Tim Jensen

Software Engineer

+49 1511 6039958
mail@timjensen.de
tjensen42

Profile

Software Engineer with an interdisciplinary background and about three years of experience in Rust. I really enjoy the challenge of writing simple yet robust code. Fascinated by the idea of combining compile-time safety with runtime performance, I have specialized in low-level Rust. I am looking for an environment that shares my curiosity, where I can collaborate on modern technologies and continuously learn.

Education

BSc Mechatronics (Focus: Computer Science) Sep 2019 – present
Technical University of Applied Sciences Mannheim

- Interdisciplinary integration of mechanical, electrical, and software engineering principles. Focused on microcontrollers, systems programming, and hardware interfacing.
- Thesis: Implementation of a software-based, OSEK-compatible real-time scheduler in Rust for preemptive control of async tasks.
- Current Grade Average: 1.4

Core Curriculum Jul 2021 – Jan 2023
42 Heilbronn

- Developed strong self-reliance through purely project-based learning of C/C++ and computer science fundamentals without instructors.
- Engaged in peer-to-peer learning within an international environment, involving mutual code reviews and discussions. Learned to approach technical problems from different perspectives and effectively share knowledge.
- Won 1st Place at MHP Mobility Hack 2022: Developed an end-to-end OTA firmware update solution for IoT devices in a 72h sprint, competing against 150+ participants.

Experience

Software Engineer (Intern, Working Student, Thesis) Feb 2023 – present
Vector Informatik GmbH | R&D Embedded Systems

- **Thesis & Continued Development:** Executed the Bachelor Thesis research. Subsequently extended the scheduler to support deadlock-free resource sharing (OSEK-PCP).
- **Async Rust:** Conducted a design-space exploration of async runtimes (e.g., Embassy, RTIC, Tokio) and investigated `io_uring` and zero-copy IPC mechanisms for automotive use cases.
- **Interoperability & Protocols:** Evaluated Rust/C++ interop strategies (e.g., CXX, Cap'n Proto) and implemented SOME/IP (de)serialization in Rust. Analyzed communication patterns in MICROSAR Adaptive versus native Rust ecosystem solutions.
- **Tooling & Infrastructure:** Co-developed a C#/Java prototype for a new declarative Software Product Line approach, implementing an ANTLR grammar for a custom DSL and analyzing Git internals. Established the team's Rust development infrastructure by creating standardized Docker images and GitLab CI templates.

Formula Student Team Member | Electronics Oct 2019 – Sep 2021
Delta Racing Mannheim Electric e.V.

- Development of the safety-critical Tractive System Active Light (TSAL): PCB design, mechanical design (CAD & 3D printing), assembly, wiring, and hardware bring-up.
- Programming distributed systems on TM4C123x MCUs using FreeRTOS; system-level debugging and error analysis on the CAN bus.

Web Design & Server Administration Feb 2016 – Dec 2022
Freelance

- Design and development of websites (HTML, CSS, JS, WordPress) and creation of design proposals (UI/print) using Figma, Affinity Designer, and Pixelmator.
- Setup and administration of Linux servers (VPS). Configuration of Nginx, DNS, mail servers, and automated backups, later utilizing Plesk for management.

Selected Projects

Autonomous 2x2 Rubik's Cube Robot Mar 2025 – June 2025
C/C++, ESP32, Computer Vision, 3D Printing

- Developed an autonomous robot using an ESP32-CAM board and 3D-printed parts.
- Implemented servo control and computer vision for state detection. Compressed the solution space into a binary lookup table for O(1) move retrieval on constrained hardware.

miniRT – Physics-Based Ray Tracer Mar 2022 – June 2022
C, Linear Algebra, Computer Graphics | Sources

- Developed a ray tracer from scratch, translating optical physical laws directly into code.
- Implemented ray-object intersections (spheres, cubes, cylinders) and simulated shadows, diffuse/specular reflections, texture mapping, and refraction.

minishell – Bash-like Shell Implementation Nov 2021 – Jan 2022
C, Unix Systems Programming | Sources

- Developed a fully functional shell from scratch, managing process lifecycles and IPC.
- Wrote a custom lexer/parser to handle command arguments, environment expansion, wildcards, and logical operators (&&, ||).

Skills & Technologies

Programming Languages:	Rust, C, C++, Bash, Typescript, Python
Development Tools:	Linux, Git, Docker, GitLab CI/CD, Cargo, Make, VS Code
Documentation:	Typst, mdBook, draw.io, HTML, CSS, Figma
Spoken Languages:	German (Native), English (Fluent, B2–C1)

Hobbies & Interests

As a counterbalance to digital work, I recharge outdoors through calisthenics, bikepacking, and hiking. I love acquiring new skills, currently training to master the freestanding handstand and kettlebell juggling. My engineering passion translates into the physical world through DIY projects like my camper van conversion and a custom bike building. Driven by curiosity, I also experiment with nutrition, cultivating sprouts and exploring fermentation.