

BACHELOR THESIS  
Awab Abdelkarim

# Mensch-Maschinen Interaktion auf einem autonomen Roboter durch visuelle Signale

---

FAKULTÄT TECHNIK UND INFORMATIK  
Department Informatik

Faculty of Engineering and Computer Science  
Department Computer Science

Awab Abdelkarim

# Mensch-Maschinen Interaktion auf einem autonomen Roboter durch visuelle Signale

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung  
im Studiengang *Bachelor of Science Angewandte Informatik*  
am Department Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Stephan Pareigis  
Zweitgutachter: Prof. Dr. Eike Langbehn

Eingereicht am: 29. August 2022

**Awab Abdelkarim**

**Thema der Arbeit**

Mensch-Maschinen Interaktion auf einem autonomen Roboter durch visuelle Signale

**Stichworte**

Mensch Maschinen Interaktion, autonome Roboter, visuelle Signale, ROS

**Kurzzusammenfassung**

Im Rahmen dieser Arbeit wurde ein Mensch-Maschine Interaktionssystem entwickelt. Das Ziel dieser Arbeit war, die Akzeptanz von Benutzern autonomer Roboter zu erhöhen. Dafür sollte ein System entwickelt werden, welches die Absichten eines autonomen Roboters visuell wiedergibt. Durch den LiDAR erkannte Personen, wurden auf einem um den Roboter herum befestigten LED-Streifen, angezeigt. Es wurde eine Umfrage mit jeweils aktiviertem und deaktiviertem LED-Streifen durchgeführt. Die Umfrage ergab, dass ein LED-Streifen zur Kommunikation das Sicherheitsgefühl der Benutzer erhöht.

**Awab Abdelkarim**

**Title of Thesis**

Human-machine interaction on an autonomous robot through visual signals

**Keywords**

Human Machine Interaction, autonomous Robots, visual Signals, ROS

**Abstract**

In this thesis, a human-machine interaction system was developed. The goal was to increase the acceptance of users of autonomous robots. For this purpose, a system was developed that visually reflects the intentions of an autonomous robot. Persons detected by the LiDAR were shown on a LED strip around the robot. A survey was conducted with each LED strip activated and deactivated. The survey indicated that an LED strip for communication increases the users' sense of safety.

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>vi</b>
<b>Tabellenverzeichnis</b>	<b>ix</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Zielsetzung . . . . .	1
1.2 Aufbau der Arbeit . . . . .	2
1.3 Verwandte Arbeiten . . . . .	2
<b>2 Grundlagen</b>	<b>4</b>
2.1 Mensch-Maschinen-Interaktion . . . . .	4
2.2 ROS . . . . .	5
2.3 Husky . . . . .	5
2.4 WLED . . . . .	5
<b>3 Funktionale Anforderungen</b>	<b>6</b>
3.1 REQ1 Personenerkennung anzeigen . . . . .	6
3.2 REQ2 Systemstatus anzeigen . . . . .	7
3.3 REQ3 Abbiegevorgang anzeigen . . . . .	8
3.4 Konfigurierbarkeit . . . . .	9
<b>4 System</b>	<b>10</b>
4.1 Hardware . . . . .	10
4.1.1 ESP32 Development Board . . . . .	10
4.1.2 RS-LiDAR-16 . . . . .	11
4.1.3 WS2812B LED . . . . .	12
4.1.4 Konzept LED-Controller . . . . .	13
4.2 Nachrichtenformate . . . . .	17
4.2.1 PointCloud2 . . . . .	17
4.2.2 MarkerArray . . . . .	18

4.2.3	PointsArray . . . . .	19
4.2.4	Segment Array . . . . .	20
4.3	Konfigurationsparameter . . . . .	22
4.3.1	API . . . . .	27
4.4	Personenerkennungskomponente . . . . .	28
4.4.1	Kommunikation mit HMI System . . . . .	28
4.5	Entwurf . . . . .	31
4.5.1	REQ1 Personenerkennung anzeigen . . . . .	31
4.5.2	REQ2 Systemstatus anzeigen . . . . .	35
4.5.3	REQ3 Abbiegevorgang anzeigen . . . . .	37
<b>5</b>	<b>Evaluierung</b>	<b>40</b>
5.1	Objektive Evaluierung . . . . .	40
5.2	Subjektive Evaluierung . . . . .	42
5.2.1	Aufbau des Fragebogens . . . . .	43
5.2.2	Auswertung . . . . .	47
5.3	Verbesserungsmöglichkeiten . . . . .	55
5.3.1	Personenerkennung funktioniert nicht auf Produktivsystem . . . . .	55
5.3.2	Eventuelle Speicherprobleme durch Mikrocontroller . . . . .	56
<b>6</b>	<b>Fazit</b>	<b>58</b>
6.1	Ausblick . . . . .	59
	<b>Literaturverzeichnis</b>	<b>60</b>
<b>A</b>	<b>Anhang</b>	<b>63</b>
A.1	Fragebogen . . . . .	63
A.2	Videos aus dem Fragebogen . . . . .	69
A.3	OpenAPI 3 Schema . . . . .	69
	<b>Glossar</b>	<b>70</b>
	Selbstständigkeitserklärung . . . . .	71

# Abbildungsverzeichnis

3.1	Husky zeigt an, dass er Personen erkannt hat. Dort wo Personen stehen, leuchtet der LED-Streifen. Die Farbe, Effektgeschwindigkeit und Helligkeit sollen über eine Konfiguration eingestellt werden können. . . . .	7
3.2	Husky zeigt einen Systemstatus an. Hier beispielhaft die Einfahrt in einen Aufzug mit dem Status warning . . . . .	8
3.3	Husky zeigt an, dass er Abbiegen möchte. Dafür werden die Blinkmodule an der Seite aktiviert, in welche der Husky fahren möchte. . . . .	8
4.1	ESP32 NodeMCU Entwicklungsboard <sup>1</sup> . . . . .	11
4.2	RS-LiDAR-16 der Firma Robosense <sup>2</sup> . . . . .	11
4.3	Datenübertragungsmethode des WS2812B. Jeder Datenzyklus enthält n * 24 Bit an Daten. n steht für die Anzahl der in Reihe geschalteten WS2812B LEDs. Die erste LED liest die ersten 24 Bit, entfernt diese und sendet den Rest weiter. Die nächste LED liest dann die nächsten ersten 24 Bit, entfernt diese und sendet den Rest weiter. Dies läuft dann über den ganzen Streifen. Zwischen jedem Zyklus müssen $\geq 50\mu\text{s}$ gewartet werden (World-Semi). . . . .	12
4.4	Platinenansicht LED Controller . . . . .	14
4.5	Schaltplan LED Controller . . . . .	15
4.6	Einfache Spannungseinspeisung des LED-Streifens. Hier wird die Spannungsversorgung des Controllers und des LED-Streifens getrennt. Links ist exemplarisch ein Arduino Uno zu sehen, welcher über den USB-Anschluss mit Spannung versorgt wird. Rechts ein LED-Streifen mit einzeln ansteuerbaren LEDs, welcher über Spannungseinspeisung mit Spannung versorgt wird. Die Leitungen sind wie folgt zugewiesen: Schwarz = Ground, Blau = Dateninput, Rot = Spannung. Die Masse wird dennoch miteinander verbunden, um das gleiche Bezugspotential zwischen dem LED-Controller und dem LED-Streifen zu haben <sup>3</sup> . (Project Tempos, 2022) . . . . .	16

4.7	Spannungsversorgung des LED-Streifens und des LED-Controllers. Hier werden der Controller und der LED-Streifen zusammen mit der gleichen Spannung versorgt. Hierfür benötigen wir nur eine Spannungsquelle. (Project Tempos, 2022)	17
4.8	Beispielhafte Darstellung vom Nachrichtenformat <code>Marker</code> , welche visuell in RViz dargestellt sind <sup>4</sup> .	18
4.9	Klassendiagramm für das Nachrichtenformat eines <code>PointsArray</code> . Jede erkannte Person erhält eine neue ID, welche von der 4.4 zugewiesen wurde. In <code>x</code> , <code>y</code> und <code>z</code> werden die Koordinaten der erkannten Person eingetragen. <code>degree</code> entspricht dem berechneten relativen Winkel der Person zum autonomen Roboter.	19
4.10	Exemplarische Darstellung der Beleuchtung von Nachbar-LEDs. LED 6 soll leuchten. Durch Einstellen der Konfiguration <code>led/adjacent</code> , dass auf jeder Seite 2 LEDs mitleuchten sollen, werden LED 4,5,7 und 8 mitbeleuchtet. <sup>5</sup>	25
4.11	Beispiel	26
4.12	Darstellung aller Endpunkte der API zum Einstellen der Konfigurationsparameter	28
4.13	Komponentenübersicht der Personenerkennung, sowie die Nachrichtenformate, die zur Kommunikation zwischen den Komponenten genutzt werden.	31
4.14	Ablaufdiagramm der Personenerkennung, sowie die dazugehörige Darstellung im LED-Strip	32
4.15	Beispielhafte Darstellung der Personenerkennung in der Simulationsumgebung Gazebo	33
4.16	Ablaufdiagramm der Darstellung des Systemstatus	36
4.17	Darstellung Systemstatus	37
4.18	Ablaufdiagramm der Blinkersteuerung	38
5.1	Exemplarische Darstellung des Prototyps.	41
5.2	Verschiedene Helligkeitsstufen des LED-Streifens. Links die schwächste Helligkeit und rechts die höchste Helligkeit	41
5.3	Exemplarische Darstellung vom <code>warning</code> -Status auf dem Prototypen	42
5.4	Roboter fährt an stehender Person vorbei	44
5.5	Ausschnitt aus dem Kurzvideo aus Frage F3, welches einen Blinker zeigen soll. Die orange markierten Bereiche blinken.	46
5.6	Verteilung des Geschlechts der Befragten	47

5.7	Alter der Befragten gruppiert anhand des Geschlechts. . . . .	48
5.8	Angabe, ob Befragten in einem Beruf arbeiten, in dem Sie mit Informatik zu tun haben . . . . .	49
5.9	Gefühl, ob der Roboter die Person erkannt hat. Blau steht für Weiblich und Rot für Männlich. Das HMI System war in diesem Szenario aus . . .	50
5.10	Gefühl, ob der Roboter die Person erkannt hat. Blau steht für Weiblich und Rot für Männlich. Das HMI System war in diesem Szenario an . . . .	51
5.11	Sicherheitsgefühl, wenn der Roboter am Befragten vorbeifährt. Das HMI System war in diesem Szenario aus . . . . .	52
5.12	Sicherheitsgefühl, wenn der Roboter am Befragten vorbeifährt. Das HMI System war in diesem Szenario an . . . . .	53
5.13	Antwort der Befragten beim Zeigen eines Videos, wo der Roboter einen Abbiegevorgang zeigen möchten. . . . .	54



# Tabellenverzeichnis

5.1 Vergleich von ESP8266 und ESP32 (Riccardi, 2021) . . . . .	56
--	----

# 1 Einleitung

Kommunikation schafft Vertrauen. Sei es der Saugroboter, der durch auditive Sprache mitteilt, dass er fertig ist, oder der Kaffeeautomat, der nach Fertigstellung des Kaffees piept. In all diesen Punkten ist Kommunikation vorhanden. Dies kann durch verschiedene Art und Weise übermittelt werden. Sei es durch visuelle Signale, wie eine blinkende Lampe oder einen LED-Streifen, der auf bestimmtes Verhalten hinweisen soll, oder durch auditive Signale, wie ein Lautsprecher in einem Flugzeug, der die letzten Meter vor der Landung wiedergibt. Überall ist Kommunikation eingebaut. Dies ist auch wichtig, um die Akzeptanz der Benutzer zu erlangen.

Insbesondere in der Thematik von autonomen Systemen ist dies wichtig. In einer von AutoScout24 beauftragten Umfrage aus dem Jahre 2021 haben sich rund 53% der Befragten gegen autonome Fahrzeuge positioniert (Innofact). In einer anderen Umfrage, durchgeführt von DHBW Ravensburg, gaben 65% der Befragten an, dass Sie ein Problem mit der Akzeptanz von autonomen Fahrzeugen, aufgrund des fehlenden Vertrauens in die Technik haben (DHBW Ravensburg).

## 1.1 Zielsetzung

In dieser Arbeit soll ein Mensch-Maschinen-Interaktions-System, im weiteren Verlauf HMI System genannt, entwickelt werden, welches in ein bereits vorhandenes System integriert werden soll. Der autonome Roboter soll durch visuelle Signale mit seiner Umwelt kommunizieren können. Durch die Sensorik des autonomen Roboters, nimmt er seine Umgebung wahr, und gibt diese Wahrnehmung in Form von visuellen Signalen wieder aus. Hierfür werden Anforderungen definiert. Das Ziel dieser Arbeit ist es, durch visuelle Signale die Akzeptanz gegenüber Dritten zu erhöhen. Dies soll durch eine verständliche Kommunikationsart des autonomen Roboters mit Dritten erzielt werden.

## 1.2 Aufbau der Arbeit

In Kapitel 2 gehen wir auf einige Grundlagen, die zum Verständnis dieser Arbeit wichtig sind, ein.

Kapitel 3 definiert die funktionalen Anforderungen. Gleichzeitig definiert dieses auch die Sprache des autonomen Roboters.

Anschließend geht es in Kapitel 4 um das Herzstück dieser Arbeit. Anfangs wird auf die benutzte Hardware eingegangen. Danach die wichtigsten Nachrichtenformate, die in der Kommunikation zwischen den Komponenten genutzt wurden. Es werden die Anpassungsmöglichkeiten des Systems erläutert, und wie diese über eine API verändert werden können. Anschließend wird die Personenerkennungskomponente vorgestellt und wie das HMI System mit dieser kommuniziert. Danach wird der Implementierungsentwurf der Anforderungen aus Kapitel 3 erläutert.

Kapitel 5 wird in zwei Teile unterteilt. In der objektiven Evaluierung wird überprüft, ob die in Kapitel 3 gestellten Anforderungen erfüllt worden sind. In der subjektiven Evaluierung wird eine Umfrage durchgeführt. Die gestellten Anforderungen werden den Befragten visuell am autonomen Roboter gezeigt und dazu Fragen gestellt. In der Auswertung wird geschaut, ob die Zielsetzung in Abschnitt 1.1 erfüllt wird. Abschließend werden aufgetretene Probleme erläutert und Verbesserungsmöglichkeiten vorgestellt.

## 1.3 Verwandte Arbeiten

Das Thema Mensch-Maschinen Interaktion hat sich in den letzten Jahren zu einer sehr interessanten Debatte entwickelt. Deshalb gibt es auch einige interessante Arbeiten, die sich mit dem Grundansatz der Kommunikation zwischen Mensch und Maschine beschäftigen. Auf einige wird hier eingegangen, da sie auch als Inspiration zur Bearbeitung dieser Arbeit geführt haben.

Petzoldt u. a. (2018) hat sich mit der Verbesserung der Sicherheit im Straßenverkehr durch Einbau eines vorne am Fahrzeug montiertem Bremslichts beschäftigt. Chang u. a. (2018) hat verschiedene visuelle Signalarten in einer Zebrastreifen-Situation verglichen und ist zu dem Entschluss gekommen, dass LED-Streifen einen sehr hohen Grad an Unverständlichkeit haben. Chang u. a. (2018) vertritt die Meinung, dass textbasierte Anzeigen die beste Wahl sind. de Clercq u. a. (2019) ist zu dem gleichen Ergebnis gekommen

wie Chang u. a. (2018). Beide Paper haben nach einer verständlichen Art gesucht, den betroffenen Passanten mitzuteilen, dass Sie den Zebrastreifen überqueren können. Burns u. a. (2019) nutzt einen Tageslichtprojektor an der Vorderseite des autonomen Roboters und projiziert auf den Boden wann der Roboter abbiegen möchte. Dies wurde mit herkömmlichen Blinkern verglichen. Burns u. a. (2019) kam zu dem Ergebnis, dass die Projektion verständlicher mitteilt, wann der Roboter abbiegen möchte. Außerdem fanden einige Befragte seiner Umfrage, dass die Projektion moderner und die Entscheidungsfindung beschleunigt wird. Nachteile waren der Blick auf den Boden und die Problematik, dass nicht bei allen Witterungsverhältnissen die Projektion gut zu sehen ist. Die Blinker, wie wir sie von Kraftfahrzeugen kennen, sind verständlich und effektiv, bringen aber den Nachteil mit, dass nicht klar ist, wann der Roboter abbiegen möchte.

## 2 Grundlagen

In diesem Kapitel wird kurz auf Grundlagen eingegangen, die zum Verständnis dieser Arbeit von Vorteil sein könnten.

### 2.1 Mensch-Maschinen-Interaktion

Die Mensch-Maschinen-Interaktion, unter anderem auch Human Machine Interaction (HMI) oder auch Mensch-Computer-Interaktion (MCI) genannt, ist ein Fachgebiet der Informatik, welches sich mit der Kommunikation und Interaktion zwischen Mensch und Maschine/Computer beschäftigt.

[...] die es Robotern oder der Robotertechnologie ermöglichen, erfolgreich mit Menschen interagieren zu können. (Bartneck u. a., 2020, S. 191)

Die Gesellschaft für Informatik e.V. (2022) definiert den Forschungsbereich der Mensch-Maschinen-Interaktion wie folgt:

Unter Mensch-Computer-Interaktion (MCI) versteht man den Bereich der angewandten Informatik, der sich mit Fragen rund um die benutzer- und kontextgerechten Gestaltung von IT-Systemen beschäftigt. (Gesellschaft für Informatik e.V., 2022)

Generell geht es darum, wie man IT-Systeme entwickelt, die einfach und verständlich Informationen Ihren Benutzern mitteilen können.

### 2.2 ROS

In dieser Arbeit wurde ROS (Version Noetic) als Basis des autonomen Roboters genutzt. ROS, Abkürzung von Robot Operating System, ist ein Framework für Roboter. ROS ist eine Sammlung von Softwarebibliotheken und Tools, welche die Entwicklung von Roboter Anwendungen vereinfachen und unterstützen soll. In ROS wird zu Kommunikation zwischen den Komponenten von ROS, ein Publisher (Veröffentlicher) und Subscriber (Abonnent) System genutzt. Ein Publisher veröffentlicht unter einem definierten Namen, Topic genannt, Daten eines definierten Typs. Verschiedene Subscriber können diese Topic's abonnieren und erhalten, sobald eine Nachricht vom Publisher veröffentlicht wurden, ein Event. In Abschnitt 4.4.1 (Kommunikation mit HMI System) wird näher darauf eingegangen.

### 2.3 Husky

Der Husky ist die Basis eines autonomen Roboters der Forschungsgruppe AutoSys<sup>1</sup> der HAW Hamburg. In dieser Arbeit wurde er zum Testen des HMI System's genutzt. Der Husky ist Teil des Projektes Testbereich Intelligente Quartiersmobilität<sup>2</sup>. Einer von vielen Anwendungsbereichen des Huskys ist die autonome Übernahme der letzten Meile in der Paketzustellung. Das Ziel des Projektes ist die Entwicklung eines Roboters, der alle Aufgaben autonom erledigen kann.

