



Roboter Orchestrierungssoftware

Exposé für die Lehrveranstaltung Systemadministration

Stefan Geiring, 30658

Marvin Müller, 32850

Nicolas Baumgärtner, 32849

Wintersemester 22/23

05.November 2022

Betreuer: M.Sc. Aykan D. Inan
Ravensburg-Weingarten University of Applied Sciences

Inhaltsverzeichnis

1 Thema	2
2 Motivation	2
3 Ziel	2
4 Eigene Leistung	2
5 Aufbau der Arbeit	3
6 Grundbegriffe	3
7 Zielsetzung und Anforderungen	3
8 Stand der Technik und Forschung	3
9 Lösungsideen	3
10 Evaluation der Lösungsideen anhand der Anforderungen	4
10.1 Wahl des Frontend Frameworks	4
10.2 Warum ROS?	4
10.3 Micro-ROS und RTOS	4
10.4 Roboter Platform Prototyp und Designentscheidungen	4
11 Implementierung	5
11.1 React Front-End	5
11.1.1 Aufbau des Frontends (Architektur)	5
11.2 ROS Back-End	5
11.2.1 Firmware Kompilierungs Toolchain	5
11.3 Roboter Platform	5
11.3.1 Elektronik	5
11.3.2 3D-Druckteile	5
12 Evaluation der Implementierung	5
12.1 Fehler und Verbesserungen Frontend	5
12.2 Fehler und Verbesserungen Backend	5
12.3 Fehler und Verbesserungen Roboter Platform	5
13 Fazit und Ausblick	5

1 Thema

Entwicklung einer Orchestrierungssoftware mit Weboberfläche auf Basis von ROS.

Roboter werden immer häufiger eingesetzt egal ob in Spezial Industrial bereichen oder auch einfach zuhause. Die einfache und übersichtliche Orchestrierung vieler Roboter ist deshalb besonders wichtig. Roboter Schwärme spielen außerdem, in der heutigen Zeit immer öfters eine wichtige Rolle. Ein Beispiel für große Roboter Schwärme die bereits eingesetzt werden sind Lieferdrohen, Drohnenshows als ersatz für Feuerwerk oder die simplere Variante davon, eine Lagerverwaltung.

2 Motivation

Wir interessieren uns alle für die Welt der Robotik. Das Thema bietet unglaublich viel Lernpotential.

Desweiteren ist das Thema fächerübergreifend, wir arbeiten in insgesamt 4 Sektoren: der reinen Softwareentwicklung, Webentwicklung sowie der Hardware/Elektronik Entwicklung und natürlich der Robotik Entwicklung/Forschung.

3 Ziel

Ziel des Projektes soll es sein eine Lauffähige Orchestrierungssoftware für Roboter auf ROS Basis zu entwickeln. Mit deren Hilfe man mehrere Roboter überwachen und auch steuern kann.

Desweiteren eine selbst designde Hardware Plattform für unsere kleinen Beispiel Roboter zu designen, welche im nachhinein durch diverse Komponenten erweitert werden können. Wie zum Beispiel Kamera, Bumber, Laser etc.

4 Eigene Leistung

Entwicklung einer einfachen Weboberfläche auf ROS Basis welche als Orchestrierungssoftware für Roboter dienen soll. Die Weboberfläche der Orchestrierungssoftware soll in erster Linie als einfache Kontroll und Debugging Schnittstelle dienen.

Um unsere Orchestrierungssoftware sinngemäß demonstrieren zu können sollen außerdem kleine Roboter auf ESP32 Basis gebaut werden. Diese Roboter sollen aus einem 3D-Gedruckten Gehäuse bestehen. Auf den ESP32 soll Micro-ROS auf Basis von FreeRTOS ausgeführt werden.

Unsere kleinen Beispiel Roboter sollen außerdem im Idealfall mit modular austauschbaren Erweiterungen ausstattbar sein. Diese Erweiterungen sollen Simple Sensorik und Aktorik zur Verfügung stellen.

