen, FABI-Box erklären und ProbandIn experimentieren lassen

Je nach verfügbarer Zeit, wird nach dem letzten Punkt der/dem Probandin/Probanden die Wahl gelassen, beliebige Funktionen erneut zu starten. An dieser Stelle wird die/der Probandin/Proband auch befragt, welche Übungen ihr/ihm am besten gefallen (hat).

## 4.4 Erster Testdurchlauf

Nach Vervollständigung der Implementierungen der Basisfunktionen wurde ein erster Testdurchlauf in der Paulusgasse durchgeführt. Wie geplant, wurde dort mit einem einzelnen Kind getestet, welches von Thomas Bettinger ausgewählt wurde. Hierbei handelt es sich um einen 12 Jahre alten Jungen (Proband 1), mit erhöhtem Förderbedarf und einer großen Technikaffinität. Ziel dieses ersten Testdurchlaufes war es, den *RoboFriend* erstmals unter realen Bedingungen und mit allen Funktionen von einem Kind auf Funktionsfähigkeit und Benutzerfreundlichkeit zu erproben.

### 4.4.1 Zwischenergebnis

Alle bestehenden Funktionen wurden nacheinander getestet, das heißt, Proband 1 spielte jedes Spiel und steuerte danach mit Benutzung der FABI-Box den Roboter. Danach wurden



alle Mängel, Wünsche und Verbesserungsvorschläge sowohl von Seiten des Kindes als auch von Thomas Bettinger besprochen und zusammengefasst.

Hierzu wurden folgende Inhalte diskutiert:

- GUI: Was muss geändert werden um eine höhere Benutzerfreundlichkeit für die Zielgruppe zu erzielen?
- Spielimplementierungen: Sind die Spielinhalte genügend aufbereitet?
- Roboter Steuerung: Fällt hier ein falsches oder ungewünschtes Verhalten des Roboters auf?
- Webserver: Ist das Webinterface ausreichend?
- allfällige Fehlfunktionen und Verbesserungsvorschläge

Das Kind hatte sehr viel Spaß und wollte kaum aufhören. Besonders die lustigen, zufälligen Geräusche sowie das Steuern des Roboters entpuppten sich schnell als Lieblingsbeschäftigung. Von den gewünschten Funktionen haben sich alle als funktionsfähig erwiesen. Lediglich bei der Robotersteuerung fiel folgendes auf: Beim Fernsteuern des Roboters konnten die Fahrbefehle, scheinbar aufgrund der Sensoren, nicht so ausgeführt werden, wie erwartet und gewünscht. Bei allen Fahrbefehlen, die den mittleren Sensor abfragten, wurde nach einem kurzen Ruck abrupt angehalten. Da die erkannten Sensorwerte ein (in der Realität nicht vorhandenes) Hindernis aufzeigten, wurde der Sensor durch Abkleben ignoriert (siehe unterer Bildbereich in Abbildung 35) (siehe dazu auch „Sensoren“). So konnte der Roboter ohne unerwarteten Störungen ferngesteuert werden. Aufgrund des nun verfälschten Sensorwertes, war die Hinderniserkennung deaktiviert und hätte nicht mehr zu einem automatischen Stehenbleiben des Roboters, wäre eine Behinderung vorhanden gewesen, führen können. Kleinere Fehler bei der Fernsteuerung innerhalb der Fahrbefehle wurden ebenfalls aufgedeckt.

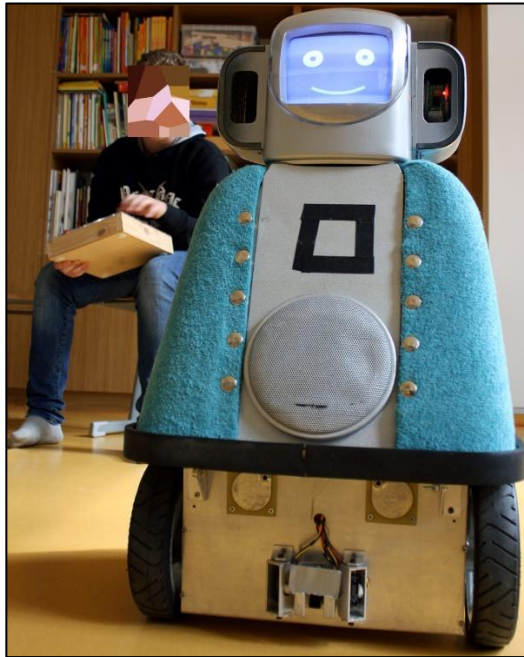


Abbildung 35: Proband 1 beim Testen

In Tabelle 7 ist eine Zusammenfassung des gesammelten Feedbacks zu sehen. Die bis zum zweiten Testdurchlauf durchgeführten Verbesserungen sind dabei gelb markiert. Wie unschwer zu erkennen ist, konnte das umfangreichste Feedback für das erste Spiel „Objekte finden“ (siehe Kapitel „Motorik“) gesammelt werden. Dies liegt daran, dass hier nicht nur am meisten, sondern auch die erste, Interaktion zwischen Kind und Roboter stattfand. Hauptsächlich fokussieren sich die Feedback-Punkte auf die Sound Files. Im Allgemeinen kann gesagt werden, dass der Roboter nicht genug Sound Files bietet: Intensiveres und vielseitigeres akustisches Feedback seitens des Roboters wurde gewünscht, genauso wie das Erweitern des Sortiments nicht nur auf Bausteine, sondern auf andere Gegenstände. Was sich ebenfalls beim Steuern des Roboters gezeigt hat, ist, dass sich bei der Ausführung mehrerer Fahrbefehle hintereinander Befehle stauen (in einer Queue gespeichert werden), und dadurch verzögert hintereinander abgearbeitet werden. Dies musste auch aus Sicherheitsgründen angepasst werden, da dadurch Fahrbefehle unerwartet ausgeführt wurden. Gelöst wurde dies auf der Mikrocontroller Firmware mittels einer Schleife, die solange Daten übertragen werden am seriellen Port liest. Erst danach wird das letzte empfangene Kommando ausgeführt und verhindert somit die Stauung mehrerer unmittelbar nacheinander ausgeführten Befehle.