### 2.4 WLED

In dieser Arbeit wurde für den LED-Controller die Software WLED genutzt<sup>3</sup>. WLED zeichnet sich durch seine Community aus, die in der Open-Source Entwicklung des Projektes, viel Erfahrung mit einfließen lässt. Durch die JSON-API stellt WLED eine Kommunikationsschnittstelle mit dem Controller/der Software bereit. WLED unterstützt eine große Anzahl an verschiedenen Mikrocontrollern und LED-Streifen-Typen<sup>4</sup>.

---

<sup>1</sup><https://autosys.informatik.haw-hamburg.de/>

<sup>2</sup><https://autosys.informatik.haw-hamburg.de/project/smartmobility/>

<sup>3</sup>Weitere Information zu WLED unter <https://kno.wled.ge/> zu finden.

<sup>4</sup>Hardware Kompatibilitätsliste WLED <https://kno.wled.ge/basics/compatible-hardware/>

## 3 Funktionale Anforderungen

In diesem Kapitel gehen wir auf die gewünschten Anforderungen an unser System ein.

### 3.1 REQ1 Personenerkennung anzeigen

Der Husky zeigt an, dass er Personen in seinem nahen Umfeld erkannt hat. Dies wird über den LED-Streifen, ein Band um den Husky mit LED-Lichtern, angezeigt. Wenn zum Beispiel vor dem Husky eine Person erkannt wird, leuchten die Lichter an dieser Stelle. Personen werden über den LiDAR erkannt und einer ID zugewiesen. Die Software speichert die erkannten Objekte und sendet diese mit der Angabe der ID, x-Koordinate, y-Koordinate und dem Winkel relativ zum Husky zur Darstellung auf dem LED-Streifen weiter. In Abbildung 3.1 ist diese Anforderung visuell dargestellt.

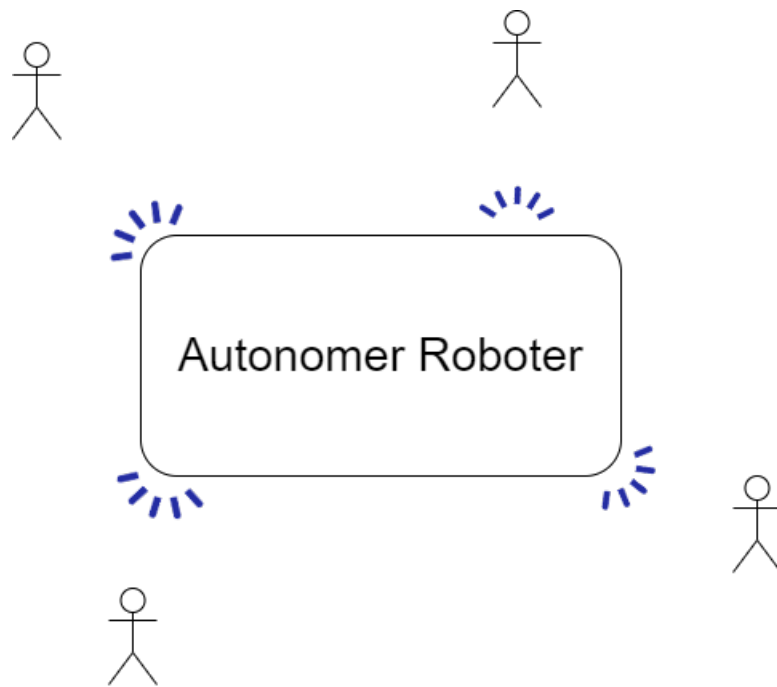


Abbildung 3.1: Husky zeigt an, dass er Personen erkannt hat. Dort wo Personen stehen, leuchtet der LED-Streifen. Die Farbe, Effektgeschwindigkeit und Helligkeit sollen über eine Konfiguration eingestellt werden können.

## 3.2 REQ2 Systemstatus anzeigen

Der Status des Huskys wird über ein state-Topic festgelegt. Je nachdem in welchem Status der Husky sich befindet, wird eine festgelegte LED-Routine abgespielt. Dies soll nicht nur für das HMI System nützlich sein, sondern auch für alle anderen Komponenten des Huskys. So kann z.B. eine Warnung als warning in den state gesendet werden. Das HMI System zeigt dann über den LED-Strip ein langsam pulsierendes Oranges Licht an. In einem Fehlerzustand, error-State, zeigt der Husky beispielsweise ein schnell pulsierendes rotes Licht an. Dies kann nach Belieben erweitert werden. Außerdem sollen die Farb- und Effekteinstellungen über eine Konfiguration einstellbar sein.



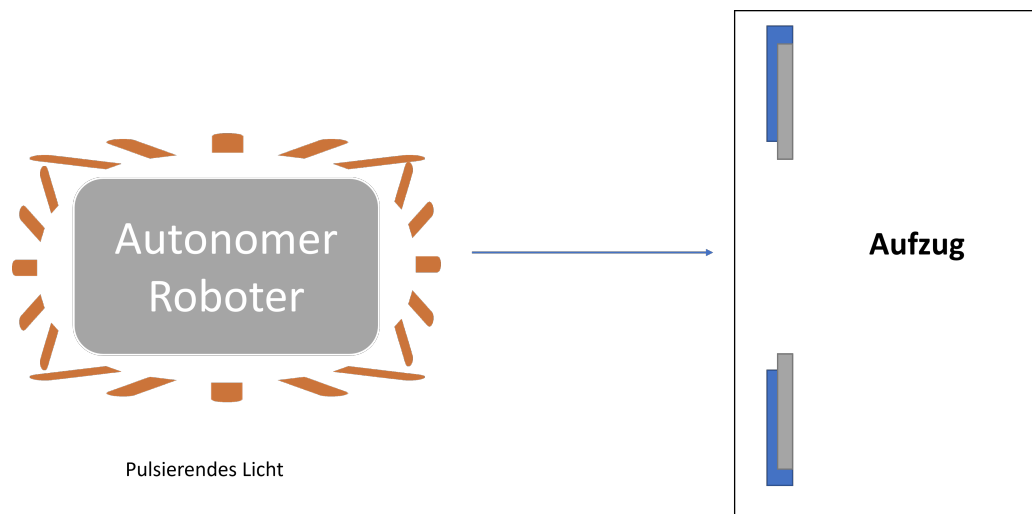


Abbildung 3.2: Husky zeigt einen Systemstatus an. Hier beispielhaft die Einfahrt in einen Aufzug mit dem Status warning

### 3.3 REQ3 Abbiegevorgang anzeigen

Der Husky soll über seine visuellen Signale anzeigen, dass er demnächst abbiegen möchte. Über ein festgelegtes Topic, welches die Richtung des Abbiegevorgangs enthält, liest das HMI System aus, auf welcher Seite die Blinker aktiviert werden sollen. Wie bei einem Kraftfahrzeug besitzt der Husky an jeder Seite Blinkermodule.

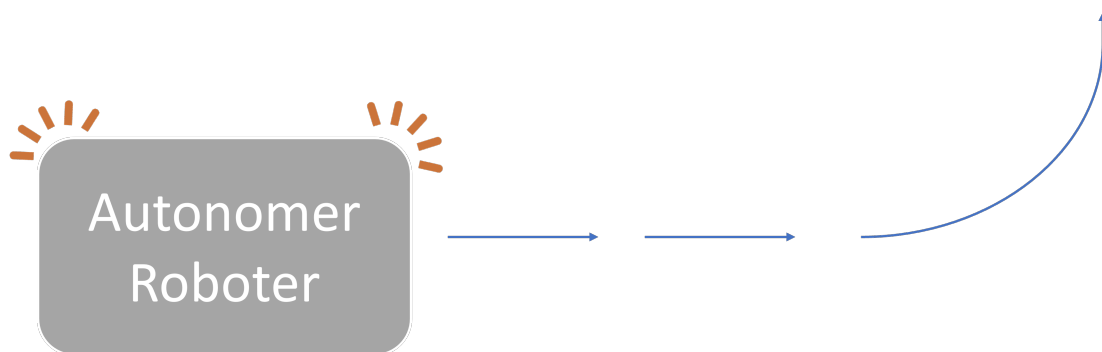


Abbildung 3.3: Husky zeigt an, dass er Abbiegen möchte. Dafür werden die Blinkmodule an der Seite aktiviert, in welche der Husky fahren möchte.

### **3.4 Konfigurierbarkeit**

Eine weitere Anforderung ist die Ermöglichung der Konfigurierbarkeit. Wesentliche Bestandteile des Systems, wie Farb- und Effekteinstellungen, Systemzustände etc. sollen über eine Konfigurationsdatei einstellbar sein. Somit wird maximale Flexibilität bei Benutzung des Systems gewährleistet.

# 4 System

In diesem Kapitel gehen wir anfangs auf die genutzte Hardware ein. Danach werden die in dieser Arbeit genutzten Nachrichtenformate erläutert. Anschließend gehen wir auf die Konfigurationsparameter ein. Danach wird auf die Personenerkennungskomponente eingegangen und wie diese technisch funktioniert. Im Anschluss wird das System vorgestellt.

## 4.1 Hardware

### 4.1.1 ESP32 Development Board

Genutzt wurde ein NodeMCU Board mit einem ESP32 Chip. Der Chip besitzt neben einem Bluetooth-Chip, eine Wireless-LAN Funktionalität, welche direkt mit dem lokalen Netzwerk des autonomen Roboters verbunden werden kann. So können beispielsweise API Aufrufe vom ESP32 entgegengenommen werden und definierte Funktionen ausgeführt werden. Der ESP32 wurde zum Ausführen von WLED genutzt. Der LED-Streifen war mit ESP32 verbunden und wurde durch WLED gesteuert. In Abbildung 4.1 ist das in dieser Arbeit verwendete Entwicklungsboard zu sehen.

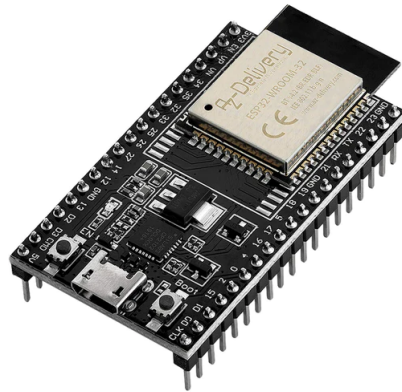


Abbildung 4.1: ESP32 NodeMCU Entwicklungsboard<sup>1</sup>.

In Abschnitt 4.1.4 wird auf die Verkabelung des ESP32's eingegangen.

#### 4.1.2 RS-LiDAR-16

Der RS-LiDAR-16 ist ein Laserscanner, welcher eine 16 schichtige Punktwolke seiner Umgebung erstellt. Der Laser strahlt Lichtimpulse aus. Das reflektierte Licht wird dann erfasst. Durch Ermittlung der Zeit, die die Reflektion bis zum Laserscanner zurück gebraucht hat, wird die Distanz ermittelt. Diese Daten werden in ROS als PointCloud2 Nachrichten gespeichert. In Abbildung 4.2 ist der in dieser Arbeit verwendete LiDAR zu sehen.



Abbildung 4.2: RS-LiDAR-16 der Firma Robosense<sup>2</sup>.

---

<sup>1</sup><https://www.az-delivery.de/products/esp-32-dev-kit-c-v4>

<sup>2</sup><https://www.robosense.ai/en/rslidar/RS-LiDAR-16>

### 4.1.3 WS2812B LED

Die WS2812B LED ist ein Vollfarben-Modul, welche auf einem Streifen mit anderen LEDs einzeln angesteuert werden kann. Sie unterscheidet sich gegenüber üblichen LED-Modulen durch den integrierten Schaltkreis, welcher die Ansteuerung einzelner LEDs ermöglicht. Dafür benötigen die einzelnen LEDs keine Information darüber, wie viel LEDs sich auf dem Streifen befinden. Dies wird, wie in Abbildung 4.3 zu sehen ist, umgesetzt.

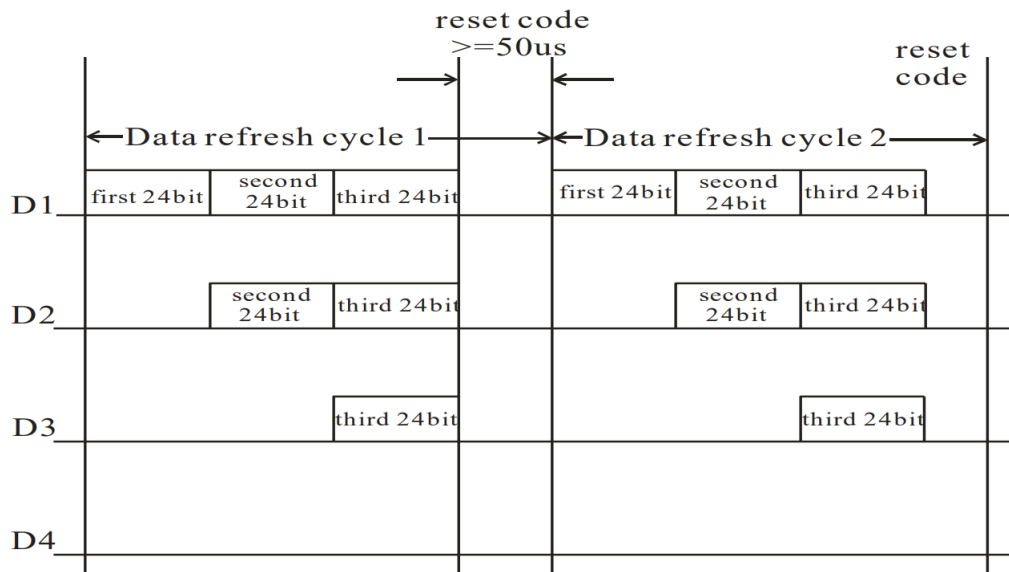


Abbildung 4.3: Datenübertragungsmethode des WS2812B. Jeder Datenzyklus enthält  $n * 24$  Bit an Daten.  $n$  steht für die Anzahl der in Reihe geschalteten WS2812B LEDs. Die erste LED liest die ersten 24 Bit, entfernt diese und sendet den Rest weiter. Die nächste LED liest dann die nächsten ersten 24 Bit, entfernt diese und sendet den Rest weiter. Dies läuft dann über den ganzen Streifen. Zwischen jedem Zyklus müssen  $\geq 50\mu\text{s}$  gewartet werden (World-Semi).

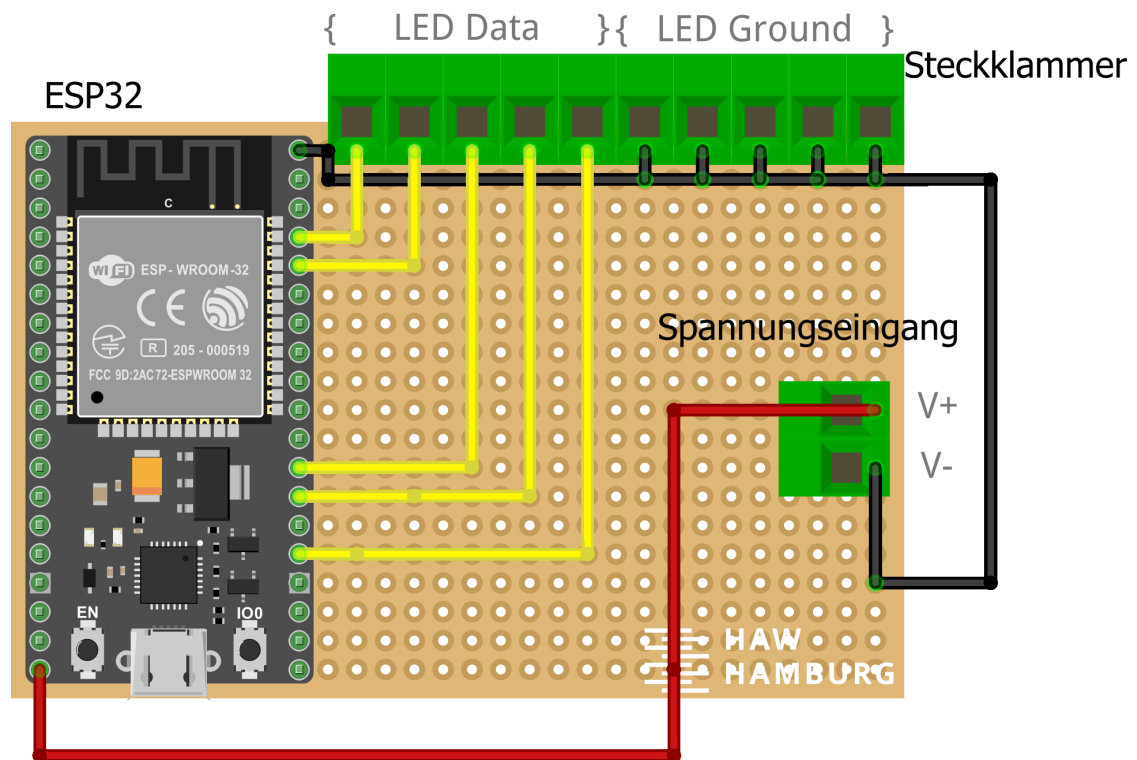
Der Datenstrom eines Datenzyklus, der zur Ansteuerung der WS2812B LEDs benötigt wird, enthält  $n * 24$  Bit.  $n$  steht hier für die Anzahl an in Reihe geschalteten LEDs. Die Kodierung der 24 Bit Daten sind die RGB Daten, welche für jeden Farbraum 8 Bit vorsieht. Dies sind die Zahlen von 0 - 255.

In Abbildung 4.3 werden 3 LEDs angesteuert. Die LED D1 liest die ersten 24 Bit ein, entfernt diese aus dem Datenstrom und sendet den Rest an LED D2 weiter. Dort werden die nächsten 24 Bit eingelesen, aus dem Datenstrom entfernt und weitergesendet bis

keine Daten mehr vorhanden sind. Ein Datenzyklus wird durch eine Pause von  $\geq 50\mu\text{s}$  beendet (World-Semi).

