AI-Assisted Programming of Electric Motor Control Systems

Graduate

Raphael Carigiet

Introduction: Programming embedded control code requires in-depth knowledge of hardware-specific APIs and the C programming language. Many users of the hardware do not possess the programming skills required to implement the control logic themselves.

This master's thesis bridged this gap by developing software that utilizes generative artificial intelligence (AI) to automatically translate natural-language descriptions of the desired hardware behavior into executable C code for the embedded platform. This simplifies the development of embedded applications, reduces development time and makes them accessible to users without in-depth programming knowledge.

Approach / Technology: A Retrieval-Augmented Generation (RAG) approach was chosen as the basis for the software architecture, as the limited amount of available data did not allow any training of a model. The hardware documentation and code examples are stored in a vector database and used as a knowledge base for code generation. For each user input, the most relevant information is retrieved from this knowledge base and provided as context to the Large Language Model (LLM) running locally.

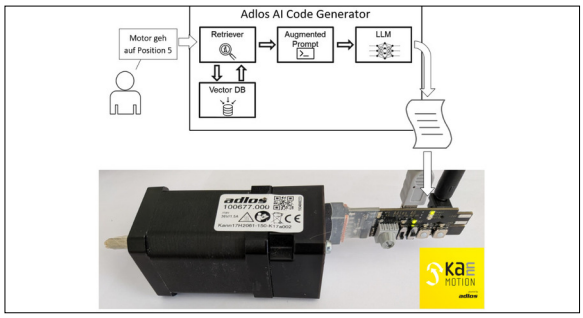
The code generation process has also been structured and expanded. The software architecture follows a multi-agent system, in which the tasks of information retrieval, code generation and quality assurance are handled by specialized agents. Two feedback loops are used to ensure code quality. In the first loop, each piece of code is automatically checked by the compiler, which automatically corrects any reported errors. The second loop comprises an LLM-based code review step, in which the generated code is compared with the original user input to identify and correct any discrepancies between the requirement and its implementation.

Conclusion: The developed software can run on a standard computer without dedicated graphics hardware and a high-performance AI workstation. Six open-source LLMs were systematically evaluated to identify the most suitable model. The "qwen3-coder:30b-a3b" model proved to be the best overall solution for the task at hand, given the available resources.

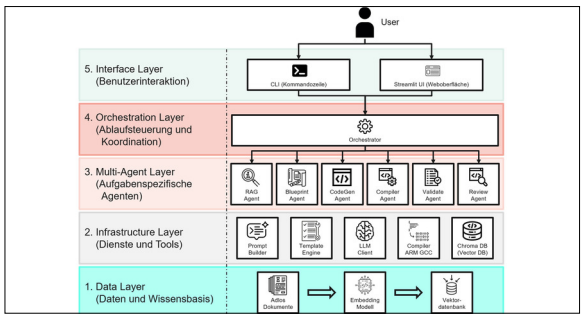
The achievable code quality and processing time on a standard computer and an AI workstation were compared using this model. The results show that code can be generated on the standard computer whose quality is comparable to that achieved on the more powerful workstation. This is particularly relevant for applications in which processing time is as important as output quality, since the generation process requires at least 14 times more time.

The software runs on cost-effective standard hardware rather than on dedicated AI infrastructure, thereby reducing the cost of using generative AI in embedded development.

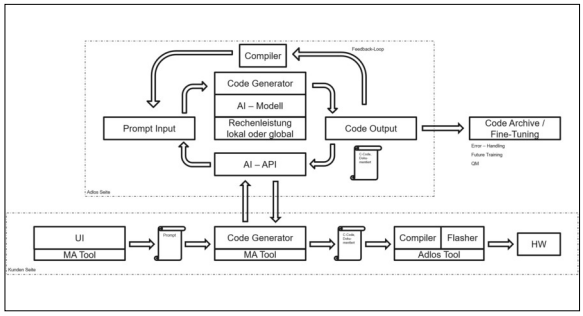
Conceptual idea behind the AI application
Own presentment



Software architecture of the multi-agent AI application
Own presentment



Integration of AI-based code generation into the user's development process
Own presentment



Advisor
Prof. René Pawlitzek

Co-Examiner
Prof. Mag. Dipl.-Ing. Dr. Kathrin Plankensteiner, FHV Vorarlberg University of Applied Sciences

Subject Area
Computer Science, Data Science

Project Partner
Adlos AG, Balzers, Liechtenstein