

Matěj Schrödl

Prague, Czech Republic

Email: matej.schrodl@gmail.com Mobile: +420 720 178 081

GitHub: github.com/schrodlm

LinkedIn:

LeetCode: leetcode.com/schrodlm

EXPERIENCE

C++ Firmware Developer - Prusa Research

January 2025 - present

- Day-to-day work on the Buddy firmware (firmware running on Prusa's FDM 3D printers - MK4, XL, MINI, CORE One): feature development, code review, and bug fixing across the printer lineup. Recent examples:
 - INDX multitool system:** fan control, inactive-nozzle heating, tool-pickup selftests.
 - Communication infrastructure:** a COBS encoder/decoder for framed serial protocols (with unit tests) and a centralized printer CRC layer integrated into the Modbus stack.
 - OpenCyphal-over-UART (ANFC):** communication bridge for legacy/limited MCU targets — peripheral configuration, HAL integration, a VCAN/UART bridge utility, and Jenkins CI lanes.

C++ Student Developer - Siemens Mobility

November 2023 – December 2024

- Product context:** firmware for a **railway point-machine controller** (the actuator that switches trains between tracks), developed under **CENELEC EN 50128 / SIL** rail-safety standards. The system was structured as about a dozen cooperating C++ components with a formal init/configure/cyclic lifecycle.
- Role:** owned two of those components end-to-end (development, code review) within the formal safety process — DOORS-based requirements, formal reviews, and the project's C++ styleguide.

Java Junior Developer - Trixi s.r.o

February 2023 - September 2023

- Smartform (Java/Spring):** built the Slovak-side ingestion service for Trixi's Smartform - a Czech form-validation / address-autocomplete API used by 1000+ Czech e-shops and enterprises (e.g. Alza, Zonky, Tipsport). Daily S3 batch pulls from the Slovak business register into PostgreSQL via Liquibase migrations, with async REST endpoints for batch retrieval.

SKILLS

- Proficient:** C++ (17/20), C; CMake, GCC arm-none-eabi, Git, gdb, Linux
- Familiar:** Python; FreeRTOS, STM32 HAL; SPI, I2C, UART, CAN, Modbus, OpenCyphal; Jenkins
- Languages:** English (fluent), Czech (native)

EDUCATION

Faculty of Mathematics and Physics, Charles University

Master's Degree in System Engineering

Prague, Czech Republic

September 2025 - present

Faculty of Information Technology, Czech Technical University

Bachelor's Degree in Software Engineering

Prague, Czech Republic

September 2021 - June 2025

PROJECTS

Bachelor's Thesis - Benchmarking Framework for Rotor Models. Platform for evaluating SMT models generated from machine code in the Selfie ecosystem. Characterizes such models, compares them against conventional SMT encodings across multiple solvers, and demonstrates the framework on sample benchmarks.

Operating System Kernel (team). Built an OS kernel from scratch in C with a 3-person team for the MSIM (MIPS-like) simulator. Implemented kernel heap allocator, cooperative and preemptive thread scheduling, frame allocator, virtual memory, synchronization primitives, and userspace process support across seven incremental milestones.

Parallel and HW-level Code Optimization. Course portfolio covering CPU-level performance engineering and parallel computing in modern C++. Implemented and benchmarked binary search and Floyd-Warshall with SSE/AVX/AVX-512 intrinsics and MPI; k-means with Intel TBB; an n-body potential simulator with SYCL on heterogeneous hardware; and PySpark deduplication on a Slurm cluster.

obsidian-md-converter — shipped Python tool. Packaged CLI that turns an Obsidian vault into clean Markdown for static-site generators (Jekyll, Hugo). Pluggable front-matter backends, YAML-driven per-directory mappings, type-safe path wrappers, structured error hierarchy, full unit + integration test suite under GitHub Actions, and a ready-to-use pre-commit hook for validating vaults. Drives my own publishing pipeline.

Additional:

- Single-cycle 32-bit CPU in Verilog** with a custom ISA and testbench.
- Embedded firmware coursework:** bare-metal drivers and FreeRTOS-based real-time scheduling, including a DMA-driven SPI peripheral backed by internal flash.
- Cache-conscious data structures (C++):** splay trees, AB-trees, cuckoo hashing, and cache-aware matrix

transpose with experimental write-ups.

- I enjoy algorithmic problems (graph theory, DP, game theory) in C++/Scala - see LeetCode.