### 4.1.4 Konzept LED-Controller

Zur verbesserten Handhabung und Austauschbarkeit des LED-Controllers, wurde eine einfache Platine entwickelt. Abbildung 4.4 zeigt den Aufbau der Platine. Der dort verwendete Controller, ist der in Abschnitt 4.1.1 vorgestellte ESP32. Abbildung 4.5 zeigt den Schaltplan der Platine. Dort ist auch die Pinbelegung des ESP32's zu sehen. Über den Spannungseingang wird der ESP32 über den VIN Pin (Voltage Input) mit 5V Betriebsspannung versorgt. Über die Steckklammern werden die LED-Streifen verbunden. LED Data versorgt den LED-Streifen mit den benötigten Daten. LED Ground synchronisiert die Masse des LED-Streifens und des LED-Controllers. Die LED-Streifen erhalten über ein separates Netzteil Spannung, können aber auch über den Spannungseingang V+ mit Spannung versorgt werden. Hier muss nur beachtet werden, dass das verwendete Netzteil eine hohe Stromstärke liefern kann, da LED-Streifen relativ viel Stromstärke benötigen. In dieser Arbeit wurde ein LED-Streifen des Typs WS2812B benutzt, welcher 50mA pro LED's bei voller Leuchtkraft verbraucht. Bei einem Streifen mit 120 LED sind dies  $120 * 50\text{mA} = 6.000\text{mA} = 6\text{A}$  maximaler Verbrauch.



fritzing

Abbildung 4.4: Platinenansicht LED Controller

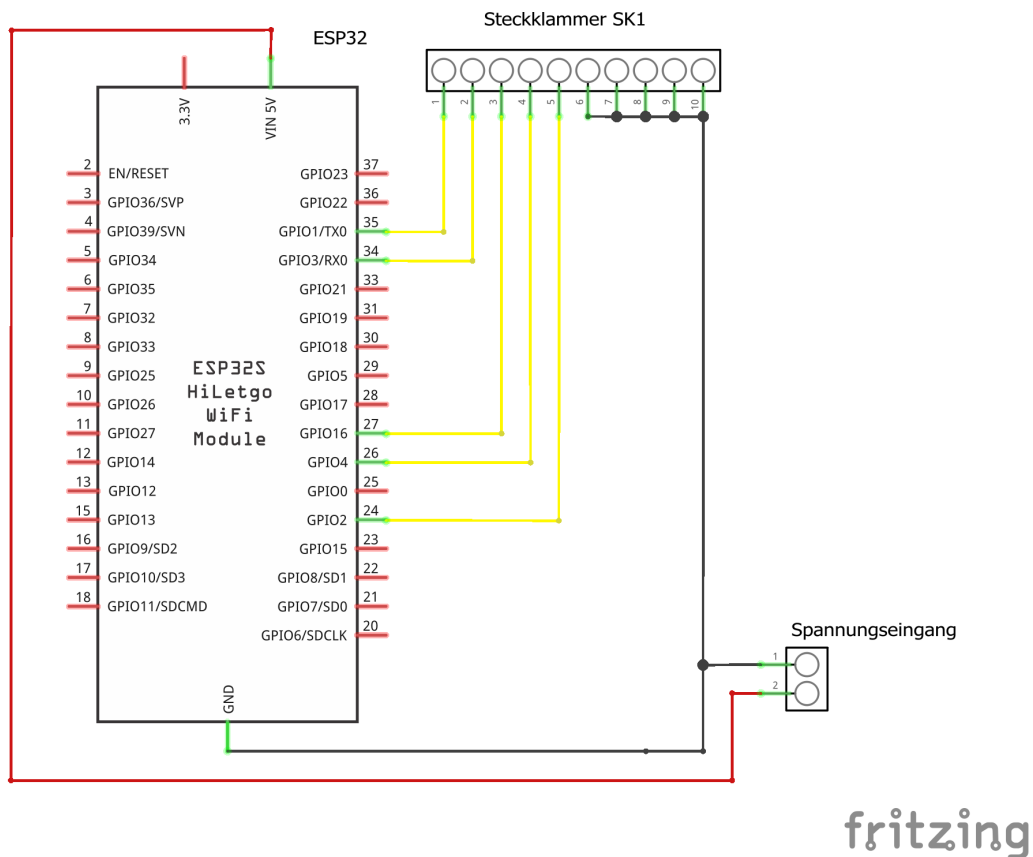


Abbildung 4.5: Schaltplan LED Controller

Da die genutzte LED-Controller Software WLED eine große Anzahl an verschiedenen LED-Streifen Typen unterstützt, die auch höhere Spannung benötigen als der ESP32, wird die Spannungsversorgung des LED-Streifens über der Platine in Abbildung 4.4 bewusst ausgelassen, um eine höhere Basis-Kompatibilität zu ermöglichen. Die Spannungsversorgung des LED-Streifens kann über Spannungseinspeisung ermöglicht werden. Abbildung 4.6 stellt diese Einspeisung dar.



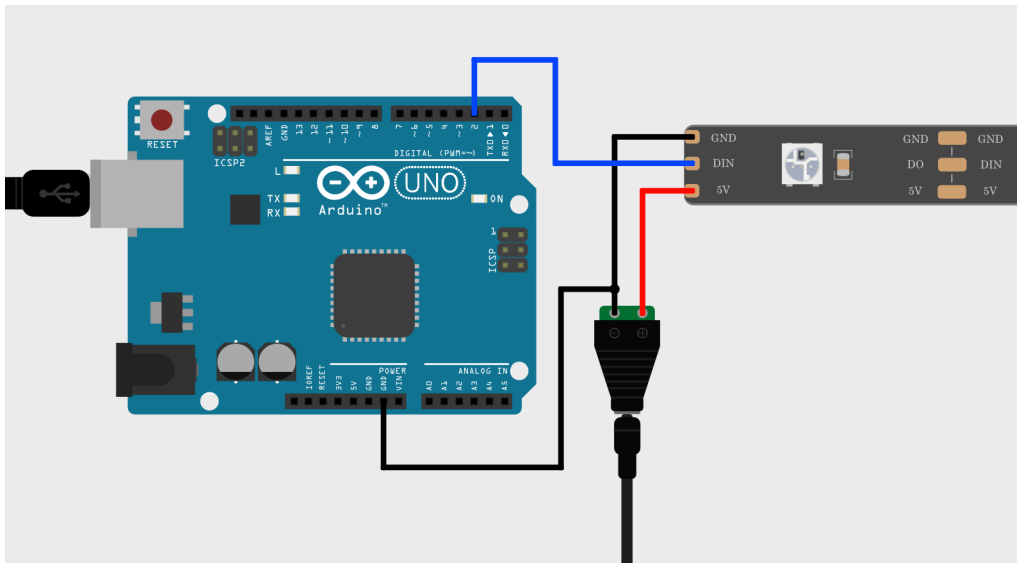


Abbildung 4.6: Einfache Spannungseinspeisung des LED-Streifens. Hier wird die Spannungsversorgung des Controllers und des LED-Streifens getrennt. Links ist exemplarisch ein Arduino Uno zu sehen, welcher über den USB-Anschluss mit Spannung versorgt wird. Rechts ein LED-Streifen mit einzeln ansteuerbaren LEDs, welcher über Spannungseinspeisung mit Spannung versorgt wird. Die Leitungen sind wie folgt zugewiesen: Schwarz = Ground, Blau = Dateninput, Rot = Spannung. Die Masse wird dennoch miteinander verbunden, um das gleiche Bezugspotential zwischen dem LED-Controller und dem LED-Streifen zu haben<sup>3</sup>. (Project Tempos, 2022)

Da die in dieser Arbeit genutzten LED-Streifen 5V benötigen, und dies bei dem ESP32 auch der Fall ist, kann hier von einer einzigen Spannungsquelle, wie in Abbildung 4.7 dargestellt ist, Gebrauch gemacht werden.

<sup>3</sup>[https://www.biancahoegel.de/technik/elektrik/masse\\_elek.html](https://www.biancahoegel.de/technik/elektrik/masse_elek.html)

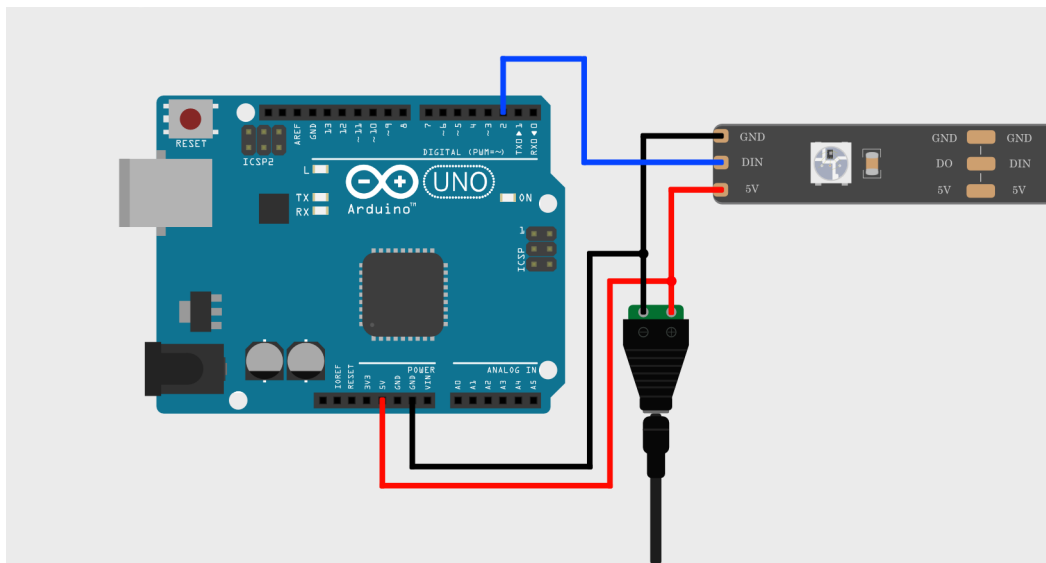


Abbildung 4.7: Spannungsversorgung des LED-Streifens und des LED-Controllers. Hier werden der Controller und der LED-Streifen zusammen mit der gleichen Spannung versorgt. Hierfür benötigen wir nur eine Spannungsquelle. (Project Tempos, 2022)

In Abbildung 4.6 und 4.7 ist exemplarisch ein Arduino Uno Mikrocontroller dargestellt. Die Verkabelung unterscheidet sich aber kaum vom ESP32.

## 4.2 Nachrichtenformate

In dieser Arbeit wird Gebrauch von verschiedenen Nachrichtenformaten gemacht. In diesem Abschnitt gehen wir genauer drauf ein, um im weiteren Verlauf verständlich die Algorithmen erklären zu können.

### 4.2.1 PointCloud2

Die PointCloud2 ist eins in ROS integriertes Nachrichtenformat, welches für Punktwolken von beispielsweise LiDAR's genutzt wird. Eine Punktwolke besteht aus vielen Punkten

in einem n-dimensionalen Raum, welches Strukturen von erkannten Objekten / Hindernissen darstellen sollen. Das Nachrichtenformat ist in der Dokumentation von ROS beschrieben<sup>4</sup>.

### 4.2.2 MarkerArray

Der MarkerArray ist eins in ROS integriertes Nachrichtenformat, welches ermöglicht verschiedene primitive geometrische Formen zu definieren (ROS, 2022). Das MarkerArray ist eine Liste, welches Nachrichten des Typs `Marker`<sup>5</sup> speichert. Diese können mit RViz visuell dargestellt werden. Die Darstellung dieser Formen/Marker ist in Abbildung 4.8 zu sehen. In Abbildung 4.15 ist auf der linken Seite RViz dargestellt. Die dort dargestellten quadratischen Boxen sind die in dieser Arbeit verwendeten Marker.

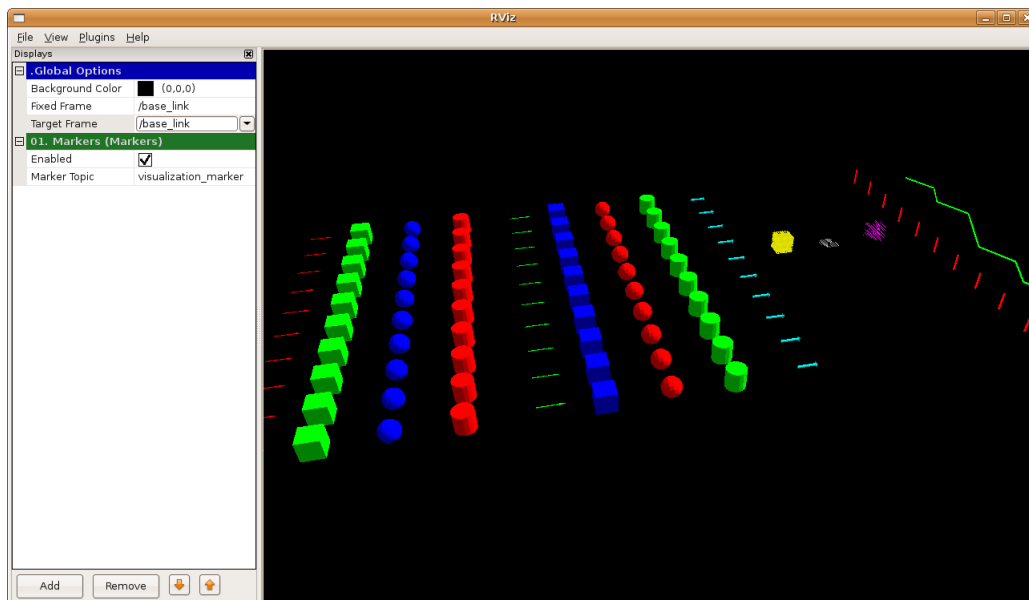


Abbildung 4.8: Beispielhafte Darstellung vom Nachrichtenformat `Marker`, welche visuell in RViz dargestellt sind<sup>6</sup>.

<sup>4</sup>PointCloud2 Nachrichtenformat [http://docs.ros.org/en/noetic/api/sensor\\_msgs/html/msg/PointCloud2.html](http://docs.ros.org/en/noetic/api/sensor_msgs/html/msg/PointCloud2.html)

<sup>5</sup>Die vollständige Spezifikation der Nachrichtendefinition `Marker`, kann unter [http://docs.ros.org/en/noetic/api/visualization\\_msgs/html/msg/Marker.html](http://docs.ros.org/en/noetic/api/visualization_msgs/html/msg/Marker.html) nachgelesen werden.

<sup>6</sup>Abbildung übernommen von <http://wiki.ros.org/rviz/DisplayTypes/Marker>

### 4.2.3 PointsArray

Das PointArray ist ein selbstdefiniertes Nachrichtenformat, welches für die Speicherung von Personenerkennung genutzt wird.

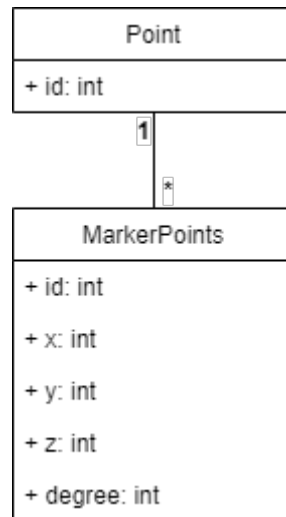


Abbildung 4.9: Klassendiagramm für das Nachrichtenformat eines PointsArray. Jede erkannte Person erhält eine neue ID, welche von der 4.4 zugewiesen wurde. In  $x$ ,  $y$  und  $z$  werden die Koordinaten der erkannten Person eingetragen. `degree` entspricht dem berechneten relativen Winkel der Person zum autonomen Roboter.

In Abbildung 4.9 sehen wir den Aufbau dieses Formates. In **MarkerPoints** wird die, von der Personenerkennungskomponente, zugewiesene ID, die  $x$ ,  $y$ ,  $z$  Koordinaten und den berechneten relativen Winkel zum autonomen Roboter eingetragen. Diese Inhalte werden vom HMI System ausgelesen und einem LED-Segment auf dem LED-Streifen zugewiesen.

```
1 {1:
2     MarkerPoints{
3         id=1,
4         x=5,
5         y=6,
6         z=10,
7         degree=75
8     },
9 2:
10    MarkerPoints{
11        id=2,
12        x=10,
13        y=2,
14        z=10,
15        degree=180
16    }
17 }
```

Code Listing 4.1: Beispielhafte Darstellung eines PointsArray, anhand der Definition von Abbildung 4.9. Hier exemplarisch zwei erkannte Personen mit der ID 1,  $x=5$ ,  $y=6$   $z=10$ ,  $degree=75$  / ID 2,  $x=10$ ,  $y=2$ ,  $z=10$ ,  $degree=180$

Das Point-Objekt in Abbildung 4.9 ist eine Liste, welche mit MarkerPoints gefüllt ist. Jedes MarkerPoints wird mit der dazugehörigen ID der erkannten Person als Schlüssel/Key in dieser Liste gespeichert. Die MarkerPoints sind Klassenobjekte, welche mit Daten der erkannten Personen befüllt sind, welche zuvor aus der Kommunikation mit der Personenerkennungskomponente extrahiert wurden. Die Kommunikation mit der Personenerkennungskomponente ist in Kommunikation mit HMI System (4.4.1) genauer beschrieben.

#### 4.2.4 Segment Array

Das SegmentArray ist ein Nachrichtenformat der Software WLED, welches zur Kommunikation mit dem LED-Controller genutzt wird.

Über eine POST Anfrage an den Endpunkt `led-controller-hostname/json` oder `led-controller-hostname/json/state` wird eine Liste mit Segmenten gesen-

det. `led-controller-hostname` steht für die IP-Adresse oder Hostname des LED-Controllers. Diese werden vom LED-Controller verarbeitet und die gewünschten LEDs werden beleuchtet.

```
1 {
2     "seg": [{
3         "id": 0
4         "start": 0,
5         "stop": 20,
6         "col": [
7             [255, 160, 0]
8         ],
9         "fx": 0,
10        "sx": 127,
11        "bri": 255
12    }]
13 }
```

Code Listing 4.2: Darstellung eines SegmentArrays, welches via JSON Anfrage an WLED gesendet wird. `seg` ist eine Liste von Segmenten. Jedes Segment wird durch einen Start und Stop definiert. Dies sind die Indexe der LEDs, die das Segment definieren. Über `col`, `fx` und `sx` können die Farb- und Effekteinstellungen eingestellt werden. Über `bri` kann die individuelle Helligkeit des Segmentes eingestellt werden <sup>7</sup>.

In Listing 4.2 ist beispielhaft ein solches Segment dargestellt. Dieses Segment erhält die ID 0 und startet an der ersten LED (`start`) auf dem Streifen bis zur LED mit dem Index 20 (`stop`). Die Farbe (`col`) des Segmentes entspricht dem RGB Code [255, 160, 0]. Der Effekt (`fx`) ist 0, welcher nach Definition von WLED einer soliden Darstellung ohne Animation entspricht <sup>8</sup>. Die Effektgeschwindigkeit (`sx`) ist 127. Die Helligkeit (`bri`) ist 255 und entspricht dem Maximum, wodurch das Segment mit voller Leuchtkraft leuchtet.

---

<sup>7</sup><https://kno.wled.ge/interfaces/json-api/#contents-of-the-segment-object>

<sup>8</sup>Alle vorhandenen Effekte können unter <https://kno.wled.ge/features/effects-palettes/> nachgelesen werden. Dort werden diese animiert zum besseren Verständnis dargestellt.

