

Diplomarbeit
Gruppe “SwarmBots”
Schuljahr 2024/25

Betreuer: **Prof. Erich Erker**
Gruppenmitglieder: **Arthur Burjak**
 : **Leander Gastgeber**
 : **Jones Soliman**
 : **Mihael Stojkovic**

Inhaltsverzeichnis

1	Zusammenfassung	1
1.1	Abstract (English)	1
2	Projektstart	1
3	Prototyp	1
4	Erster Roboteraufbau	1
4.1	Elegoo Tumbler Kit	1
4.2	Guide	2
4.3	Tamerlan & Bambi	3
5	Programmierung	3
5.1	ESP32-CAM	3
5.2	Sensoren	4
5.2.1	LiDAR	4
5.2.2	Gyroskop	4
5.2.3	Encoder	4
6	Backend-Server	4
6.1	Datenverwaltung	4
6.2	Roboterpositionen	4
7	Web-Interface	4
7.1	LiDAR-Karte	4
7.2	Anzeigen der Sensordaten	4
7.3	Fernüberwachung per Kamera	4
8	Probleme	4
8.1	Docker	4

Abbildungsverzeichnis

Tabellenverzeichnis

1 Zusammenfassung

Ziel dieser Diplomarbeit ist es, drei Roboter zu entwickeln, welche kooperativ die Umgebung erkunden können. Hierbei ist ein Roboter (“Guide”) mit einem LiDAR-Sensor ausgestattet, welcher Entfernungsmessungen durchführt. Die anderen beiden Roboter (getauft “Tamerlan” und “Bambi”) sollen komplett “blind” sein. Koordiniert wird das ganze über einen zentralen Server, welcher die gesammelten Daten zusätzlich über ein Webinterface darstellt. Als zusätzliche Aufgabe sollen sich die Roboter auf nur einer Achse balancieren, da wir Kits für balancierende Roboter verwenden, welche wir für unsere Zwecke modifiziert haben. Die Software basiert auf den Robotern, die wir letztes Jahr im Zuge der Projektwoche als Vorbereitung auf die Diplomarbeit gebaut haben.

1.1 Abstract (English)

TODO: translate

2 Projektstart

3 Prototyp

Im Zuge der Projektwoche im Schuljahr 2023/24 haben wir bereits begonnen, einen ersten Prototypen unseres SwarmBots-Systems zu bauen. Dieser Prototyp bestand aus nur zwei Robotern, welche auf Basis eines fertigen Fahrgestells zu sehr wackligen Gefährten wurden. Sehr viel Autonomie hatten die früheren Roboter auch noch nicht, der LiDAR war noch nicht funktionstüchtig, stattdessen wurden die Roboter mittels Videospiel-Controller ferngesteuert. Diese Fernsteuerung wurde aber auch schon damals über eine Websocket-Verbindung implementiert. Außerdem wurde während der Projektwoche der Code für die ESP32-CAMs fast komplett fertiggestellt, diese verwenden jetzt immer noch größtenteils das gleiche Programm. Trotz der Schwächen des damaligen Systems wurde unserem Projekt der erste Preis in der Kategorie “Experten” verliehen.

4 Erster Roboteraufbau

4.1 Elegoo Tumbler Kit

Um den Hardwareaufbau so einfach wie möglich zu gestalten, entschieden wir uns dafür, fertig entwickelte Kits online zu bestellen und dann zu modifizieren. Die Wahl des Kits fiel letztendlich auf den “Tumbler” von Elegoo (Siehe Abbildung 1). Der Tumbler ist ein zweirädriger Roboter, welcher auf einer Achse balanciert. Zur Kontrolle des unmodifizierten Kits gibt es eine Smartphone-App, welche die Roboter über Bluetooth

fernsteuern kann. Da wir die Roboter über WLAN steuern wollten, und die Tumbler-Kits standardmäßig nur eine Bluetooth-Erweiterung eingebaut haben, haben wir die mitgelieferten Arduino Nano durch ESP32-Boards im Arduino Nano-Format ersetzt.



Abbildung 1: Rendering des Elegoo Tumbler

4.2 Guide

Die Aufgabe von *Guide* ist es, mithilfe eines LiDAR-Sensors (Siehe Kapitel 5.2.1) die Umgebung nach Hindernissen und den anderen Robotern abzusuchen. Die vom LiDAR gesammelten Abstandsdaten werden über eine TCP/IP Websocket-Verbindung an einen zentralen Server weitergegeben, welcher diese weiter verarbeitet. Um den LiDAR-Sensor zu montieren, haben wir das Fahrgestell leicht modifiziert und die oberste Ebene (mit dem Akku) erhöht, um Platz für Schrauben zu schaffen.

4.3 Tamerlan & Bambi

5 Programmierung

Für die Programmierung der Mikrocontroller verwenden wir PlatformIO. PlatformIO ist eine Alternative zur Arduino-IDE, welche mithilfe von Plugins in viele IDEs wie z.B. VSCode und CLion integriert werden kann. Alternativ kann man PlatformIO auch über die Kommandozeile bedienen. Vorteile von PlatformIO gegenüber der Arduino-IDE sind u.A. schnelleres Kompilieren, ordentliche Auto-Vervollständigung, statische Code-Analyse (Linting), und ein schön geregeltes System für Bibliotheken.

5.1 ESP32-CAM

Wir verwenden jeweils ein ESP32-CAM AI-Thinker Modul zur erweiterten Fernüberwachung der Roboter. Dieses verbindet sich über WLAN mit dem IoT-Netzwerk und bietet über HTTP einen MJPEG-Videostream an. Die drei Videostreams (einer pro Roboter) werden dann im Web-Interface zusätzlich zu den LiDAR-Umgebungsdaten angezeigt, um das räumliche Vorstellungsvermögen der Benutzer zu unterstützen.



Abbildung 2: Erstes empfangenes Bild der ESP32-CAM

5.2 Sensoren

5.2.1 LiDAR

5.2.2 Gyroskop

5.2.3 Encoder

6 Backend-Server

6.1 Datenverwaltung

6.2 Roboterpositionen

7 Web-Interface

7.1 LiDAR-Karte

7.2 Anzeigen der Sensordaten

7.3 Fernüberwachung per Kamera

8 Probleme

8.1 Docker

Am Laptop von Herr Gastgeber war es am Beginn des Schuljahres nicht möglich, jegliche Geräte im Netzwerk der HTL zu erreichen. Grund dafür war eine Software namens Docker, welche außerhalb dieses Projekts zur Containerisierung von anderen Anwendungen dient. Docker war standardmäßig so konfiguriert, dass es containerspezifische Subnets im IPv4-Adressbereich 172.17.0.0/16 erstellt. Diese Subnets hatten dann am Laptop eine höhere Priorität als das Schulnetzwerk, was dafür sorgte, dass das weltweite Internet noch erreichbar war, nicht aber das lokale Schulnetzwerk.

Abbildungsverzeichnis

1	Rendering des Elegoo Tumbler	2
2	Erstes empfangenes Bild der ESP32-CAM	3

Tabellenverzeichnis