

Zwischenvortrag: ROS 2 Projekt “waymo”

Parameter	Kursinformationen
Veranstaltung:	Robotik Projekt
Semester	Sommersemester 2025
Hochschule:	Technische Universität Berkakademie Freiberg
Inhalte:	Exposé Vortrag
Link auf GitHub:	https://github.com/Bigfire3/waymo/blob/documentation/presentation/zwischenvortrag.md
Autoren	Fabian Zänker, Lucas Adler, Simon Hörtzsch

- Gruppenmitglieder: Fabian Zänker, Lucas Adler, Simon Hörtzsch
- Studiengang: Robotik | Mathematik in Wirtschaft, Engineering und Informatik | Angewandte Informatik
- Betreuer: Prof. Dr. Sebastian Zug, Gero Licht
- Datum: 28.05.2025

1. Projektstand

! Meilensteine

Zeitstrahl

Titel	Status	Type	Zieldatum	Tage	Material
Abgabe Aufgabe 1	Completed	Hausaufgabe	16. April 2025 16:15	Vor 34 Tagen	
Exposé Vortrag	Completed	Vortrag	23. April 2025 16:15	Vor 27 Tagen	
Abgabe Aufgabe 2	Completed	Hausaufgabe	30. April 2025 16:15	Vor 20 Tagen	
Abgabe Aufgabe 3	Completed	Hausaufgabe	7. Mai 2025 16:15	Vor 13 Tagen	
Abgabe Aufgabe 4	Completed	Hausaufgabe	21. Mai 2025 16:15	Heute!	
Zwischenvortrag	On Track	Vortrag	28. Mai 2025 16:15	Noch 7 Tage	
Abgabe Aufgabe 5	On Track	Hausaufgabe	4. Juni 2025 16:15	Noch 14 Tage	
Abgabe Aufgabe 6	On Track	Hausaufgabe	18. Juni 2025 16:15	Noch 28 Tage	
Abgabe Große Aufgabe	Not started	Hausaufgabe	2. Juli 2025 16:15	Noch 42 Tage	
waymo	On Track	Projekt	9. Juli 2025	Noch 49 Tage	Aufgaben.pdf Übungen-OPAL

Übersicht über Aufgaben und Fristen zum Robotik Projekt in Notion-Datenbank

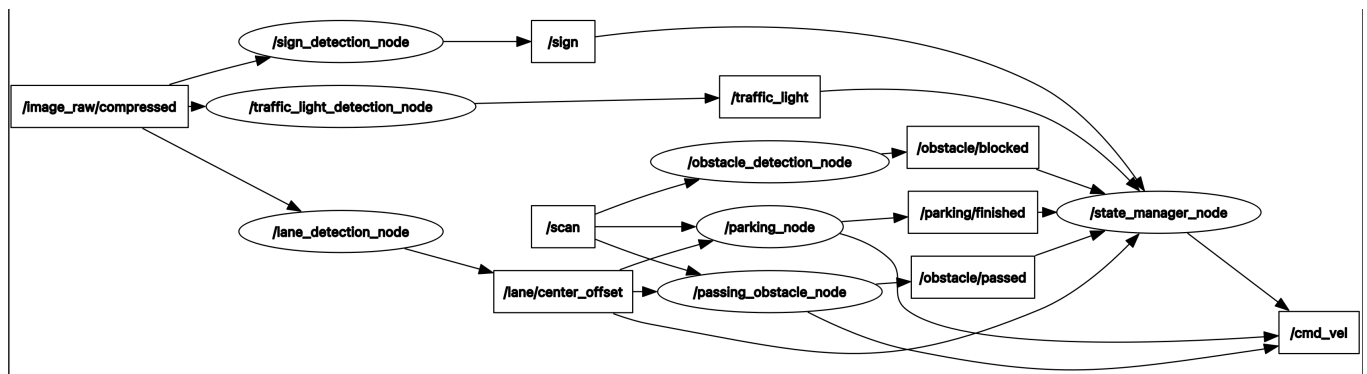
2. Systemarchitektur

Überlegungen:

- Struktur des Projektes als einzelnes Package
- Erweiterung durch hinzufügen von Nodes mit neuem Funktionsumfang
- Planung von Publisher und Subscriber, um gewollte Funktionalität bestmöglich umzusetzen

Topics und Nodes:

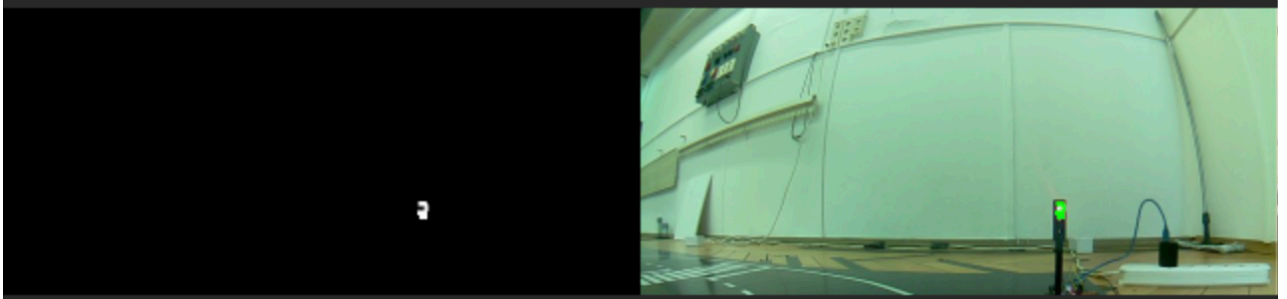
- Visualisierung der Node-Interaktionen (Nodes und Topics):



Übersicht der ROS2-Nodes und Datenflüsse, erstellt mit rqt_graph

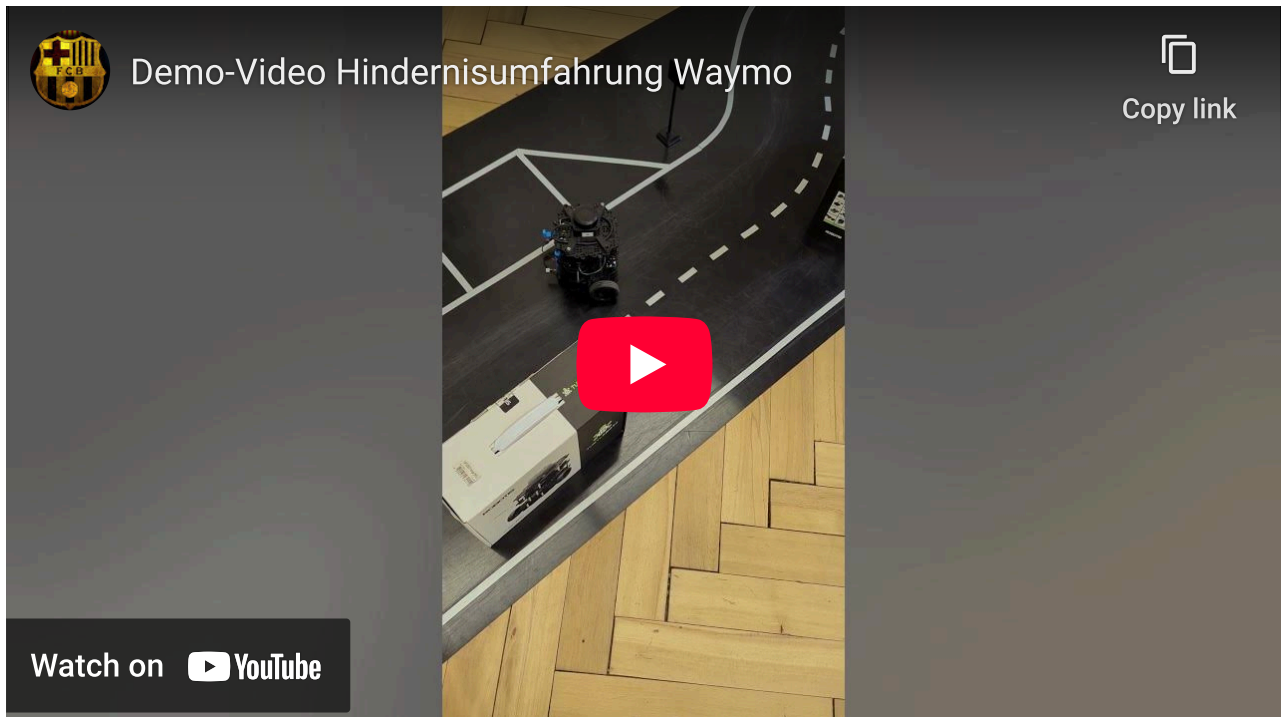
3. Kernfunktionen und Demonstration

- Ampelerkennung:



Debug Bilder zur Ampelerkennung

- Hindernisumfahrung:

