

Ronak Jain

U.S. Citizen | 617-840-1096 | [ronak @ ronakpjain.com](mailto:ronak@ronakpjain.com) | github.com/ronakpjain | linkedin.com/in/ronakpjain

EDUCATION

Purdue University

B.S. in Electrical Engineering, GPA: 3.45/4.0

May 2028

West Lafayette, IN

Relevant Coursework: *Microprocessor Systems and Interfacing, Computer Design and Prototyping, ASIC Design Lab, Operating Systems, Compilers and Translator Writing Systems*

EXPERIENCE

Software Engineering Intern

May 2026 – August 2026

Tesla

Palo Alto, CA

- Designed and integrated V2G PowerShare firmware for new charging products in a shared monorepo, defining structured component interfaces with Protocol Buffers.
- Developed firmware for Arm Cortex-M33 and Cortex-A55 platforms, integrating FreeRTOS, Embedded Linux, Buildroot, CAN communication, and gRPC service interfaces.
- Orchestrated asynchronous Rust components with Tokio to coordinate concurrent charging-system operations.
- Streamlined firmware integration and delivery by automating builds and test execution through CI/CD and AI-assisted development workflows, improving reliability.
- Authored unit and processor-in-the-loop tests to validate firmware behavior before hardware deployment, reducing test setup time by approximately 30%.

PROJECTS

Purdue Electric Racing

September 2024 – Present

- Architected a CAN bootloader and DaqApp update workflow for six STM32G474 nodes with generated transports, CRC/vector validation, metadata commits, and reset-safe recovery.
- Automated firmware packaging with Python and Rust tooling to generate manifest-driven firmware archives, pad images, compute STM32 CRCs, and validate sizes, addresses, paths, and CAN IDs.
- Implemented STM32G4 HALs for flash, ADC, USB, SPI, and USART with DMA-backed transfers, input validation, and dedicated hardware test targets.
- Integrated firmware testing with CMake, GoogleTest/CTest, C23 shims, sanitizers, coverage, and GitHub Actions.
- Expanded the CAN library to support FDCAN, route and filter messages, and handle standard and extended identifiers.
- Configured a Yocto/KAS Raspberry Pi CM5 PREEMPT_RT image with 1 kHz timers and PPS support, then automated cached WIC builds in GitHub Actions.

FIL

July 2026 – Present

- Built FIL from scratch in C++20 to run real ARM ELF firmware at the instruction level on an emulated STM32G4/Cortex-M4F, modeling CPU exceptions and memory alongside DMA, timers, ADC, USART, and FDCAN.
- Connected six simulated boards over virtual CAN with deterministic scheduling and simulated-time events, enabling repeatable vehicle-network scenarios with live CAN frame injection and JSONL execution traces.
- Accelerated instruction-level firmware execution with caching, exact-state loop batching, and transactional parallel slices, achieving 1.37x real-time throughput on a 96M-instruction workload.
- Validated FIL with GoogleTest/CTest and sanitizers; used simulated-time event inputs, live CAN frame injection, and JSONL traces to reproduce and inspect firmware behavior across the six-board vehicle network.

Ronfire

May 2025 – Present

- Built an asynchronous Tokio-based Rust HTTP server with modular transport and static-file layers, serving plugin routes over Unix sockets at 4,100+ requests/sec and 1.65 GB/s on image workloads in benchmark testing.
- Refactored HTTP, transport, static-file, runtime, and plugin layers into feature-gated APIs; implemented compiled-in handlers with factory registration, configuration validation, and exact-route dispatch.
- Implemented a proxy plugin with host-restricted redirects, response-size limits, timeouts, TTL caching, and stale fallback.
- Hardened HTTP handling with bounded header reads and pipelined-request framing; supported byte-range and HEAD requests while containing static files within the document root against traversal and symlink escapes.

TECHNICAL SKILLS

Languages: C, C++, Rust, Python, Lua

Embedded & RTOS: STM32, Arm Cortex-M, FreeRTOS, CAN/FDCAN, SPI, USART, DMA, ADC, USB

Systems & Build: Embedded Linux, Yocto, Buildroot, KAS, PREEMPT_RT, CMake, Docker, GitHub Actions

Frameworks & Tools: Tokio, gRPC, Protocol Buffers, GoogleTest, CTest, unit/PIL testing, Git, Neovim