Perry's Résumé - Embedded Linux & Audio Systems

Perry Kundert <*perry@kundert.ca*> +1-780-970-8148

2026-11-17

Abstract

C++ engineer on embedded Linux with an audio pedigree: a software audio mixer for priority-scaled SCADA alert audio at HP, when mixing was done entirely in software; years of building Linux kernels through the 2.4/2.6 days for advanced driver support; and real-time signal processing at rates that make audio look leisurely — a 25 Msps SDR receiver decoding 64 concurrent channels on commodity ARM hardware. Networking and IPC at scale, signed release delivery, CI/CD, and a garage robotics project with a firmware layer of its own. (PDF, Text)

Skills

- **C/C++ & real-time:** C++ (25+ yrs), C, Python (15+ yrs), Rust (5+ yrs), Go (working fluency); SIMD (Intel AVX, ARM NEON); lock-free concurrency; multi-threaded and multi-process designs with hard latency budgets.
- **Embedded Linux:** kernel building through the 2.4/2.6 years for advanced driver support; device drivers; cross-compilation — ARMv8 with hand-optimized NEON hot paths, C++ to WebAssembly via Emscripten; hardware bring-up, PCB debug, serial protocols, microcontrollers (ESP32, AVR-class).
- **Audio & DSP:** software audio mixer for priority-scaled SCADA alert audio (HP) — all-software mixing before hardware offload was the norm; polyphase filterbanks, window/DFT design, sub-sample timing estimation; real-time pipelines sustained at multi-Msps — audio rates are the leisurely end.
- **Networking & services:** TCP/IP, UDP, HTTP(S), TLS 1.3/mTLS, QUIC, WebSockets; custom datagram protocols (JPEG frames + IMU samples over UDP); service discovery; Unix sockets, messaging systems, architectures of hundreds of concurrent processes.
- **Build, CI & delivery:** Git, CI/CD (GitHub and GitLab pipelines), Docker build images, Nix reproducible builds, Make-based cross-compilation toolchains, signed release delivery (Ed25519).
- **Diagnostics:** Linux system-level debugging across processes, networking, memory and performance; regression harnesses measured against recorded field data; the 2 AM stack trace.

Experience

IT/OT Solutions Engineer - *Vantage Engineering* (Spruce Grove, AB 2026–present)

- Built Ignition SCADA CI/CD: pre-deployment testing of Ignition Python scripts against CPython 2.7, Jython and Ignition's own Jython over an offline runtime simulation; Git-managed deployments, dev/staging/production environments with automated regression testing.
- Developed a secure DNP3 data diode supplying real-time field data to IT systems while leaving OT field equipment unreachable from the IT side.

R&D Consultant - *Dominion R&D Corp.* (Remote, 2009–present)

- Built and operate `BUCKy`, a production local-LLM service (llama.cpp) grounding a ~100K-token knowledge base: precomputed once into a KV cache and cloned copy-on-write per session, ~10x per-query cost reduction, running on commodity Mac hardware.
- Deployed and operate a fleet of hundreds of remotely monitored Starlink-connected routers across remote sites: indoor/outdoor WiFi, camera monitoring, and on-site PCs running AI image analysis (Blue Iris) that triggers event reporting and annotates video segments.
- Developed `ezpwd-reed-solomon`, high-throughput C++ error-control codecs with SIMD inner loops, cross-compiled to JavaScript/WebAssembly via Emscripten; used by aerospace and defence developers globally.
- Created `cpppo`, the Python EtherNet/IP and CIP protocol library used worldwide as the integration layer between analytics and the PLCs on the plant floor.
- Built `datasim`, a large-scale PLC simulation environment — hundreds of concurrent simulated PLCs, each its own process and network stack — with field-data replay and latency, loss and bandwidth injection.
- Designed, fabbed and bench-tested a prototype 1-Wire sensor board for intrinsically safe grain-bin monitoring; the pilot did not proceed.

Software Engineer - *clearGRID Ltd.* (Kelowna, BC 2017–2022)

- Built an end-to-end real-time receiver that polyphase-filterbank-channelized a 25 Msps I/Q stream into 200 kHz channels — 125 in the full band, 65 in the production band — decoding SCM reports on 64 concurrent 900 MHz ISM-band channels, sustained, with no dropped samples, on commodity x86 and ARMv8.
- Built a recorded-signal regression harness so algorithm changes were measured against captured field data rather than synthetic benchmarks.

Distributed Systems R&D - *Holo Ltd.* (Kelowna, BC 2018–2020)

- Architected and validated HoloFuel's partition-tolerant transaction engine in Rust: agents advance shared state whenever they can communicate, with no total-order or quorum lockup.

Sr. IT Advisor - *Enbridge Pipelines* (Edmonton, AB 2002–2009)

- Developed solid-state RTU firmware for continuous 24/7 control in remote locations, sustaining fewer than 1 defect per 10,000 lines of code, measured in the field.
- Architected LiveWIRE, a cryptographically secure, Reed-Solomon erasure-coded, multi-route protocol for real-time SCADA telemetry over marginal and intermittent links.

Software Developer - *Hewlett-Packard* (Calgary, AB 1989–1996)

- Developed the software audio mixer for RTAP's priority-scaled SCADA alert audio — audio mixing done entirely in software, in an era when the hardware did it; shipped in the platform still deployed continent-wide, 30+ years on.
- Re-engineered the core alarm engine of the RTAP SCADA platform around DFA state machines; end-users configure complex, distributed safety-critical workflows.

Personal Projects

BB-8 Droid - *with my daughter* (2026)

- Working on converting a Disney Droid Depot BB-8 into an autonomous, AI-driven robot: an ESP32-S3 camera head streaming MJPEG over a custom UDP protocol (each datagram carries one JPEG frame and its accumulated IMU samples, atomic by design), BLE motor control, monocular visual odometry, GyroFlow stabilization (nobody's done this with a real-time video feed apparently!), and a local LLM model under llama-cpp answering natural-language commands on a commodity Mac. Work in progress (I think perhaps I'm more excited about this project than my girl ;)

Open Source

- cpppo - Industrial EtherNet/IP, CIP and Modbus protocol implementation (Python)
- ezpwd-reed-solomon - Reed-Solomon and BCH FEC (C++/JS/Python)

Education

B.Sc. Computer Science - *University of Calgary* (Calgary, AB 1984-1989)