### 4.3 Konfigurationsparameter

Um maximale Flexibilität zu gewährleisten, besteht die Möglichkeit über eine Konfigurationsdatei, verschiedenste Parameter zu verändern. Außerdem können bestimmte Parameter zur Laufzeit über eine API angepasst werden, welche im Anschluss automatisch gespeichert werden. In Abschnitt 4.3.1 wird genauer darauf eingegangen. Im nachfolgenden Teil werden die unterstützten Konfigurationsparameter genauer erläutert.

```
1 {
2     "node_name": "hmi_node",
3     "default_state": "normal",
4     "api_port": 8001,
5     "api_password": "HuskyHAW",
6     "led": {
7         "controller_hostname": "http://192.168.178.200/
8             json/",
9         "count": 120,
10        "adjacent": 2,
11        "components": {
12            "detect_people": {
13                "color": [[0, 0, 255]],
14                "fx": 0,
15                "sx": 1,
16                "brightness": 120
17            },
18            "turning_indicator": {
19                "color": [[255, 165, 0]],
20                "fx": 1,
21                "sx": 180,
22                "brightness": 120,
23                "top_left": [106,112],
24                "top_right": [8,14],
25                "bottom_left": [69,75],
26                "bottom_right": [46,52]
27            }
28        }
29    }
30 }
```

```
28     },
29     "topic": {
30         "turning": "/turning_indicator",
31         "state": "/hmi_status"
32         "hdl_people_tracking_markers": "/
           hdl_people_tracking_nodelet/markers"
33     },
34     "system_states": {
35         "normal": {
36             "color": [[0, 0, 0]],
37             "fx": 0,
38             "sx": 1,
39             "brightness": 50
40         },
41         "attention": {
42             "color": [[0, 0, 255]],
43             "fx": 2,
44             "sx": 180,
45             "brightness": 50
46         },
47         "warning": {
48             "color": [[232, 174, 21]],
49             "fx": 2,
50             "sx": 180,
51             "brightness": 50
52         },
53         "error": {
54             "color": [[195, 52, 72]],
55             "fx": 2,
56             "sx": 180,
57             "brightness": 50
58         }
59     }
60 }
```



Code Listing 4.3: Konfigurationsdatei zur Definition der Parameter des HMI Systems. Exemplarisch in Zeile 56 ist die Definition des error-Systemstatus zu sehen. Dort wird in `color` die RGB-Werte für die Darstellung definiert, in `fx` die ID des Effekts von WLED, in `sx` die Effektgeschwindigkeit und in `brightness` die Helligkeit bei der Darstellung des Status.

Im weiteren Verlauf wird jeder einzelner Parameter erläutert:

### **node\_name**

Knoten Namen, welcher zum Abonnieren eines Topic's gebraucht wird. Wird bei Initialisierung des HMI System's angegeben.

```
1 rospy.init_node(self.config.node_name, anonymous=True)
```

Code Listing 4.4: Initialisierung von `rospy`. `rospy` ist eine Python Bibliothek, welche zur Kommunikation mit ROS benötigt wird. Bei `init_node` wird anfangs der Name der Node festgelegt. Dieser wird aus der Konfiguration (`node_name`) ausgelesen. `anonymous` gibt an, ob es mehrere Nodes mit dem gleichen Namen geben kann.<sup>9</sup>

### **default\_state**

Standard Status, welcher bei Initialisierung, als auch nicht erkannten Status oder Ausnahmefällen, angenommen wird.

### **api\_port**

Der Port, über welchen die API zugänglich ist.

### **api\_password**

Passwort zur Authentifizierung mit dem API-Gateway.

---

<sup>9</sup><http://wiki.ros.org/rospy/Overview/Initialization%20and%20Shutdown>

### led/count

Die Anzahl der LEDs auf dem gesamten Streifen.

### led/adjacent

Anzahl der benachbarten LEDs, die bei Darstellung von erkannten Personen mit leuchten sollen. In Abbildung 4.10 ist dieses Konzept exemplarisch dargestellt. Es soll die LED mit dem Index 6 leuchten. Durch Definieren von led/adjacent auf den Wert 2, werden LED 4, 5, 7 und 8 auch mit beleuchtet.



Abbildung 4.10: Exemplarische Darstellung der Beleuchtung von Nachbar-LEDs. LED 6 soll leuchten. Durch Einstellen der Konfiguration led/adjacent, dass auf jeder Seite 2 LEDs mitleuchten sollen, werden LED 4,5,7 und 8 mitbeleuchtet.<sup>10</sup>

### led/components

Definition von Farben, Effekten, Effektgeschwindigkeit und Helligkeit für bestimmte Komponenten, wie die Erkennung von Personen.

**color** Zweidimensionales Array mit RGB Farbcodes. Format: [[255, 255, 255]]

**fx** ID des gewünschten Effektes. Eine Liste der Effekte ist hier einsehbar <sup>11</sup>.

**sx** Effektgeschwindigkeit. Wert zwischen 0 - 255.

**brightness** Helligkeit des LED-Bereiches. Wert zwischen 0 - 255.

### led/components/detect\_people

Farbeinstellungen für die Personenerkennung. Enthält Einstellungen wie in led/components

---

<sup>10</sup>Angepasste Grafik von <https://kno.wled.ge/features/segments>

<sup>11</sup>Liste der unterstützten Effekte <https://kno.wled.ge/features/effects-palettes>

### **led/components/turning\_indicator**

Ergänzend zu den in led/components genannten Einstellungen, müssen die LEDs für die Blinker definiert werden. Jede Ecke des autonomen Roboters wird mit einem Index für die Startled und dem Index der Endled als 2er Tupel angegeben. In Abbildung 4.11 ist das Konzept visuell dargestellt. Jede Ecke, also vorne links, vorne rechts, hinten links und hinten rechts, wird als 2er Tupel in der Konfiguration angegeben. Wie in Abbildung 4.11 dargestellt, soll eine Ecke von LED-Index 4 bis LED-Index 8 als Blinker definiert werden. Somit tragen wir in der korrespondierenden Ecke den Wert [4, 8] ein.



Abbildung 4.11: Beispielhafte Darstellung zum Festlegen der Blinkbereiche. In der Abbildung sind die LED 4 bis 8 markiert. Somit muss in der Konfiguration in led/components/turning\_indicator an der korrespondierenden Ecke [4, 8] eingetragen werden.<sup>12</sup>

### **topic/turning\_indicator**

Topic, in welches die zu blinkenden Blinker bestimmt werden. Mögliche Werte: left, right, all.

### **topic/state**

Topic, in welchem die Status des autonomen Roboters verwaltet werden soll. In dieses Topic können andere Komponenten bestimmte Status triggern. Die Farb- und Effekteinstellungen der Status werden in system\_states definiert.

### **topic/hdl\_people\_tracking\_markers**

Topic, in welches die Personenerkennungskomponente die Marker der erkannten Personen veröffentlicht.

---

<sup>12</sup>Angepasste Grafik von <https://kno.wled.ge/features/segments>

### `system_states`

In diesem Bereich werden alle unterstützten Status des autonomen Roboters definiert. Zu jedem Status muss wie in `led/components` die Farbeinstellungen mit eingetragen werden.

Alle Parameter, bis auf die einzelnen System States, sind verpflichtend und werden beim Starten des HMI System überprüft. Bei Fehlern wird eine Fehlermeldung ausgegeben und das HMI System wird beendet.

#### 4.3.1 API

Um die Werte auch zur Laufzeit anpassen zu können, werden für bestimmte Konfigurationsparametern eine API zu Verfügung gestellt. Es können die LED Farbeinstellung (`led/components`) der Komponenten und der Systemstatus angepasst werden. Außerdem können die Zuweisung der LEDs, welche mit dem Blinker assoziiert sind, angepasst werden.

Wie in Abbildung 4.12 zu sehen, existieren für die 3 wesentlichen Anforderungen Endpunkte. Jeder Endpunkt besitzt einen `GET-Request` zum Anfordern der aktuellen Konfiguration dieser Komponente. Die dort enthaltenen Werte können verändert werden und an den gleichen Endpunkt mittels `POST-Request` übermittelt werden.

Der Zugriff zur API ist über ein Passwort gesichert. Dieses kann in den Konfigurationsparametern (`api_password`) eingestellt werden. Über den Header »Authorization« kann das Passwort im Aufruf mitgegeben werden. Nicht autorisierte Zugriffe werden mit dem HTTP 401 Status blockiert.

Zu besseren Validierung der Eingaben, wurde ein Schema in OpenAPI 3 erstellt. Mit diesem Schema können einerseits alle möglichen Endpunkte, sowie Ihre Ein- und Rückgaben, dargestellt werden, sondern auch automatisiert die Endpunkte anhand des Schemas abglichen werden. Nicht valide Anfragen werden mit dem HTTP Status 400 Bad Request, sowie mit einer Fehlermeldung, mit Angabe des fehlerhaften Eintrags beantwortet.

config		Changing the configuration values		See documentation of LED Controller for possible values related to fx etc.: <a href="#">http...</a>	
GET	/config	Get current config		▼	🔒 ↩
POST	/config/components/detect_people	set people detection config values		▼	🔒 ↩
GET	/config/components/detect_people	get people detection config values		▼	🔒 ↩
POST	/config/components/turning_indicator	set turning indicator config values		▼	🔒 ↩
GET	/config/components/turning_indicator	get turning indicator config values		▼	🔒 ↩
POST	/config/system_states/{state}	set specific system states config values		▼	🔒 ↩
GET	/config/system_states/{state}	get specific system states config values		▼	🔒 ↩

Abbildung 4.12: Darstellung aller Endpunkte der API zum Einstellen der Konfigurationsparameter

Abbildung 4.12 zeigt alle verfügbaren Endpunkte der API zum Einstellen der Konfigurationsparameter. Dieser Ansatz erlaubt in Zukunft eine Applikation zu entwickeln, die unabhängig von ROS, mit dem HMI System kommunizieren kann.

## 4.4 Personenerkennungskomponente

Das in Kapitel 4.5.1 genutzte Package zur Personenerkennung wurde von Koide u. a. (2019) entwickelt. In deren Arbeit geht es um die Analyse und Messung von Personenverhalten, um passende Interaktionssysteme zu entwickeln. Der zum Paper dazugehörige Source-Code wurde in GitHub veröffentlicht<sup>13</sup>. Der dort verfolgte Ansatz stammt von Haselich u. a. (2014). Die Personenerkennung in der LIDAR Punktwolke erfolgt nach dem Kidono u. a. (2011) Ansatz.

### 4.4.1 Kommunikation mit HMI System

Die Personenerkennungskomponente speichert die erkannten Personen in einem Topic, welches standardmäßig `/hdl_people_tracking_nodelet/markers` ist. Sollte dieses verändert werden, kann in der Konfiguration das neue richtige Topic unter dem Konfigurationspunkt `topic/hdl_people_tracking_markers` angegeben werden.

<sup>13</sup>[https://github.com/koide3/hdl\\_people\\_tracking](https://github.com/koide3/hdl_people_tracking)

Im `/hdl_people_tracking_nodelet/markers` Topic, im weiteren Verlauf Tracking-Marker-Topic genannt, werden Nachrichten des Typs `MarkerArray` (4.2.2) eingetragen. Diese wurden bei Initialisierung des HMI System angegeben. In Listing 4.5 wird der Code zum Abonnieren eines Topic's gezeigt.

```
1  rospy.Subscriber(  
2  self.config.topic.hdl_people_tracking_markers,  
3  MarkerArray,  
4  self.callback_position,  
5  queue_size=1)
```

Code Listing 4.5: Beispielcode zum Abonnieren eines Topics. Hier wurde das Topic abonniert, welches die Marker der erkannten Personen enthält. Im Konstruktor von `rospy.Subscriber` übergeben wir in Zeile 2 den Namen des zu abonnierenden Topics, danach in Zeile 3 den Typ der enthaltenen Nachricht, anschließend in Zeile 4 die Methode die ausgeführt werden soll beim Eingang von Nachrichten und am Schluss in Zeile 5 wie groß die Warteschlange sein soll.

In Zeile 2 übergeben wir den Namen des Tracking-Marker-Topic's, welches aus der Konfiguration ausgelesen wird. In Zeile 3 wird der Typ der Nachrichten in diesem Topic angegeben. In Zeile 4 der Name der Funktion, der bei Eingang von Nachrichten im Tracking-Marker-Topic ausgeführt werden soll. Zeile 5 definiert die Größe der Warteschlange. In dem Fall wird immer nur eine Nachricht gespeichert. Alles andere wird verworfen.

Für jede erkannte Person wird dann ein Marker erstellt. Dafür ermittelt die Personenerkennungskomponente, die Position der erkannten Person. In ROS wird ein imaginäres Koordinatensystem erstellt, dessen Mittelpunkt der autonome Roboter ist. Somit befindet sich der autonome Roboter auf der Position  $x = 0$ ,  $y = 0$ ,  $z = 0$ . Die Personenerkennungskomponente trägt die ermittelten Koordinaten der erkannten Person in `Marker.Pose.Position` ein<sup>14</sup>. Unter Position befinden sich dann die  $x$ ,  $y$  und  $z$  Koordinaten der erkannten Person, welche vom HMI System ausgelesen werden. In Listing 4.6 wird exemplarisch der Code zum Auslesen der Position und das Eintragen ins `PointsArray` dargestellt.

---

<sup>14</sup>Die Nachrichtendefinition von `Marker` kann unter [http://docs.ros.org/en/noetic/api/visualization\\_msgs/html/msg/Marker.html](http://docs.ros.org/en/noetic/api/visualization_msgs/html/msg/Marker.html) nachgelesen werden.

```
1 def callback_position(self, data):
2     for mark in data.markers:
3
4         x = mark.pose.position.x
5         y = mark.pose.position.y
6         z = mark.pose.position.z
7
8         marker_id = mark.text.split(":")[1]
9
10        degree = round(math.atan2(y, x) * (180 / math.pi))
11
12        self.pointArray[marker_id] = MarkerPoints(marker_id, x, y,
13            z, degree)
14    [...]
```

Code Listing 4.6: Beispielcode zum Auslesen der Positionen der erkannten Personen durch die Personenerkennungskomponente. Diese werden dann ins PointsArray eingetragen, um im weiteren Verlauf durch das HMI System ausgelesen werden zu können.

Durch das Abonnieren des Tracking-Marker-Topic's, wird bei jeder eingefügten Nachricht in dieses Topic, die Funktion in Listing 4.6 ausgeführt. Wir erhalten im Funktionsaufruf die Variabel `data`, welche den Typ `MarkerArray` hat. Da `MarkerArray` eine Liste ist, welche viele einzelne Marker enthalten kann, wird eine `for`-Schleife, wie in Zeile 2 gesehen eingebaut. Somit können wir auf jeden einzelnen Marker zugreifen. In Zeile 4,5 und 6 werden die `x`, `y` und `z` Koordinaten zwischengespeichert. In Zeile 8 die enthaltene ID, die bereits durch die Personenerkennungskomponente zugewiesen wurde. In Zeile 10 wird der Winkel relativ zum autonomen Roboter berechnet, welcher in Formel 4.1 noch erläutert wird. Diese Informationen werden dann in einem `PointsArray` eingefügt und an den LED-Controller weitergeleitet. Hierauf wird in Abschnitt 4.5.1 noch genauer eingegangen.

## 4.5 Entwurf

Die Sprache des autonomen Roboters wird nach den in Kapitel 3 gestellten Anforderungen modelliert. Im folgendem Kapitel wird die Funktionsweise jeder Anforderung erläutert.

### 4.5.1 REQ1 Personenerkennung anzeigen

Der autonome Roboter soll signalisieren können, dass er Personen in seinem Umfeld erkennt. Dafür wird ein auf dem Dach montierter LiDAR der Firma RoboSense genutzt. Dieser gibt eine Punktwolke zurück.

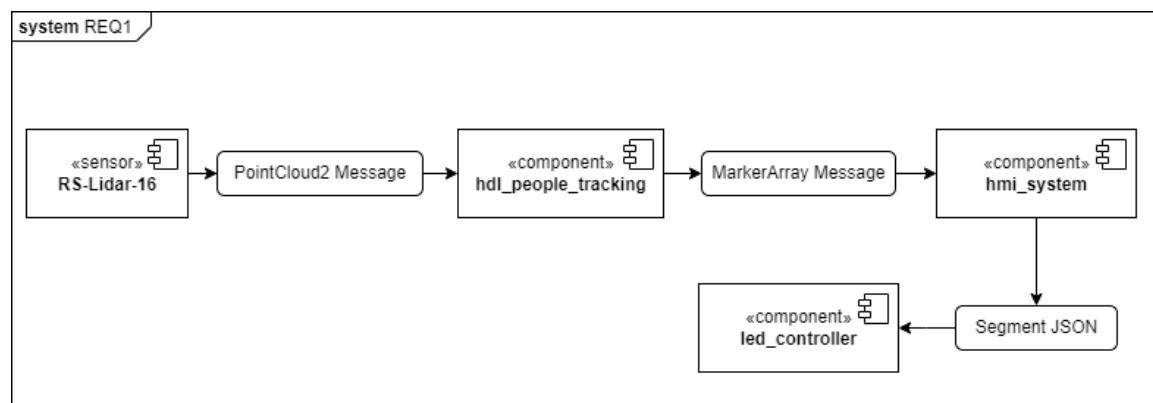


Abbildung 4.13: Komponentenübersicht der Personenerkennung, sowie die Nachrichtenschemata, die zur Kommunikation zwischen den Komponenten genutzt werden.

In Abbildung 4.13 wird die Übersicht, der in dieser Anforderung genutzten Komponenten, angezeigt. Der LiDAR gibt eine Punktwolke zurück, welche als PointCloud2 Nachricht an die Personenerkennungskomponente gesendet wird. Die Personenerkennungskomponente weist jeder erkannten Person eine ID zu und erstellt einen Marker, wie in Abschnitt 4.2.2 beschrieben, um diese Person herum. Dieser Marker enthält die Koordinaten der Person. Eine Liste der erkannten Personen wird als MarkerArray Nachricht an das HMI System gesendet. Im Anschluss wird ein Segment Array, wie in Abschnitt 4.2.4 beschrieben, als JSON an den LED-Controller gesendet. Dieser aktiviert die gewünschten LEDs auf dem Streifen.



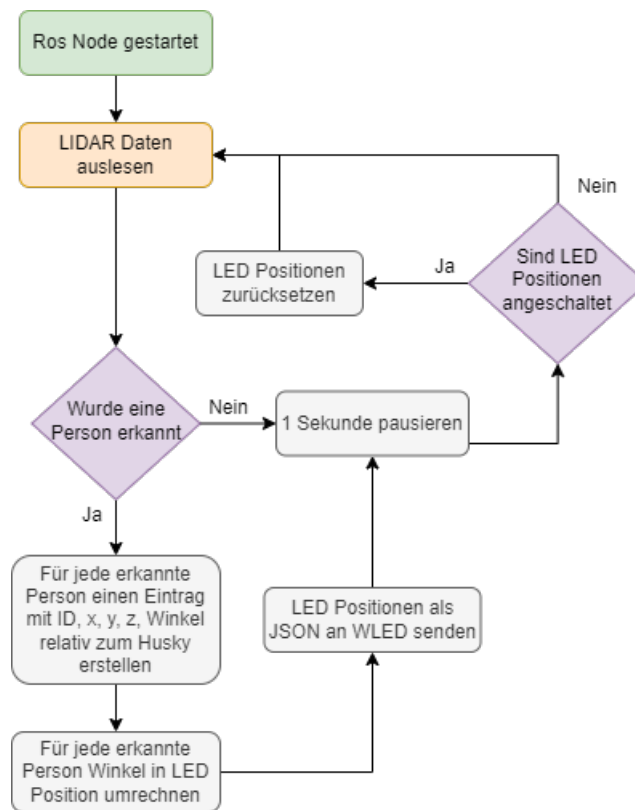


Abbildung 4.14: Ablaufdiagramm der Personenerkennung, sowie die dazugehörige Darstellung im LED-Strip

In Abbildung 4.14 wird zum besseren Verständnis den Algorithmus der Personenerkennung, sowie die Kommunikation mit dem LED-Controller, dargestellt.