<b>Spiel 1</b>	<b>Roboter Steuerung</b>	<b>allg. GUI</b>	<b>Webinterface</b>	<b>Sonstiges</b>
Sounderweiterungen bei falschem Baustein	Ev. PW sichern; nicht so leicht zugänglich machen	Buttons: Groß- & Kleinschreibung	Funktion „Sensorwert einstellen“	Lautstärken erhöhen
Sounderweiterungen bei richtigem Baustein	Mikrofon einbauen	„Zurück“ Button: Home Button (Symbol)	Funktion „Sensoren ON/OFF“	Fahrbefehle über-schreiben
Nach 5s das Gesuchte wiederholen	mit Mikrofon neue Sounds aufnehmen	zusätzlich Symbole auf Buttons		
andere Gegenstände wie Autos, Flugzeug, etc.	Buttons beschriften (FABI-Box) inkl. Symbolen			
Zusätzliche Sounds	Steuern als Belohnung bei Spiel			
Stimmung Levels				
Schwierigkeitsgrad (mit Ohren)				
Einführung mit Fotos				

Tabelle 7: Zusammenfassung der Wünsche, Anregungen und Verbesserungsvorschläge (gelbe Markierungen zeigen in dieser Arbeit umgesetzte Änderungen)

Zum Zeitpunkt des ersten Testdurchlaufs war die Robotersteuerung, deren Implementierung erst nach Verringerung des starken Fokus auf Kinder mit ASS entstanden war, noch genauso zugänglich wie die Spiele (gleich großer, beschrifteter Button). Dies wurde als Kritikpunkt genannt, da es passieren könnte, dass ein Kind den Roboter unbeaufsichtigt darauf los flitzen lässt. (An dieser Stelle ist anzumerken, dass es kein Ziel dieses Projektes ist oder war, eine Plattform zu schaffen, die Kinder vollkommen unbeaufsichtigt nutzen, daher lag auch die Idee für den Webserver/Telepräsenz nahe). Um dieser Gefahr entgegenzuwirken wurde die Robotersteuerung in die „Einstellungen“ (symbolisiert durch ein Zahnrad) versetzt (siehe dazu Abbildung 36). Das User-Interface musste ebenfalls noch einigen Änderungen unterzogen werden, um passend für die Zielgruppe gestaltet zu sein. Thomas Bettinger brachte hier relevante Punkte ein, die vor allem die Schreibweise und kindgerechte Darstellung (durch Symbole) betrafen. Das Ergebnis hierfür ist ebenfalls im Hauptmenü (Abbildung 36), oder auch in Untermenüs (Home-Button, links oben), wie beispielsweise in Abbildung 33 zu sehen.

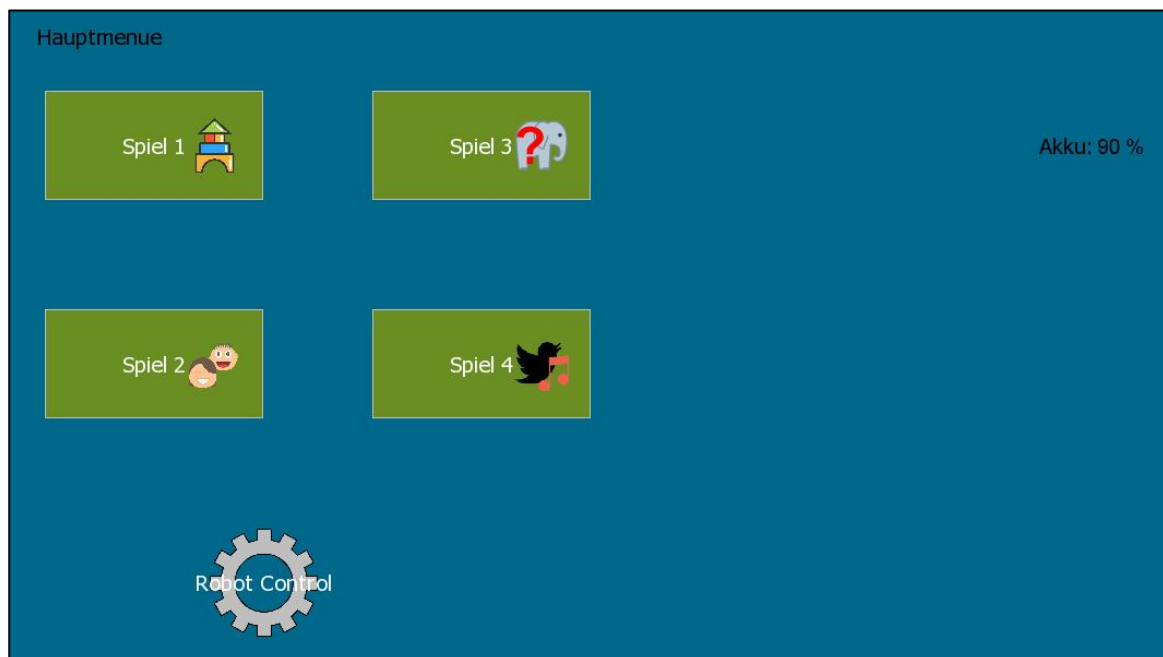


Abbildung 36: Überarbeitetes Hauptmenü

Aus zeitlichen Gründen konnte nicht allen Punkten nachgegangen werden, welche jedoch nun eine gute Grundlage für nachfolgende Arbeiten liefert. Dazu mehr in „Ergebnisse“ und „Zusammenfassung und Ausblick“.

## 4.5 Zweiter Testdurchlauf

Nach der Implementierung von einigen Kritikpunkten, sowie anderen Änderungen wurde eine zweite Evaluierung mit Kindern durchgeführt. Ziel des zweiten Durchlaufs war es, den Roboter mehreren Kindern nacheinander zu überlassen und zu evaluieren, wie diese, nach Einarbeitung des Feedbacks aus dem ersten Testlauf, mit ihm zurechtkommen.

Wichtige Fragen waren hierbei:

- Wie verständlich ist die Funktionsweise des Roboters für die Kinder?
- Wie verhalten sie sich gegenüber dem Roboter?
- Wie gut/schlecht kommt welches Spiel an und warum?
- Wie schnell vergeht die Lust, sich mit dem Roboter zu beschäftigen?
- Gesamteindruck

Bei diesem Testdurchlauf kamen insgesamt sieben Kinder innerhalb von zwei Tagen mit dem Roboter in Kontakt.

