

Bachelorarbeit

NeoPixel Rings for Robot State Visualization

Benedikt Deike

benedikt.deike@informatik.uni-hamburg.de
Studiengang B.Sc. Informatik
Matr.-Nr. 7080666

Erstgutachter: Dr. Norman Hendrich
Zweitgutachter: Dr. Andreas Mäder

Abgabe: 12.2021

Abstract

This project tries to visualize the internal state of a PR2 robot with the help of ring-shaped LEDs. In particular, the question of how intuitively colorful animations can represent the internal state of the robot is devoted to. A detailed literature search is presented on this first. Then two visualization concepts are presented, each of which prioritizes intuitiveness and information content to different degrees. A system for determining the robot status, as well as a hardware and software construct for displaying the visualization concepts is additionally described.

Kurzfassung

Dieses Projekt versucht mithilfe von ringförmigen LEDs den internen Zustand eines PR2-Roboters zu visualisieren. Insbesondere stellt sich die Frage, wie intuitiv animierte Farbmuster den internen Zustand des Roboters darstellen können. Die Arbeit beginnt mit einer ausführlichen Literaturrecherche zu diesem Thema. Anschließend sind zwei Visualisierungskonzepte dargelegt, die Intuitivität und Informationsgehalt jeweils unterschiedlich stark priorisieren. Ein System zur Bestimmung des Roboterzustands, ebenso wie ein Hard- und Softwarekonstrukt zur Darstellung der Visualisierungskonzepte wird außerdem beschrieben.

Inhaltsverzeichnis

| | | |
|----------|---|-----------|
| 1 | Einführung | 1 |
| 1.1 | Aufbau und Struktur | 1 |
| 1.2 | Hintergrund und Motivation | 1 |
| 1.3 | Projektziel | 2 |
| 1.4 | Projektverlauf | 2 |
| 2 | Terminologie | 3 |
| 3 | Stand der Forschung | 5 |
| 3.1 | Intuitive Visualisierung | 5 |
| 3.2 | Visualisierung von Emotionen | 6 |
| 3.3 | Visualisierung von Zuständen | 9 |
| 3.4 | Implikationen | 12 |
| 4 | Implementation | 14 |
| 4.1 | Elektronische Bauteile | 15 |
| 4.2 | Kommunikation | 16 |
| 4.3 | Roboterzustand | 21 |
| 4.4 | Visualisierung | 22 |
| 4.4.1 | Mikrocontroller Quellcode | 22 |
| 4.4.2 | ROS-Paket (Visualising) | 23 |
| 4.5 | Gesamtsystem | 26 |
| 5 | Visualisierungskonzepte | 27 |
| 5.1 | Emotionen und Gesichtsausdrücke | 27 |
| 5.2 | Statuslichter und Buchstaben | 29 |
| 6 | Ausblick | 32 |
| | Literaturverzeichnis | 33 |

Abbildungsverzeichnis

| | | |
|----|---|----|
| 1 | PR2-Roboter | 3 |
| 2 | Taxonomie des Vorwissens | 5 |
| 3 | Roboter Maru und NAO Roboter | 8 |
| 4 | Rudimentärer Roboter und Gesichtsausdrücke | 10 |
| 5 | Roboter HERA und Roboter CoBot | 12 |
| 6 | Farbverläufe | 14 |
| 7 | NeoPixel-Ringe und Arduino Nano | 15 |
| 8 | Verschaltung der elektronischen Komponenten | 16 |
| 9 | Imaginärer ROS-Graph | 17 |
| 10 | Nachrichtenformat | 19 |
| 11 | Protokollabläufe | 20 |
| 12 | Beobachtersystem | 21 |
| 13 | Klassendiagramm | 24 |
| 14 | Gesamtsystem | 26 |
| 15 | NeoPixel-Ringe und PR2-Roboter | 29 |

Tabellenverzeichnis

| | | |
|---|--|----|
| 1 | Zuordnung von Emotionen zu Farben | 7 |
| 2 | Metaphern, Eigenschaften und Farben | 9 |
| 3 | Farben und deren Bedeutung nach ISO-Standard IEC 60204-1 | 11 |
| 4 | Gesichtsausdrücke | 28 |
| 5 | Buchstaben | 31 |

Quellcodeverzeichnis

| | | |
|---|---|----|
| 1 | Hauptschleife des Mikrocontrollers | 22 |
| 2 | Funktion zum Darstellen animierter Farbmuster | 23 |
| 3 | Farbmuster definierende Datei | 25 |

1 Einführung

Diese Bachelorarbeit ist die schriftliche Ausarbeitung des Bachelorprojekts *Neo-Pixel Rings for Robot State Visualization*. Die Begriffe Bachelorprojekt und Bachelorarbeit seien mit Projekt und Arbeit abgekürzt.

[Abschnitt 1](#) dient der Einführung in die Thematik des Projekts. Zu Beginn ist der Aufbau und die Struktur dieser Arbeit in [Unterabschnitt 1.1](#) aufgeführt. Anschließend sind die Hintergründe und die Motivation für das Thema in [Unterabschnitt 1.2](#) erläutert. [Unterabschnitt 1.3](#) beschreibt die Ziele des Projekts, bevor abschließend in [Unterabschnitt 1.4](#) der Projektverlauf dargestellt ist.

1.1 Aufbau und Struktur

Der Rahmen des Projekts ist im [Abschnitt 1](#) festgelegt. Anschließend sind im [Abschnitt 2](#) wichtige Begriffe, welche in dieser Arbeit Verwendung finden, definiert. [Abschnitt 3](#) diskutiert die Bedeutung und den Umfang des Projektziels und präsentiert im Anschluss den Forschungsstand relevanter Forschungsgebiete. Elektronische Bauteile und Softwarekomponenten, die im Verlauf des Projekts verwendet und entwickelt wurden, sind im [Abschnitt 4](#) dargelegt. Zwei Visualisierungskonzepte, die den internen Zustand des PR2-Roboters kommunizieren, sind im [Abschnitt 5](#) aufgeführt. Mögliche Erweiterungen und Verbesserungen der Projektergebnisse sind abschließend im [Abschnitt 6](#) aufgelistet.

1.2 Hintergrund und Motivation

Die Forschungsgruppe TAMS der Universität Hamburg verwendet Roboter für verschiedene Forschungsprojekte. Einer dieser Roboter ist der in [Abbildung 1](#) dargestellte humanoide PR2-Roboter von Willow Garage. Der PR2-Roboter sieht sich oft mit schwierigen und komplexen Aufgaben der Forschungsgruppe konfrontiert. Die Verarbeitung und Ausführung dieser Aufgaben durch den Roboter nimmt üblicherweise einen variablen Zeitraum in Anspruch. In manchen Fällen ist es dem Roboter jedoch nicht möglich, die gegebene Aufgabe erfolgreich abzuschließen. In dieser Situation kann die Forschungsgruppe allerdings nicht sofort erkennen, ob der Roboter noch mit der Lösung der Aufgabe beschäftigt ist, oder ob sich der Roboter in einem Fehlerzustand befindet. Dies herauszufinden ist nur durch das Einsehen von Statusinformationen möglich, die über einen mit dem Roboter verbundenen Monitor einsehbar sind. Die unstrukturierte und unübersichtliche Darstellung der Statusinformationen machen deren Verständnis aber häufig mühsam und zeitaufwendig. Um den aktuellen internen Zustand des Roboters beim Ausführen von Aufgaben schneller erkennen zu können, soll dieser beim Betrachten des Roboters sofort ersichtlich sein. An den Roboter angebrachte LEDs, welche animierte Farbmuster abspielen, stellen einen möglichen Lösungsansatz für das beschriebene Problem dar.

1.3 Projektziel

Ziel dieses Projekts ist die Visualisierung der internen Zustände des PR2-Roboters mithilfe von zwei LED-Ringen. Da die LED-Ringe humanoide Augen imitieren sollen, ist eine Installation um die Kameras des Roboters beabsichtigt. Die Umsetzung des Projekts sieht einen mit dem PR2-Roboter über ein USB-Kabel verbundenen Mikrocontroller vor, welcher mit den LEDs kommuniziert. Anhand dieses Arrangements ist es dem Roboter möglich, die beiden LED-Ringe in Echtzeit anzusprechen. Konkret soll das Ansprechen der LEDs ein Robot-Operating-System-Knoten (ROS-Knoten) im Netzwerk des PR2-Roboters übernehmen. Ein weiterer ROS-Knoten soll, basierend auf Diagnosedaten aus dem ROS-Netzwerk des Roboters, dessen internen Zustand in Echtzeit ermitteln. Mithilfe beider ROS-Knoten soll es möglich sein, je nach ermitteltem Zustand, das Abspielen eines animierten Farbmusters auf den LED-Ringen zu veranlassen.

Der Fokus dieses Projekts soll auf dem Finden von intuitiven, animierten Farbmustern zur Visualisierung der internen Zustände des PR2-Roboters liegen. Dabei können unter anderem menschliche Gesichtsausdrücke als Inspirationsquelle dienen.

1.4 Projektverlauf

Die Ergebnisse dieses Projekts wurden in einem Zeitraum von drei Monaten erarbeitet. Ausgangspunkt war der aktuelle Stand der Forschung. Zunächst wurden für die Projektumsetzung notwendige, elektrische Bauteile studiert und auf ihre Tauglichkeit geprüft. Nach dem Vertrautsein mit dem wichtigsten elektronischen Bauteil, dem Mikrocontroller, konnte die notwendige Funktionalität für diesen programmiert werden. Um den Mikrocontroller ansprechen zu können, ist zunächst eine Kommunikationssoftware erarbeitet worden, welche dann in das Robot-Operating-System (ROS) des PR2-Roboters eingegliedert wurde. Das Definieren von relevanten, internen Roboter Zuständen ermöglichte im Anschluss die Entwicklung einer reaktiven Softwarekomponente, welche das Abspielen von animierten Farbmustern mithilfe des Mikrocontrollers anstößt. Nach Fertigstellung des Hardware- beziehungsweise Software-Konstrukts, konnte mit dem Erstellen multipler animierter Farbmuster für die Visualisierung der internen Zustände des Roboters begonnen werden.



Abbildung 1: Der PR2 ist aktuell einer der fortschrittlichsten Forschungsroboter. Seine leistungsstarken Hard- und Softwaresysteme ermöglichen es ihm, Tische aufzuräumen, Handtücher zu falten und Getränke aus dem Kühlschrank zu holen. Der PR2-Roboter der Forschungsgruppe TAMS trägt zudem an seinem rechten Arm die sogenannte Shadow Dexterous Hand der Firma The Shadow Robot Company und auf seinem Kopf eine Microsoft Xbox One Kinect-Sensorleiste.

2 Terminologie

Das Erstellen intuitiver Visualisierungen für die Zustände des PR2-Roboters setzt zunächst eine Definition von Intuitivität voraus. Intuitivität ist in dieser Arbeit entsprechend der von Mohs et al. [7] gegebenen [Definition 1](#) zu verstehen.

Definition 1. *Ein technisches System ist intuitiv benutzbar, wenn es durch nicht bewusste Anwendung von Vorwissen durch den Benutzer zu effektiver Interaktion führt.*

Zusätzlich ist in [Anmerkung 1](#) erläutert, was in diese Arbeit unter dem Zustand eines Roboters verstanden wird.

Anmerkung 1. *Mikrocontroller, Prozessoren, Kameras, Aktuatoren, Programme und eine Vielzahl weiterer Komponenten machen einen Roboter aus. Der Zustand einer Komponente lässt sich anhand einiger individueller Eigenschaften zu jedem Zeitpunkt charakterisieren. Produzierte Abwärme und Auslastung charakterisieren unter anderem beispielsweise den Zustand eines Prozessors. Der Zustand des gesamten Roboters ist durch die Summe der Charakterisierungen aller Komponenten gegeben.*

Hat ein Roboter eine bestimmte Tätigkeit zu bewerkstelligen, etwa das Lösen der in [Unterabschnitt 1.2](#) beschriebenen Aufgaben, so kann der Zustand eines Roboters zudem wie in [Anmerkung 2](#) verstanden werden.

Anmerkung 2. *Der Zustand eines Roboters zeigt den Fortschritt bei der Erfüllung einer bestimmten Aufgabe.*

[Anmerkung 2](#) lässt sich im Falle des PR2-Roboters der Forschungsgruppe TAMS jedoch in [Anmerkung 1](#) überführen. Einzelne Programme beschreiben die vom PR2-Roboter ausführbaren Tätigkeiten. Phasen, die derartige Programme während ihrer Ausführung durchlaufen, sind somit Indikatoren für den Fortschritt beim Bewerkstelligen einer Aufgabe. Des Weiteren sind Laufzeitphasen Eigenschaften, anhand derer ein laufendes Programm charakterisiert werden kann. Nach [Anmerkung 1](#) umfasst der Begriff Komponente auch ein, für den Roboter notwendiges, Programm. Der Fortschritt bei der Erfüllung einer Aufgabe lässt sich dementsprechend als Eigenschaft einer Komponente auffassen.

Mithilfe von animierten Farbmustern, die in [Anmerkung 3](#) erklärt werden, versucht dieses Projekt die Zustände des PR2-Roboters zu visualisieren.

Anmerkung 3. *Animierte Farbmuster werden auf LED-Feldern abgespielt. Ein LED-Feld besteht aus einzelnen LEDs. Ein Farbmuster speichert für jede dieser LEDs einen Farbwert. Durch das Abspielen mehrerer Farbmuster hintereinander entsteht ein animiertes Farbmuster.*

Im Folgenden sind einige Begriffe aus dem Forschungsgebiet der Mensch-Computer-Interaktion definiert, welche in dieser Arbeit Verwendung finden. Alle Definitionen, bis auf die Definition einer künstlichen Ausgabemodalität, sind der von Kopp [6] gehaltenen Vorlesung Human-Computer-Interaction entnommen.