Beim Starten der ROS Node werden die Daten des LiDAR ausgelesen und in das ROS Topic "rslidar\_points- veröffentlicht.

Die Personenerkennungskomponente (Kapitel 4.4) abonniert dieses Topic und bekommt alle Daten des LiDAR's als Nachricht im PointCloud2 Format. Diese erkennt nach Hase-lich u. a. (2014) Personen in der Punktwolke und erstellt für jede erkannte Person einen Marker, welcher visuell als Box dargestellt wird. In Abbildung 4.15 sind diese Boxen beispielhaft dargestellt. Jede Box auf der linken Seite von Abbildung 4.15 ist mit einer Person auf der rechten Seite der gleichen Abbildung assoziiert. Diese Boxen werden in ROS im Nachrichtenformat MarkerArray dargestellt.

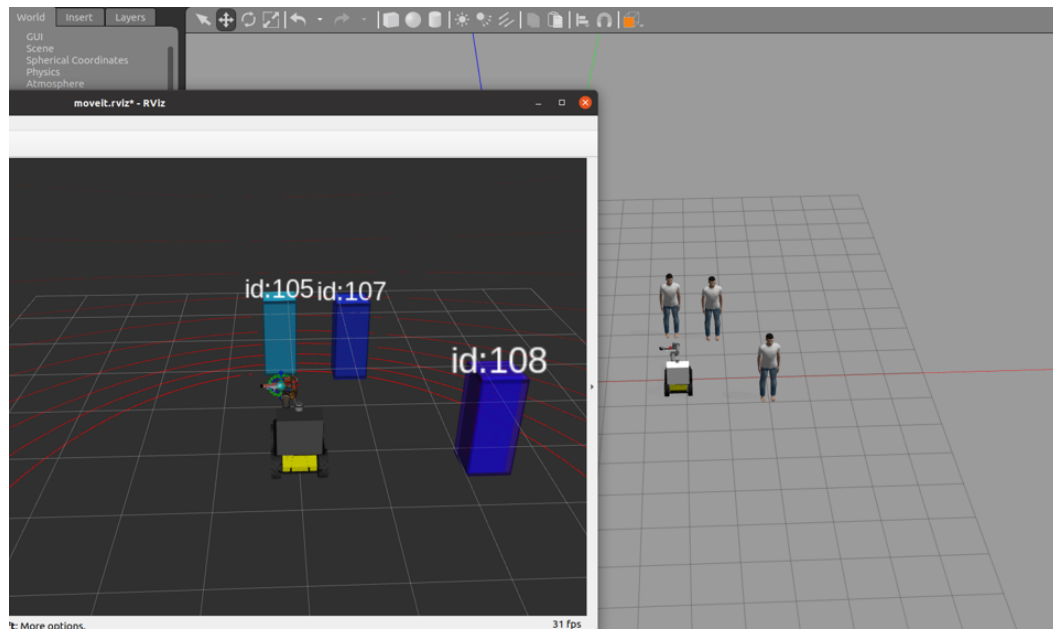


Abbildung 4.15: Beispielhafte Darstellung der Personenerkennung in der Simulationsumgebung Gazebo

Alle erkannten Personen werden in eine Liste des Typs `MarkerArray` eingefügt und über ein eingestelltes Topic veröffentlicht. Dieses liest das HMI System aus. Über `topic/hdl_people_tracking_markers` in der Konfiguration kann das zu abonnierende Topic eingestellt werden, welches das `MarkerArray` beinhaltet.

Das HMI System iteriert nun über die Liste `MarkerArray` und berechnet für jede Person/jeden Marker, den relativen Winkel zum autonomen Roboter basierend auf den  $x$  und  $y$  Koordinaten der Marker. In Formel 4.1 wird die Berechnung des relativen Winkels dargestellt. Die Formel 4.1 ist eine leicht abgewandelte Version, als die von Miller (2016) vorgestellt wurde<sup>15</sup>. Hier werden die Koordinaten des Markers eingesetzt. Der autonome Roboter ist der Mittelpunkt, von dem in der Berechnung ausgegangen wird. Er erhält die Koordinate  $(0, 0)$ , wodurch dieser in der Berechnung keine Rolle spielt. Dies wurde bereits in Abschnitt 4.4.1 beschrieben.

$$\text{relativerWinkel} = \text{atan2}(Y, X) * \frac{180}{\pi} \quad (4.1)$$

<sup>15</sup><https://math.stackexchange.com/a/1596518>

Nach der Berechnung des relativen Winkels, werden diese mit den dazugehörigen Koordinaten und der zugewiesenen ID, durch die Personenerkennungskomponente, in eine interne Datenstruktur, nach Abschnitt 4.2.3 PointsArray, des HMI System's eingetragen. Dieses PointsArray wird dann an die LED Komponente gesendet.

Die LED Komponente ist für die Kommunikation zwischen HMI System und dem WLED LED-Controller zuständig. Die LED Komponente iteriert nun über das PointsArray, welches die erkannten Personen enthält. Für jeden Eintrag wird der enthaltene Winkel auf die Anzahl der verfügbaren LEDs auf dem Streifen abgebildet. Dies wird nach Formel 4.2 umgerechnet.

$$winkelAlsLED = x * y / 360 \quad (4.2)$$

Bei Formel 4.2 gilt das x als Winkel des aktuellen Eintrags aus dem PointsArray, und das y als die Anzahl der LEDs auf dem Streifen, welche aus der Konfiguration entnommen wird (led/count). Das Ergebnis der Multiplikation von x und y wird dann durch 360 dividiert, um das Ergebnis der Multiplikation auf den Zahlenraum eines Kreises umzuwandeln. Danach erhalten wir den Index der LED, welche mit dem Winkel der erkannten Person zusammenhängt. Es gibt außerdem die Möglichkeit, über die Konfigurationsparametern einzustellen, wie viele benachbarte LEDs, bei der Darstellung der Erkennung der Person, mit angezeigt werden sollen. In Abbildung 4.10 ist dieses Prinzip bereits erläutert worden.

Dafür muss eine untere und obere Grenze berechnet werden, die den Bereich der zu leuchtenden LEDs abbildet. Hier kann es aber vorkommen, dass beispielsweise bei einem Winkel 0, welcher in der Formel 4.2 einen Wert von 0 ausgibt, der Index auf die erste LED festlegt wird. In dem Fall muss beachtet werden, dass die untere Grenze der benachbarten LEDs negativ ist und somit beim maximalen LED-Index anfängt. Für die Abdeckung dieses Falles schafft Formel 4.3 Abhilfe.

$$untereGrenze = \begin{cases} winkelAlsLED - z, & \text{if } winkelAlsLED - z > 0 \\ a + (winkelAlsLED - z), & \text{ansonsten} \end{cases} \quad (4.3)$$

In Formel 4.3 wird der aus Formel 4.2 errechnete Wert eingesetzt.  $z$  ist der Wert des Konfigurationsparameters `led/adjacent`.  $a$  ist der Wert des Konfigurationsparameters `led/count`.

Die Berechnung in Formel 4.3 ist davon abhängig, ob die Subtraktion des `led/adjacent` vom `winkelAlsLED` einen negativen Wert hervorruft. Wenn dieser Fall zutrifft, wird die gesamte Anzahl der LEDs des Streifens addiert.

$$obereGrenze = \begin{cases} \text{winkelAlsLED} + z, & \text{if } \text{winkelAlsLED} + z < a \\ (\text{winkelAlsLED} + z) - a, & \text{ansonsten} \end{cases} \quad (4.4)$$

Formel 4.3 berechnet den Index der Start-LED und Formel 4.4 den Index der End-LED. Somit wird für diese erkannte Person, der Bereich der zu leuchtenden LEDs, festgelegt. Die Farben und Effekte des Bereiches können über die Konfigurationsparameter definiert werden (`led/components/detect_people`).

Die in dieser Arbeit genutzte LED-Controller Software WLED nutzt Segmente zur Einteilung des LED-Streifens in unterschiedliche Bereiche. Jedes Segment kann unterschiedliche Farbeinstellungen, wie in den Konfigurationsparametern `led/components` aufgezeigt wird, darstellen. Außerdem muss ein Start-Index und End-Index für die LEDs definiert werden, mit welchem der Bereich des Segmentes beschränkt wird. Jede erkannte Person wird als Segment eingetragen.

Nach Durchlauf der Iterierung der erkannten Personen, wird das Segment Array via JSON API an den LED-Controller gesendet. Dies wurde bereits in Segment Array (4.2.4) beschrieben.

#### 4.5.2 REQ2 Systemstatus anzeigen

Der autonome Roboter soll seiner Umgebung bestimmte Systemstatus anzeigen. Beispielsweise, wenn der Roboter in einen Aufzug fährt, bietet das HMI System die Möglichkeit, dass andere Packages mit dem System interagieren können. Dafür bietet das HMI System ein Topic in ROS an. Dort können die Packages fest definierte Werte eintragen, die

vom HMI System erkannt und dargestellt werden. In den Konfigurationsparametern können beliebig viele Status eingefügt werden, mit verschiedenen LED-Farben und Effekten (Konfiguration `system_states`).

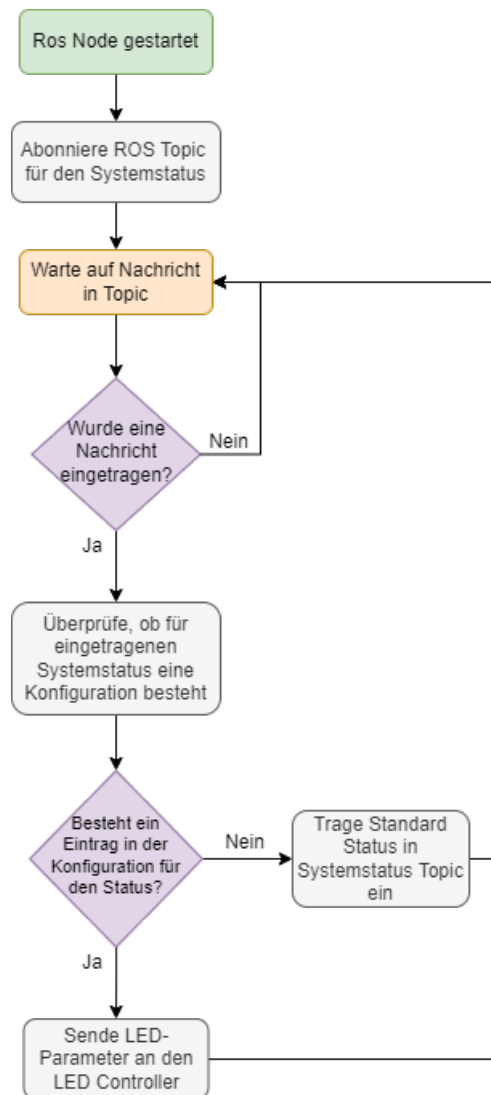


Abbildung 4.16: Ablaufdiagramm der Darstellung des Systemstatus

In Abbildung 4.16 wird die Logik der Darstellung des Systemstatus dargestellt.

In der Konfiguration, können wir angeben, in welchem Topic die Systemstatus eingetragen werden (`topic/state`). Dieses Topic wird vom HMI System abonniert. Sobald eine Nachricht in dieses Topic eingetragen wird, wird ein Event im HMI System getriggert.

Die ausgelöste Funktion überprüft, ob der gewünschte Systemstatus in der Konfiguration bereits definiert wurde (`system_states`). Sollte dies nicht der Fall sein, wird der Standard-Systemstatus, welcher in der Konfiguration definiert wurde (`default_state`), eingetragen. Bei vorhandenen Konfigurationsparametern für den gewünschten Status, werden die definierten LED-Parameter an den LED-Controller gesendet. Dieser spricht dann alle LEDs des LED-Streifens an, wobei zu beachten ist, dass die Darstellung des Systemstatus als Hintergrund dargestellt wird. Damit ist gemeint, dass beispielsweise die Darstellung von erkannten Personen priorisiert wird und die Darstellung des Systemstatus überschreibt.



- (a) Beispielhafte Darstellung des Fehlerzustands auf dem gesamten LED-Streifen. Alle LEDs leuchten anhand der in der Konfiguration definierten Einstellungen.



- (b) Überschreibung der Darstellung des Fehlerzustands durch den Blinker. Die Darstellung von Systemstatus werden nicht priorisiert und wird durch andere Komponenten überschrieben.

Abbildung 4.17: Darstellung von Systemstatus und Überschreibung durch andere Komponenten wie der Personenerkennung und des Blinkers. <sup>16</sup>

### 4.5.3 REQ3 Abbiegevorgang anzeigen

Der autonome Roboter soll Abbiegevorgänge anzeigen. Da bereits Fahrzeugblinker etabliert und weit verbreitet sind, nutzen wir diesen Vorteil. Komponenten des autonomen Roboters können in das `topic/turning_indicator` Topic einen String eintragen. Mögliche Werte sind `right`, `left` oder `all`. Dies beschreibt, welche Seite des Blinkers leuchten soll. Über die Konfiguration können die LED-Bereiche der Blinkerseiten definiert werden. Als Beispiel hier Abbildung 4.11. Hier werden beispielhaft die LED 4 bis 8 als Blinker definiert. Somit muss in der Konfiguration in `led/components/turning_indicator` an der korrespondierenden Ecke [4, 8] eingetragen werden. Dies wird für alle 4 Ecken in der Konfigurationsdatei eingetragen.

Der Ablauf zum Aktivieren der Blinker wird in Abbildung 4.18 veranschaulicht.

<sup>16</sup>Angepasste Grafik von <https://kno.wled.ge/features/segments>

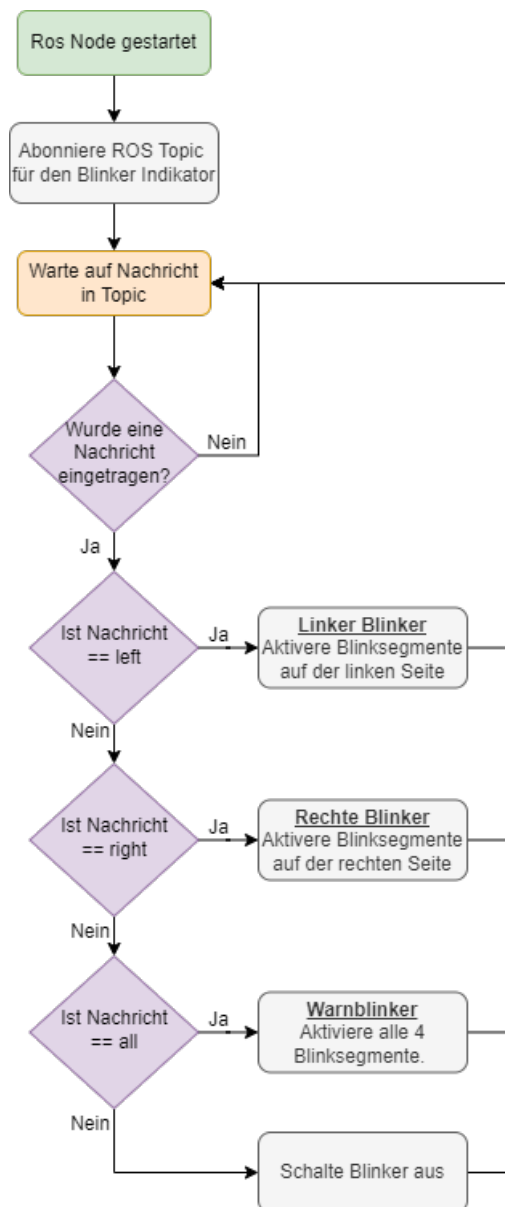


Abbildung 4.18: Ablaufdiagramm der Blinkersteuerung

Beim Starten der ROS Node, wird das mit dem Blinker Indikator korrespondierende Topic abonniert. Dieses wird über die Konfiguration definiert (`topic/turning_indicator`). Es wird geprüft, ob die erhaltene Nachricht, `left`, `right` oder `all` entspricht.

### **Left**

Die Nachricht `left` aktiviert die Blinker, die mit der linken Seite assoziiert sind. In der Konfiguration sind dies die Bereiche `top_left` und `bottom_left`. Je nach Farb- und Effekteinstellungen werden diese entsprechend beleuchtet.

### **Right**

Die Nachricht `right` aktiviert die Blinker, die mit der rechten Seite assoziiert sind. In der Konfiguration sind dies die Bereiche `top_right` und `bottom_right`. Je nach Farb- und Effekteinstellungen werden diese entsprechend beleuchtet.

### **All**

Die Nachricht `all` aktiviert die Blinker, die mit der linken und rechten Seite assoziiert sind. Vergleichsweise wie ein Warnblinker. In der Konfiguration sind dies die Bereiche `top_left`, `top_right`, `bottom_left` und `bottom_right`. Je nach Farb- und Effekteinstellungen werden diese entsprechend beleuchtet.

### **Ansonsten**

Sollte eine Nachricht eingehen die nicht bekannt ist, wird dies wie ein Reset gewertet und die Blinker werden deaktiviert.



# 5 Evaluierung

Die Evaluierung wird in zwei Teile aufgeteilt. Anfangs wird auf die objektive Evaluierung eingegangen. Hier wird überprüft, ob die fachlichen Anforderungen erfüllt worden sind. In der subjektiven Evaluierung wird das HMI System auf seine Nützlichkeit überprüft. Dafür wird eine Umfrage durchgeführt, wo die Teilnehmer das HMI System auf dem Roboter begutachten können, um Ihre Meinung äußern zu können.

## 5.1 Objektive Evaluierung

In der objektiven Evaluierung gehen wir auf die fachlichen Anforderungen, die in Kapitel 3 definiert wurden, ein. Es geht darum zu schauen, ob die Anforderungen erfüllt worden sind.

### **REQ1 Personenerkennung anzeigen**

Die Personenerkennung durch die Personenerkennungskomponente funktioniert wie erwartet in der Simulation. In Abbildung 4.15 ist in der Simulationsumgebung rechts Gazebo zusehen. Links die Darstellung der Erkennung der Personenerkennungskomponente in RViz. In Gazebo werden an verschiedenen Stellen rund um den autonomen Roboter Personen eingefügt. Diese werden erkannt und erhalten eine ID zugewiesen. Bewegt man diese Personen leicht, verschiebt sich die Position äquivalent in RViz und die ID wird nicht verändert. Wird eine Person weit vom Ausgangspunkt verschoben, erhält die Person eine neue ID und die alte Erkennung wird entfernt. Es wurde zur Verifizierung der Funktionalität der LED-Komponente, ein Prototyp gebaut. Der Prototyp ist eine rechteckige Holzplatte, welche mit einem LED-Streifen komplett umrandet wurde. In Abbildung 5.1a ist eine exemplarische Darstellung des Prototyps. Abbildung 5.1b zeigt eine exemplarische Darstellung von 2 erkannten Personen an. In Abbildung 5.2a und Abbildung 5.2b sind die verschiedenen Helligkeitsstufen dargestellt.