Eine Übersicht der Kinder, die den Roboter beim zweiten Testdurchlauf ausprobiert haben, ist Tabelle 8 zu entnehmen. Proband 2, sowie Probandin 3 und Probandin 4, testeten die Roboterplattform am ersten Tag, die übrigen ProbandInnen am darauffolgenden.

Proband	Alter	Diagnose	Auffälligkeiten
Proband 2	14	ASS	technikaffin
Probandin 3	15	erhöhter Förderbedarf	interessiert, ruhig
Probandin 4	14	erhöhter Förderbedarf	schüchtern, ängstlich
Probandin 5	12	wahrscheinlich ASS	interessiert
Proband 6	11	erhöhter Förderbedarf	scheinbare Verständnisprobleme
Probandin 7	15	Trisomie 21	aggressiv
Probandin 8	14	erhöhter Förderbedarf	ruhig

Tabelle 8: Übersicht der ProbandInnen

#### 4.5.1 Zwischenergebnis

Für Proband 2 waren die Spiele „ok“, er hat alle Spiele erfolgreich „durchgespielt“ und viel Zeit anschließend damit verbracht den Roboter fernzusteuern, was ihm auch am meisten Spaß gemacht hat, das fand er „super“. (Siehe Abbildung 37) Proband 2 war, wie Proband 1, sehr wissbegierig und fragte viel über den Roboter und Technik.



Abbildung 37: Proband 2 beim Steuern mit der FAB-Box

Probandin 3 zeigte ein schnelles Auffassungsvermögen und zeigte Freude an der Verwendung der Spiele. An der Steuerung des Roboters fand sie gleich starken Gefallen

wie an den Spielen selbst. Jedes Spiel wurde mehrmals von ihr gespielt (Siehe Abbildung 38). Ihre Sitznachbarin, Probandin 4, hingegen fürchtete sich vor dem Roboter und traute sich weder eine alleinige Steuerung zu noch selbständig mit ihm zu spielen. Probandin 4 verwendete stets immer eine Person als Zwischenkommunikationsmittel, um die Spiele des Roboters nutzen zu können. Probandin 3 hat Probandin 4 dazu motiviert die Bausteine zu finden, das Tiergeräusch zu erraten, die Emotion richtig zuzuordnen und Ähnliches. Probandin 4 erkannte das meiste richtig, traute sich jedoch nicht selbst den Baustein an den Roboter zu halten oder auf das Touchpad zu klicken. Mit der Zeit traute sie sich immer näher und benutzte zum Beispiel die Hand einer anderen Person, um auf dem Touchpad die entsprechende Antwort auszuwählen.



Abbildung 38: Probandin 3 beim Testen

Bereits beim ersten Testdurchlauf begegnete Probandin 5 dem Roboter auf dem Gang, wo sie ein sehr überschwängliches Verhalten ihm gegenüber aufwies, in dem sie ihn gleich mit einem „Klaps“ begrüßt hat. Andere Roboter für Kinder mit ASS sind bereits vorbereitet auf eventuellen körperlichen Kontakt, sodass sich weder der Roboter noch das Kind verletzen können. Diese Eigenschaft weist der hier verwendete *RoboFriend* leider nicht auf und kann auch nicht ohne weiteres dahingehend angepasst werden. Während des zweiten Testdurchlaufs hatte die Probandin 5 eine ruhigere Laune und ging alle Spiele inklusive der Steuerung durch, ohne größere Schwierigkeiten zu haben.

Bei Proband 6 konnte davor beobachtet werden, dass er sich gerne mit dem iPad beschäftigte, weshalb das Nachfolgende eventuell auch als nachvollziehbar erachtet werden könnte. Anstatt das Tablet als Interaktionsmittel mit dem Roboter zu nutzen, verbrachte er

die Zeit damit, unentwegt auf dem Touchscreen und den dortigen vorhandenen Buttons zu klicken. Da sich der Screen jedoch nicht jedes Mal sofort ändert (sondern erst Feedback des Roboters abwartet) sind die Klicks gespeichert und gereiht worden. Aufgrund dieses Befehlstaues kam es zu folgender Fehlermeldung von Windows: „python.exe reagiert nicht mehr“. Zwei Möglichkeiten stehen an dieser Stelle zur Verfügung: zu warten oder das Programm zu beenden. Die Auswahloptionen wurden erprobt und zeigten, dass beim Auswählen des Wartens das Tablet nach einigen Sekunden ohne weiteren Problemen, zur Bedienung des *RoboFriends* verwendet werden konnte. Was dem Jungen, zusätzlich zum Drücken, auch Spaß machte, war das Steuern des Roboters mit dem Joystick.

Probandin 7 hatte ein etwas auffälligeres aggressives Verhalten und zeigte dies während der Steuerung durch stärkeres Rütteln an der FABI-Box, diese hielt jedoch erfolgreich stand. Es war deutlich zu erkennen, dass ihr die Steuerung nicht leichtfiel. Bei Probandin 8 konnten ebenfalls Schwierigkeiten bei der Steuerung des Roboters festgestellt werden.

Es liegt die Vermutung nahe, dass die Steuerung aus der Perspektive der Mädchen stattfand, anstatt aus der des Roboters. Daher verhielt sich der *RoboFriend* selten so, wie es die beiden Probandinnen wollten oder erwarteten.

Insgesamt gab es beim zweiten Testdurchlauf keinerlei Komplikationen, Schwierigkeiten, Fehlfunktionen oder Fehlverhalten. Die einzige Problematik, die beim Roboter erkannt wurde, war der mittlere Sensor. Der Grund dafür, dass die Werte des mittleren Sensors ein Hindernis indizieren, liegt an der Bodenbeschaffenheit des Raumes: Ein glatter, heller Boden, der durch den Lichteinfall spiegelt und so die Messung des Sensors verfälscht. Dieser dürfte zusätzlich etwas zu weit nach unten geneigt sein. Um dieses Problem zu beheben, könnte versucht werden, die Neigung des Sensors zu verändern, oder zukünftig auf Teppichen zu arbeiten.

Die zusätzliche Erkenntnis darüber, dass es Kinder gibt, die Schwierigkeiten mit dem Steuern des Roboters haben, sollte genutzt werden, um alternative Steuerungsmöglichkeiten in Betracht zu ziehen.