5 Aufbau der Arbeit

- ROS in Docker Container lauffähig bekommen.
- Micro-ROS auf ESP32 oder anderer Hardware lauffähig bekommen. Hardware ist noch nicht final definiert, deshalb muss erst noch evaluiert werden ob der ESP32 unseren Anforderungen und Wünschen gerecht wird.
- Frontend für ROS auf Web-Basis programmieren.
- Evaluation des Frameworks für Web-Frontend (evtl. Software zwischenlayer nötig).
- Design des Roboter Gehäuses entwerfen, welches 3D-Gedruckt wird.
- Antriebsstrang des Roboters entwerfen.
- Evaluation und Testing welcher Antrieb unseren Anforderungen entspricht.

6 Grundbegriffe

7 Zielsetzung und Anforderungen

Genauere Erläuterung vom Ziel und den Anforderungen.

8 Stand der Technik und Forschung

Technologischer Standpunkt Software:

ROS bietet bereits ein starkes und recht einfach nutzbares Framework, allerdings vermisst man eine schöne und einfach bedienbare "graphische Oberfläche. Es existieren kleinere Projekte welche sich dieser Frontend entwicklung annehmen. Allerdings hat uns keines dieser Projekte zufrieden gestellt. Vor allem was die Orchestrierungsmöglichkeit mehrerer Roboter angeht.

Technologischer Standpunkt Hardware:

Es existieren bereits viele Forschungen und Beispiele für diverse Roboter und deren verschiedensten Antriebskinematiken. Für unser Projekt werden wir allerdings eine eigene Roboter Plattform designen und 3D-Drucken. Viele der bereits vorhandenen Roboter sind entweder zu Groß und teuer oder viel zu klein und deshalb ebenfalls nicht gut für eine demonstration geeignet. Unter anderem wollen wir das unser Roboter den Vorteil bietet weitestgehend 3D-gedruckt zu sein. Die ETH Zürich hat bereits eine kleine Zusammenfassung über die wichtigsten Antriebarten mobiler Roboter zur verfügung gestellt. Für unser Projekt haben wir uns vorerst für einen simplen Diferentialantrieb entschieden.

9 Lösungsideen

Micro ROS ist eine fork von ROS und damit auch lauffähig auf schwächeren Systemen. Als Microcontroller für unsere Roboter Plattform fiel die Entscheidung auf einen ESP32. Der

ESP32 bietet für seine Kosten und seine Größe ausreichend Rechenpower und desweiteren unterstützt das Micro-ROS Framework bereits den ESP32. Als alternativen für den ESP32 könnten ansonsten noch ein Raspberry PI oder ein Arduino bzw. ATmega Microcontroller eingesetzt werden.

Für das Front-End kommen viele Frameworks in Frage aber wir wollten unsere Auswahl vorerst auf React oder Vue beschränken. Mit diesen Frameworks haben bereits alle unsere Gruppenteilnehmer gearbeitet und somit hatten wir alle schon Erfahrung damit gesammelt. Und einsteig wurde erleichtert.

Als Notlösung für unser Frontend kämme ansonsten Plain HTML und CSS in frage, was aber höchstwahrscheinlich in einem sehr großen Aufwand enden würde.

10 Evaluation der Lösungsideen anhand der Anforderungen

10.1 Wahl des Frontend Frameworks

Beispiel an Mainsail OS

10.2 Warum ROS?

10.3 Micro-ROS und RTOS

10.4 Roboter Plattform Prototyp und Designentscheidungen

Die Anforderungen an die Roboter Plattform für dieses Projekt waren recht klein und einfach gehalten.

Das wichtigste bei der Neuentwicklung und Design einer Roboter PLatofr

11 Implementierung

11.1 React Front-End

11.1.1 Aufbau des Frontends (Architektur)

11.2 ROS Back-End

11.2.1 Firmware Kompilierungs Toolchain

11.3 Roboter Plattform

11.3.1 Elektronik

11.3.2 3D-Druckteile

12 Evaluation der Implementierung

12.1 Fehler und Verbesserungen Frontend

12.2 Fehler und Verbesserungen Backend

12.3 Fehler und Verbesserungen Roboter Plattform

13 Fazit und Ausblick