Definition 2. *Eine Modalität bezeichnet ein kommunikatives System, welches durch die Art und Weise, wie Information kodiert und interpretiert wird, gekennzeichnet ist. Modalitäten betreffen sowohl die Informationsübertragung vom Menschen zur Maschine (technisch: Eingabemodalität) als auch von der Maschine zum Menschen (technisch: Ausgabemodalität).*

Definition 2.1. *Eine Schnittstelle, die genau eine Eingabemodalität oder Ausgabemodalität zur Verfügung stellt, ist unimodal. Stellt eine Schnittstelle jedoch mehrere Eingabemodalitäten und/oder Ausgabemodalitäten zur Verfügung, so ist diese multimodal.*

Definition 2.2. *Eine Ausgabemodalität ist natürlich, wenn sie Teil der kommunikativen Grundkompetenz eines (sozialisierten) Menschen ist. Hierzu gehören: Sprache (Laute), Gestik, Mimik und Körpersprache (Proxemik). Ist eine Ausgabemodalität nicht natürlich und von einem Roboter ausdrückbar, so wird sie künstlich genannt. Hierzu gehören zum Beispiel blinkende Lampen oder Lichter.*

3 Stand der Forschung

[Abschnitt 3](#) diskutiert zunächst in [Unterabschnitt 3.1](#) die Bedeutung und den Umfang des Projektziels. Anschließend werden auf Basis der vorangegangenen Diskussion relevante Forschungsgebiete und deren Forschungsstand in [Unterabschnitt 3.2](#) und [Unterabschnitt 3.3](#) präsentiert. Der Literaturrecherche entnommene Kriterien, die als Orientierung für den Lösungsentwurf des Projektziels dienen, sind abschließend in [Unterabschnitt 3.4](#) dargelegt.

3.1 Intuitive Visualisierung

Der [Unterabschnitt 1.3](#) nennt die Visualisierung der internen Zustände des PR2-Roboters als zentrales Ziel für dieses Projekt. Eine Codierung der Zustände mithilfe von animierten Farbmustern, welche mit allen Mitgliedern der Forschungsgruppe abgesprochen ist, kommuniziert den Zustand des Roboters auf simple und effektive Weise. Eine Lösung, die eine derartigen Codierung implementiert, ist jedoch möglicherweise nicht intuitiv zu verstehen. Das Nachschlagen einzelner Codes, aufgrund von fehlendem oder vergessenem Vorwissen, könnte demnach zur Notwendigkeit werden. Solch eine Situation ist definitiv nicht im Sinne des Projektziels.

Intuitivität nach [Definition 1](#) erfolgt aufgrund der unbewussten Anwendung von Vorwissen. Die Herkunft von Vorwissen lässt sich nach Mohs et al. [7] in eine aus fünf Ebenen bestehende Taxonomie gliedern, die in [Abbildung 2](#) dargestellt ist. Die Anzahl der Personen, welche das Vorwissen einer Ebene besitzen, nimmt dabei mit fortschreitenden Ebenen ab, da der Spezialisierungsgrad des Vorwissens mit jeder Ebene zunimmt. So umfasst zum Beispiel die erste Ebene angeborenes Vorwissen, wohingegen Vorwissen, welches durch Sinneserfahrungen erlangt wurde, der zweiten Ebene zugeschrieben wird. Interaktionen zwischen Mensch und Roboter, welche sich nur auf das Vorwissen der ersten beiden Ebenen stützen, sind demnach für eine Vielzahl von Personen intuitiv.

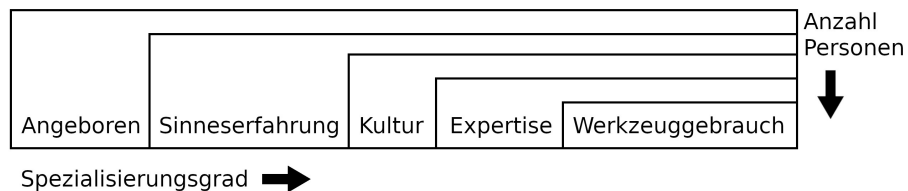


Abbildung 2: Die Horizontalachse zeigt die Taxonomie des Vorwissens, wohingegen die Vertikalachse die Anzahl der Personen, welche das Vorwissen einer Ebene besitzen, schematisch andeutet. Die Höhe der Zeilen in der Vertikalachse ist dabei willkürlich gewählt [7].

Unter Forschenden herrscht Uneinigkeit, ob grundlegende Emotionen [3] kulturspezifische soziale Konstrukte oder biologisch primitive Reaktionen sind [11].

Ekman et al. [2] legen jedoch nahe, dass grundlegende Emotionen in menschlichen Gesichtern universal erkannt werden. Demnach ist anzunehmen, dass die Herkunft von zur Erkennung grundlegender Emotionen notwendigen Vorwissens in die ersten beiden Ebenen, der in [Abbildung 2](#) dargestellten Taxonomie, einzugliedern ist. Die internen Zustände des PR2-Roboters zunächst auf menschliche Emotionen abzubilden und anschließend diese Emotionen mithilfe von animierten Farbmustern darzustellen, könnte demnach zu einer intuitiven Codierung der Zustände führen. Dieser Ansatz ist zusätzlich durch die angedachte Platzierung der LED-Ringe um die Kameras des PR2-Roboters begünstigt. Diese wirken in der anthropomorphen Interpretation des Roboters nämlich wie dessen Augen, welche wesentlich für das Erkennen von Emotionen sind [15]. Im [Unterabschnitt 3.2](#) sind daher einige Erkenntnisse zur Visualisierung von Emotionen durch Roboter präsentiert.

Die Codierung der Zustände des PR2-Roboters nur von menschlichen Emotionen abzuleiten, birgt jedoch eine große Limitierung. Lediglich sechs grundlegende Emotionen sind ausgemacht, weshalb nur maximal sechs unterscheidbare Zustände mit einer solchen Codierung darstellbar sind. Nach [Anmerkung 1](#) setzt sich der Zustand eines Roboters aus den Zuständen der einzelnen Komponenten zusammen. Der Zustand einer einzelnen Komponente hat allerdings isoliert betrachtet oft bereits mehr als sechs Ausprägungen. Dieser Umstand verdeutlicht, dass eine intuitive Codierung nur auf Grundlage von menschlichen Emotionen nicht möglich ist. Die Mitglieder der Forschungsgruppe TAMS besitzen aber eine Fülle an fachspezifischem Vorwissen, das in die Ebene der Expertise aus [Abbildung 2](#) einzuordnen ist. Eine Codierung, die dieses Vorwissen berücksichtigt, kann einerseits nach [Definition 1](#) intuitiv sein und andererseits eine Vielzahl von Zuständen codieren. Erkenntnisse zur intuitiven Visualisierung von internen Roboterzuständen aufgrund von fachspezifischem Vorwissen werden daher im [Unterabschnitt 3.3](#) präsentiert.

Vorwissen, das in [Abbildung 2](#) der Ebene Expertise zugeteilt ist, basiert oft auf Vorwissen niedrigerer Ebenen. Dementsprechend ist Intuitivität, die auf fachspezifischem Vorwissen fußt, häufig durch Intuitivität, dessen Grundlage Vorwissen niedrigerer Ebenen ist, bedingt. Angesichts dieser Abhängigkeiten ist eine strikte Trennung zwischen den Erkenntnissen aus [Unterabschnitt 3.2](#) und jenen aus [Unterabschnitt 3.3](#) nicht möglich. Ungeachtet dessen ist eine solche Trennung in dieser Arbeit vorgenommen worden. Begründet ist dies mit der Hoffnung, eine bessere Einordnung der Erkenntnisse durch den Leser zu ermöglichen.

3.2 Visualisierung von Emotionen

Die Visualisierung menschlicher Emotionen mithilfe von Robotern ist in der Forschung von großem Interesse. Soziale Roboter [10] mit einem anthropomorphen Erscheinungsbild und einer Vielzahl an komplexen, natürlichen Ausgabemodalitäten sind hierfür prädestiniert. Angesichts der Beschränkung auf nur zwei LED-Ringen zur Visualisierung der internen Zustände des PR2-Roboters sind

jedoch viele der im Zusammenhang mit diesen Robotern gewonnenen Erkenntnisse [4], [8], [9] weniger relevant für dieses Projekt. Die Präsenz von nicht humanoiden Robotern mit rudimentären Ausgabemodalitäten in unserem Alltag ist jedoch in den letzten Jahren stetig gestiegen. Ein zunehmendes Interesse, Emotionen mithilfe von rudimentären menschlichen und insbesondere rudimentären künstlichen Ausgabemodalitäten auszudrücken, ist demnach nicht verwunderlich. Die Verwendung von Geräuschen, Farben und simplen Bewegungen scheint in dieser Hinsicht besonders vielversprechend zu sein [16], [18].

Im Folgenden sind einige Forschungsarbeiten vorgestellt, die Erkenntnisse beim Ausdrücken von menschlichen Emotionen durch Roboter mithilfe verschiedener Ausgabemodalitäten dokumentieren. Dabei finden insbesondere diejenigen Aspekte der jeweiligen Forschungsprojekte Beachtung, welche sich mit der Vermittlung von Emotionen durch Farben oder animierter Farbmuster befassen.

Song und Yamada [16] versuchen mithilfe des simplen Roboters Maru die vier menschlichen Emotionen Wut, Fröhlichkeit, Traurigkeit und Gelassenheit darzustellen. Dazu verwenden sie einerseits nur Farben, nur Vibrationen oder nur Geräusche, aber auch Kombinationen aus diesen Ausgabemodalitäten. [Abbildung 3a](#) zeigt den Roboter Maru und deutet dessen Ausgabemodalitäten an. Die verwendete Zuordnung von Emotionen zu Farben ist in [Tabelle 1](#) zu sehen. Die Forscher merken jedoch an, dass sich die Fachliteratur nicht wirklich ei-

| Emotion | Wut | Fröhlichkeit | Traurigkeit | Gelassenheit |
|----------------|-----|--------------|-------------|--------------|
| Farbe | Rot | Grün | Blau | Weiß |

Tabelle 1: Die Zuordnung von Emotionen zu Farben, welche von Song und Yamada verwendet wurde [16].

nig ist, welche Emotionen sich eindeutig mit welchen Farben assoziieren lassen. Nach einem Experiment mit dem Roboter Maru und 24 Probanden ist absehbar, dass Maru die Emotionen Wut, Traurigkeit und Gelassenheit verständlich mithilfe einer oder mehrerer Ausgabemodalitäten ausdrücken kann, wohingegen die Emotion Fröhlichkeit von den Probanden nicht eindeutig erkannt werden konnte.

Der Roboter NAO, welcher sich mithilfe von Bewegungen, Geräuschen und verschiedenen Augenfarben verständigen kann, dient Häring et al. [12] als Plattform für den Versuch, die Emotionen Wut, Traurigkeit, Fröhlichkeit und Angst mithilfe eines Roboters auszudrücken. [Abbildung 3b](#) zeigt den Roboter NAO, der von SoftBank Robotics entwickelten wird. Für jede der genannten Emotionen erarbeitete die Forschungsgruppe jeweils zwei von NAO ausführbare Darstellungen, welche die vier Ausgabemodalitäten des Roboters kombinieren. Die erarbeiteten Darstellungen wurden anschließend 67 Probanden präsentiert, um deren Aussagekraft einzuschätzen. Hierfür waren die Probanden aufgefordert, zu jeder präsentierten Darstellung jeweils eine Emotion zu nennen. Die große

Mehrheit der Probanden konnte jeder Darstellung die angedachte Emotion zuordnen. Ebenso wie im Forschungsprojekt von Song und Yamada fiel es den Probanden am schwersten, die Darstellung der Emotion Fröhlichkeit zu erkennen. Um die Aussagekraft der jeweils einzelnen Ausgabemodalitäten zu bewerten, führte die Forschungsgruppe im Folgenden ein Experiment mit 42 Probanden durch. Die Bewegungen NAOs sind demnach sehr aussagekräftig für das Erkennen der jeweiligen Emotion, was überwiegend auch für NAOs Geräusche gilt. Die variablen Augenfarben jedoch konnten als unzuverlässig für das Ausdrücken von Emotionen identifiziert werden.



(a) Roboter Maru (übernommen aus [16], Abb. 1) (b) NAO Roboter ([13] entnommen)

Abbildung 3: Der Roboter Maru ist von Song und Yamada in Eigenproduktion für deren Experimente hergestellt worden, wohingegen der Roboter NAO vom Unternehmen SoftBank Robotics kommerziell vertrieben wird.