## **4.6 Ergebnisse der Evaluierung mit Proband/innen**

Im vorangehenden Kapitel „Zweiter Testdurchlauf“ wird bereits auf einige wichtige Ergebnisse eingegangen. Zusammenfassend ist jedoch klar zu sagen, dass der Roboter bei fast allen Kindern sehr gut ankommt. Je mehr Zeit die Kinder mit dem Roboter verbracht haben, desto mehr freundeten sie sich mit dem *RoboFriend* an. Die Verbesserung in der Steuerung des Roboters war essentiell und brachte eine ganz andere Fahrqualität mit sich. Das Steuern des Roboters machte allen Kindern Spaß, unabhängig von Alter oder Einschränkungen. Lediglich eine Testerin wurde etwas aggressiv, da sie die Steuerung nicht unter Kontrolle hatte; sie wollte jedoch nicht etwas „zerstören und weggehen“, sondern viel mehr ihrer Frustration freien Lauf lassen und es zum Funktionieren bringen. Der Tabelle 9

kann eine Übersicht der Zufriedenheit der ProbandInnen entnommen werden. Die Bewertung der Spiele wurde mittels Schulnotensystem durchgeführt, mit 1 als Sehr Gut und 5 als Sehr Schlecht. Zusätzlichen wurden Zwischenwerte um die mittlere Note (3, Mittel) benutzt. Gewählt wurden die Bewertungen anhand der Gespräche mit den ProbandInnen sowie der Beobachtungen.

ProbandIn→ Funktion↓	P1 (m)	P2 (m)	P3 (w)	P4 (w)	P5 (w)	P6 (m)	P7 (w)	P8 (w)	Ø
<b>Spiel 1</b>	3	2,5	1	2,5	2	3,5	4	2	<b>2,56</b>
<b>Spiel 2</b>	3,5	3	1	2,5	2	3,5	4	2	<b>2,69</b>
<b>Spiel 3</b>	2,5	2,5	1	2,5	2	3	3	2	<b>2,31</b>
<b>Spiel 4</b>	3	3	1	3	2	3	3	2	<b>2,5</b>
<b>Steuerung</b>	1	1	1	2,5	2	2	5	3	<b>2,19</b>
<b>Ø</b>	<b>2,6</b>	<b>2,4</b>	<b>1</b>	<b>2,6</b>	<b>2</b>	<b>3</b>	<b>3,8</b>	<b>2,2</b>	
<b>Legende:</b> sehr gut (1), gut (2), mittel bis gut (2,5), mittel (3), mittel bis schlecht (3,5), schlecht (4), sehr schlecht (5)									

Tabelle 9: Zufriedenheit der ProbandInnen mit den getesteten Funktionen

Da bereits die Erfahrung gemacht wurde, dass die Abstandssensoren bei unterschiedlichen Bodenbeschaffenheiten keine zuverlässigen Werte liefern, wäre eine Idee, im Webinterface einstellen zu können, ob die Sensoren überhaupt verwendet werden sollen (ON/OFF Switch). Weiteres könnte überdacht werden, auch verschiedene Schwellwerte zu implementieren, die über da Web-Interface eingestellt werden könnten. Für die schrittweisen Fahrbefehle, die über den Touchscreen ausgewählt werden, wäre eine Anpassung der Schrittgröße sinnvoll.

Die Spiele kamen insgesamt bei den weiblichen Probandinnen besser an als bei den männlichen. Das erste Spiel „Objekte finden“ war bei der Suche nach Farben sehr einfach für die Kinder, die Frage nach der Form erwies sich jedoch in manchen Fällen als eine Herausforderung. Was sich bei allen Kindern als Problem erwies, war das Objekt „Zylinder“. Erst nach wiederholtem Zeigen, welcher Baustein der Zylinder ist, konnten die Kinder diesen wiedererkennen, was nicht unbedingt im Sinne dieses Projektes ist. Der Roboter soll dazu dienen, die Kinder zu unterstützen und ihnen Lernspiele zu präsentieren, die Spaß machen. Ein Nachteil des ersten Spieles ist, dass relativ wenige Objekte verwendet werden, was zu schneller Langeweile führt. Ein größerer Mangel bei dem ersten Spiel ist auch noch die Einführung. Dafür wäre es gut, wenn am Tablet eine Erklärung mit Fotos zu sehen wäre. Aus zeitlichen Gründen konnte das in dieser Arbeit nicht mehr umgesetzt werden.

Das zweite Spiel „Emotionen erkennen“ kam bei den Kindern gut an. Einmal die Geschichte durchzuspielen hat ihnen Spaß gemacht, aber dieselbe Geschichte ein zweites Mal durchzugehen, reizte sie eher nicht, vor allem, wenn beim ersten Durchgang bereits alles



richtig beantwortet wurde. Hier wäre es sinnvoll, mehrere Geschichten zu vertonen und ihnen Titel zu geben, um im Menü zwischen den Geschichten auswählen zu können. Diese sollten außerdem unterschiedlich lang und mit unterschiedlichen Schwierigkeitsstufen gestaltet werden, wie zum Beispiel durch die Verwendung von mehr als nur drei Emotionen. Das dritte Spiel „Tiergeräusche zuordnen“ kam am besten an. Die Tiergeräusche wurden als lustig empfunden, die Kinder können die Geräusche selbst nachahmen und lernen schnell, welches Tier wie klingt. Die meisten Kinder hatten keine Schwierigkeiten damit. Bei Proband 6 ist aufgefallen, dass ihm manche Tiere nicht wirklich bekannt waren, bzw. das Geräusch für ihn nicht zuordenbar war, dies betraf vor allem den Elefanten und die Kuh. Da das vierte Spiel „Tiergeräusche machen“ ausschließlich zur Unterhaltung dient, konnten ähnliche Ergebnisse beobachtet werden. Es macht den Kindern einerseits Spaß, die Tiergeräusche zu hören, andererseits auch selbst bestimmen zu können, welches Geräusch als nächstes abgespielt werden soll. Hierbei beinhaltet der Spielablauf auch den Aufbau von direkten Kontakt der Kinder mit dem Roboter, um ihm die entsprechende Karte an den Bauch zu halten.

