

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

May 2028

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

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

- Engineered and integrated firmware features, including Vehicle-to-Grid (V2G) PowerShare capabilities, for new charging products within a large, shared monorepo, using Protocol Buffers for structured interfaces.
- Worked across Arm Cortex-M33 and Cortex-A55 cores using FreeRTOS, Embedded Linux, and Buildroot, with CAN-based communication and gRPC service interfaces connecting charging system components.
- Developed asynchronous components in Rust with Tokio to coordinate concurrent charging-system operations.
- Leveraged CI/CD frameworks and AI-assisted development tools to automate builds and test execution, enabling reliable integration and delivery of firmware changes.
- Created and maintained unit and processor-in-the-loop (PIL) tests to validate firmware behavior, reducing hardware-test setup time by approximately 30%.

PROJECTS

Purdue Electric Racing

September 2024 – Present

- Architected a production bootloader framework for STM32 G4/F4 controllers, including CRC-validated firmware updates, metadata-managed boot flow, and safe application handoff logic.
- Engineered an FDCAN messaging library in Python with routing/filtering for standard and extended identifiers, enabling reliable inter-controller communication across vehicle subsystems.
- Built firmware tooling to automate bootloader/app builds, CRC generation, combined recovery image creation, and release packaging for deployment workflows.
- Contributed to DaqApp2 (Rust) to support real-time CAN telemetry visualization and bootloader update operations, improving serviceability and debugging speed.
- Developed robust USART/SPI HALs with DMA and migrated scheduling from bare-metal loops to FreeRTOS to improve determinism for safety-critical control tasks.

Ronfire

May 2025 – Present

- Engineered a high-performance asynchronous HTTP server in Rust with Tokio, serving static and plugin-backed routes over Unix domain sockets at over 4,100 requests/sec (1.65 GB/s) for image workloads.
- Refactored the server into focused HTTP, transport, static-file, runtime, and plugin modules while preserving root-level APIs and supporting feature-gated builds with no default features.
- Designed a compiled-in plugin architecture with object-safe handlers, factory-based registration, strict configuration validation, exact-route dispatch, and a feature-gated reverse proxy using reqwest and rustls.
- Hardened request and proxy handling with bounded header/body reads, timeout and redirect policies, TTL caching with stale fallback, byte-range/HEAD support, and document-root containment against traversal and symlink escapes.

Ronq

June 2025 – Present

- Built a C++23 command runner to execute multiple shell commands in parallel using `fork`, `exec`, and pipes, with labeled, colorized output.
- Designed a move-only RAII file descriptor wrapper to enforce exclusive ownership and prevent resource leaks across process boundaries.
- Added process group management and signal handling (SIGINT/SIGTERM/SIGKILL) to support graceful shutdown and robust cleanup of foreground and background processes.

TECHNICAL SKILLS

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

Embedded Systems: STM32, FreeRTOS, FDCAN, SPI, USART, DMA

Tools & Practices: Git, Docker, CI/CD, unit testing, PIL testing, AI-assisted development, **NeoVim**