Während Song und Yamada erfolgreich mithilfe des Roboters Maru die Ausgabemodalitäten Farben, Vibrationen und Geräusche zum Ausdrücken von Emotionen kombinieren, bewerten Häring et al. Farben als unzuverlässig beim Versuch, Emotionen mithilfe des Roboters NAO auszudrücken. Nach der Einschätzung von Löffler et al. [18] zeigen die gegensätzlichen Resultate, dass Darstellungen von Emotionen nicht uneingeschränkt zwischen Plattformen beziehungsweise Robotern mit ähnlichen Ausgabemodalitäten austauschbar sind. Für die Forschungsgruppe ist ein Grund hierfür das Fehlen einer Methodik zum Abbilden von Emotionen auf rudimentäre, menschliche und insbesondere rudimentäre, künstliche Ausgabemodalitäten. Löffler et al. versuchen unter anderem angesichts solcher Widersprüche wesentliche Empfehlungen für das Ausdrücken der grundlegenden Emotionen Fröhlichkeit, Traurigkeit, Angst und Wut mithilfe der Ausgabemodalitäten Bewegung, Farbe und Geräusche zu erarbeiten. Die genannten Ausgabemodalitäten sind von der Forschungsgruppe aufgrund deren breiten Unterstützung in gängigen Forschungsrobotern gewählt worden. Für die darzustellenden Emotionen wiederum hat man sich aufgrund deren universalen Erkennbarkeit entschieden. Zunächst wurden Metaphern aus dem alltäglichen Sprachgebrauch zusammengetragen, die als Vorlage für die Visualisierung jeder Emotion dienen. Anschließend sind basierend auf den gesammelten Metaphern unimodale Darstellungen für jede Emotion abgeleitet worden. In [Tabelle 2](#) ist das Ergebnis dieses Prozesses für die Ausgabemodalität Farbe zu sehen. Die zwölf er-




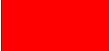
| Emotion | Metapher | Eigenschaft | Farbe |
|--------------|------------------------------------|--------------|---|
| Fröhlichkeit | strahlendes Lächeln, herzerwärmend | hell, warm |  |
| Traurigkeit | tieftraurig | dunkel, kalt |  |
| Angst | von Angst überschattet | dunkel |  |
| Wut | rotschén | rot |  |

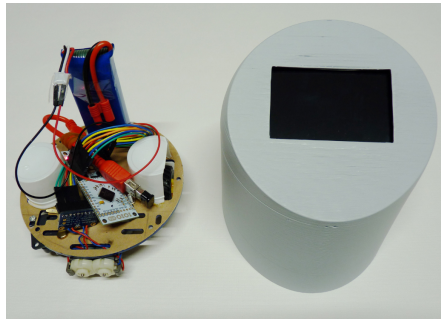
Tabelle 2: Metaphern und deren abgeleitete Eigenschaften für die Wahl der zu den Emotionen korrespondierenden Farben [18].

arbeiteten Darstellungen, je eine Darstellung für jede Emotion und jede Ausgabemodalität, sind in einer Studie mit 33 Probanden getestet worden. Dabei war von besonderem Interesse, wie aussagekräftig die einzelnen Ausgabemodalitäten für jede Emotion sind. [Abbildung 4a](#) zeigt den simplen Roboter, bestehend aus einem Smartphone mit fahrbarem Untersatz, der zur Präsentation der Darstellungen diente. Es wurden sowohl ausschließlich unimodale als auch multimodale Kombinationen der Darstellungen präsentiert. Die Auswertung der Studie ergab unter anderem, dass die Ausgabemodalität Bewegung zur unimodalen Darstellung von Emotion am effektivsten ist. Stehen jedoch zwei Ausgabemodalitäten zur Verfügung, verspricht eine multimodale Darstellung mittels Bewegung und Farbe den meisten Erfolg. Die ausgewerteten Ergebnisse dienen Löffler et al. als Grundlage, einige Empfehlungen beim Darstellen von Emotionen mithilfe von rudimentären Ausgabemodalitäten zu geben. Zunächst ist klarzustellen, dass verschiedene Ausgabemodalitäten unterschiedlich effektiv beim Darstellen von Emotion sind. Dementsprechend sollten Ausgabemodalitäten für jede darzustellende Emotion individuell gewählt werden, wobei multimodale Darstellungen von Emotionen die Verständlichkeit dieser verbessern. Eine multimodale Darstellung mittels Bewegung und Farbe weist dabei das beste Verhältnis zwischen Kosten und Nutzen auf.

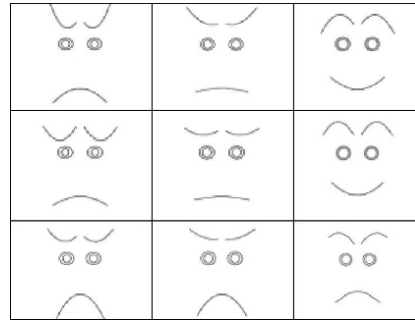
3.3 Visualisierung von Zuständen

Die Zustände eines Roboters mithilfe von Gesichtsausdrücken zu visualisieren ist von Archevapanich et al. [5] implementiert worden. Die Arbeit der Forschungsgruppe kann somit als Bindeglied zwischen dem [Unterabschnitt 3.2](#) und dem [Unterabschnitt 3.3](#) gesehen werden, da technische Zustände visualisiert werden, dies aber mithilfe von Gesichtsausdrücken beziehungsweise Emotionen bewerkstelligt wird. Zudem entspricht dieser Ansatz der in [Unterabschnitt 3.1](#) formulierten Hoffnung, mithilfe von Emotionen eine intuitive Codierung zu erarbeiten. Die Darstellung der Gesichtsausdrücke ist mithilfe eines reduzierten Gesichts, bestehend aus Strichen für Augen, Augenbrauen und den Mund, umgesetzt worden. Ein Generator, der auf Fuzzylogik basiert, erzeugt zu jedem Roboterzustand einen passenden Gesichtsausdruck. Verschiedene Gesichtsaus-

drücke sind in [Abbildung 4b](#) zu sehen. Die Möglichkeit, Augen, Augenbrauen und einen Mund zur Visualisierung von Gesichtsausdrücken zu verwenden, ist deutlich wirksamer als ausschließlich zwei LED-Ringe. Demnach sind Erkenntnisse von Archevapanich et al. nur bedingt relevant für dieses Projekt.



(a) Smartphone mit Abdeckung und fahrbarer Untersatz (übernommen aus [18], Abb. 2)



(b) Gesichtsausdrücke (übernommen aus [5], Tab. 3)

Abbildung 4: Ihrem Roboter keine humanoiden Züge zu verleihen, war eine bewusste Entscheidung von Loeffler et al., um Verzerrungen beim Bewerten der Ausgabemodalitäten vorzubeugen.

Einen ausdrucksstarken Kopf für den Assistenzroboter HERA zu entwickeln ist von van der Grinten et al. [19] dokumentiert worden. Der Roboter soll unter anderem als autonomes Liefersystem in öffentlichen Gebäuden eingesetzt werden. Aufgrund des Fehlens jeglicher Manipulatoren, ist dieser dabei besonders auf die Hilfe von in der Nähe stehenden Personen angewiesen. So müssen zum Beispiel Türen geöffnet oder Aufzüge für HERA bedient werden. Die Komponenten des entwickelten Kopfes sind in [Abbildung 5a](#) zu sehen. Insbesondere die LED-Felder sind für dieses Projekt von Interesse, da diese den internen Zustand des Roboters darzustellen versuchen und, im Falle von benötigter Hilfe, danebenstehende Personen auf HERA aufmerksam machen sollen. Die LED-Felder können zwölf verschiedene animierte Farbmuster wiedergeben, wobei jedes animierte Farbmuster mit einer variablen Geschwindigkeit und mit einer beliebigen Farbe abgespielt werden kann. Da zu HERAs Einsatzgebieten auch Fertigungsstätten zählen sollen, wurde der ISO-Standard IEC 60204-1 [23] bei der Wahl der Farben für die animierten Farbmuster eingehalten. [Tabelle 3](#) zeigt einen Ausschnitt, der im ISO-Standard IEC 60204-1 definierten Farben. Um HERAs neuen Kopf zu testen, wurde ein digitaler Fragebogen verwendet, der den Probanden einen neutralen, einen traurigen und einen konzentrierten Gesichtsausdruck des Roboters präsentierte. Diese Gesichtsausdrücke sind von besonderem Interesse, da sie die Roboterzustände neutral, hilfsbedürftig oder beschäftigt ausdrücken sollen. Durch das Auswerten der Fragebögen konnte unter anderem die Signifikanz der LED-Beleuchtung für das Erkennen des neutralen und des konzentrierten Gesichtsausdrucks gezeigt werden.

| Farbe | Bedeutung |
|-------|---|
| Rot | kritischer Fehler |
| Gelb | ungewöhnlicher Zustand, anbahnender kritischer Fehler |
| Grün | Normalzustand |
| Blau | Handlung zwingend notwendig |

Tabelle 3: Farben und deren Bedeutung nach ISO-Standard IEC 60204-1 [19].

Baraka und Veloso [17] versuchen mithilfe von LED-Lichtern den Zustand des Serviceroboters CoBot, der in [Abbildung 5b](#) dargestellt ist, zu visualisieren. Für CoBot müssen, ebenso wie für HERA, Türen geöffnet und Aufzüge bedient werden, weshalb der Roboter im Alltag oft auf die Unterstützung von in der Nähe befindlichen Personen angewiesen ist. Demzufolge muss CoBot, im Fall von benötigter Hilfe, auf sich aufmerksam machen und die benötigte Unterstützung verständlich kommunizieren können. Nach Ansicht der Forschenden ist es mithilfe von LED-Lichtern möglich, benötigte Unterstützung über eine große Distanz zu kommunizieren und gleichzeitig beistehende Personen nicht zu stören. Weitere von CoBot unterstützte Ausgabemodalitäten, wie beispielsweise Sprache oder auf Bildschirmen dargestellte Texte, haben in dieser Hinsicht jeweils Defizite. So ist die Kommunikation via Geräuschen beziehungsweise Sprache nicht über einen langen Zeitraum und gleichzeitig über eine große Distanz möglich, ohne Personen im Umkreis zu belästigen. Texte, die auf einem Bildschirm dargestellt werden, sind zwar weniger störend als lautstarke Sprache oder Geräusche, können aus weiter Distanz aber nur schwer erkannt werden.

Baraka und Veloso formalisieren die Visualisierung von Roboterzuständen in ihrer Arbeit und zeigen anhand von CoBot wie sich die erarbeitete Formalisierung praktisch umsetzen lässt. Zunächst ist klarzustellen, welche Aufgaben ein Roboter zu bewerkstelligen hat. Ausgehend von diesen können Zustände formuliert werden, die ausdrücken, ob Aufgaben erfüllt, in Bearbeitung oder gescheitert sind. Die Visualisierung der Zustände via LED-Lichtern ist im Formalismus mithilfe von Animationsmatrizen, Intensitätsfunktionen und Animationstupeln umgesetzt. Eine Animationsmatrix weist jeder LED drei Intensitätsfunktionen zu, jeweils eine für jeden Farbkanal der LED. Jede Intensitätsfunktion gibt abhängig von der parametrisierten Zeit einen bestimmten Intensitätswert aus. Baraka und Veloso präsentieren sechs grundlegende Intensitätsfunktionen in ihrer Arbeit, womit sich in Kombination mit verschiedenen Animationsmatrizen moderat komplexe Animation beschreiben lässt. Animationstupel bündeln mehrere LEDs und deren zugewiesene Intensitätsfunktionen. Ein Algorithmus, dem verschiedene Animationstupel und Ausprägungen der formulierten Zustände vorliegen, generiert sogenannte Animationsepisoden, die als abgeschlossene Teileinheiten einer Animation definiert sind. Animationsepisoden ermöglichen auf-

grund ihrer abgeschlossenen Natur einen ungestörten Animationsfluss bei wechselnden Zuständen, was für die dauerhafte und reaktive Darstellung des Roboterzustands gewünscht ist. In Situationen, in denen mehrere Zustände dargestellt werden müssen, entscheidet eine Funktion, welche Darstellung den Vorrang erhält. Somit ist gewährleistet, dass immer lediglich genau ein Zustand visualisiert wird.

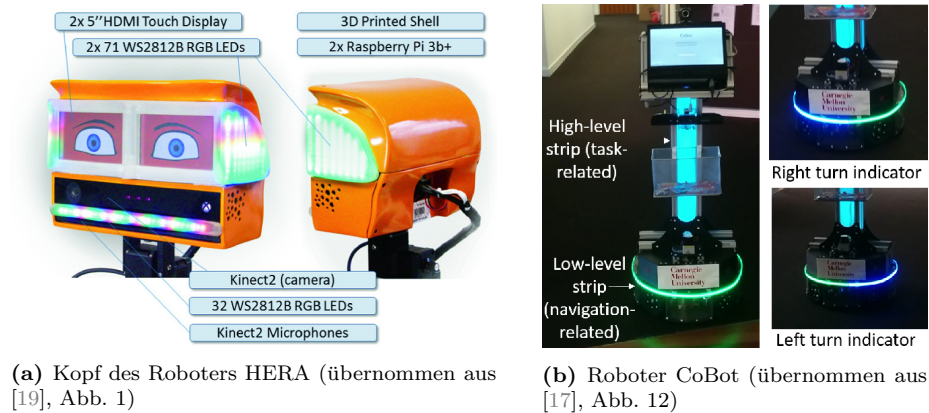


Abbildung 5: Der Roboter CoBot hat vermehrt Navigationsaufgaben zu bewältigen, weshalb eine separate LED-Leiste für die Visualisierung des Zustands dieser speziellen Aufgabe angebracht wurde.

3.4 Implikationen

Sowohl die Arbeit von Song und Yamada als auch die Arbeit von Löffler et al. bestätigten, dass Emotionen durch Farben mit variierendem Erfolg kommuniziert werden können. Dieser Umstand verleiht der in [Unterabschnitt 3.1](#) formulierten Hoffnung, mithilfe von Farben Emotionen zu visualisieren und somit eine intuitive Codierung der Roboterzustände zu ermöglichen, Validität.

Häring et al. und Löffler et al. zeigen zudem, dass Emotionen mittels Bewegung verständlicher als mittels Farbe zu kommunizieren sind. Bewegungen zur Visualisierung von Emotionen für dieses Projekt zu nutzen, sprengt jedoch einerseits dessen Rahmen und ist andererseits kaum hilfreich für die Forschenden der Forschungsgruppe TAMS. Häufig erfordert das Lösen, der in [Unterabschnitt 1.2](#) erwähnten Aufgaben, die Verwendung der Extremitäten des PR2-Roboters. Dementsprechend sind hinderliche Bewegungen des Roboters zur Visualisierung seines Zustands beim Lösen von Aufgaben nicht erwünscht. In seltenen Fällen können solche unerwarteten Bewegungen zudem ein Sicherheitsrisiko darstellen.