## 5 Zusammenfassung und Ausblick

Aufgrund der ständig verbesserten Technologien kann sich die Therapierobotik stets weiterentwickeln. Dank Open Source und Mikrocontrollern wie Arduino und dem Einplatinencomputer Raspberry Pi gibt es immer mehr kostengünstige Möglichkeiten, Roboter zu entwickeln. Um sinnvolle Therapieroboter zu designen, ist eine Zusammenarbeit mit Therapeuten, Angehörigen, Betroffenen oder anderen erfahrenen Bezugspersonen unumgänglich. Derzeit ist die Therapierobotik noch ein Forschungsgebiet, das noch vielen Herausforderungen gegenübersteht und ständiger Weiterentwicklung bedarf.

Die Ergebnisse zeigen deutlich, dass die Roboterplattform *RoboFriend* hohes Potenzial hat, eine sinnvolle Lieblingsbeschäftigung für Kinder mit ASS bzw. erhöhtem Förderbedarf zu werden. Bei regelmäßigem Gebrauch können Ziele, wie Verbesserungen im Erkennen von Emotionen, sicherlich erreicht werden. Auf der anderen Seite zeigen die Ergebnisse auch, dass es, aufgrund des großen Spektrums an Beeinträchtigungen der Zielgruppe, beinahe unmöglich ist einen „Allround-Roboter“ für Kinder mit ASS zu entwickeln.

Einige Anregungen für Erweiterungen wären zum Beispiel der Einbau eines Mikrofons um die Sound-Datenbank möglichst einfach zu erweitern bzw. Live-Audiofeedback zum Videostream hinzuzufügen. Dies würde die zusätzliche Möglichkeit bieten, beim Fernsteuern zu hören, was in der Umgebung des Roboters gesprochen wird. Weiter könnten verschiedene Reaktionen bei richtigen und falschen Antworten durch zufällig ausgewählte Sounds erzielt werden. Um das Spiel „Objekte finden“ spannender und abwechslungsreicher zu gestalten, müssen zusätzliche Objekte miteinbezogen werden. Welche dafür geeignet sind, sollte unbedingt genauer mit einem Experten wie Thomas Bettinger besprochen werden, um unter anderem zu vermeiden, dass für die Kinder gefährliche Gegenstände verwendet werden.

Verschiedene Stimmungs-Modi einzubauen, wie etwa „lieb-vorsichtig“ oder „aggressiv-penetrant“ würden die Interaktionsmöglichkeiten des Roboters erhöhen. Es gäbe durchaus Kinder, die einen großen Spaß daran hätten, wenn der Roboter eine aggressive oder penetrante Persönlichkeit zeigen könnte und statt einem lieben „Bitte bring mir blau“ ein „Bringst du mir jetzt endlich einen blauen Baustein?!!“ ertönt. Zusätzlich wäre die Einbindung der LED Ohren ein erdenkbares, künftiges Feature. Sie könnten nicht nur verwendet werden, um Feedback zu Antworten zu geben, sondern auch um Stimmungen zu repräsentieren oder zum Beispiel die Schwierigkeit bestimmter Spiele erhöhen (beispielsweise verlangt der Roboter nach einem blauen Baustein, währenddessen seine Ohren aber rot leuchten). Zusätzliche Funktionen für das Einstellen von Sensorwerten sind ebenfalls erstrebenswert. Durch die Verwendung von OpenCV<sup>16</sup>, einer Bildverarbeitungssoftware, könnte zum Beispiel eine Gesichtserkennung verwendet werden, damit der Roboter zusätzliche Interaktionsfähigkeiten erhält (z.B. Kinder erkennen und anhand ihrer Gesichter

---

<sup>16</sup> <http://opencv.org/>

entsprechend reagieren, sie grüßen oder anders ansprechen). Werden zusätzliche IR Sensoren eingebaut, könnten weitere Funktionen, wie z.B. Line-Following, realisiert werden. Im Gesamten zeigen die Ergebnisse dieser Arbeit, dass durch die *RoboFriend* Plattform und durch die implementierten Funktionen für Interaktion (insbesondere RFID, Bewegungsfernsteuerung, emotionale Gesichtsausdrücke und akustisches Feedback) sowie die modulare Gestaltung von Hardware und Software (Tablet-GUI als Nutzerinterface, WebServer mit Live-Videostream und Python-Middleware mit flexiblem Kommunikationsprotokoll) viele sinnvolle Möglichkeiten geschaffen worden sind, bestehende Therapiespiele zu verbessern und weitere Therapiespiele effektiv zu erstellen und diese testen zu können. Dies soll auch durch nachfolgende Arbeiten, in denen die Roboterplattform erweitert und evaluiert wird, geschehen. Hierzu sollte von Beginn an der Schwerpunkt auf Personen mit erhöhtem Förderbedarf gelegt werden, um möglichst vielen Kindern in der „Schule Paulusgasse“ und anderen Einrichtungen die Möglichkeit zu bieten, von der Roboterplattform zu profitieren.

---

# Literaturverzeichnis

- Amazon, 2017. *10X 125KHz ID Tag Sticker Wasserdicht Klebe EM4100 Access Control*. [online] <[https://www.amazon.de/dp/B00MO9K5MS/ref=cm\\_sw\\_r\\_other\\_apa\\_z17SzbM5D9NKC](https://www.amazon.de/dp/B00MO9K5MS/ref=cm_sw_r_other_apa_z17SzbM5D9NKC)> [Zugriff am: 05.09.2017]
- American Honda Motor Co. Inc, 2017. *ASIMO*. [online] <<http://asimo.honda.com/>> [Zugriff am: 08.10.2017]
- Arduino, 2016. *A runningAverage Class for Arduino*. [online] <<http://playground.arduino.cc/Main/RunningAverage>> [Zugriff am 08.11.2016]
- Asterics, 2017. *FABI*. [online] <<https://github.com/asterics/FABI>> [Zugriff am: 12.09.2017]
- Atmel Corporation, 2009. *8-bit AVR Microcontroller with 64/128K Bytes of ISP Flash and USB Controller*. [online] <<https://www.pjrc.com/teensy/at90usb1286.pdf>> [Zugriff am: 01.07.2016]
- Autismus-Kultur, 2006. [online] <<http://autismus-kultur.de/>> [Zugriff am: 10.09.2017]
- Bach, H., 2006. *Wer tauscht mit mir? Kommunikationsförderung autistischer Menschen mit dem "Picture Exchange Communication System"*. Stuttgart: Ibidem. ISBN: 978-3-89821-692-0
- Baird et.al., 2006. *Prevalence of disorders of the autism spectrum in a population cohort of children in South Thames: the Special Needs and Autism Project (SNAP)*. London: Lancet
- Bettinger, 2017. [online] <<https://www.bettinger.at/bettinger/>> [Zugriff am: 01.08.2017]
- Bölte, S., 2009. *Autismus, Spektrum, Ursachen, Diagnostik, Interventionen, Perspektiven 1.* Auflage. Bern: Hans Huber.
- Bonarini A., et al, 2016: *Playful interaction with Teo, a Mobile Robot for Children with Neurodevelopmental Disorders*. Portugal, 2016: Proceedings of the 7th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion.
- Bremach-Reisemobile, 2013. *LiFePO-Akkus*. [online] <<http://www.bremach-reisemobile.org/technik/technik-nach-themen/fahrzeugelektrik/lifepo-akkus>> [Zugriff am: 10.09.2017]

