

# Projektangabe MRK: LLM als Steuerzentrale

Wilfried Wöber

August 2025

## 1 Angabe

Erstellen Sie einen Chatbot zur Steuerung Ihrer Roboteranwendung. Dieser Chatbot soll folgende Funktionen beinhalten:

1. **LLM Interface:** Implementieren Sie ein Interface zu einem kostenfreien large language model (z.B.: Gemini 1.5 flash).
2. **Retrieval-Augmented Generation:** Implementieren Sie frei zugängliche Chunking und word embedding Methoden für einen Retrieval-Augmented Generation (RAG) Ansatz. Nutzen Sie einen lokalen Vectorstore.
3. **Chatbot:** Implementieren Sie basierend auf den beiden vorhergehenden Punkten einen Chatbot. Für diese Implementierung müssen Sie eine Chat-History implementieren.
4. **Tooling:** Geben Sie via *Tooling* Ihrem Chatbot die Möglichkeit den Roboter zu steuern. Bauen Sie Ihr System auf dem agentic Prinzip auf (keine hardgecodedtes Tooling). Das Tool soll Goals akzeptieren und den Roboter dementsprechend anteuern.

Die Anwendung soll schlussendlich durch natürlicher geschriebener Sprache in Konversation zwischen dem Chatbot und den Anwender\*innen den Roboter zu Zielen schicken. Die Zielen sollen nicht via Koordinaten, sondern via Namen der Zielen (siehe unten) bezeichnet werden. Ein Beispiel ist

*Fahren bitte in dein Zuhause.*

Adaptieren Sie die System- und Userprompt für Ihre Anwendung. Der Chatbot soll so lange nachfragen bis Zielen abgeleitet werden können. Erstes dann soll autonom (!) das Tool angesprochen werden.

### 1.1 Details Chatbot

Sie können Orchestrierungsframeworks wie *Langchain* oder *Haystack* zur Entwicklung verwenden. Weiters können Sie bestehende Module für LLM-Interfaces, den Vectorstore, Chunking, Text Embedder, Chat History und Tools verwenden. Weitere Hilfestellungen sind:

1. LLM Interface und Hyperparameter: Beachten Sie die Hyperparameter *Temperature* und *top\_p* bei der Integration des LLM-Interfaces. Beachten Sie weiters, dass bei den meisten LLM Implementierungen der API-Key als Umgebungsvariable vorhanden sein muss. Diese kann via `$export name=Variable=Key` im Terminal dem Programm zur Verfügung gestellt werden.  
Hier finden Sie Tipps.
2. RAG: Sie müssen mindestens diese Angabe dem RAG-System zur Verfügung stellen. Beachten Sie, dass die Chunk-Größe und Überlappung bei Standardverfahren essentielle Parameter sind, welche die Qualität der Anwendung signifikant beeinflussen.  
Hier finden Sie Beispiele.
3. Chatbot: Sie können einen Chatbot selbstständig oder mit bestehenden Modulen (siehe RAG-Link) implementieren. Sollten Sie bestehende Graphen-basierte Lösungen nutzen werden diesbezüglich Fragen bei der Endpräsentation gestellt werden.
4. Tools: Für den Tool Aufruf können Sie bestehende Lösungen nutzen (siehe MCP) bzw. Implementierungen in Orchestrierungsframeworks (siehe hier (empfohlen)) nutzen.

## 1.2 Umgebung des Roboters

Die ROS Umgebung finden Sie im Moodlekurs. Sie können frei wählen, ob Sie ROS1 (Support vom Lektor) oder ROS2 (freiwillig) nutzen wollen.

Prinzipiell benötigen Sie keine neuen ROS-Fähigkeiten. Der inhaltliche Schwerpunkt dieser LV ist die Entwicklung des Chatbots, welcher eine bestehende ROS Umgebung ansteuern soll.

Folgende Posen sind in der Karte definiert:

Es folgen hier vier Posen, welche Sie selbstständig basierend auf Ihrer ROS-Umgebung in der im Moodlekurs zur Verfügung stehenden tex Datei hinzufügen müssen. Diesen Schritt benötigen Sie für das RAG-System. Füllen Sie daher selbstständig die folgenden Posen aus. Löschen Sie im Anschluss diesen roten Text.

1. Home-Position:  $x : 0 \ y : 0.5 \ \theta : 0$
2. Pose A:  $x : 2 \ y : -0.5 \ \theta : 1.75$
3. Pose B:  $x : -1 \ y : 0.5 \ \theta : 3.14$
4. Ziel:  $x : -1 \ y : -0.5 \ \theta : -1$

**Anmerkung:** Experimentieren Sie. Versuchen Sie Anweisungen wie "Düse zur Home-Position, fahre dann von Punkt B zur Pose A und dann zum Ziel.".

### 1.3 Entwicklung und Dokumentation

Nutzen Sie Github zur Entwicklung. Die *README.md* Datei soll eine vollständige Dokumentation Ihres Projekts enthalten. Weiters sollen Ihre Erkenntnisse dokumentiert sein. Diese müssen mindestens folgende Aspekte beinhalten:

- Auswirkung unterschiedlicher Sprache (z.B.: Deutsch, Englisch)
- Auswirkung unterschiedlicher Chunking Hyperparameter bzw. Chunking Ansätzen
- Ihre Empfehlung für industrielle Nutzung. Wäre ein Chatbot zur Steuerung eines Industrieroboters aus Ihrer Sicht empfehlenswert? Was wären die Vorteile/Nachteile.
- (optional, empfohlen) Die Auswirkung von unterschiedlichen LLM.

Bereiten Sie für die letzte Einheit dieser LV eine Präsentation (Slides und eine Livepräsentation) vor. Die Benotung ist individuell. Die Basis der Benotung ist:

1. Korrekte und vollständige Implementierung des Chatbots inkl. RAG und Tooling. (60%)
2. Funktions der Anwendung. (20%)
3. Dokumentation (20%)

Alle Teammitglieder werden über alle Aspekte und den Quellcode befragt. Sollte ein Teammitglied Fragen nicht ausreichend beantworten können wird für dieses Teammitglied das Projekt negativ bewertet. Ein neues, individuelles Projekt ist dann als Zweitbeitritt durchzuführen. Sollte dieses Projekt wieder negativ bewertet werden wird ein drittes Projekt definiert, dessen Lösung vor einer Kommission in einer kommissionellen Prüfung präsentiert werden muss.