Welche Farben zum Visualisieren von Zuständen Verwendung finden sollten, ist durch die präsentierten Erkenntnisse nicht eindeutig geklärt. Die Farbe Rot ist jedoch im [Unterabschnitt 3.2](#) einvernehmlich der Emotion Wut zugeordnet. Die Sinnhaftigkeit dieser Zuordnung ist durch die Studie von Löffler et al. gezeigt, in der Probanden die Farbe Rot überwiegend mit der Emotion Wut assoziierten. Des Weiteren kann, mutmaßlich aufgrund der Sozialisierung des modernen Menschen, die Farbe Rot als ein Warnsignal verstanden werden, was sich in der Bedeutung der Farbe im ISO-Standard IEC 60204-1 widerspiegelt. Die negative Konnotation der Farbe Rot macht diese zu einem guten und insbesondere intuitiven Indikator von Zuständen, die ein Abweichen vom Normalbetrieb signalisieren.

Eine, in [Anmerkung 1](#) beschriebene Komponente, kann einerseits voll funktionsfähig sein oder andererseits beliebig gravierende Warnungen, Fehler oder Defekte aufweisen, welche schlimmstenfalls einen Komplettausfall der Komponente verursachen. Eine Skala, die funktionsfähige Komponenten den Wert null und funktionsunfähige Komponenten den Wert eins zumisst, ermöglicht, mittels Zahlen zwischen null und eins, die Beurteilung der Zustände von Komponenten. Zahlen, die nicht dem Wert null entsprechen, indizieren demnach, dass die entsprechende Komponente Warnungen, Fehler oder Defekte aufweist.

Die Werte einer solchen Skala zu visualisieren ist mithilfe zweier Farben möglich. Hierzu ist einer Startfarbe der Wert null und einer Endfarbe der Wert eins zuzuordnen. Alle Zahlen zwischen null und eins können anschließend als Anteile der beiden Farben dargestellt werden. Ist der Zustand einer Komponente beispielsweise mit einem Drittel bezifferbar, so leitet sich die zugehörige Farbe aus zwei Dritteln der ersten Farbe und einem Drittel der zweiten Farbe ab.

Eine Farbe, die neutrale Zustände intuitiv visualisiert, lässt sich anhand der präsentierten Erkenntnisse nicht eindeutig bestimmen. Song und Yamada versuchen jedoch unter anderem mithilfe der Farbe Weiß die Emotion Gelassenheit auszudrücken, was ihnen in ihrem Experiment faktisch gelingt. Neutrale Zustände mit der Emotion Gelassenheit zu assoziieren ist sicherlich nicht abwegig. [Abbildung 6a](#) zeigt das Farbspektrum, das mit der Startfarbe Weiß und der Endfarbe Rot zur Visualisierung von Roboterzuständen zur Verfügung stünde.

Neben der Farbe Weiß scheint auch die Farbe Grün zur Visualisierung neutraler Zustände geeignet zu sein. Die Farbe Grün ist nach Herings [1] Gegenfarbentheorie die Gegenfarbe zur Farbe Rot. Naheliegender ist dementsprechend, der Farbe Grün die gegenteilige Bedeutung der Farbe Rot zuzuweisen. Aufgrund der hohen Intuitivität der Farbe Rot könnte auch die Bedeutung der Gegenfarbe Grün intuitiv verstanden werden. Des Weiteren spricht der ISO-Standard IEC 60204-1 der Farbe Grün die Bedeutung Normalzustand zu, was der Bedeutung eines neutralen Zustands stark ähnelt. [Abbildung 6b](#) zeigt das Farbspektrum, das mit der Startfarbe Grün und der Endfarbe Rot zur Visualisierung von Roboterzuständen zur Verfügung stünde.



(a) Linearer Farbverlauf mit Startfarbe Weiß und Zielfarbe Rot



(b) Linearer Farbverlauf mit Startfarbe Grün und Zielfarbe Rot

Abbildung 6: Die Farbwerte der Farbverläufe im RGB-Farbraum sind mittels folgender Funktion berechnet: $F(x) = x \cdot F_1 + (1 - x) \cdot F_2$. F steht für die RGB-Farbe, welche bei dem Mischungsverhältnis x der zwei Farben F_1 und F_2 erzeugt wird.

Geeignete animierte Farbmuster aus den präsentierten Erkenntnissen zu synthetisieren ist nicht möglich. Dieser Umstand ist den sehr speziellen Limitationen zuzuschreiben, welche mit der Verwendung von LED-Ringen zur Visualisierung von Roboterzuständen einhergehen. Animierte Augen finden zwar beispielsweise bei der Entwicklung von HERAs Kopf durch Grinten et al. ihren Platz, sind dort jedoch in Kombination mit Augenbrauen und mithilfe eines Displays dargestellt. Die Bedeutsamkeit von Augenbrauen zum Visualisieren von Emotionen kann anhand der Arbeit von Archevapanich et al. erkannt werden. [Abbildung 4b](#) zeigt, dass nur die Augenbrauen und der Mund, nicht aber die Augen, zur Visualisierung verschiedener Gesichtsausdrücke Verwendung finden.

Baraka und Veloso präsentieren in ihrer Arbeit eine interessante Formalisierung sowie einen interessanten Algorithmus zum Generieren von Animationsepisoden. Bedauerlicherweise sieht der Algorithmus und Teile des Formalismus kein Einlesen von externen, individuell gestalteten Animationen vor, weshalb beide Aspekte in dieser Arbeit keine Anwendung finden. Das Konzept von Animationsepisoden wiederum ist eine attraktive Überlegung, da ein ungehinderter Animationsfluss während der Visualisierung von Roboterzuständen auch in diesem Projekt erwünscht ist.

4 Implementation

[Abschnitt 4](#) befasst sich mit der erarbeiteten Hard- und Softwarelösung zum Erreichen des im [Unterabschnitt 1.3](#) formulierten Projektziels. Zunächst sind in [Unterabschnitt 4.1](#) die verwendeten elektronischen Bauteile vorgestellt und anschließend deren Zusammenspiel dargelegt. Das Kommunikationsprotokoll zwischen den elektrischen Bauteilen und dem PR2-Roboter, welches im Verlauf dieses Projekts erarbeitet wurde, ist schwer ohne grundlegende ROS-Kenntnisse einzuordnen. Ein Überblick ist demnach zunächst über ROS und anschließend über das erdachte Kommunikationsprotokoll in [Unterabschnitt 4.2](#) gegeben. [Unterabschnitt 4.3](#) stellt das System, das die Zustände einzelner Komponenten des PR2-Roboters ermittelt und diese anschließend aggregiert, vor. Ausgehend von

dem ermittelten Roboterzustand, ist in [Unterabschnitt 4.4](#) aufgezeigt, wie animierte Farbmuster abgespielt werden. [Unterabschnitt 4.5](#) präsentiert abschließend eine Übersicht des Gesamtsystems.

4.1 Elektronische Bauteile

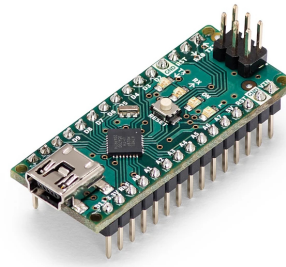
Ein NeoPixel, auch als WS2812 Integrated-Light-Source bezeichnet, ist eine simple, skalierbare und günstige LED-Einheit, welche das gesamte Farbspektrum darstellen kann. Ein Anschluss, den jeder NeoPixel zur Verfügung stellt, ermöglicht mithilfe eines Kontrollkabels das Einstellen beliebiger Farben. Die Freiheit bei der Wahl des Farbwerts macht jedoch den Gebrauch eines Mikrocontrollers, wie beispielsweise einem Arduino Mikrocontroller, und etwas Software für die Inbetriebnahme erforderlich. Adafruit, der Hersteller der NeoPixel, stellt diverse, quelloffene Bibliotheken für die Arduino Mikrocontroller Familie bereit, um die Kommunikation mit NeoPixeln zu vereinfachen.

Mehrere NeoPixel lassen sich in Reihe schalten, indem deren Versorgungsspannung, sowie deren Kontrollsignal an den jeweils nächsten NeoPixel weitergegeben wird. Ein Mikrobaustein, den jeder NeoPixel verbaut, ermöglicht in Reihe geschalteten NeoPixeln weiterhin individuelle Farbwerte anzunehmen. Diese Eigenschaft erlaubt NeoPixeln, kombiniert in den verschiedensten Formen aufzutreten. Dieses Projekt verwendet zwei NeoPixel-Ringe, welche jeweils aus 16 NeoPixeln bestehen, und von denen einer in [Abbildung 7a](#) dargestellt ist.

Der Arduino Nano, ein kleiner, Steckbrett freundlicher Mikrocontroller, welchem der ATmega328 Prozessor zugrunde liegt, ist in diesem Projekt eingesetzt. 16 digitale, sechs digitale und PWM fähige und acht analoge Ein-/Ausgabepins stellt dieser unter anderem zur Verfügung. Ferner ist der Nano mit 32KB FLASH-Speicher, wovon 2KB für dessen Bootloader reserviert sind, und 2KB SRAM ausgestattet. Der Mikrocontroller bezieht Strom über ein USB-Kabel, welches zudem serielle Kommunikation via des UART-Protokolls ermöglicht. [Abbildung 7b](#) zeigt den in diesem Projekt verwendeten Arduino Nano.



(a) NeoPixel Ring ([14] entnommen)



(b) Arduino Nano ([20] entnommen)

Abbildung 7: NeoPixel-Ringe sind mit bis zu 24 NeoPixeln pro Ring zu erwerben. Der Arduino Nano ist einer der kleinsten Mikrocontroller der Arduino Familie, weshalb er sich für dieses Projekt anbietet.

Beide NeoPixel-Ringe sind jeweils über zwei Kabel mit dem Arduino Nano verbunden. Ein Kabel gewährleistet die Kommunikation der einzelnen NeoPixel mit dem Mikrocontroller, während das andere Kabel die Stromversorgung sichert. Die Versorgungsspannung beziehen beide NeoPixel-Ringe über den 5V-Pin des Nanos. Sollten zusätzliche NeoPixel Verwendung finden, so müsste ein externes Netzteil zum Einsatz kommen, das die erhöhte Stromlast bewältigen kann. Zwischen den NeoPixel-Ringen und der Stromquelle ist nach Adafruit [21] ein $500\ \mu\text{F}$ - $1000\ \mu\text{F}$ bei $6,3\text{V}$ Kondensator vorgesehen, welcher mutmaßlich Spannungsspitzen glätten und somit gleichbleibende Helligkeiten aller NeoPixel gewährleisten soll. Mit ausschließlich 32 NeoPixeln, jeweils 16 NeoPixeln pro NeoPixel-Ring, scheinen solche Kondensatoren jedoch nicht notwendig zu sein, weshalb diese nicht eingesetzt sind. Ein NeoPixel-Ring ist an Pin 6, während der andere NeoPixel Ring an Pin 7, mit dem Arduino Nano, via eines Kommunikations- beziehungsweise Kontrollkabels, verbunden ist. Ein $300\ \Omega$ - $500\ \Omega$ Widerstand zwischen Kontrollleitung und NeoPixel-Ring ist nach Adafruit zudem vorgesehen. Beide Widerstände sind nach Angaben des Herstellers [22] jedoch bereits in die Ringe integriert. [Abbildung 8](#) zeigt den beschriebenen Aufbau.

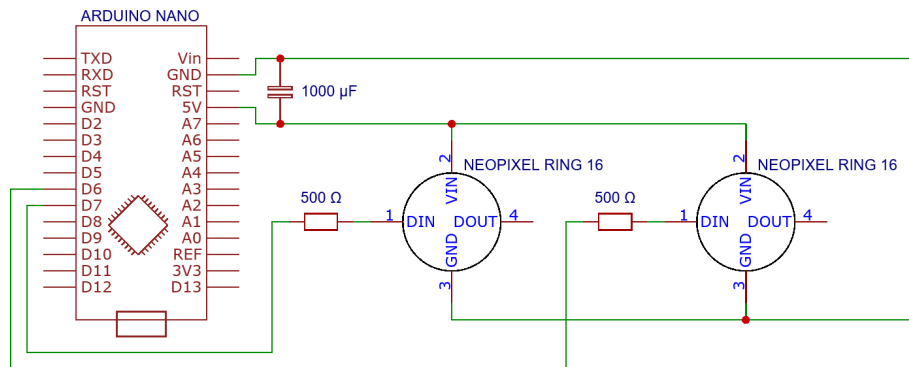


Abbildung 8: Empfohlene Verschaltung des Arduinos mit zwei NeoPixel-Ringen nach Adafruit. Widerstände in den Datenleitungen sind aufgrund Adafruits Empfehlung dargestellt, obwohl diese laut dem Hersteller bereits in NeoPixel-Ringen verbaut sind. Negative Auswirkungen sind durch zusätzliche Widerstände in den Datenleitungen nicht zu erwarten. Die Verschaltung, die in diesem Projekt gewählt ist, tilgt den $1000\ \mu\text{F}$ Kondensator sowie beide $500\ \Omega$ Widerstände.

4.2 Kommunikation

ROS, ein Akronym für Robot-Operation-System, ist ein flexibles, quelloffenes Framework, das überwiegend in Kombination mit Robotern Verwendung findet. Die Bezeichnung Operation-System, also Betriebssystem, ist jedoch irreführend, da ROS lediglich als Plattform zum Verarbeiten und Austauschen von Infor-

mationen dient. ROS übernimmt folglich nicht die Aufgaben eines klassischen Betriebssystems, sondern delegiert diese an ein zugrundeliegendes Linux Derivat.

