

SOFTWARE & EMBEDDED SYSTEMS ENGINEERING

Seeking software and embedded systems roles. Available for onsite work, relocation, travel, and internships.

Embedded & hardware: ARM Cortex-M, STM32, bare-metal firmware, CAN, UART, SPI, I2C, PWM; Altium PCB design, board bring-up, oscilloscopes, logic analyzers

Software: C, C++, Rust, Python, Haskell, Go, TypeScript; Linux, Git, Docker, Nix; systems integration and debugging

Flight software & autonomy: ROS 2, Nav2, sensor fusion, GNSS/IMU, SITL/HITL and recorded-run replay; PX4/MAVLink source analysis

Engineering Experience

Ibrium Studios

09/2025 - 05/2026

Independent Embedded Systems Engineer | Production systems

- Architected distributed real-time control for a multi-axis motor system delivered for production use; implemented the CAN protocol, ARM Cortex-M node firmware, and central controller for synchronized motion.
- Designed multilayer motor-control PCBs in Altium and wrote bare-metal firmware for feedback, fault detection, and safe-stop behavior; owned board bring-up and hardware/software integration.
- Resolved an intermittent CAN-control latency fault using a logic analyzer to isolate a control-loop deadline overrun; moved scheduling to a fixed-rate timer interrupt and revalidated the motion sequence.
- Integrated SLAM registration with Unreal Engine models; delivered robotic controls under a fixed production deadline, documented operating procedures, and trained production operators.

Saddleback College Robotics

06/2021 - 06/2023

Co-Software Lead, Mars Rover Team | University Rover Challenge

- Co-led ROS 2 rover autonomy through field trials across two competition seasons; integrated Nav2 behavior trees, waypoint navigation, ArUco detection, and obstacle avoidance using stereo cameras, LiDAR, GPS, and IMU fusion.
- Connected autonomous and teleoperated control to STM32 through an asynchronous Rust driver using COBS/postcard messaging; integrated ODrive motor controllers and modeled CAN drive control in MATLAB/Simulink.
- Built React/TypeScript and Foxglove interfaces for telemetry, waypoint commands, and robot control, with Jetson H.264 video; coordinated subsystem integration through GitHub reviews.
- Built Gazebo/ros2_control and Isaac Sim environments, rosbag field-run replay, and a Docker test harness to exercise navigation changes without vehicle hardware.

AI-Assisted Aircraft Design

02/2026

Independent Software Engineer | Aircraft design and analysis

- Connected Python, NumPy, SciPy, OpenVSP, and CalculiX in a repeatable workflow from parametric geometry through aerodynamic, structural, mass, and aeroelastic analysis.
- Built an event-backed validation chain across geometry and analysis handoffs; retained artifacts, checked outputs against published references, and flagged uncertain results for engineering review.
- Screened hundreds of design variants, validated three finalists with higher-fidelity tools, and identified a trade-study design with approximately 24% lower structural mass than its baseline.

Shipyard Operations Software

03/2026

Independent Software Engineer | Shipyard operations software

- Built and deployed a shipyard-operations MVP, connecting parts, equipment, workers, and task dependencies in a PostgreSQL ontology through Rust and Python/FastAPI services and a React/TypeScript operator interface.
- Integrated OPC-UA telemetry and operational-data adapters; used OR-Tools CP-SAT to evaluate schedules against resource and task constraints, with AI-assisted analysis supporting planning decisions.
- Packaged the system for local deployment with Docker; implemented role-based access, event history, and tamper-evident audit records so operational changes and recommendations remained reviewable.

Selected Engineering Projects

Flight-Software Static Analysis (PX4)

2026

Self-directed systems-security project | Rust / PX4 v1.16

- Built a Rust static analyzer to study C/C++ flight-software architecture, with function discovery, direct-call extraction, and command-line lookup of PX4 MAVLink handlers and their callees.
- Implemented heuristic checks for missing authentication references, unchecked memory-access patterns, and navigation-state writes outside expected state-machine functions; produced findings for source review.
- Validated call-graph behavior with unit tests and PX4 function lookups; compared vulnerable and corrected fixtures to check rule behavior. Recorded validation included 29 passing tests and a successful release build.

Counter-UAS Electronic-Warfare Decision Engine

03/2026 - Present

Independent EW and maritime decision-support software | Simulation and replay

- Built a Haskell signal and decision pipeline that processes synthetic and SigMF replay IQ through FFT and CA-CFAR detection, extracts emitter features, and matches observations against a threat library.
- Implemented constrained EW response selection and plan comparison using explicit mission state, spectrum and resource checks, and confidence-based abstention; preserved plan-scoped decisions and replay evidence for debugging and evaluation.
- Used Nix to define reproducible build environments for offline-capable software; evaluated behavior in adversarial simulation and replay scenarios so changes could be reviewed against recorded inputs and decisions.

Additional Experience

Tata

12/2025 - 02/2026

Independent Field Researcher and Developer | Apple AR/LiDAR research partnership

Conducted AR/LiDAR research across field sites, traveling throughout the engagement to configure sensors and validate captures. Developed C software and repeatable data-collection protocols for consistent indoor scanning.

MedBWS

05/2020 - 06/2023

Automation Lead | MedBWS, hospital deployments

Built a computer-vision and Swift iOS workflow for pharmaceutical identification and disposal, deployed across hospitals in a HIPAA-compliant environment; integrated authentication, authorization, secure APIs, and audit logging for controlled-medication records.

Technical Skills

Firmware and hardware: C, bare-metal and RTOS firmware, ARM Cortex-M, STM32, CAN, UART, SPI, I2C, PWM; Altium PCB design, board bring-up, JTAG/SWD, oscilloscopes, logic analyzers

Flight software and autonomy: C++, Rust, Python; ROS 2, Nav2, behavior trees, sensor fusion, robot_localization, GNSS/IMU integration, ODrive motor control, PX4/MAVLink source analysis

Test and simulation: Gazebo, Isaac Sim, ros2_control, rosbag replay, Docker hardware-in-the-loop harness, MATLAB/Simulink, Nix reproducible builds, unit and fixture testing

Systems and interfaces: Linux, NVIDIA Jetson, Git, TCP/IP, UDP, ROS 2 DDS, message passing, COBS/postcard framing, H.264 video links

Other languages and tools: Haskell, Go, TypeScript, FastAPI, PostgreSQL, OR-Tools, OPC-UA, Swift

Coding assessment: CodeSignal Industry Coding Assessment, 600/600, August 2026

Education

Irvine Valley College | Computer Engineering coursework

All A grades in C++, Computer Organization and Assembly, MATLAB, Python, Calculus, Linear Algebra, and Differential Equations. Finalist, CLA Hacks 2023.