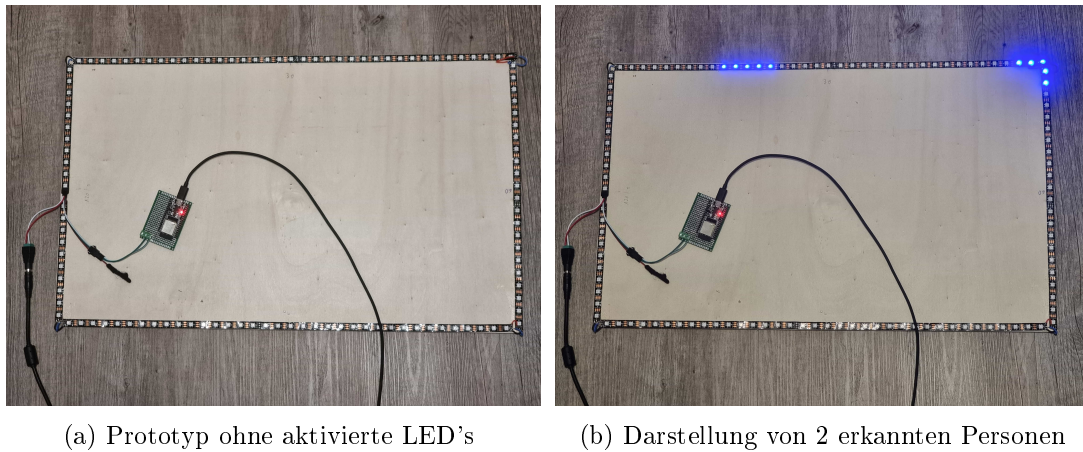


Abbildung 5.1: Exemplarische Darstellung des Prototyps.

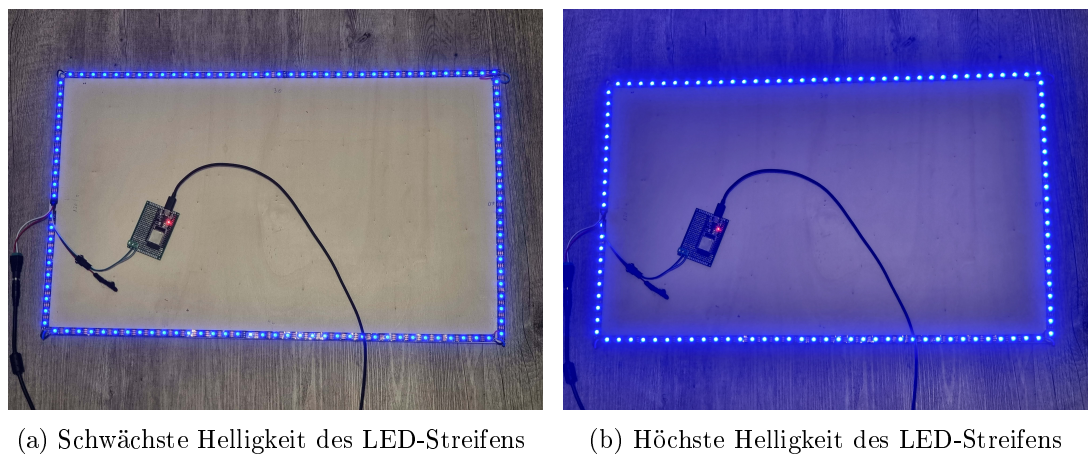


Abbildung 5.2: Verschiedene Helligkeitsstufen des LED-Streifens. Links die schwächste Helligkeit und rechts die höchste Helligkeit

Auf dem Produktivsystem funktioniert die Personenerkennung nicht. Es gibt Probleme mit der Erkennung der Punktwolke durch die Personenerkennungskomponente. Auf dieses Problem wird in Abschnitt 5.3 Verbesserungsmöglichkeiten eingegangen.

### **REQ2 Systemstatus anzeigen**

Systemstatus, die in der Konfiguration definiert wurden, werden korrekt angezeigt. Über das konfigurierte ROS Topic (topic/state), werden Nachrichten abgegriffen. Beim Eintragen einer Nachricht, die mit einem vorhandenen Systemstatus übereinstimmt, werden

die konfigurierten Farb- und Effekteinstellungen wiedergegeben. Bei nicht konfigurierten Systemstatus, wird auf den Standardstatus zurückgegriffen. In Abbildung 5.3 ist eine exemplarische Darstellung des warning-Status zu sehen.

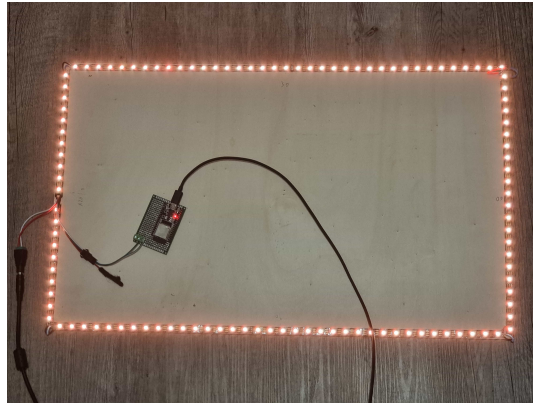


Abbildung 5.3: Exemplarische Darstellung vom warning-Status auf dem Prototypen

### REQ3 Abbiegevorgang anzeigen

Die Blinker, welche in der Konfiguration definiert wurden, werden korrekt angesteuert und angezeigt. Über das konfigurierte ROS Topic (`topic/turning_indicator`), werden Nachrichten abgegriffen. Beim Eintragen einer Nachricht, wird überprüft, ob diese `left`, `right` oder `all` entspricht. Wenn dies der Fall ist, leuchtet die korrespondierende Ecke des Prototyps. Ansonsten werden die Blinker zurückgesetzt und es leuchtet keine Ecke mehr.

## 5.2 Subjektive Evaluierung

Zur Validierung der Nützlichkeit des HMI System wurde eine Umfrage durchgeführt. Dafür wurden die gleichen Fragen in zwei verschiedenen Szenarien gestellt. In Szenario 1 wird der Roboter ohne aktivem HMI System gezeigt. In Szenario 2 wird das HMI System aktiviert und die gleichen Muster wie in Szenario 1 durchgeführt. Zur Durchführung dieser Umfrage, wurden Kurzvideos erstellt, welche Inhalte von Dritten enthalten. Das 3D-Modell des gezeigten Roboters ist von Andrey Novichkov (2022). Das 3D-Modell der gezeigten Person ist von Dmitriy Nikonov (2019). Die Kurzvideos sind im Anhang zu finden.

### 5.2.1 Aufbau des Fragebogens

Der Fragebogen ist in 3 Teile unterteilt.

Der erste und zweite Teil enthalten geschlossene Fragen. Damit sind Fragen mit festen Antwortmöglichkeiten gemeint. Somit wird ein einheitlicher Referenzrahmen zwischen allen Befragten ermöglicht.

Geschlossene Fragen sollten verwendet werden, wenn man allen Befragten bei der Beantwortung denselben Referenzrahmen vorgeben möchte. (Thielsh u. a., 2012, S. 4)

Im dritten Teil ist eine offene Frage zur persönlichen Einschätzung des HMI System's zu finden. Hier wird versucht, eine Möglichkeit der Mitteilung einer Bemerkung oder Frage zu ermöglichen. Damit wird dem Befragten ermöglicht, seine Meinung ohne Einschränkung äußern zu können.

Offene Fragen einzubauen ist von Vorteil, wenn man ablehnendes oder sensibles Verhalten erfassen möchte. (Thielsh u. a., 2012, S. 4)

Im ersten Teil werden Fragen zum Alter, Geschlecht und der beruflichen Orientierung gestellt. Es wird geschaut, ob diese Merkmale einen Einfluss auf die Beurteilung des HMI System's haben.

Im zweiten Teil werden die Personenerkennung und die Abbiegevorgang Anzeige vorgestellt.

In der Personenerkennung werden zwei gleiche Szenarien gezeigt. Ein autonomer Roboter fährt an einer stehenden Person vorbei. Im ersten Szenario fährt der Roboter ohne aktivem HMI System vorbei. Im zweiten Szenario fährt der Roboter mit aktivem HMI System vorbei. Diese sind als Kurzvideos in der Umfrage eingebaut. In Abbildung 5.4a und Abbildung 5.4b sind Ausschnitte der Kurzvideos zu sehen. In Abbildung 5.4a ist Szenario 1 mit deaktiviertem HMI System zu sehen. In Abbildung 5.4b ist Szenario 2 mit aktiviertem HMI System zu sehen.



(a) Roboter ohne aktivem HMI System



(b) Roboter mit aktivem HMI System

Abbildung 5.4: Roboter fährt an stehender Person vorbei

Hierzu werden jeweils 2 Fragen gestellt.

**F1: Hatten Sie das Gefühl, dass der Roboter die Person erkannt hat?**

- (-) Stimme voll und ganz zu
- (-) Stimme zu
- (-) Stimme weder zu noch lehne ab
- (-) Stimme nicht zu
- (-) Stimme überhaupt nicht zu

Mit Frage F1 wird versucht herauszufinden, ob der autonome Roboter im Kurzvideo davor, dem Befragten den Eindruck vermittelt hat, die Person erkannt zu haben. Hier gibt es eine Abstufung von Stimme voll und ganz zu bis Stimme überhaupt nicht zu, um konkreter zu ermitteln, wie sicher sich die Befragten sind.

**F2: Würden Sie sich in dieser Situation sicher fühlen, wenn der Roboter an Ihnen vorbeifährt?**

- (-) Stimme voll und ganz zu

- (-) Stimme zu
- (-) Stimme weder zu noch lehne ab
- (-) Stimme nicht zu
- (-) Stimme überhaupt nicht zu

Mit Frage F2 wird versucht, die persönliche Meinung über das Vertrauen gegenüber autonomen Roboter zu ermitteln. Hier wird geschaut, ob das HMI System einen Unterschied gemacht hat.

Anschließend wird ein Kurzvideo gezeigt, welches den autonomen Roboter, mit aktiven Blinkern an der rechten Vorder- und Rückseite zeigt. In Abbildung 5.5 ist ein Ausschnitt dieses Videos zu sehen. Folgende Frage wird dazu gestellt.

**F3: Was vermuten Sie, hat/hatte der Roboter im gezeigten Video vor?**

- (-) Der Roboter hat was erkannt
- (-) Der Roboter möchte abbiegen
- (-) Der Roboter ist kaputt
- (-) Ich verstehe nicht, was der Roboter mir signalisieren möchte

Mit Frage F3 wird geschaut, ob die gezeigte Animation das richtige Verhalten verständlich zeigt.



Abbildung 5.5: Ausschnitt aus dem Kurzvideo aus Frage F3, welches einen Blinker zeigen soll. Die orange markierten Bereiche blinken.

In Teil 3 wird eine offene Frage gestellt, welche lautet:

**F4: Wie bewerten Sie persönlich das visuelle Verhalten des Roboters im Allgemeinen? Fühlen Sie sich durch die Lichtsignale sicherer? Geben Sie uns gerne Feedback dazu**

Mit Frage 4 wird dem Befragten die Möglichkeit gegeben, sich offen zu äußern. Es wird versucht herauszufinden, ob die Befragten eher kritisch oder positiv gegenüber autonomen Systemen im Allgemeinen eingestellt sind.

Alle Antworten werden über alle Teilnehmer der Umfrage verglichen. Es wird überprüft, ob Alter, Geschlecht und berufliche Orientierung eine Rolle in der Beantwortung der Fragen von Teil 2 und 3 spielen. Außerdem wird geschaut, ob das HMI System verständlich mit seiner Umwelt kommuniziert und das vorhandene Akzeptanzproblem dadurch verbessert werden kann.

### 5.2.2 Auswertung

In diesem Abschnitt gehen wir auf die Auswertung der Umfrage ein.

An der Umfrage haben 52 Personen teilgenommen, wovon 44 den Fragebogen komplett ausgefüllt haben und berücksichtigt werden.

Von den 44 Personen waren 72% Männlich und 28% Weiblich.(Abbildung 5.6).

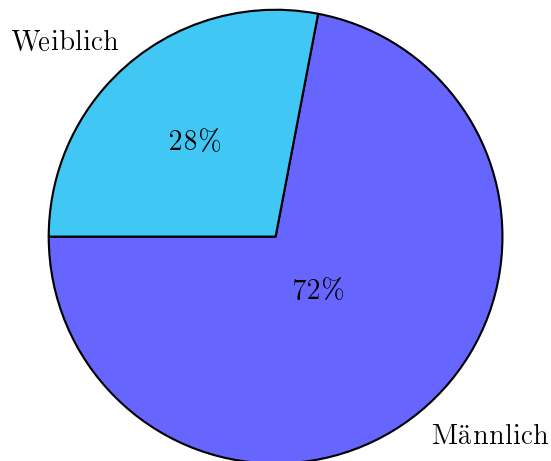


Abbildung 5.6: Verteilung des Geschlechts der Befragten

Die Befragten hatten ein Alter zwischen 19 und 55 Jahren. Das Durchschnittsalter aller Befragten lag bei 27 Jahren. In Abbildung 5.7 ist die Aufteilung der Altersgruppen der Befragten dargestellt. Zu sehen ist, dass die meisten Befragten unter 30 Jahren waren. Das Durchschnittsalter der weiblichen Befragten lag bei ca. 30 Jahren. Bei den männlichen Befragten lag das Durchschnittsalter bei ca. 26 Jahren.



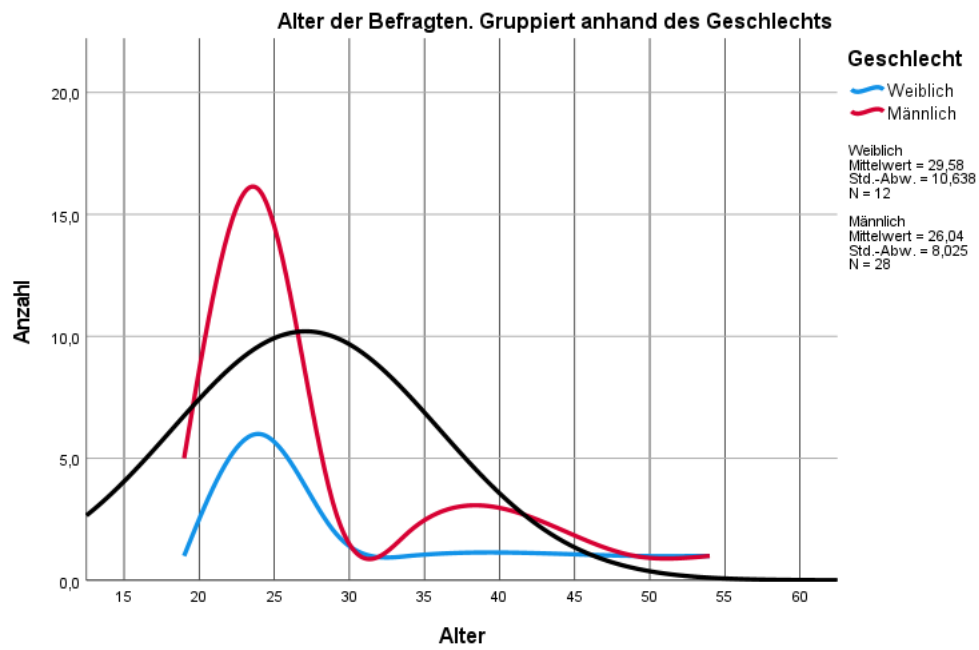


Abbildung 5.7: Alter der Befragten gruppiert anhand des Geschlechts.

56% der Befragten gaben an, dass Sie in einem Beruf arbeiten, welcher mit Informatik zu tun hat. 42% verneinten diese Frage. 2% haben keine Angaben dazu gemacht. (Abbildung 5.8).

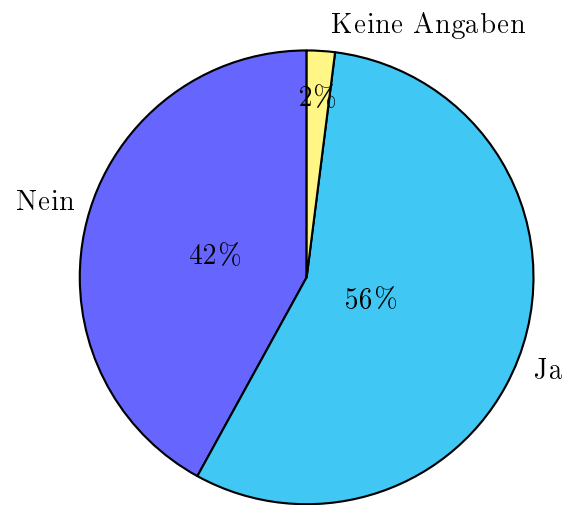


Abbildung 5.8: Angabe, ob Befragten in einem Beruf arbeiten, in dem Sie mit Informatik zu tun haben

Anschließend wurden die Kurzvideos gezeigt. In Abbildung 5.9 war das HMI System ausgeschaltet. 42% der Befragten waren sich nicht sicher, ob der Roboter die Person erkannt hat. 52% der Befragten waren sich sicher, dass der autonome Roboter die Person nicht erkannt hat. 7% waren der Meinung, dass der Roboter die Person erkannt hat.

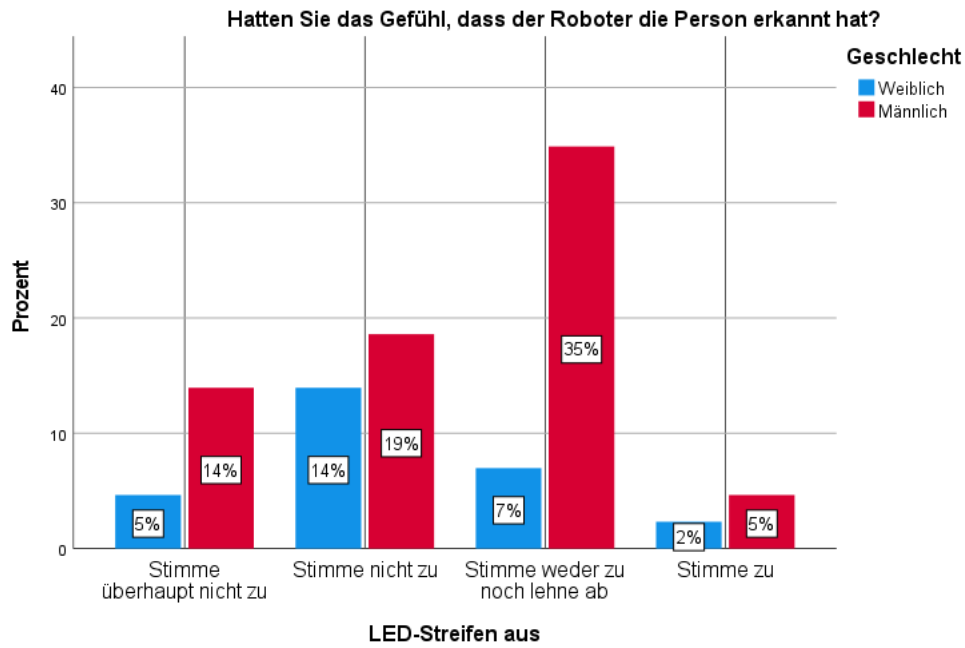


Abbildung 5.9: Gefühl, ob der Roboter die Person erkannt hat. Blau steht für Weiblich und Rot für Männlich. Das HMI System war in diesem Szenario aus

Bei Aktivierung des HMI System's zeigte sich die Effektivität des Systems. In Abbildung 5.10 zeigt sich Verteilung der Antworten. 89% aller Befragten gaben an, dass der autonome Roboter die Person erkannt hat. 11% waren sich nicht sicher oder verneinten diese Frage.