Ein Roboter besteht nach [Anmerkung 1](#) aus einer Vielzahl von Komponenten. Ein Großteil der Komponenten ist Sensoren, Programmen oder Aktuatoren zuzuordnen. Sensoren bilden in der Regel die Umwelt eines Roboters auf abstrakte Informationen ab. Kameras des PR2-Roboters beschreiben beispielsweise Objekte in dessen Sichtfeld als dreidimensionale Punktwolken. Die durch Sensoren virtuell beschriebene Umwelt des Roboters wird fortlaufend von kompakten, hoch spezialisierten Programmen analysiert. Auf Grundlage der getätigten Analysen sind weitere Programme kontinuierlich damit beschäftigt, das Verhalten des Roboters in der physischen Welt zu steuern. Ein solches Unterfangen ist mithilfe der vielen Aktuatoren des Roboters zu bewerkstelligen. Programmweisungen erwirken Reaktionen derartiger Aktuatoren, was beispielsweise das Greifen eines Objekts oder das Heben eines Arms ermöglicht.

Der Informationsfluss zwischen Sensoren und Programmen als auch zwischen Programmen und Aktuatoren ist mit einer Vielzahl an Komponenten unverständlich. Probleme, die damit einhergehen, wie beispielsweise ein inkompatibler Quellcode interdisziplinärer Projekte, versucht ROS durch die Systematisierung des Informationsaustauschs zwischen Komponenten zu lösen. Jeder austauschfreudigen Komponente wird das Erzeugen eines ROS-Knotens ermöglicht, mittels welchem diese Informationen, in Form von sogenannten Nachrichten, empfangen und versenden kann. Die Programmiersprachen PYTHON und C++ sind von ROS für das Verfassen von ROS-Knoten vorgesehen. ROS-Nachrichten, die als Behälter für Informationen dienen, lassen sich über eine ROS eigene Syntax beschreiben. ROS-Pakete, die Knoten und Nachrichten bündeln, ermöglichen das Definieren kompakter Einheiten, welche wiederverwendbar und übersichtlich sind.

ROS-Knoten und deren Kommunikationspfade sind als Graph darstellbar. Die Herkunft der Bezeichnung ROS-Knoten ist auf diese Eigenschaft zurückzuführen. ROS-Knoten entsprechen den Knoten, Kommunikationspfade den Kanten derartiger Graphen. [Abbildung 9](#) zeigt einen imaginären ROS-Graphen.

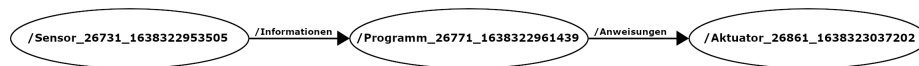


Abbildung 9: Ein imaginärer ROS-Graph, welcher den Informationsfluss von Sensoren über Programme zu Aktuatoren darzustellen versucht. Das Rückschlagsymbol zu Beginn jedes Bezeichners ist Teil dessen globalen Namens und gibt zudem Auskunft über dessen Namensraum.

Nach [Abschnitt 5](#) bestehen animierte Farbmuster aus einzelnen hintereinander abgespielten Farbmustern. Wie viele Farbmuster in einem bestimmten Zeitraum abzuspielen sind und wie oft ein animiertes Farbmuster zu wiederholen ist, entsprechen Parametern, die ein animiertes Farbmuster zusätzlich beschreiben. Der Speicherplatzbedarf eines animierten Farbmusters, welches beispielsweise 16 einzelne Farbmuster zählt, ist unter Berücksichtigung aller Parameter mit 773 Bytes anzugeben. Die Verwendung zweier NeoPixel-Ringe erfordert jedoch zwei animierte Farbmuster, was eine Verdopplung des Speicherplatzbedarfs zur Folge hat, welcher demnach mit 1546 Bytes bezifferbar ist. [Gleichung 1](#) zeigt den Hergang dieser Werte. Die Speicherkapazität des Arduino Nanos ist mit zwei jeweils

$$\underbrace{2}_{(1)} \cdot \left(\underbrace{16 \cdot 16 \cdot 3 \cdot 1 \text{ B}}_{(2)} + \underbrace{4 \text{ B} + 1 \text{ B}}_{(3)} \right) = 1546 \text{ B} \quad (1)$$

(1) Zwei NeoPixel-Ringe; (2) 16 Farbmuster mit jeweils 16 NeoPixeln, die drei Farbwerte von je einem Byte erwarten; (3) Vier Byte, welche die Wiedergabegeschwindigkeit des animierten Farbmusters bestimmen, sowie ein Byte für die Anzahl an Wiederholungen

16 Farbmuster umfassenden animierten Farbmuster somit bereits zum Großteil erschöpft. Wird der Programmcode, der zur Laufzeit im Arbeitsspeicher untergebracht ist, zudem berücksichtigt, so ist festzustellen, dass umfangreichere animierten Farbmuster dort nicht vollständig unterzubringen sind. Ferner ist die Möglichkeit, eine Vielzahl an animierten Farbmuster im Arbeitsspeicher des Arduino Nanos abrufbereit zu hinterlegen, aufgrund der Speicherknappheit auszuschließen. Eine ununterbrochene Wiedergabe von Farbmustern, welche für die Visualisierung der Zustände des PR2-Roboters erwünscht ist, erfordert demnach ein kontinuierliches Nachladen von Farbmustern. Das Laden animierter Farbmuster aus dem FLASH-Speicher des Arduino Nanos oder das Zuführen dieser via der USB-Verbindung, entspricht hierbei dem Lösungsspielraum. Letzterer Lösungsansatz ist in diesem Projekt umgesetzt, da eine derartige Lösung eine hohe Flexibilität in Aussicht stellt.

Der PR2-Roboter ist mittels eines USB-Kabels mit dem Arduino Nano verbunden, welches eine Kommunikation über das serielle UART-Protokoll ermöglicht. Diese Kommunikation ist aufseiten des Roboters mithilfe eines ROS-Knoten umgesetzt, der in der Programmiersprache PYTHON verfasst ist und sich der Bibliothek PYSERIAL bedient. Dem Arduino ist es ohne zusätzliche Bibliotheken möglich seriell via UART zu kommunizieren.

Die Bibliothek ROSSERIAL, welche diversen Mikrocontrollern zur Verfügung steht und Kommunikation zwischen ROS-Instanzen und Mikrocontrollern ermöglicht, findet keine Verwendung, da deren Gebrauch die geringe Speicherkapazität des Nanos unverhältnismäßig stark strapaziert.

Das Kommunikationsprotokoll zwischen dem Arduino Nano und dem ROS-Knoten sieht drei Arten von Nachrichten vor, welche in [Abbildung 10](#) gezeigt sind. Eine Instruktion weist den Mikrocontroller an, diverse Tätigkeiten auszuführen. Das Abspielen, Pausieren und Nachladen animierter Farbmuster ist dem Mikrocontroller hierdurch beispielsweise mitteilbar. Datennachrichten dienen als Behälter für einzelne Farbmuster, die im Anschluss einer Nachlade Instruktion versandt werden.

Synchrone Protokollstrukturen erhöhen die Robustheit beim Kommunizieren. Ein Erwidern von Nachrichten erwirkt geringfügige Synchronität während der Kommunikation. Ein einigermaßen robuster Nachrichtenaustausch ist erwünscht, weshalb der Arduino Nano auf jede Instruktion mit einer Statusmeldung, der dritten Nachrichtenart, antwortet. Instruktionen und Datennachrichten fließen von ROS-Knoten zum Arduino Nano, während Statusmeldungen den entgegengesetzten Weg nehmen. [Abbildung 11](#) zeigt Automaten, welche das erarbeitete Kommunikationsprotokoll beschreiben.

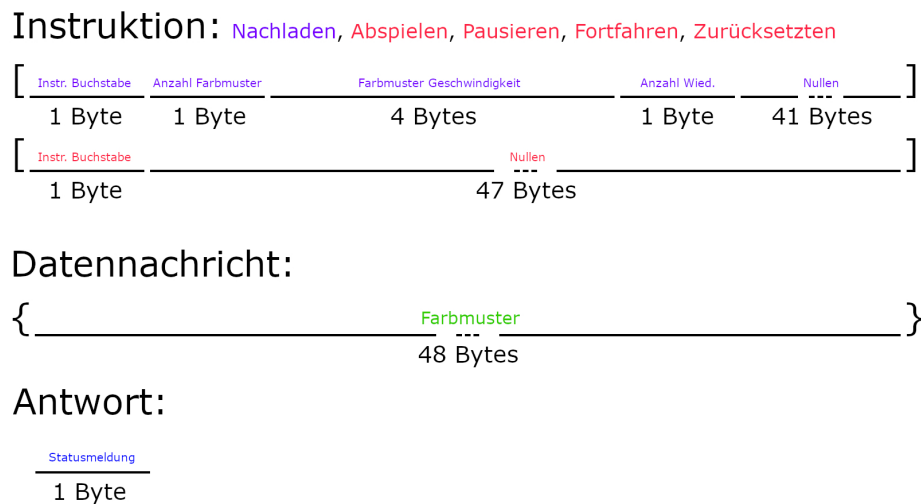


Abbildung 10: Instruktionen als auch Datennachrichten setzen sich jeweils aus exakt 50 Bytes zusammen. Diese Eigenschaft vereinfacht den Mechanismus zum Empfangen von Nachrichten aufseiten des Arduino Nanos, was wiederum dessen Quellcode übersichtlicher gestaltet. Der Umfang von 50 Bytes ist mit den 48 Bytes großen Farbmustern plus den zwei Bytes großen, ASCII codierten Begrenzungszeichen zu erklären. Instruktionen beginnen und enden mit eckigen Klammern, während Datennachrichten geschweiften Klammern umfassen. Antworten des Arduino Nanos bestehen ausschließlich aus einem Byte, das genau eine Statusmeldung codiert. Jede der insgesamt fünf verschiedenen Instruktionen ist mit einem ASCII codierten Großbuchstaben eindeutig bezeichnet. Aufgrund jener Bezeichner vermag der Arduino mit der jeweiligen Instruktion umzugehen.

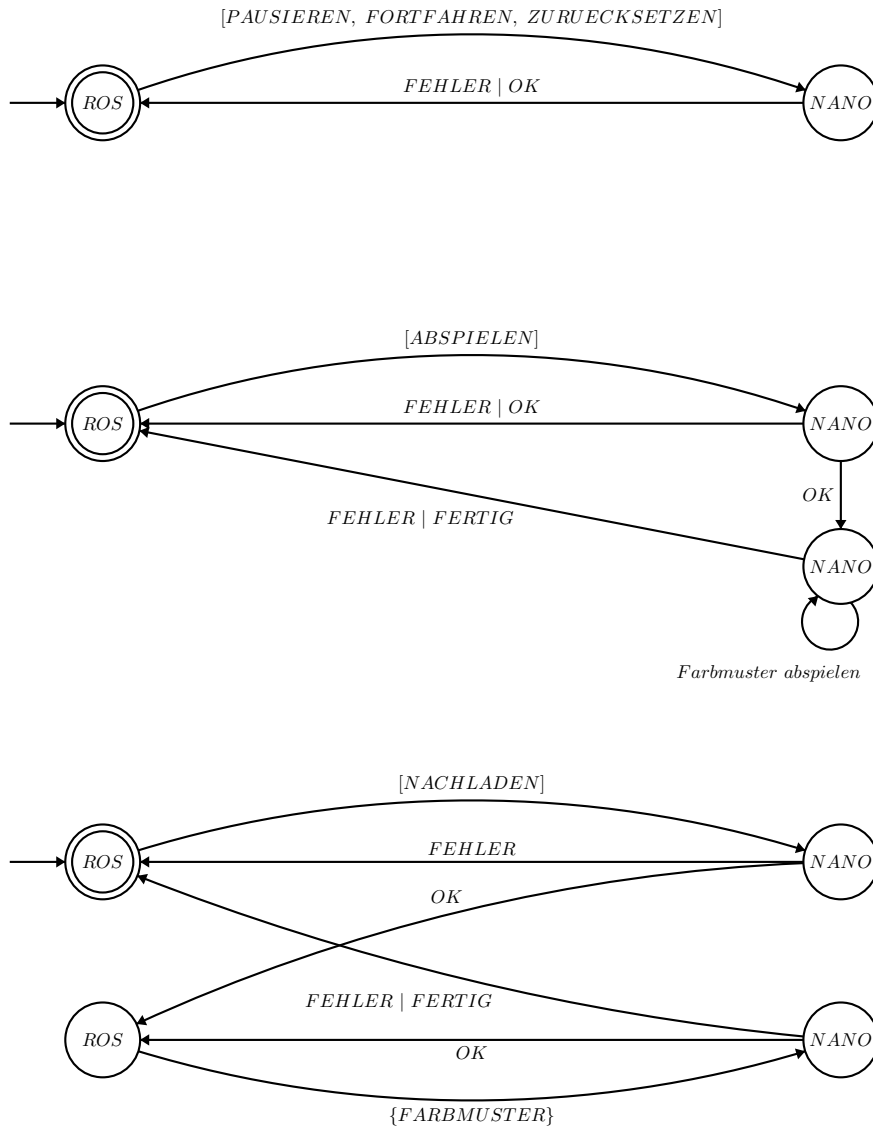


Abbildung 11: Instruktionen, die eckige Klammern umgeben, sind ausschlaggebend für den jeweiligen Protokollverlauf. Statusmeldungen des Nanos sind in den präsentierten Automaten zugunsten der Übersichtlichkeit auf OK, FEHLER und FERTIG reduziert. Da Statusmeldungen des Nanos keine begrenzenden Symbole umschließen, sind diese ebenso an den Kanten der Automaten inexistent. Senkrechte Striche trennen mögliche Statusmeldungen voneinander. Die Schleife des zweiten Automaten beschreibt einen internen Prozess des Mikrocontrollers und ist dementsprechend nicht als Statusmeldung zu verstehen.

