

**Dávid Juhász** | Software Engineer | Bridging Hardware-Software Co-Design with Clarity  
[+46 735 88 21 59](tel:+46735882159) | [hello@davidjuhasz.dev](mailto:hello@davidjuhasz.dev) | [davidjuhasz.dev](http://davidjuhasz.dev) | [linkedin.com/in/juhaszdavid](https://linkedin.com/in/juhaszdavid) | Sweden

## Summary

I am a Systems Engineer building the core layers of the hardware-software stack—from compilers and simulators to production firmware. I specialize in the **0-to-1 phase** of co-design, bringing software engineering rigor to resource-constrained platforms. By bridging the gap between disciplines, I enable teams to move beyond fragmented handovers toward a **joint ecosystem** of efficient, high-impact solutions.

## Skills

- Compilers & Tool Development: LLVM, TVM, code optimization, scheduling, code generation, static analysis
- Embedded & Systems Engineering: Hardware bring-up, bare-metal, firmware, RTOS, PCB- and IC-level understanding
- Communication Protocols: UART, SPI, I2C, CAN, Ethernet, TCP, UDP, sockets, custom protocol stacks
- Programming Languages: C, C++, Python, Rust, Assembly
- Toolchains & Build Systems: LLVM, Clang, GCC, Make, CMake
- Debugging & Optimization: GDB, LLDB, hardware probes, waveform analysis, performance profiling and tuning
- Development Tools & Environments: Git, GitHub, Azure DevOps, VSCode, Bash, Linux CLI, SSH, Docker, Podman
- Software Architecture: Resource-constrained, multi-threaded, real-time systems, interpreters, virtual machines
- Modeling & Simulation: Performance modeling, system simulation, hardware-software interaction
- Hardware-Software Co-Design: FPGA integration, hardware acceleration, architecture optimization, edge AI
- Machine Learning & AI Acceleration: PyTorch, Tensorflow, ONNX, interpretation and compilation-based deployment
- Technical Leadership & Mentorship: Guiding engineers, fostering knowledge sharing, driving architectural decisions
- Cross-Functional Collaboration: Strategic goal alignment, bridging engineering, product, and leadership teams

## Experience

**Senior Software Engineer** | *Frontgrade Gaisler* | *Gothenburg, Sweden (Remote)* | *June 2025 – ongoing*

- Enabled neuromorphic edge AI on space-grade RISC-V platforms via Akida runtime integration and benchmarking.
- Contributed to FPGA verification and RTL reviews for custom SoC designs, aligning hardware and software teams

**Senior Architectural Modeling Engineer** | *FlexAI* | *Paris, France (Remote)* | *July 2024 – November 2024*

- Shaped team vision, roadmap, and modeling strategies to support next-generation AI accelerator development.
- Led profiling of PyTorch-based LLMs to assess execution time, memory usage, and optimization opportunities.

**Expert Software Engineer – Edge AI Systems** | *Imys* | *Stockholm, Sweden (Hybrid)* | *November 2022 – May 2024*

- Directed embedded software and AI toolchain development, aligning technical execution with strategic goals.
- Delivered runtime libraries, simulation frameworks, and code generation pipelines for a custom edge AI accelerator.
- Contributed to system architecture design for RISC-V integration and evaluated accelerator-software interfaces.

**Head of Software Engineering** | *Imys* | *Stockholm, Sweden (Hybrid)* | *March 2021 – October 2022*

- Led software department strategy, team growth, and cross-functional coordination across edge AI projects.
- Defined system architecture for a scalable edge AI platform, guiding execution model and system co-design.
- Contributed to code generation and simulation tools enabling early-stage validation of accelerator architectures.

**Lead Software Architect** | *Imys* | *Stockholm, Sweden (Hybrid)* | *January 2018 – March 2021*

- Spearheaded LLVM-based toolchain and embedded software stack development for custom processor platforms.
- Delivered full toolchain solutions, including compiler, linker, debugger, and integrated VSCode extensions.
- Supervised system architecture, board bring-up, FreeRTOS porting, and hardware driver development.

**R&D Software Engineer** | *Imys* | *Stockholm, Sweden* | *August 2014 – January 2018*

- Developed an LLVM-based code generator and tooling to optimize embedded architecture workflows.
- Built low-level debugger, simulator, and assembler in modern C++ to enhance embedded system development.

## Education

**MSc and BSc in Computer Science** | *Eötvös Loránd University* | *Budapest, Hungary* | *2007 – 2012*

- Specialized in compilers, formal methods, and programming languages; graduated with distinction for both degrees.

**PhD Graduate Work (Industry-Based, Unfinished)** | *Vienna University of Technology* | *Vienna, Austria* | *2016 – 2020*

- Conducted research at the intersection of compiler optimization, machine learning, and embedded systems.