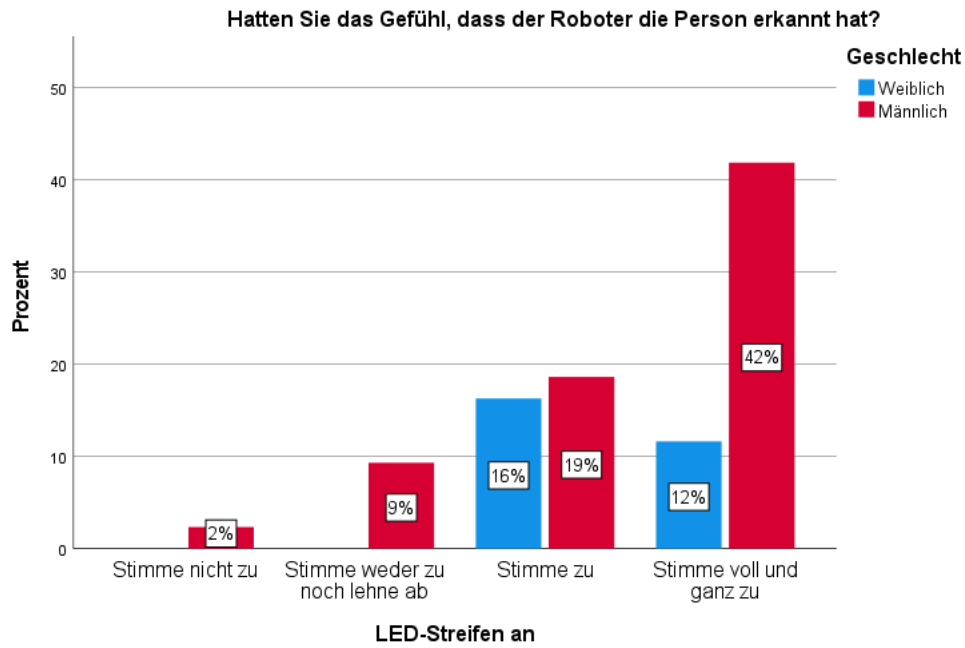


Abbildung 5.10: Gefühl, ob der Roboter die Person erkannt hat. Blau steht für Weiblich und Rot für Männlich. Das HMI System war in diesem Szenario an

Bei deaktiviertem HMI System gaben 53% der Befragten an, sich sicher zu fühlen, wenn der Roboter an Ihnen vorbeifährt. 25 % waren sich nicht sicher. 21% würden sich nicht sicher fühlen. Abbildung 5.11 zeigt dies.

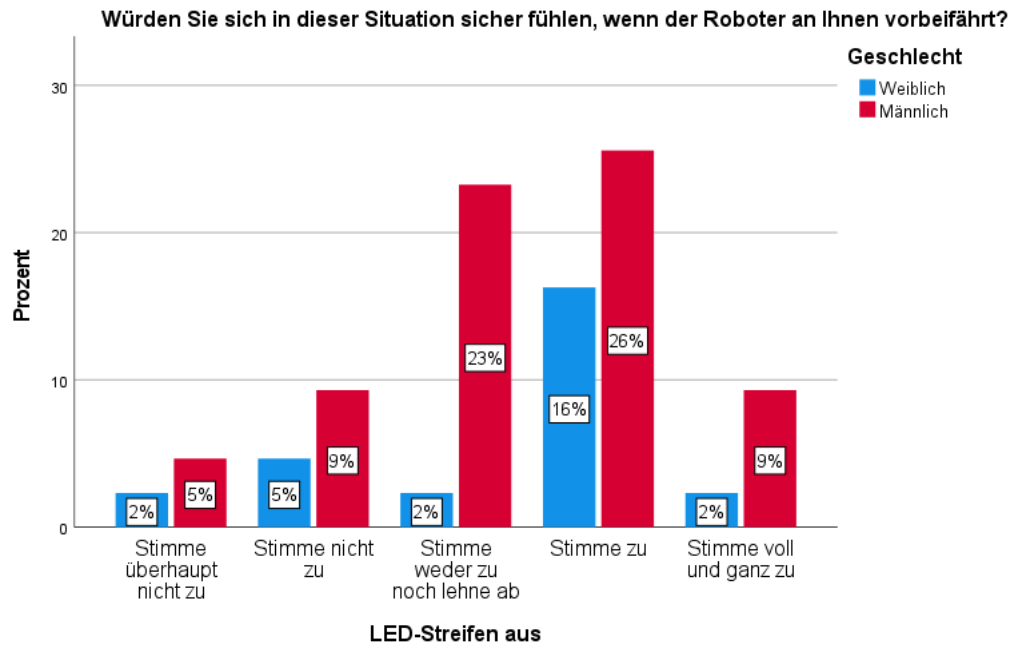


Abbildung 5.11: Sicherheitsgefühl, wenn der Roboter am Befragten vorbeifährt. Das HMI System war in diesem Szenario aus

Bei aktiviertem HMI System gaben 81% der Befragten an, sich sicher zu fühlen. 9% waren sich unsicher. 8% würden sich nicht sicher fühlen. Abbildung 5.12 zeigt dies.

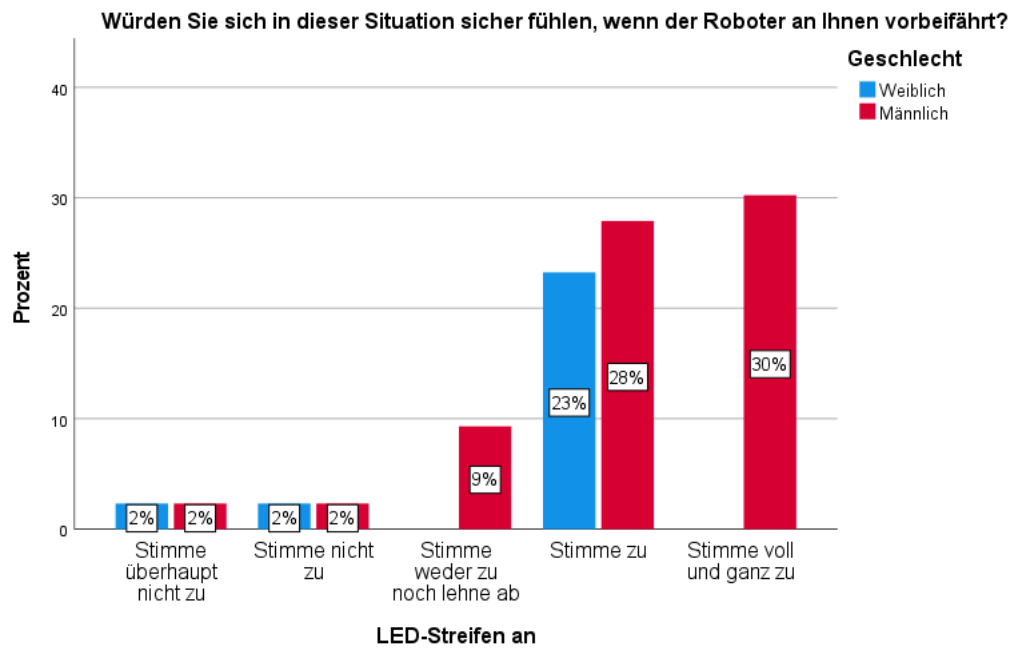


Abbildung 5.12: Sicherheitsgefühl, wenn der Roboter am Befragten vorbeifährt. Das HMI System war in diesem Szenario an

Bei Frage F3 (Abschnitt 5.2.1), welche ein Video zeigte, wo der Roboter die Blinker auf der rechten Seite blinkend anzeigte, war beabsichtigt einen Abbiegevorgang anzuzeigen. In Abbildung 5.13 ist die Auswertung dieser Frage zu sehen. 41% der Befragten haben verstanden was beabsichtigt war. 23% gaben an, dass der Roboter etwas erkannt hat. 20% sind der Meinung der Roboter sei defekt. 15% verstehen nicht, was der Roboter signalisieren möchte.

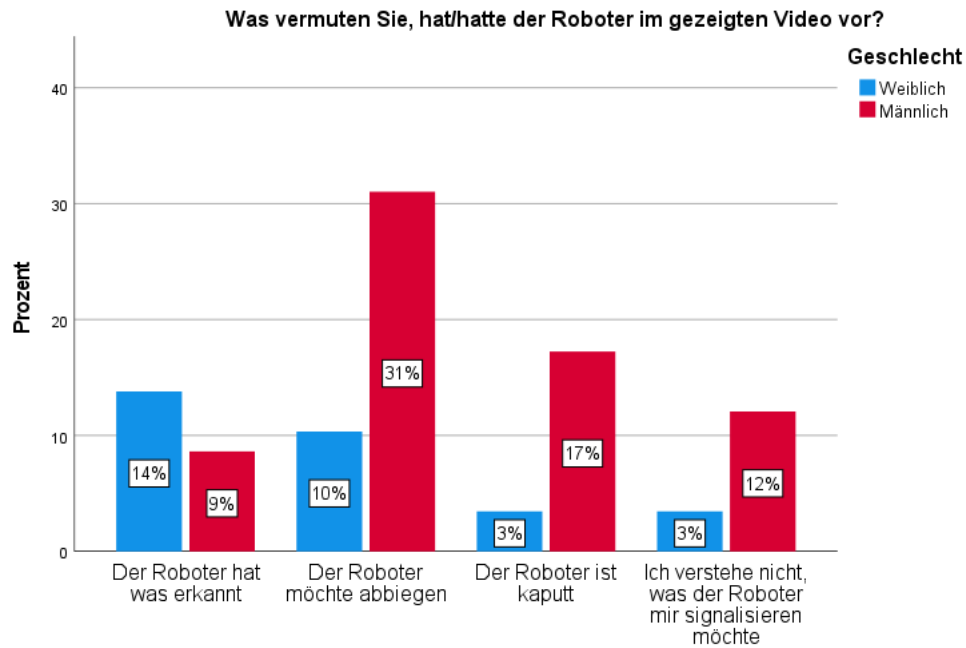


Abbildung 5.13: Antwort der Befragten beim Zeigen eines Videos, wo der Roboter einen Abbiegevorgang zeigen möchten.

Einige der Befragten haben in Frage F4, welche eine offene Frage war, ein paar interessante Anmerkungen abgegeben, die in diesem Teil anonymisiert und zusammengefasst werden. Im nachfolgendem Teil werden für die Befragten beider evaluierter Geschlechter der Begriff Proband benutzt. Dieser gilt für Männlich als auch Weiblich.

### Proband C56

C56 empfand die LED-Streifen als hilfreich, um überhaupt zu verstehen, was der Roboter vorhat. Problematisch könnte es aber werden, wenn mehrere Lichtsignale mit verschiedenen Bedeutungen auf einem Streifen angezeigt werden.

### Probanden C59, C73 und C75

Probanden C59, C73 und C75 fühlten sich durch die LED-Streifen sicherer, da der Roboter mitteilt, was er vorhat. Außerdem hatten Sie das Gefühl, dass der Roboter sie erkannt hat.

### **Proband C67**

C67 findet, dass der LED-Streifen eine vorteilhafte Verbesserung für den Roboter ist. Der Proband schlägt vor, dass der Blinker und die Personenerkennung nicht auf dem gleichen LED-Streifen dargestellt werden sollte, da dies für Verwirrung sorgt.

### **Probanden C163 und C165**

C163 nimmt die Leuchtsignale als Warnsignale wahr. C163 und C165 würden sich nicht sicher fühlen, wenn der Roboter an Ihnen vorbeifährt.

Überwiegend war eine positive Rückmeldung bei den Probanden zu sehen, was die Anzeige der Personenerkennung über den LED-Streifen anging. Der Blinker wurde eher negativ bewertet. Er wurde außerdem als verwirrend und nicht aussagekräftig bezeichnet.

In der Auswertung haben die in Teil 1 erfassten Daten, keinen aussagekräftigen Unterschied bei der Beantwortung der Fragen gemacht.

## **5.3 Verbesserungsmöglichkeiten**

Während der Erstellung dieser Arbeit sind einige Probleme aufgetaucht. Auf diese wird in diesem Abschnitt eingegangen und eventuelle Lösungsansätze werden vorgestellt.

### **5.3.1 Personenerkennung funktioniert nicht auf Produktivsystem**

Die Komponente zur Erkennung von Personen nutzt Punktwolken im Format Point-Cloud2, die durch einen LiDAR erzeugt werden.

Werden auf dem Produktivsystem Daten vom LiDAR bezogen, stürzt die Personenerkennungskomponente ab, sobald sich eine Person im Sichtbereich des LiDAR's befindet. Dies ist in der Simulation nicht der Fall.

Der Unterschied vom Produktivsystem zur Simulation ist der Hersteller des LiDAR's. Im Produktivsystem wird ein LiDAR der Firma RoboSense genutzt. In der Simulation wird ein LiDAR der Firma Velodyne simuliert. Beide geben Ihre Daten als Punktwolke im Format PointCloud2 zurück.



Zur Analyse des Problems wurde im Produktivsystem ein Rosbag aufgenommen. Ein Rosbag ist eine Aufnahme aller Daten, die in ROS verarbeitet werden. Somit können bei Problemen, die gleichen Daten zu einem späteren Zeitpunkt abgespielt werden, um gleiche Voraussetzungen schaffen zu können. Beim Abspielen des Rosbag's in der Simulationsumgebung, welches vorher im Produktivsystem aufgenommen wurde, traten keine Probleme auf und die Personenerkennungskomponente erkannte Personen in der Punktwolke.

Eine mögliche Fehlerquelle ist der Hersteller des LiDAR's, da im Versuchsaufbau der Personenerkennungskomponente ein LiDAR der Firma Velodyne genutzt wurde. Somit könnte zur Eingrenzung des Problems auf dem Produktivsystem, ein LiDAR der Firma Velodyne installiert werden.

### 5.3.2 Eventuelle Speicherprobleme durch Mikrocontroller

Für den LED-Controller wurde ein Mikrocontroller der ESP Familie genutzt.

ESP-Mikrocontroller eignen sich aufgrund Ihrer Bauform und Stromverbrauches für Projekte, wo Platz und Strom nicht unendlich vorhanden sind. Die ESP-Familie besteht wesentlich aus zwei verschiedenen Controllern

	ESP8266	ESP32
CPU-Kerne	1	2
CPU-Frequenz	160 MHz	240 MHz
SRAM	160 KB	520 KB

Tabelle 5.1: Vergleich von ESP8266 und ESP32 (Riccardi, 2021)

Wie in Tabelle 5.1 zu sehen ist, besitzt der ESP32 2 Kerne, die auf einer höheren Frequenz als auf dem ESP8266 laufen. Dies erlaubt dem ESP32 parallel zu arbeiten, sowie schneller Aufgaben zu lösen. Beide hier verglichenen ESP's verbrauchen durchschnittlich 80mA (AZ-Delivery, a,b). Zum Beginn der Bearbeitung dieser Arbeit wurde als LED-Controller der Vorgänger des ESP32 genutzt. Der ESP8266 konnte aufgrund seines geringen SRAM's in Verwendung mit WLED nur maximal 16 Segmente unterstützen. Außerdem kam es vor, dass der ESP8266 bei Verwendung von mehr als 10 Segmenten, starke Verzögerungen bei der Darstellung von aktualisierten Segmenten aufwies. Jede

Sekunde, werden die Segmente aktualisiert. Es kam vor, dass der ESP8266 3 Segmentaktualisierungen verzögert war. Auch die Effekte der einzelnen Segmente wurde teilweise nicht flüssig angezeigt. Ein Upgrade auf einen ESP32 konnte diese Probleme beheben. Der ESP32 unterstützt in Verwendung mit WLED (Version 0.13.2) 32 Segmente. Durch den 2 Kern Prozessor des ESP32, ist die nicht flüssige Darstellung von Effekten, die beim 1 Kern Prozessor des ESP8266 auftraten, behoben.

## 6 Fazit

In dieser Arbeit wurde auf Basis von ROS, eine Komponente mit Basisfunktionen für eine Mensch-Maschinen-Interaktion entwickelt. Das Ziel dieser Arbeit war es, ein System zu entwickeln, welches dem autonomen Roboter die Möglichkeit gibt, mit seiner Umwelt zu kommunizieren, um somit die Akzeptanz zu erhöhen.

Durch die Personenerkennungskomponente wurden über den LiDAR Personen erkannt. Diese wurden über das HMI System ausgelesen und an den Positionen, wo Personen stehen, leuchtete der LED-Streifen, um den nahestehenden Personen zu zeigen, dass der autonome Roboter sie erkannt hat. Andere Komponenten des autonomen Roboters können über ROS Topic's mit dem HMI System kommunizieren. So können beispielsweise an den Seiten des autonomen Roboters Blinklichter aktiviert werden, um Abbiegevorgänge mit seiner Umwelt zu kommunizieren. Außerdem können Systemzustände, welche über konfigurierte Farb- und Effekteinstellungen definiert werden, über den LED-Streifen angezeigt werden. Auch hier können andere Komponenten des autonomen Roboters mit dem HMI System über ROS Topic's diese Zustände / States aktivieren.

Die Evaluierung hat gezeigt, dass die Befragten grundlegend positiv gegenüber dem HMI System eingestellt waren. Bei deaktiviertem HMI System waren sich die Befragten unsicher, wie sie sich fühlen sollen, wenn der Roboter an Ihnen vorbeifährt (Abbildung 5.11). Als im zweiten Szenario das HMI System aktiviert wurde, stieg das Sicherheitsgefühl der Befragten an (Abbildung 5.12). Dies zeigt, dass das HMI System eine Verbesserung des Sicherheitsgefühls und somit der Akzeptanz bewerkstelligt hat. Die Darstellung der Personenerkennung über den LED-Streifen war für 89% der Befragten verständlich (Abbildung 5.10). Die Darstellung des Blinkers, über den gleichen LED-Streifen, empfanden einige Befragte als verwirrend und nicht aussagekräftig. Es wurde vorgeschlagen, den Blinker nicht auf dem LED-Streifen, wo die Personenerkennung angezeigt wird, darzustellen.

## 6.1 Ausblick

Die Evaluierung hat gezeigt, dass das HMI System eine Verbesserung der Akzeptanz mit Menschen hervorruft. Das in dieser Arbeit entwickelte System kann durch eine auditive Schnittstelle erweitert werden. Außerdem kann eine LED-Controller-Software entwickelt werden, welche genau auf die Zwecke des HMI System zugeschnitten ist, wobei man aber hier erwähnen muss, dass WLED bereits eine befriedigende Performance abliefert. In Burns u. a. (2019) wurde ein Tageslichtprojektor an der Vorderseite des autonomen Roboters befestigt, welcher durch visuelle Ablichtung auf den Boden vor dem Roboter seine Intentionen kommunizieren konnte. Dies ist ein sehr interessanter Ansatz, da dies die Menge der Darstellungsmöglichkeiten deutlich erweitert. In Rouchitsas und Alm (2019) wurden verschiedene Konzepte zur Umsetzung einer Mensch-Maschinen-Interaktion vorgestellt, die zur Erweiterung und Verbesserung des in dieser Arbeit entwickeltem System genutzt werden können.