- Burga, C.G., Lum P.S., Shor, P.C, Van der Loos M., 2000. *Development of robots for rehabilitation therapy: The Palo Alto VA/Stanford experience*. Volume 37, No. 6. Journal of Rehabilitation Research and Development.
- Burton, A., 2013: *Dolphins, dogs, and robot seals for the treatment of neurological disease*. Volume 12, No. 9. The Lancet Neurology: Elsevier
- Code Mercenaries Hard- und Software GmbH, 2013. *IO-Warrior: Generic universal I/O Controller for USB*. [online] <[http://www.codemerics.com/downloads/iowarrior/IOW24\\_40\\_Datasheet.pdf](http://www.codemerics.com/downloads/iowarrior/IOW24_40_Datasheet.pdf)> [Zugriff am: 12.09.2017]
- Dautenhahn, K., Nehaniv, C.L, Walters, M.L., Robins, B., Kose-Bagci, H., Mirza, N.A, Blow, M, 2009. *KASPAR - a minimally expressive humanoid robot for human-robot interaction research*. Volume 6, Issue 3-4. Journal: Applied Bionics and Biomechanics. DOI: 10.1080/11762320903123567
- Dautenhahn, K., 2017. *Prof. Dr. Kerstin Dautenhahn*. [online] <<https://homepages.herts.ac.uk/~comqkd/>> [Zugriff am: 16.08.2017]
- Daynurseries, 2016. *KASPAR the friendly robot teaches children human feelings*. [online] <<https://www.daynurseries.co.uk/news/article.cfm/id/1577637/Kaspar-the-friendly-robot-teaches-children-human-feelings>> [Zugriff am: 12.09.2017]
- Diehl, J., et al, 2012. *The clinical use of robots for individuals with Autism Spectrum Disorders: A critical review, Research in Autism Spectrum Disorders*. Volume 6, Issue 1. Journal: Research in Autism Spectrum Disorders: Elsevier.
- Eggshell Robotics, 2009. *Uncanny-Valley: Models wanted for 3d-Facescanner Artproject*. [online] <<http://www.eggshell-robotics.com/blog/336-uncanny-valley-models-wanted-for-3d-facescanner-artproject>> [Zugriff am: 08.10.2017]
- Electrical Engineering, 2012. *How to read high voltages on microcontroller?* [online] <<https://electronics.stackexchange.com/questions/42710/how-to-read-high-voltages-on-microcontroller>> [Zugriff am: 01.02.2017]
- Finkenzeller, K., 2010. *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication*. 3. Auflage. Chichester: Wiley
- Frizing, 2017. [online] <<http://fritzing.org/home/>> [Zugriff am: 08.09.2017]
- Gelsomini et al, 2017. *Puffy - an Inflatable Mobile Interactive Companion for Children with Neurodevelopmental Disorders*. Denver: Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems.

- Hackl, F., 2016. *RoboFriend – An interactive multi-purpose robotic platform*.
- Häußler, A., et. al., 2003. *SOKO Autismus: Gruppenangebote zur Förderung Sozialer Kompetenzen bei Menschen mit Autismus*. Dortmund: modernes lernen.
- Häußler, A., 2008. *Der TEACCH Ansatz zur Förderung von Menschen mit Autismus*. Dortmund: modernes lernen.
- Howlin, P., Baron-Cohen, S. & Hadwin J.A., 1999. *Teaching Children with Autism to Mind-Read*. Chichester: Wiley.
- Huijnen C., Lexis M. & de Witte L., 2016. Matching Robot KASPAR to Autism Spectrum Disorder (ASD) Therapy and Educational Goals. Volume 8, Issue 4, S.445-455. International Journal of Social Robotics
- ICD, 2017. 2017 ICD-10-CM Diagnosis Code F84.0. [online] <<http://www.icd10data.com/ICD10CM/Codes/F01-F99/F80-F89/F84-/F84.0>> [Zugriff am: 10.09.2017]
- IEE, 2005. *Radio Frequency Identification Device Technology (RFID)*. Stevenage: The IEE.
- ISO/IEC 18000-2:2009, 2009. *Information technology -- Radio frequency identification for item management -- Part 2: Parameters for air interface communications below 135 kHz*. 2. Edition
- IT Wissen, 2016. *DC/DC-Wandler*. [online] <<http://www.itwissen.info/DC-DC-Wandler-DC-to-DC-converter.html>> [Zugriff am: 07.10.2017]
- KidsSoup Inc. Und Tochtergesellschaften, 2004-2017. [online] <<http://www.kindersuppe.de/>> [Zugriff am: 15.03.2017]
- Kramer et al, 2012. *Hacking the Kinect*. 1. Ausgabe: apress
- MacDorman, K.F., 2005a. *Androids as an experimental apparatus: Why is there an uncanny valley and can we exploit it?* In CogSci-2005 Workshop: Toward Social Mechanisms of Android Science, pages 106– 118.
- MacDorman, K.F., Minato T., Shimada M., Itakura, S., Cowley, S., Ishiguro, H., 2005b. *Assessing human likeness by eye contact in an android testbed*. In Proc. VII Ann. Meeting of the Cognitive Science Society
- MacDorman, K.F. & Ishiguro, H., 2006. *The uncanny advantage of using androids in social and cognitive science research*. Interaction Studies, 7(3), 297-337.
- MIT, 2017. *Kismet*. [online] <<http://www.ai.mit.edu/projects/humanoid-robotics-group/kismet/kismet.html>> [Zugriff am: 08.10.2017]