4.3 Roboterzustand

Komponenten, die beabsichtigen Informationen bereitzustellen oder zu empfangen, können ROS-Knoten implementieren. Der ROS-Knoten einer Komponente fungiert als Indikator für dessen Zustand. Einerseits, nach [Anmerkung 2](#), dienen hierzu die Laufzeitphasen von ROS-Knoten, andererseits die Informationen beziehungsweise Nachrichten, welche diese versenden. Nach [Unterabschnitt 3.4](#) ist das Bewerten einzelner Komponenten mithilfe einer Skala möglich. Software-schnipsel, sogenannte Beobachter, welche entweder Laufzeitphasen analysieren oder Nachrichtenströme abhören, ordnen ROS-Knoten und somit deren korrespondierende Komponenten in besagte Skala ein. Beobachter, die Laufzeitphasen analysieren, sind in ROS-Knoten integriert. Nicht integrierte Beobachter, also jene, die Nachrichtenströme analysieren, sind als alleinstehende ROS-Knoten umgesetzt. Alle Beobachter senden ihrer Bewertungen an einen zentralen ROS-Knoten, welcher aus allen Einsendungen kontinuierlich einen Gesamtsystem beschreibenden Wert synthetisiert. Die Zustände der einzelnen Komponenten bleiben dabei sichtbar, was Flexibilität bei der Visualisierung gewährleistet. Wie die verschiedenen Zustände der Komponenten zu aggregieren sind oder ob dies überhaupt geschehen soll, ist mittels ROS-Parametern regulierbar.

Das System, welches soeben dargelegt wurde, ist von den Forschenden des Fachgebiets Echtzeitsysteme der Leibniz Universität Hannover implementiert und unter folgendem Link veröffentlicht worden: <https://github.com/luhrts/monitoring>. Die Lösung, die von diesem Projekt umgesetzt ist, entspricht im Allgemeinen deren Implementation. Im Detail sind jedoch einige Vereinfachungen vorgenommen worden, um die Komplexität zu minimieren. Ein Übersetzerknoten ist zudem im Einsatz, welcher ermöglicht, beide Systeme parallel zu betreiben und somit von den bereits erdachten Beobachtern des Vorbildsystems zu profitieren. [Abbildung 12](#) zeigt schematisch den beschriebenen Aufbau.

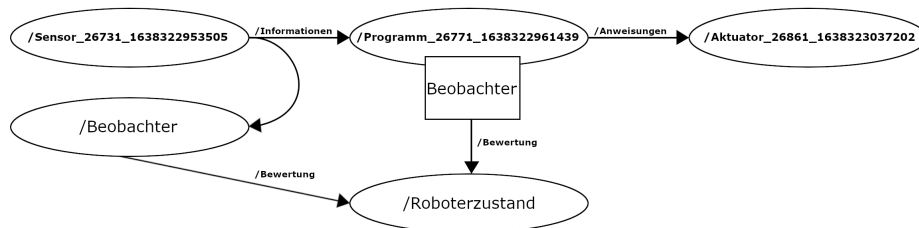


Abbildung 12: Als Rechteck angedeutet, ein Beobachter der in einem bestehenden ROS-Knoten integriert ist. Zudem angedeutet, ein Beobachter der Informationen eines Sensors analysiert.

4.4 Visualisierung

Das Gesamtsystem zum Visualisieren des PR2-Roboterzustands besteht aus elektrischen Bauteilen, Quellcode des Arduino Nanos und einem ROS-Paket. Die verwendeten elektronischen Bauteile und deren Verschaltung ist bereits in [Unterabschnitt 4.1](#) dargestellt. [Unterunterabschnitt 4.4.1](#) erklärt knapp den Quellcode des Arduino Nanos zum Abspielen animierter Farbmuster. [Unterunterabschnitt 4.4.2](#) geht anschließend etwas ausführlicher auf die entwickelten ROS-Paket ein.

4.4.1 Mikrocontroller Quellcode

Die Hauptschleife, welche der Arduino Nano kontinuierlich durchläuft, ist in [Quellcode 1](#) gezeigt. Zeilen 3 - 14 befassen sich mit der Abarbeitung eingehender Nachrichten. In den Zeilen 8 und 9 ist zu sehen, dass mithilfe des in [Unterabschnitt 4.2](#) beschriebenen Nachrichtenaufbaus, insbesondere mit den Begrenzungszeichen, identifiziert wird, wie mit den eingegangenen Nachrichten umzugehen ist. Falls keine vollständige Nachricht vorliegt, wird in Zeile 15 entweder eine eingehende Nachricht gelesen oder in Zeile 16 ein geladenes animiertes Farbmuster abgespielt, falls dies dem Arduino Nano mitgeteilt wurde. Die für

```
1 void loop()
2 {
3   if (rcvd_msg)
4   {
5     char startMarker = msg[0];
6     char endMarker   = msg[MSG_SIZE - 1];
7
8     if      (startMarker == '[' && endMarker == ']') handle_instruction();
9     else if (startMarker == '{' && endMarker == '}') handle_data();
10    else                                           handle_wrong_format();
11
12    rcvd_msg = false;
13    msg_size = 0;
14  }
15  else if (Serial.available() > 0)      read_serial();
16  else if (playback && animation_loaded) play_animation();
17 }
```

Quellcode 1: Hauptschleife im Quellcode des Arduino Nanos.

die Visualisierung zuständige Funktion `play_animation()` ist in [Quellcode 2](#) dargestellt. Zeile 5 gewährleistet die Einhaltung einer spezifizierten Wiedergabegeschwindigkeit. Die Funktion `delay()` findet dabei keine Verwendung, da deren Einsatz die Responsivität des Systems negativ beeinträchtigt. Die Zeilen 8 bis 10 deuten den Quellcode zur Darstellung zweier Farbmuster auf je einem NeoPixel-Ring mithilfe der in [Unterabschnitt 4.1](#) erwähnten Bibliotheken der Firma Adafruit an. Aus Gründen der Übersichtlichkeit ist jener Quellcodeabschnitt lediglich angedeutet. Zeile 21 ermöglicht die fortlaufende Wiederholung des aktuell geladen animierten Farbmusters, falls dies erwünscht ist. Die Angabe des maximal erlaubten Werts, 255, für die Anzahl an Wiederholungen beim Nachladen eines animierten Farbmusters aktiviert diese Funktionalität.

```

1 void play_animation()
2 {
3   if (curr_ensemble < num_ensembles)
4   {
5     if (millis() > frame_time + start_time)
6     {
7
8       ...
9       // display frame
10      ...
11
12      start_time = millis();
13      curr_ensemble++;
14    }
15  }
16
17  if (curr_ensemble >= num_ensembles)
18  {
19    if (millis() > frame_time + start_time)
20    {
21      if (num_iterations != 0xFF) curr_iteration++;
22      if (curr_iteration >= num_iterations)
23      {
24        reset_playback();
25        respond(0x4F);
26      }
27      else curr_ensemble = 0;
28    }
29  }
30 }

```

Quellcode 2: Funktion, die das Darstellen animierter Farbmuster ermöglicht, im Quellcode des Arduino Nanos.

4.4.2 ROS-Paket (Visualising)

Das ROS-Paket, das die Darstellung des Roboterzustands bewerkstelligt, besteht aus der Implementation eines ROS-Knotens, mehreren ausführbaren Skripten und Funktionalität bereitstellenden PYTHON Klassen. Der ROS-Knoten, ebenso wie die Skripte, bedienen sich der in den Klassen angebotenen Methoden und im Falle der Skripte, stellen deren Funktionalität dem Nutzer zur Verfügung. Ein Klassendiagramm besagter Klassen ist in [Abbildung 13](#) gezeigt.

Animierte Farbmuster, die sich aus einzelnen Farbmustern, einer Wiedergabegeschwindigkeit und der Anzahl ihrer Wiederholungen zusammensetzen, sind durch die Klassen `Animation`, `Ensemble`, `Frame` und `Pixel` definiert. Ein Objekt der Klasse `Pixel` beschreibt den Farbwert eines NeoPixels, die in Objekten der Klasse `Frame`, je 16 Stück, zu einem Farbmuster zusammengefasst sind. Angesichts des Gebrauchs von zwei NeoPixel-Ringen besteht die Notwendigkeit jeweils zwei Farbmuster parallel darzustellen. Ein Objekt der Klasse `Ensemble` enthält dementsprechend jeweils zwei Farbmuster, eines für den linken und eines für den rechten NeoPixel-Ring. Objekte der Klasse `Animation` führen eine Liste an beliebig vielen `Ensemble` Objekten und speichern zudem die Wiedergabegeschwindigkeit dieser und ihre gemeinsamen Wiederholungen. Eine Menge von einzelnen Farbmustern lässt sich, wie in [Quellcode 3](#) dargestellt, als Da-

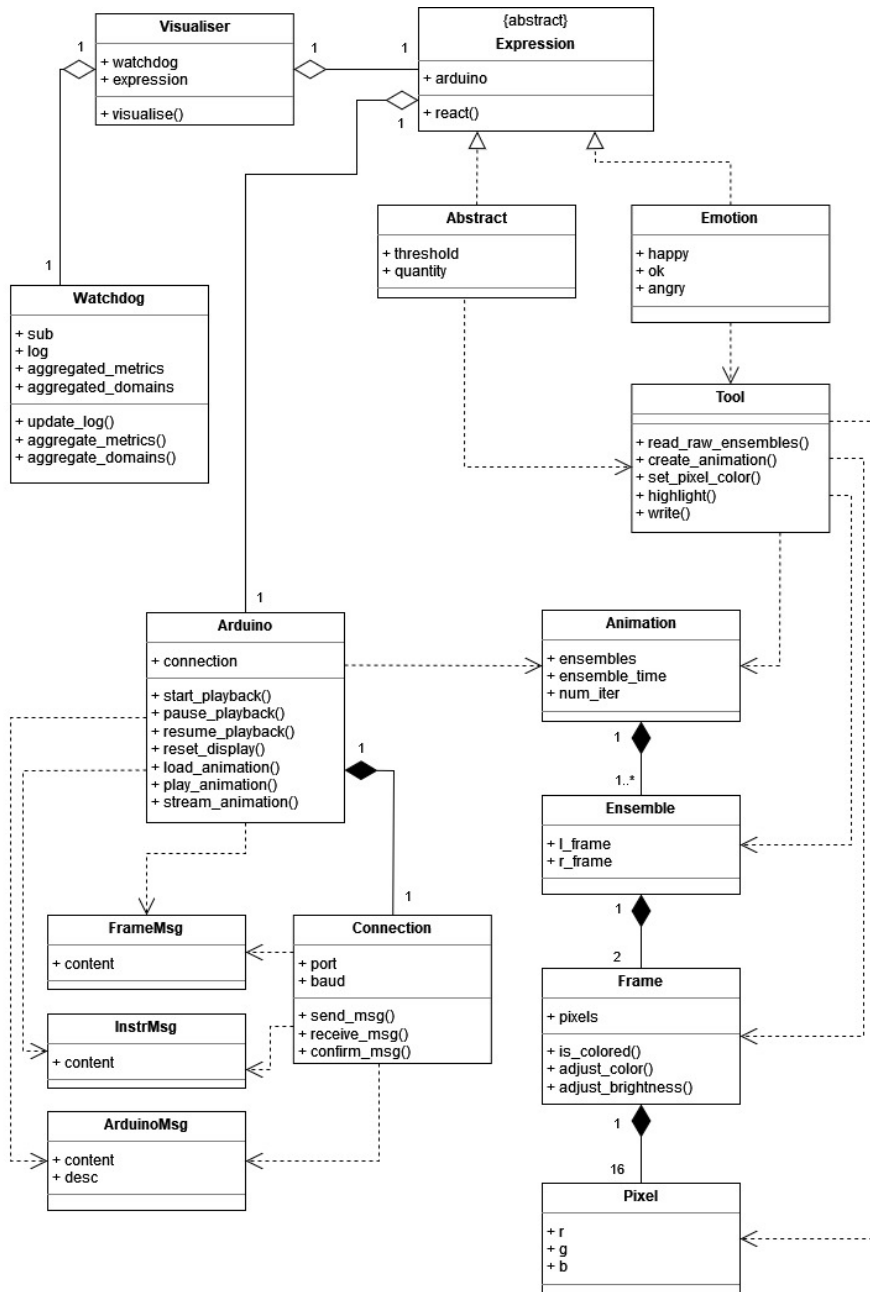


Abbildung 13: UML Klassendiagramm, das die Beziehungen zwischen den Klassen des ROS-Pakets beschreibt. Die abstrakte Klasse `Expression` ist zunächst nur aus Gründen der Verständlichkeit dem Klassendiagramm beigelegt worden, sie ist momentan nicht im ROS-Paket enthalten. Zudem sind nicht alle Attribute und Methoden der jeweiligen Klassen aufgeführt, um die Übersichtlichkeit der Darstellung zu erhöhen.

4.5 Gesamtsystem

Ein Überblick über das entwickelte System ist in [Abbildung 14](#) schematisch gegeben. Dessen Quellcode kann über folgenden Link abgerufen werden: <https://gitlab.rrz.uni-hamburg.de/bav6096>.