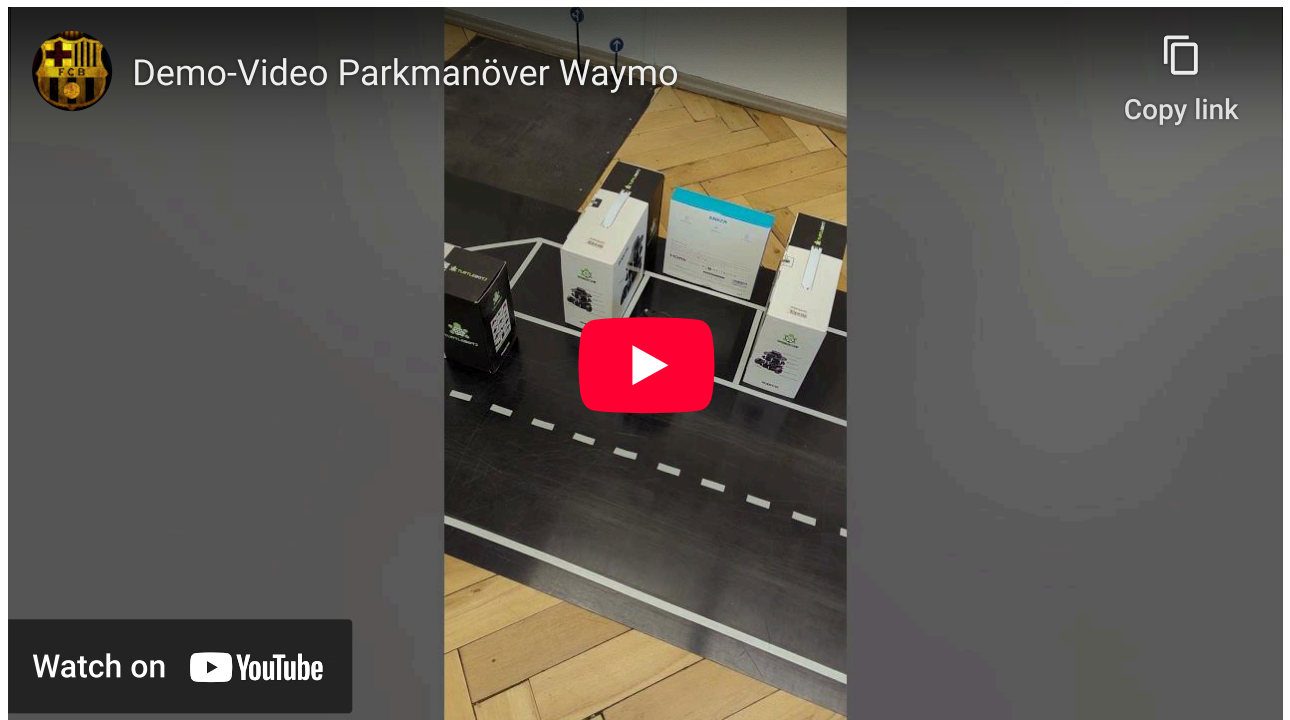


- Park-Schild-Erkennung:

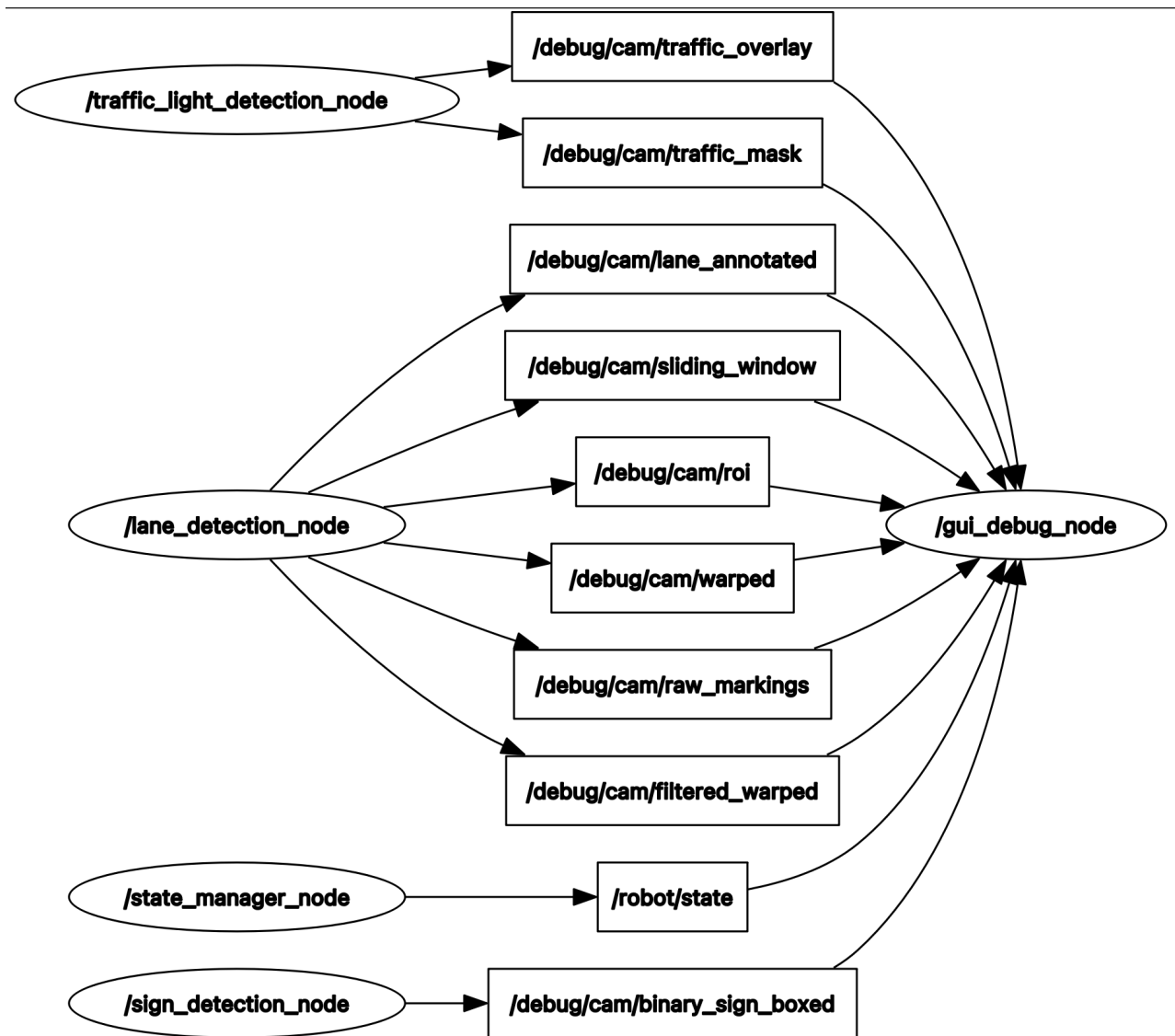


Debug Bild zur Park-Schild-Erkennung

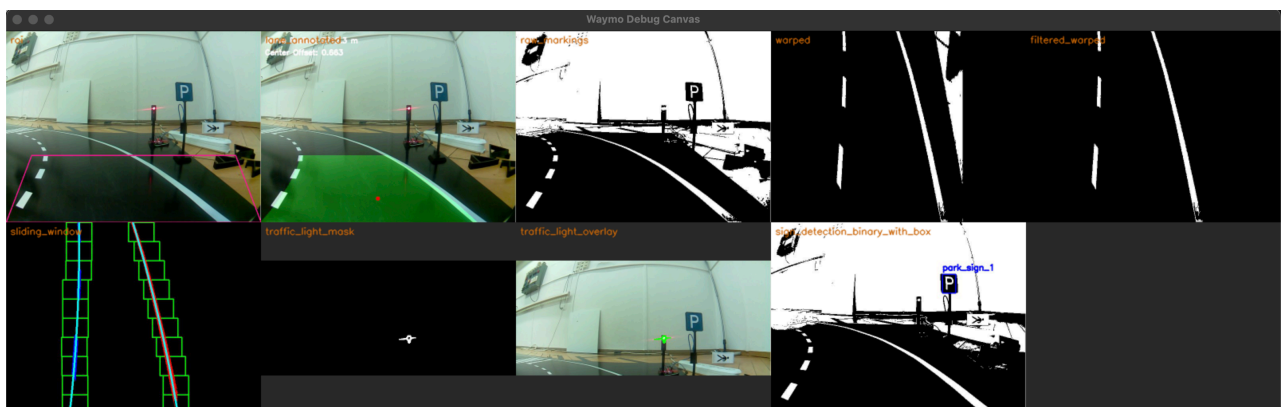
- Park-Manöver:



- GUI Debug Node und Keyboard Handler:



Übersicht der Topics und Nodes, die für das Debug-Canvas genutzt werden, erstellt mit rqt_graph



Aktuelles Debug Canvas

- Keyboard Handler: Möglichkeit zum manuellen Stoppen des Roboters
- in separatem Terminal mit den Tasten 's' für Stop/Start und 'd' zum Togglen des Debug Canvas

4. Ausblick

- Spiegelung und Reflexion auf der Fahrbahn müssen noch besser behandelt werden
- Erweiterung um Kreuzungsmanöver

Vielen Dank für Ihre Aufmerksamkeit!

Fragen?