- Mori, M., 1970. *Bukimi no tani*. [englische Übersetzung „*The Uncanny Valley*“ von MacDorman, K.F. & Kageki N., 2012, online] <<https://spectrum.ieee.org/automaton/robotics/humanoids/the-uncanny-valley>> [Zugriff am: 08.10.2017]
- Open Source Initiative, 2017. [online] <<https://opensource.org/>> [Zugriff am: 09.09.2017]
- PARO Robots U.S., Inc., 2017. [online] <<http://www.parorobots.com/>> [Zugriff am: 10.09.2017]
- Paulusgasse, 2017. [online] <<http://paulusgasse.at/>> [Zugriff am: 01.08.2017]
- Pichler, H. & Pichler, M., 2014. *Montessori-Praxis: Leichter Lernen durch „Sehen-Fühlen-Erkennen“*. Sensory-Nr. 580, 8. Auflage. Pullach/München: Sensor Verlag Pichler GmbH.
- Pixabay, 2017. *Beautiful Free Images*. [online] <<https://pixabay.com/>> [Zugriff am: 15.03.2017]
- Platus Learning Systems GmbH, 2017a. [online] <<http://www.platus.at/>> [Zugriff am: 10.09.2017]
- Platus Learning Systems GmbH, 2017b. *TOM Lernsystem*. [online] <<http://tom.platus.at/>> [Zugriff am: 10.09.2017]
- Python Software Foundation, 2001-2017a. [online] <<https://www.python.org/about/>> [Zugriff am: 09.09.2017]
- Python Software Foundation, 2001-2017b. [online] <<https://www.python.org/psf-landing/>> [Zugriff am: 09.09.2017]
- Raspberry Pi Foundation, 2017. Raspberry Pi 2 Model B. [online] <<https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>> [Zugriff am: 12.09.2017]
- Robotics Academy, 2015. *What is an IR Sensor?* [online] <[http://education.rec.ri.cmu.edu/content/electronics/boe/ir\\_sensor/1.html](http://education.rec.ri.cmu.edu/content/electronics/boe/ir_sensor/1.html)> [Zugriff am: 04.09.2017]
- RobTillaart, 2016. *RunningAverage*. [online] <<https://github.com/RobTillaart/Arduino/tree/master/libraries/RunningAverage>> [Zugriff am: 08.11.2016]
- Roy, M., Dillo, W., Emrich, H.M., Ohlmeier, M.D., 2009. *Das Asperger-Syndrom im Erwachsenenalter*. Dtsch Arztebl Int. Nr. 106(5). DOI: 10.3238/arztebl.2009.0059

- Shopify, 2017. [online]  
 <[https://cdn.shopify.com/s/files/1/0059/3932/products/Paro\\_Therapeutic\\_Robot\\_04\\_1024x1024\\_6e563953-6ee3-4269-a584-5e1f8ad4b5e4.jpg?v=1411458784](https://cdn.shopify.com/s/files/1/0059/3932/products/Paro_Therapeutic_Robot_04_1024x1024_6e563953-6ee3-4269-a584-5e1f8ad4b5e4.jpg?v=1411458784)>  
 [Zugriff am: 10.09.2017]
- Sparkfun Electronics®, 2017a. *SparkFun RFID Starter Kit*. [online]  
 <<https://www.sparkfun.com/products/13198>> [Zugriff am: 01.09.2017]
- Sparkfun Electronics®, 2017b. *General Purpose Type Distance Measuring Sensors*. [online]  
 <<https://www.sparkfun.com/datasheets/Components/GP2Y0A21YK.pdf>> [Zugriff am: 05.10.2017]
- Swinburne University, 2017a. *Matching Emotions*. [online]  
 <<http://www.autismgames.com.au>> [Zugriff am: 01.02.2017]
- Swinburne University, 2017b. *Robbie the Robot*. [online]  
 <<http://www.whizkidgames.com/?game=robbie>> [Zugriff am: 01.02.2017]
- The National Autistic Society 2017, 2017. *How to talk about autism*. [online]  
 <<http://www.autism.org.uk/get-involved/media-centre/how-to-talk.aspx>> [Zugriff am: 08.10.2017]
- TRACKS autism, 2017. [online] <<http://www.tracks-autism.org.uk/>> [Zugriff am: 11.09.2017]
- University of Hertfordshire, 2016. *Kaspar on Today Programme*. [online]  
 <<http://www.herts.ac.uk/kaspar/news-and-media/kaspar-news/kaspar-on-today-programme>> [Zugriff am: 12.09.2017]
- University of Hertfordshire, 2017a. *Kaspar the social robot*. [online]  
 <<http://www.herts.ac.uk/kaspar/the-social-robot>> [Zugriff am: 11.09.2017]
- University of Hertfordshire, 2017b. *Lead researchers for Kaspar*. [online]  
 <<http://www.herts.ac.uk/kaspar/research-and-development/lead-researchers>>  
 [Zugriff am: 12.09.2017]
- Volkmar, F.R., Paul, R., Klin, A., Cohen, D., 2005. *Handbook of Autism and Pervasive Developmental Disorders*. Volume 2. Third Edition. John Wiley & Sons. ISBN: 9780471755661
- Ward, M. et al., 2006. *RFID: Frequency, standards, adoption and innovation*. JISC Technology and Standards Watch