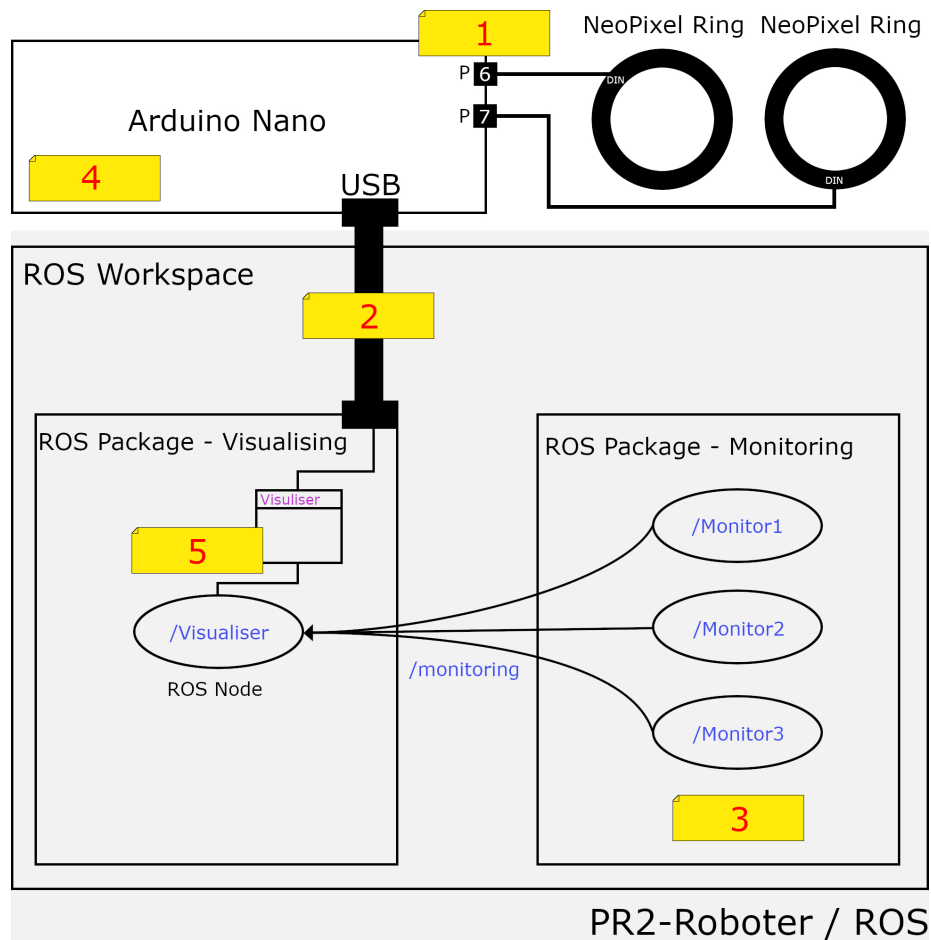


Abbildung 14: Systeme, die aufseiten des PR2-Roboters agieren, sind mit grauer Farbe hinterlegt, wohingegen Systeme, die unabhängig von diesem existieren, mit weißer Farbe untermalt sind. **1:** Die Verschaltung der elektronischen Bauteile kann dem [Unterabschnitt 4.1](#) entnommen werden. **2:** Die Kommunikation zwischen dem Arduino Nano und ROS ist in [Unterabschnitt 4.2](#) beschrieben. **3:** Das Beobachtersystem, in [Unterabschnitt 4.3](#) erläutert, ist im ROS-Paket Monitoring untergebracht. **4:** Der Quellcode zum Darstellen animierter Farbmuster ist in [Unterabschnitt 4.4.1](#) präsentiert. **5:** Der ROS-Knoten Visualiser instanziiert lediglich ein Objekt der Klasse Visualiser. Jene Objekte empfangen und aggregieren einzelne Komponentenzustände und visualisieren diese parallel dazu. Wie die genannten Tätigkeiten intern gehandhabt werden, ist in [Unterabschnitt 4.4.2](#) beschrieben.

5 Visualisierungskonzepte

Zwei Konzepte zur intuitiven Visualisierung eines Roboterzustands sind im [Abschnitt 5](#) vorgestellt. Diese Dualität ist der in [Unterabschnitt 3.1](#) gewonnen Erkenntnis geschuldet, Intuitivität und Informationsgehalt gegeneinander abwägen zu müssen. [Unterabschnitt 5.1](#) präsentiert ein Konzept, das mithilfe von Gesichtsausdrücken einen Roboterzustand zu visualisieren versucht. Die Verwendung von Gesichtsausdrücken und somit die Visualisierung von Emotionen erwirkt für eine Vielzahl von Personen eine hohe Intuitivität, ist jedoch wenig hilfreich beim Bestimmen fehlerhafter Komponenten; der vermittelte Informationsgehalt ist somit gering. Unterstützt ist die Entwicklung eines derartigen Konzepts durch die Existenz des in [Unterabschnitt 4.3](#) erwähnten Wertes, der Roboter in ihrer Gesamtheit beschreibt. [Unterabschnitt 5.2](#) erläutert hingegen ein Visualisierungskonzept, das versucht, den Zustand einzelner Roboterkomponenten zu kommunizieren. Aufgrund der Fülle an Roboterkomponenten ist der zu vermittelnde Informationsgehalt hoch, was zu einer weniger intuitiven Visualisierung führt. Die Zustände einzelner Komponenten sind durch das in [Unterabschnitt 4.3](#) beschriebene Beobachtersystem gegeben und entsprechen Werten der in [Unterabschnitt 3.4](#) angeführten Skala.

5.1 Emotionen und Gesichtsausdrücke

Dem in [Unterabschnitt 3.1](#) formulierten Gedanken, Roboterzustände auf Emotionen abzubilden, kommt das folgende Konzept nach. Hierbei ist die Entscheidung, lediglich drei Emotionen in Betracht zu ziehen, anzumerken, welche aus den notwendigen Abwägungen zwischen Intuitivität und Informationsgehalt der Visualisierung hervorgeht.

Der Einleitung von [Abschnitt 5](#) ist zu entnehmen, dass ein Roboterzustand in Form einer Zahl, die Werte zwischen null und eins annimmt, vorliegt. Der Wertebereich dieser Zahl, ein Intervall mit der Untergrenze null und der Obergrenze eins, kann ebenso wie in [Unterabschnitt 3.4](#) als Skala aufgefasst werden. Solch eine Skala mithilfe von drei Emotionen zu visualisieren, erfordert deren Unterteilung in drei diskrete Abschnitte. Jene Abschnitte sind in diesem Projekt jeweils in einheitlicher Größe gewählt. Anschließend ist das Definieren einer Abbildung, die jedem Teilintervall der Skala eine Emotion zuordnet, möglich. Wie [Unterabschnitt 3.4](#) anführt, ist mit dem Wert null eine fehlerfreie und mit dem Wert eins eine zutiefst fehlerhafte Komponente, in einer entsprechenden Skala, beschrieben. Jener [Unterabschnitt](#) legt des Weiteren dar, dass sich die Emotionen Gelassenheit und Wut plausibel mit ebendiesen Werten, null und eins, identifizieren lassen. Das Unterteilen der Skala in drei Abschnitte erfordert allerdings neben Gelassenheit und Wut eine dritte Emotion. Hierbei ist die Wahl auf die Emotion Fröhlichkeit gefallen, da diese mithilfe der NeoPixel-Ringe nachvollziehbar zu visualisieren ist. Jene Entscheidung verschiebt jedoch die soeben angesprochene Zuordnung von Werten zu Emotionen. Fröhlichkeit ist nun dem Wert null und somit dem ersten, null enthaltenden Teilintervall zugeordnet. Der Emotion Gelassenheit ist das zweite, mittlere Teilintervall zu-

geschrieben. Teilintervall drei, das den Wert eins enthält, ist, der Argumentation von [Unterabschnitt 3.4](#) weiterhin genügend, die Emotion Wut zugeteilt.

Gesichtsausdrücke zur Visualisierung von Emotionen zu verwenden ist nahelegend. Die Suche konkreter Gesichtsausdrücke für die Visualisierung der drei Emotionen hat sowohl tierische als auch menschliche Expressionen einbezogen. Zur Darstellung der Gesichtsausdrücke findet, trotz der Einführung einer dritten Emotion, das in [Abbildung 6a](#) dargestellt Farbspektrum Anwendung. Ist der Roboterzustand beispielsweise mit einem Viertel gegeben, so fällt dieser Wert in das erste Teilintervall der Bewertungsskala, was die Darstellung des Fröhlichkeit repräsentierenden Gesichtsausdrucks auf den NeoPixel-Ringen erwirkt. Zudem wird jene Darstellung in der nach [Abbildung 6a](#) dem Wert ein Viertel entsprechenden Farbe koloriert. Fröhlichkeit wird von Song und Yamada einerseits der Farbe Grün und von Löffler et al. andererseits der Farbe Gelb zugeordnet. Dient die Arbeit ersterer Autoren als Vorlage, so ist [Abbildung 6b](#) als Farbspektrum zu verwenden. Eine Anlehnung an die Arbeit letzterer Forschungsgruppe lässt jedoch [Abbildung 6a](#) als Farbspektrum, aufgrund der Ähnlichkeit zwischen der Farbe Geld und der Farbe Weiß, angebracht erscheinen. Ersteres Farbspektrum wurde bereits in [Unterabschnitt 5.2](#) verwendet, weshalb letzteres nun in diesem Konzept Anwendung findet. [Tabelle 4](#) zeigt sowohl die Vorlagen der Gesichtsausdrücke, welche die drei gewählten Emotionen visualisieren, als auch deren Umsetzung auf den NeoPixel-Ringen. [Abbildung 15](#) zeigt die Emotionen Fröhlichkeit und Wut durch die NeoPixel-Ringe visualisiert, während diese provisorisch am PR2-Roboter befestigt sind.







| Fröhlichkeit | Gelassenheit | Wut |
|---|---|---|
|  |  |  |
|  |  |  |

Tabelle 4: Die Emojis, welche Fröhlichkeit und Gelassenheit repräsentieren, entsprechen ohne Anpassungen denen der Firma Apple. Das Emoji, welches Wut visualisiert, ist aus drei verschiedenen Emojis zusammengesetzt. Das originale, Kreuzaugen enthaltende Emoji, erschien nicht wütend genug.



Abbildung 15: Zustandsvisualisierung des PR2-Roboters mithilfe von NeoPixel-Ringen und einem Visualisierungskonzept, das sich auf Emotionen und Gesichtsausdrücke stützt.

Im Verlauf der Implementierung ist die Bedeutung kenntlicher Zustandsaktualisierungen hervorgetreten. Insbesondere geringe Zustandsänderungen sind unter Verwendung eines stetigen Farbverlaufs zur Visualisierung kaum sichtbar. Das periodische Abspielen eines animierten Farbmusters zur Indizierung eines aktualisierten Roboterzustands, ist dementsprechend als Lösung umgesetzt. Ein visueller Hinweis, der einen aktualisierten Roboterzustand andeutet, ist nicht der einzige Vorteil, den eine solche Lösung mit sich bringt. Das Definieren von Animationsepisoden, die mit einer animierten Zustandsaktualisierung beginnen und mit einer Darstellung des zum Roboterzustand passenden Gesichtsausdrucks enden, ist zudem möglich. Wie in [Unterabschnitt 3.4](#) erwähnt, ermöglichen jene Animationsepisoden einen ununterbrochenen Animationsfluss, der wiederum angenehm zu betrachten ist.

5.2 Statuslichter und Buchstaben

Eine Vielzahl von Komponentenzuständen darzustellen ist mit ausschließlich zwei NeoPixel-Ringen ein kniffliges Unterfangen. Eine Strategie, die vermeintlich vielversprechend erscheint, besteht darin, zunächst beide NeoPixel-Ringe zur Visualisierung des Zustands einer Komponente zu verwenden. Ein Überblick über die Zustände aller interessanten Komponenten ist anschließend mit der periodischen, sequenziellen Wiedergabe der korrespondierenden Visualisierungen zu erlangen. Strategien dieser Art haben den Vorteil, den Zustand jeder Komponente mit dem gesamten Ausdrucksspektrum beider NeoPixel-Ringe visualisieren zu können, was deren intuitiven Verständlichkeit dienlich ist. Ein gravierender Nachteil besteht jedoch darin, nicht auf einen Blick den gesamten Roboterzustand als Anhäufung einzelner Komponentenzustände betrachten zu

können. Dieser Umstand hat zur Konsequenz, dass die rasche Identifizierung der fehlerbehaftetsten Komponenten, welche meist das größte Interesse hervorrufen, oft nicht möglich ist.

Für dieses Projekt ist demnach eine Visualisierung gewählt worden, die beide NeoPixel-Ringe zur zeitgleichen Visualisierung von maximal 32 Komponenten verwendet. Jedem NeoPixel ist dabei eine Komponente zugeordnet. Überschüssige NeoPixel, die bei weniger als 32 zu beobachteten Komponenten keine Verwendung finden, sind stets ausgeblendet. Wie in der Einleitung von [Abschnitt 5](#) erwähnt, ist der Zustand jeder Komponente als Wert zwischen null bis eins gegeben. Die einzelnen NeoPixel stellen individuell jene Werte via des in [Abbildung 6b](#) gezeigten Farbspektrums dar, das aufgrund der technischen Natur dieses Visualisierungskonzepts angemessen erscheint. Eine Problematik, welche solch ein Konzept begleitet, ist die Unwissenheit des Beobachters über die Zuordnung von NeoPixeln zu Komponenten. Diese Information dem Betrachter zu vermitteln soll dem jeweils zum Komponentenzustand darstellenden konträren NeoPixel-Ring zukommen. Sei beispielsweise der erste NeoPixel des linken NeoPixel-Rings mit dem Darstellen eines Komponentenzustands beschäftigt, dann obliegt dem rechten NeoPixel-Ring die Aufgabe über dessen korrespondierende Komponente aufzuklären.