# Literaturverzeichnis

- [Andrey Novichkov 2022] ANDREY NOVICHKOV: *Delivery Rover - Download Free 3D model by Andrey Novichkov (@hophobb)*. 2022. – URL <https://sketchfab.com/3d-models/delivery-rover-519c0d935e214921aeaf5b66fd9cfd6a>. – Zugriffsdatum: 21.08.2022
- [AZ-Delivery a] AZ-DELIVERY: ESP-32 DevKit C V4 Datenblatt. . – URL [https://cdn.shopify.com/s/files/1/1509/1638/files/ESP-32\\_DevKit\\_C\\_V4\\_Datenblatt\\_AZ-Delivery\\_Vertriebs\\_GmbH\\_24ec770f-c65e-4bd3-92c9-cd64b4d070b8.pdf?v=1615364587](https://cdn.shopify.com/s/files/1/1509/1638/files/ESP-32_DevKit_C_V4_Datenblatt_AZ-Delivery_Vertriebs_GmbH_24ec770f-c65e-4bd3-92c9-cd64b4d070b8.pdf?v=1615364587). – Zugriffsdatum: 22.08.2022
- [AZ-Delivery b] AZ-DELIVERY: NodeMCU Lua Lolin V3 Modul mit ESP8266 12E Datenblatt. . – URL [https://cdn.shopify.com/s/files/1/1509/1638/files/NodeMCU\\_Lua\\_Lolin\\_V3\\_Modul\\_mit\\_ESP8266\\_12E\\_Datenblatt.pdf?342081239282763366](https://cdn.shopify.com/s/files/1/1509/1638/files/NodeMCU_Lua_Lolin_V3_Modul_mit_ESP8266_12E_Datenblatt.pdf?342081239282763366). – Zugriffsdatum: 22.08.2022
- [Bartneck u. a. 2020] BARTNECK, Christoph (Hrsg.) ; BELPAEME, Tony (Hrsg.) ; EYSSEL, Friederike (Hrsg.) ; KANDA, Takayuki (Hrsg.) ; KEIJERS, Merel (Hrsg.) ; ŠABANOVIĆ, Selma (Hrsg.): *Mensch-Roboter-Interaktion*. München : Carl Hanser Verlag GmbH & Co. KG, 2020. – ISBN 978-3-446-46412-4
- [Burns u. a. 2019] BURNS, Christopher G. ; OLIVEIRA, Luis ; THOMAS, Peter ; IYER, Sumeet ; BIRRELL, Stewart: Pedestrian Decision-Making Responses to External Human-Machine Interface Designs for Autonomous Vehicles. In: *2019 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2019, S. 70–75. – ISBN 978-1-7281-0560-4
- [Chang u. a. 2018] CHANG, Chia-Ming (Hrsg.) ; TODA, Koki (Hrsg.) ; IGARASHI, Takeo (Hrsg.) ; MIYATA, Masahiro (Hrsg.) ; KOBAYASHI, Yasuhiro (Hrsg.): *A Video-based Study Comparing Communication Modalities between an Autonomous Car and a Pedestrian*. 2018

- [de Clercq u. a. 2019] CLERCQ, Koen de ; DIETRICH, Andre ; NÚÑEZ VELASCO, Juan P. ; WINTER, Joost de ; HAPPEE, Riender: External Human-Machine Interfaces on Automated Vehicles: Effects on Pedestrian Crossing Decisions. In: *Human Factors: The Journal of the Human Factors and Ergonomics Society* 61 (2019), Nr. 8, S. 1353–1370. – ISSN 1547-8181
- [DHBW Ravensburg ] DHBW RAVENSBURG ; DHBW RAVENSBURG (Hrsg.): *Welche Akzeptanzprobleme sehen Sie beim Autonomen Fahren?*. – URL <https://de.statista.com/statistik/daten/studie/270612/umfrage/nachteile-von-autonomen-fahrzeugen/>. – Zugriffsdatum: 21.08.2022
- [Dmitriy Nikonov 2019] DMITRIY NIKONOV: *Low Poly Person - Download Free 3D model by Dmitriy Nikonov (@dimanikonov)*. 2019. – URL <https://sketchfab.com/3d-models/low-poly-person-bfe451f06bba4a6baa4bae9f4b0b112e>. – Zugriffsdatum: 21.08.2022
- [Gesellschaft für Informatik e.V. 2022] GESELLSCHAFT FÜR INFORMATIK E.V.: *Was ist Mensch-Computer-Interaktion?* 2022. – URL <https://fb-mci.gi.de/fachbereich/was-ist-mci>. – Zugriffsdatum: 26.08.2022
- [Haselich u. a. 2014] HASELICH, Marcel ; JOBGEN, Benedikt ; WOJKE, Nicolai ; HEDRICH, Jens ; PAULUS, Dietrich: Confidence-based pedestrian tracking in unstructured environments using 3D laser distance measurements. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2014, S. 4118–4123. – ISBN 978-1-4799-6934-0
- [Innofact ] INNOFACT ; AUTOScout24 (Hrsg.): *Was halten Sie von autonom fahrenden Fahrzeuge auf deutschen Straßen?*. – URL <https://de.statista.com/statistik/daten/studie/1232664/umfrage/umfrage-zu-autonom-fahrenden-fahrzeugen-in-deutschland/>. – Zugriffsdatum: 21.08.2022
- [Kidono u. a. 2011] KIDONO, Kiyosumi ; MIYASAKA, Takeo ; WATANABE, Akihiro ; NAITO, Takashi ; MIURA, Jun: Pedestrian recognition using high-definition LIDAR. In: *2011 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2011, S. 405–410. – ISBN 978-1-4577-0890-9
- [Koide u. a. 2019] KOIDE, Kenji ; MIURA, Jun ; MENEGATTI, Emanuele: A portable three-dimensional LIDAR-based system for long-term and wide-area people behavior measurement. In: *International Journal of Advanced Robotic Systems* 16 (2019), Nr. 2. – ISSN 1729-8814

- [Miller 2016] MILLER, K.: *Find the bearing angle between two points in a 2D space*. 2016. – URL <https://math.stackexchange.com/q/1596518>. – Zugriffsdatum: 24.08.2022
- [Petzoldt u. a. 2018] PETZOLDT, Tibor ; SCHLEINITZ, Katja ; BANSE, Rainer: Potential safety effects of a frontal brake light for motor vehicles. In: *IET Intelligent Transport Systems* 12 (2018), Nr. 6, S. 449–453. – ISSN 1751-9578
- [Project Tempos 2022] PROJECT TEMPOS: *Powering Any WS2812b Setup — Project Tempos*. 2022. – URL <https://www.temposlighting.com/guides/power-any-ws2812b-setup>. – Zugriffsdatum: 28.07.2022
- [Riccardi 2021] RICCARDI, Fabiano ; GITHUB GIST (Hrsg.): *Comparison table for ESP8266/ESP32/ESP32-S2/ESP32-S3/ESP32-C3/ESP32-C6*. 2021. – URL <https://gist.github.com/fabianoriccardi/cbb474c94a8659209e61e3194b20eb61>. – Zugriffsdatum: 24.08.2022
- [ROS 2022] ROS: *rviz/DisplayTypes/Marker - ROS Wiki*. 2022. – URL <http://wiki.ros.org/rviz/DisplayTypes/Marker>. – Zugriffsdatum: 16.07.2022
- [Rouchitsas und Alm 2019] ROUCHITSAS, Alexandros ; ALM, Håkan: External Human-Machine Interfaces for Autonomous Vehicle-to-Pedestrian Communication: A Review of Empirical Work. In: *Frontiers in Psychology* 10 (2019), S. 2757
- [Thielsh u. a. 2012] THIELSH, Meinald T. ; LENZNER, Timo ; MELLES, Torsten: *Wie gestalte ich gute Items und Interviewfragen?* (2012). – URL [http://www.thielsch.org/download/wirtschaftspsychologie/Thielsch\\_2012b.pdf](http://www.thielsch.org/download/wirtschaftspsychologie/Thielsch_2012b.pdf). – Zugriffsdatum: 21.08.2022
- [World-Semi ] WORLD-SEMI: *WS2812 Datasheet*. – URL <https://cdn-shop.adafruit.com/datasheets/WS2812B.pdf>. – Zugriffsdatum: 23.08.2022

# A Anhang

## A.1 Fragebogen



### 1. Herzlich Willkommen

Im Rahmen meiner Bachelorarbeit über Mensch Maschinen Interaktion möchte ich gerne herausfinden, wie visuelle Signale die Kommunikationsmöglichkeiten von autonomen Robotern erweitern können. Dafür werden in diesem Fragebogen eine Reihe von Kurzvideos gezeigt und danach jeweils 1–2 Fragen gestellt.

Die Bearbeitungsdauer beträgt etwa 5 bis 10 Minuten. Für das Gelingen dieser Umfrage ist es wichtig, dass Sie alle Fragen vollständig ausfüllen.

Die Daten können Ihrer Person nicht zugeordnet werden, sind vollständig anonym und werden nur im Rahmen meiner Arbeit verwendet. Sollten in Freitextfelder Daten eingetragen werden, die eventuell Ihrer Person zugeordnet werden könnten, werden diese anonymisiert.

Für Ihre Teilnahme bedanke ich mich herzlich.

Awab Abdelkarim

---

### 2. Wie alt sind Sie?

Ich bin ... Jahre alt.

Alter	<input type="text"/>	<input type="checkbox"/> Möchte ich nicht sagen
-------	----------------------	---

---

### 3. Welches Geschlecht haben Sie?

- Weiblich
  - Männlich
  - Divers
- 

Möchte ich nicht sagen

---

**4. Arbeiten Sie in einem Beruf, indem Sie mit Informatik zu tun haben?**

- Ja  
 Nein

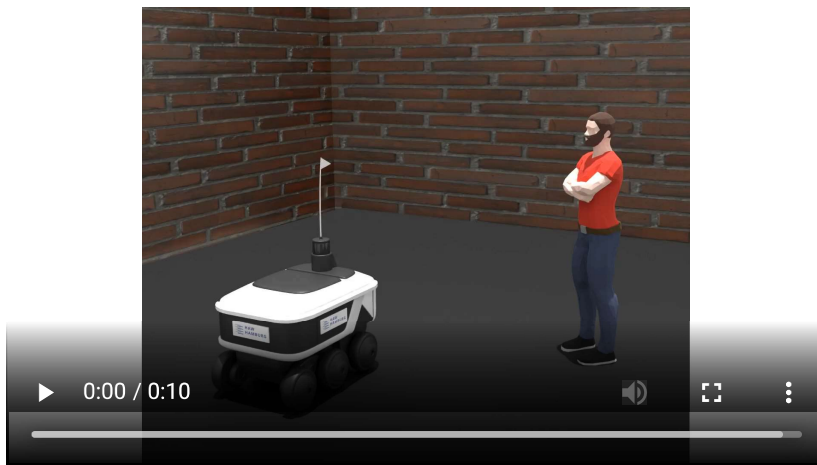
- Möchte ich nicht sagen

## Videovorschau

Ihnen werden jetzt einige Videos gezeigt, welche maximal 15 Sekunden gehen. Nach jedem Video werden Ihnen 1-2 Fragen gestellt.

### 5. Vorschau


Bitte schauen Sie sich das 10-sekündige Video an, und klicken Sie auf Weiter.



Videoquellen:

- <https://sketchfab.com/3d-models/low-poly-person-bfe451f06bba4a6baa4bae9f4b0b112e>
  - <https://sketchfab.com/3d-models/delivery-rover-519c0d935e214921aeaf5b66fd9cfd6a> (added university Logo and added LED Lights)
-

**6. Hatten Sie das Gefühl, dass der Roboter die Person erkannt hat?**



1  Stimme überhaupt nicht zu

2  Stimme nicht zu


3  Stimme weder zu noch lehne ab

4  Stimme zu

5  Stimme voll und ganz zu

**7. Würden Sie sich in dieser Situation sicher fühlen, wenn der Roboter an Ihnen vorbeifährt?**

In dieser Frage geht es explizit um Ihre Gefühle in dieser Situation.



1  Stimme überhaupt nicht zu

2  Stimme nicht zu

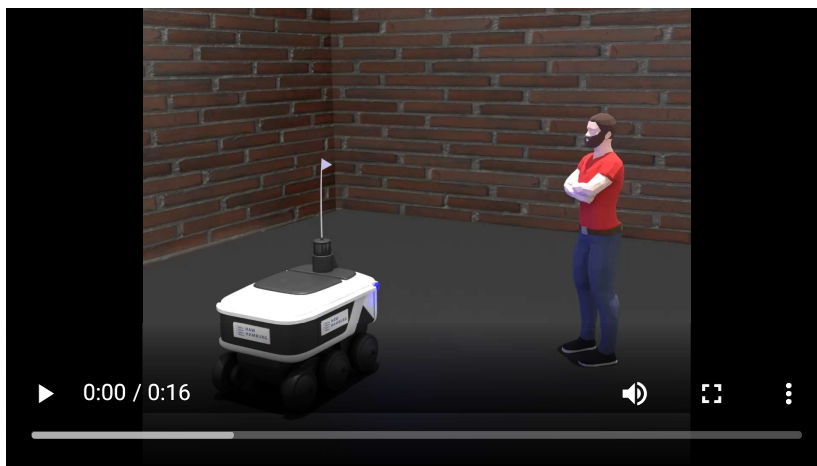
3  Stimme weder zu noch lehne ab

4  Stimme zu

5  Stimme voll und ganz zu

**8. Vorschau**

Bitte schauen Sie sich das 15-sekündige Video an, und klicken Sie auf Weiter.




Videoquellen:

- <https://sketchfab.com/3d-models/low-poly-person-bfe451f06bba4a6baa4bae9f4b0b112e>

- <https://sketchfab.com/3d-models/delivery-rover-519c0d935e214921aeaf5b66fd9cfd6a> (added university Logo and added LED Lights)

---

**9. Hatten Sie das Gefühl, dass der Roboter die Person erkannt hat?**



1

2

3

4

5

Stimme überhaupt nicht zu

Stimme nicht zu


Stimme weder zu noch lehne ab

Stimme zu

Stimme voll und ganz zu

**10. Würden Sie sich in dieser Situation sicher fühlen, wenn der Roboter an Ihnen vorbeifährt?**

In dieser Frage geht es explizit um Ihre Gefühle in dieser Situation.



1

2

3

4

5

Stimme überhaupt nicht zu

Stimme nicht zu

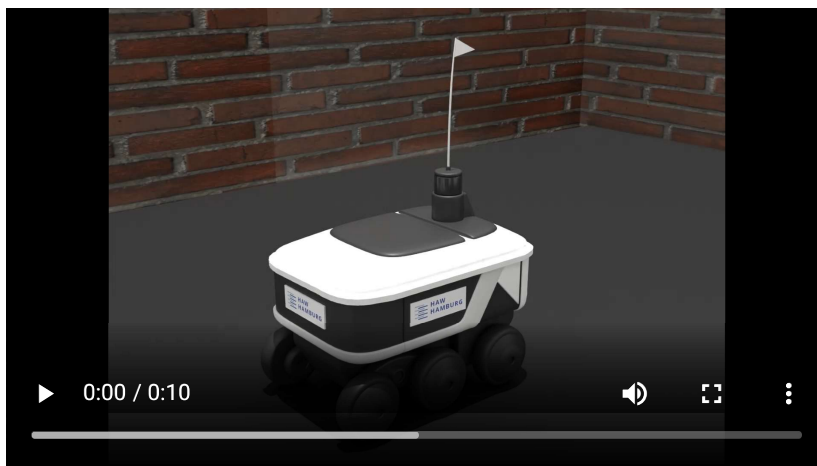
Stimme weder zu noch lehne ab

Stimme zu

Stimme voll und ganz zu

**11. Vorschau**

Bitte schauen Sie sich das 10-sekündige Video an, und klicken Sie auf Weiter.



Videoquellen:

- <https://sketchfab.com/3d-models/low-poly-person-bfe451f06bba4a6baa4bae9f4b0b112e>

- <https://sketchfab.com/3d-models/delivery-rover-519c0d935e214921aeaf5b66fd9cfd6a> (added university Logo and added LED Lights)

---

**12. Was vermuten Sie, hat/hatte der Roboter im gezeigten Video vor?**

Sie können auch mehrere Antworten auswählen

- Der Roboter hat was erkannt
- Der Roboter möchte abbiegen
- Der Roboter ist kaputt
- Ich verstehe nicht, was der Roboter mir signalisieren möchte

**13. Wie bewerten Sie persönlich das visuelle Verhalten des Roboters im Allgemeinen? Fühlen Sie sich durch die Lichtsignale sicherer? Geben Sie uns gerne Feedback dazu**

Optional

**Vielen Dank für Ihre Teilnahme!**

Wir möchten uns ganz herzlich für Ihre Mithilfe bedanken.

Ihre Antworten wurden gespeichert, Sie können das Browser-Fenster nun schließen.

## A.2 Videos aus dem Fragebogen

Die im Fragebogen gezeigten Videos sind auf der mitgegebenen CD zu finden.

HMI-Aus.mp4 zeigt den Roboter ohne aktivem HMI System an.

HMI-An.mp4 zeigt den Roboter mit aktivem HMI System an.

Blinker.mp4 zeigt den Blinker, der einen Abbiegevorgang anzeigen soll, an.

## A.3 OpenAPI 3 Schema

Die OpenAPI 3 Schema Datei für die API befindet sich auf der mitgegebenen CD. Der Dateiname dieses Schemas lautet `openapi3.json`. In `json_schema.json` befindet sich die ins JSON Schema konvertierte OpenAPI3 Schema.

# Glossar

**API** Eine API, Application Programming Interface, ist eine Schnittstelle die anderen Programmen die Möglichkeit erlaubt mit dieser Software zu kommunizieren.

**Gazebo** Gazebo ist eine 3D Simulationsumgebung, welche in Kombination mit ROS zum Testen von verschiedener Robotikanwendungen genutzt wird.

**HMI System** Software, welche in dieser Arbeit entwickelt wurde.

**LiDAR** Ein LiDAR, Light Detection and Ranging, ist ein Sensor welcher durch Laserstrahlen eine dreidimensionale Punktwolke seiner Umgebung erstellt.

**ROS** ROS, Robot Operating System, ist eine Sammlung von Software Bibliotheken und Tools, welche die Entwicklung von Roboteranwendungen vereinfachen und unterstützen.

**SRAM** SRAM, Static random-access memory, gehört wie der DRAM (Dynamic random-access memory), den wir aus Computern kennen, zu den flüchtigen Speichertypen, welche zur Haltung von gespeicherten Daten eine Spannungsversorgung benötigen. Der DRAM muss regelmäßig aktualisiert werden, um die gespeicherten Daten nicht zu verlieren. Dies ist beim SRAM nicht nötig.<sup>1</sup>

---

<sup>1</sup><https://www.techtarget.com/whatis/definition/SRAM-static-random-access-memory>

## **Erklärung zur selbstständigen Bearbeitung**

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

---

Ort

Datum

Unterschrift im Original