# Abbildungsverzeichnis

Abbildung 1: ASS Symptome (Quelle: (Autismus-Kultur, 2006)).....	12
Abbildung 2: PARO (Quelle: (Shopify, 2017)).....	15
Abbildung 3: Kaspar (Quelle: (University of Hertfordshire, 2016)) .....	16
Abbildung 4: deutschsprachige Version des „Uncanny Valley“ (Quelle: (Eggshell Robotics, 2009)) .....	17
Abbildung 5: Kaspar imitiert Kind (Quelle: (Daynurses, 2016)) .....	18
Abbildung 6: Therapieroboter Teo (Quelle: (Bonarini et al, 2016)).....	19
Abbildung 7: Puffy: (a) mit Projektion auf dem Korpus (b) Veränderung des inneren Lichts (Quelle: (Gelsomini et al., 2017)).....	21
Abbildung 8: „Robbie the Robot“, erste Aufgabe (Swinburne University, 2017b) .....	22
Abbildung 9: TOM Systeminhalte (Quelle: (Platus Learning Systems GmbH, 2017b)) .....	23
Abbildung 10: RoboFriend mit Beschriftung .....	24
Abbildung 11: Hardware Übersicht.....	27
Abbildung 12: Teensy++ 2.0 Schaltung.....	28
Abbildung 13: gelötete Teensy++ 2.0 Platine .....	29
Abbildung 14: OPV Schaltplan (Quelle: (Electrical Engineering, 2012)) .....	30
Abbildung 15: gelötete OPV Schaltung .....	30
Abbildung 16: Raspberry Pi 2 Model B, Ansicht von oben (Quelle: (Raspberry Pi Foundation, 2017)) .....	31
Abbildung 17: Funktionsweise eines IR Sensors (Quelle: (Robotics Academy, 2015)).....	33
Abbildung 18: Gegenüberstellung von Ausgangsspannung zum Objektstand (Quelle: (Sparkfun Electronics®, 2017b)).....	34
Abbildung 19: Software Übersicht .....	36
Abbildung 20: Code Ausschnitt Robot Middleware – Weiterverarbeitung des erhaltenen Befehls.....	40
Abbildung 21: Codebeispiel „Vorwärtsfahren“ .....	42
Abbildung 22: Lesen und Verarbeitung von RFID Daten .....	44
Abbildung 23: Entladungskurve (bearbeitet (Quelle bearbeitet: (Bremach-Reisemobile, 2013)) .....	45
Abbildung 24: ADC-Wert/Prozent Kurve.....	46
Abbildung 25: Codeausschnitt HTML-File .....	46
Abbildung 26: Decodierung von Befehlen des Web Frontends.....	47
Abbildung 27: Screenshot des Webservers.....	48
Abbildung 28: verschiedene Gesichtskonfigurationen .....	49
Abbildung 29: Bausteine mit RFID Sticker.....	50
Abbildung 30: Screenshot von Spiel 2 „Gefühle erkennen“ .....	51
Abbildung 31: Screenshot von Spiel 3 „Tiergeräusche zuordnen“ .....	52
Abbildung 32: RFID Karten .....	53

Abbildung 33: Screenshot der Roboter Steuerung am Tablet.....	53
Abbildung 34: Verwendete FABI Box .....	54
Abbildung 35: Proband 1 beim Testen .....	58
Abbildung 36: Überarbeitetes Hauptmenü.....	60
Abbildung 37: Proband 2 beim Steuern mit der FABI-Box.....	61
Abbildung 38: Probandin 3 beim Testen.....	62

# Tabellenverzeichnis

Tabelle 1: Spezifikationen des Teensy++ 2.0 Mikrocontrollers .....	27
Tabelle 2: Übersicht der SW Module mit Beschreibung.....	37
Tabelle 3: Befehlsliste von der GUI zur Robot Middleware .....	38
Tabelle 4: Befehlsliste vom Python Hauptprogramm zur GUI .....	40
Tabelle 5: verwendete Python Module und ihre Verwendung .....	44
Tabelle 6: Übersicht der implementierten (Spiel-) Funktionen .....	49
Tabelle 7: Zusammenfassung der Wünsche, Anregungen und Verbesserungsvorschläge (gelbe Markierungen zeigen in dieser Arbeit umgesetzte Änderungen) .....	59
Tabelle 8: Übersicht der ProbandInnen .....	61
Tabelle 9: Zufriedenheit der ProbandInnen mit den getesteten Funktionen .....	64

# Abkürzungsverzeichnis

ADC	Analog Digital Converter
ASIMO	Advanced Step in Innovative Mobility
ASS	Autismus Spektrum Störung
CRT	Cathode Ray Tube
CSI	Camera Serial Interface
DC	Direct Current
EOL	End Of Line
FABI	Flexible Assistive Button Interface
GUI	Graphical User Interface
HID	Human Interface Device
HW	Hardware
IR	Infrarot
KI	Künstliche Intelligenz
NDD	NeuroDevelopmental Disorder
NFC	Near Field Communication
OPV	Operationsverstärker
OS	Operating System
RFID	Radio Frequency IDentification
SW	Software
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus

## Anhang A – Antragsformular Stadtschulrat Wien

### Ansuchen um Bewilligung einer wissenschaftlichen Erhebung (§ 46 Abs. 2 SchUG)

An \_\_\_\_\_ die 

--

  
Schulleitung der \_\_\_\_\_  
Eingangsstempel der Schule

Name des Ansuchenden	E-Mail-Adresse	Telefonnummer

Anschrift des Ansuchenden

Titel der Arbeit	Betreuer	Kontaktdaten Betreuer

Hochschule	Fakultät

#### Art der Arbeit:

☐ Bachelorarbeit / ☐ Masterarbeiten / ☐ Dissertationen / ☐ sonstige wissenschaftliche Arbeit  
von erheblicher Bedeutung: \_\_\_\_\_

#### Angaben zur Erhebung:

Schulstufe(n): \_\_\_\_\_  
Anzahl der Klassen: \_\_\_\_\_  
geschätzte Anzahl der Befragten: \_\_\_\_\_

Erhebungszeitraum	Dauer der Erhebung
von	
bis	

#### Folgende Beilagen sind dem Ansuchen in der angegebenen Reihenfolge gesammelt beizulegen:

- a) Ausgefülltes Antragsformular;
- b) Organisationsplan der Durchführung der wissenschaftlichen Erhebung;
- c) Kurzer Abriss der theoretischen Grundlagen für die geplante wissenschaftliche Erhebung;
- d) Operationalisierung der Fragestellungen bzw. der zu überprüfenden Arbeitshypothesen;
- e) Vorlage des Untersuchungsmaterials (z.B. Fragebogen, Interviewleitfaden, Tests, etc.);
- f) Angaben hinsichtlich der Dauer der Untersuchung pro Schüler und Klasse und Angaben nach welchen Kriterien die zu befragenden Schüler ausgewählt werden;

- g) Angaben zur geplanten Auswertungsmethode;
- h) Elternbrief;
- i) Informationsschreiben für den Lehrer;
- j) Bestätigung der Betreuung durch den Hochschulprofessor;
- k) Verpflichtungserklärung zur Wahrung des Datenschutzes und der Anonymität.

Ort/Datum \_\_\_\_\_

\_\_\_\_\_  
Unterschrift Ansuchender