Eine Komponente mit diversen Farbmustern intuitiv zu beschreiben ist, vor allem unter Einbeziehung des in [Unterabschnitt 3.1](#) erwähnten Fachwissens möglich, jedoch definitiv kein Selbstläufer. Insbesondere im Hinblick auf die ROS-Instanz eines Roboters, welche unter anderem ROS-Knoten für Batterien, Kameras, Aktuatoren und interne, verarbeitende Logik implementiert. Der den Batteriezustand anzeigende NeoPixel ließe sich beispielsweise mit einem überwiegend gelben Farbmuster beschreiben, in der Hoffnung, dass der Betrachter den Implikationen zwischen Batterie, Strom, Blitz und abschließend der gelben Farbe kognitiv folgen kann. Ist hingegen eine Komponente zum analysieren eingehender Bilddaten zu visualisieren, so existieren hierfür keine intuitiven Metaphern. Je abstrakter eine Komponente, desto weniger Möglichkeiten bieten sich folglich, auch unter Einbeziehung von Fachwissen, diese intuitiv darzustellen. Eine Komponente eindeutig zu benennen ist jedoch mit einer Bezeichnung möglich, also einer Kette von aufeinanderfolgenden lateinischen Buchstaben. Da jedem Mitglied der Forschungsgruppe TAMS das lateinische Alphabet einigermaßen geläufig ist, besteht die Hoffnung, jenes (Vor)Wissen für eine intuitive Visualisierung nutzen zu können. Eine derartige Bezeichnung auf einem NeoPixel-Ring darzustellen, erfordert das sequenzielle Abspielen der einzelnen Buchstaben, welche diese ausmachen. Die erarbeitete Darstellung einiger lateinischer Buchstaben auf einem NeoPixel-Ring ist in [Tabelle 5](#) gezeigt. Um einen einzelnen NeoPixel mit dem dargestellten Text in Verbindung zu bringen, sind für die Dauer der Textwiedergabe alle bis auf jenen NeoPixel ausgeblendet.

Bei maximal 32 beobachteten Komponenten ist die vollständig sequenzielle Benennung dieser sehr zeitintensiv, was einen langen Zeitraum zwischen den Aktualisierungen der Komponentenzustände zur Folge hätte; zunächst sind alle Komponenten zu beschreiben, bevor deren Zustände aktualisiert werden könnten. Demnach existieren zwei Parameter, die einerseits einen Schwellen-

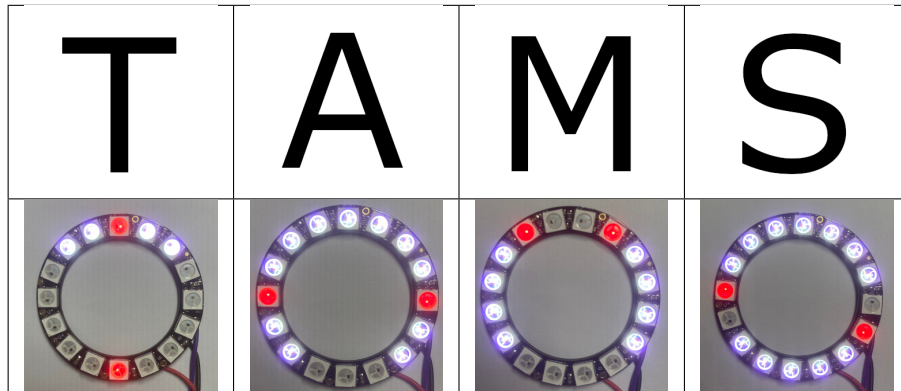


Tabelle 5: Rote NeoPixel weisen den Betrachter an, imaginär eine Strecke zwischen dem Mittelpunkt des NeoPixel-Rings und dem jeweils rot aufleuchtenden NeoPixel zu ziehen. Die Kombination aus realen, weißen und imaginären Strecken ergibt schlussendlich einen lateinischen Buchstaben.

wert definieren und zudem eine Prioritätenliste etablieren. Der Schwellenwert bestimmt zunächst, ab welchem Zustandswert die Beschreibung eines individuellen NeoPixels infrage kommt. Diesem Schwellenwert liegt der Gedanke zugrunde, dass Komponenten, die kaum Fehler aufweisen, nicht relevant sind. Der zweite Parameter, eine natürliche Zahl, gibt an, wie viele der fehlerbehaftetsten Komponenten zu benennen sind. Ist dieser Parameter beispielsweise mit dem Wert drei gegeben, dann werden ausschließlich die drei Komponenten mit den höchsten Fehlerwerten, in absteigender Reihenfolge, benannt. Das Parameterpaar ermöglicht einerseits die fehlerbehaftetsten Komponenten zu identifizieren und sorgt andererseits für eine zeitnahe Aktualisierung der Komponentenzustände.

Der in [Unterabschnitt 5.1](#) dargelegten Begründung entsprechend, ist das Kennlichmachen einer Zustandsaktualisierung gegenüber dem Betrachter erwünscht. Hierzu dient als Transition ein Farbmuster, das alle NeoPixel für einen kurzen Zeitraum vollständig ausblendet. Mithilfe einer solchen Zustandsänderungskennzeichnung ist es abermals möglich Animationsepisoden zu definieren. Eine derartige Animationsepisode beginnt mit der Visualisierung aller beobachteten Komponenten, je ein NeoPixel pro Komponente. Anschließend sind alle, den Parametern entsprechend erwähnte NeoPixel wie folgt beschrieben: Jeder NeoPixel, bis auf den zu beschreibenden, ist ausgeblendet, während parallel deren repräsentierende Buchstaben sequenziell abgespielt werden. Zwischen der Beschreibung einzelner NeoPixel sind jeweils alle beobachteten Komponenten erneut für eine kurze Zeit dargestellt. Abschließend wird die Transition, die das Aktualisieren der Komponentenzustände symbolisiert, dargestellt.

6 Ausblick

Die erdachten Visualisierungskonzepte in einer Studie zu erproben ist zunächst die interessanteste Fortführung dieses Projekts. Insbesondere die Lesbarkeit der über die NeoPixel-Ringe dargestellten Texte ist von großem Interesse. Hierfür ist auch eine Erweiterung des in [Unterabschnitt 4.3](#) beschriebenen Beobachtersystems angebracht. Einerseits müssen zusätzliche Beobachter implementiert werden, da momentan ausschließlich jene der Universität Hannover und ein Beobachter zur Messung des Abstands im Einsatz ist, andererseits weitere Aggregationsmöglichkeiten der einzelnen Komponentenzustände erdacht werden. Eine permanente Halterung der elektronischen Komponenten ist des Weiteren notwendig, um das System in realen Situation zu erproben. Jene Halterung kann beispielsweise mit dem 3D-Druckverfahren erstellt werden. Eine Schicht, die das Licht der NeoPixel streut, kann im Entwurf einer solchen Halterung ebenfalls berücksichtigt werden, was dargestellte animierte Farbmuster weniger blendend erscheinen lässt. Auch eine bessere Spezifikation des Gesamtsystems ist in naher Zukunft vorgesehen, was dessen Verwendung auch außerhalb des Testlabors der Universität Hamburg attraktiver macht.

Ein naheliegendes, aber zugleich zeitaufwendiges Unterfangen ist die Entwicklung einer Videoschnittprogramm ähnlichen Applikation, die das Erstellen einzigartiger, animierter Farbmuster drastisch vereinfacht.

Ein Beobachtersystem sowie ein entsprechendes Visualisierungskonzept, das der in [Anmerkung 2](#) verfassten Vorstellung eines Roboterzustands stärker entspricht, ist abschließend als wünschenswert zu nennen. Zwar ist es möglich, mit dem bestehenden System den Fortschritt bei der Lösung anstehender Aufgaben darzustellen, das Einbinden spezieller, handgefertigter animierter Farbmuster ist aber mühsam. So wäre bei einer Navigationsaufgabe, bei der der PR2-Roboter im Begriff ist, sich um 90° zu drehen, ein animiertes Farbmuster, das den Zustand des Drehens visualisiert, wünschenswert, aber nur schwer im jetzigen System umsetzbar.

Literaturverzeichnis

- [1] E. Hering, *Zur Lehre vom Lichtsinne, Sechs Mittheilungen an die Kaiserl. Akademie der Wissenschaften in Wien*, 2. Aufl. Wien: Gerold, 1878.
- [2] P. Ekman, W. Friesen, M. O’Sullivan, A. Chan, I. Diacoyanni-Tarlatzis, K. Heider, R. Krause, W. LeCompte, T. Pitcairn und P. Ricci Bitti, „Universals and Cultural Differences in the Judgments of Facial Expressions of Emotion,“ *Journal of personality and social psychology*, Jg. 53, S. 712–7, 1987.
- [3] P. Ekman, „An argument for basic emotions,“ *Cognition and Emotion*, Jg. 6, Nr. 3-4, S. 169–200, 1992.
- [4] K. Itoh, H. Miwa, M. Matsumoto, M. Zecca, H. Takanobu, S. Roccella, M. C. Carrozza, P. Dario und A. Takanishi, „Various emotional expressions with emotion expression humanoid robot WE-4RII,“ in *IEEE Conference on Robotics and Automation, 2004. TExCRA Technical Exhibition Based.*, 2004, S. 35–36.
- [5] T. Archevapanich, B. Purahong, M. Klingajay und P. Sooraksa, „Facial Visualization for Robotic Indicator by using Fuzzy Emotional System,“ in *2006 SICE-ICASE International Joint Conference*, 2006, S. 5634–5637.
- [6] S. Kopp. (2006). „Human-Computer Interaction,“ Adresse: https://www.techfak.uni-bielefeld.de/techfak/~skopp/Lehre/MMI_SS06/Termin11.pdf (besucht am 05.12.2021).
- [7] C. Mohs, J. Hurtienne, J. H. Israel, A. B. Naumann, M. C. Kindsmüller, H. A. Meyer und A. Pohlmeier, „IUUI – Intuitive Use of User Interfaces,“ in *Tagungsband UP06*, T. Bosenick, M. Hassenzahl, M. Peissner und M. Müller-Prove, Hrsg., Stuttgart: Fraunhofer Verlag, 2006, S. 130–133.
- [8] N. Endo, S. Momoki, M. Zecca, M. Saito, Y. Mizoguchi, K. Itoh und A. Takanishi, „Development of whole-body emotion expression humanoid robot,“ in *2008 IEEE International Conference on Robotics and Automation*, 2008, S. 2140–2145.
- [9] D.-W. Lee, T.-G. Lee, B. So, M. Choi, E.-C. Shin, K. Yang, M.-H. Baek, H.-S. Kim und H.-G. Lee, „Development of an android for emotional expression and human interaction,“ in *Seventeenth world congress the international federation of automatic control*, 2008, S. 4336–4337.
- [10] F. Hegel, C. Muhl, B. Wrede, M. Hielscher-Fastabend und G. Sagerer, „Understanding Social Robots,“ in *2009 Second International Conferences on Advances in Computer-Human Interactions*, 2009, S. 169–174.
- [11] J. H. Turner, „The Sociology of Emotions: Basic Theoretical Arguments,“ *Emotion Review*, Jg. 1, Nr. 4, S. 340–354, 2009.
- [12] M. Häring, N. Bee und E. André, „Creation and Evaluation of emotion expression with body movement, sound and eye color for humanoid robots,“ in *2011 RO-MAN*, 2011, S. 204–209.

- [13] Softbank Robotics Europe. (2014). „NAO Robot (bleu et rouge) (cropped),“ Adresse: [https://commons.wikimedia.org/wiki/File:NAO_Robot_\(bleu_et_rouge\)_cropped.jpg](https://commons.wikimedia.org/wiki/File:NAO_Robot_(bleu_et_rouge)_cropped.jpg) (besucht am 15.12.2021).
- [14] SparkFun Electronics. (2014). „12664-02a,“ Adresse: <https://commons.wikimedia.org/wiki/File:12664-02a.jpg> (besucht am 15.12.2021).
- [15] M. Cascio, M. Guarnera, Z. Hichy und S. Carrubba, „Facial Expressions and Ability to Recognize Emotions From Eyes or Mouth in Children,“ *Europe’s Journal of Psychology*, Jg. 11, S. 183–196, 2015.
- [16] S. Song und S. Yamada, „Expressing Emotions Through Color, Sound, and Vibration with an Appearance-Constrained Social Robot,“ in *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2017, S. 2–11.
- [17] K. Barka und M. M. Veloso, „Mobile service robot state revealing through expressive lights: Formalism, design, and evaluation,“ *International Journal of Social Robotics*, Jg. 10, Nr. 1, S. 65–92, 2018.
- [18] D. Löffler, N. Schmidt und R. Tscharn, „Multimodal expression of artificial emotion in social robots using color, motion and sound,“ in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 2018, S. 334–343.
- [19] T. van der Grinten, S. Müller, M. Westhoven, S. Wischniewski, A. Scheidig und H.-M. Gross, „Designing an Expressive Head for a Help Requesting Socially Assistive Robot,“ in *Human-Friendly Robotics 2019*, F. Ferraguti, V. Villani, L. Sabbatini und M. Bonfè, Hrsg., Springer International Publishing, 2020, S. 88–102.
- [20] Arduino AG. (2021). „Arduino Nano,“ Adresse: https://cdn.shopify.com/s/files/1/0438/4735/2471/products/A000005_01_iso_1000x750.jpg?v=1628695103 (besucht am 15.12.2021).
- [21] P. Burgess. (2021). „Adafruit NeoPixel Überguide, Best Practices,“ Adresse: <https://learn.adafruit.com/adafruit-neopixel-uberguide/best-practices> (besucht am 30.11.2021).
- [22] P. Burgess. (2021). „Adafruit NeoPixel Überguide, Basic Connections,“ Adresse: <https://learn.adafruit.com/adafruit-neopixel-uberguide/basic-connections> (besucht am 30.11.2021).
- [23] „Safety of machinery - Electrical equipment of machines - Part 1: General requirements,“ International Organization for Standardization, Geneva, CH, Standard, 2016.

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe angefertigt und mich anderer als der im beigefügten Verzeichnis angegebenen Hilfsmittel nicht bedient habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe und die eingereichte schriftliche Fassung der auf dem elektronischen Speichermedium entspricht.

Ich bin mit einer Einstellung in den Bestand der Bibliothek des Fachbereiches einverstanden.

Hamburg, den 15.12.2021 Unterschrift: Benedikt Deike