

Nick Armstrong

n2armstr@uwaterloo.ca | linkedin.com/in/nick-armstrong-eng | github.com/n-arms

EXPERIENCE

- Embedded Software Development Intern** | *Onsemi* Sept. 2025 - Dec. 2025
- Built a thread-safe SPI HAL using fine-grained locking and a deterministic job queue to arbitrate concurrent hardware access
 - Accelerated FPGA bring-up, analyzing Verilog & register maps to write bare-metal driver code, bypassing hardware defects
 - Architected a custom co-operatively scheduled RTOS, designing for production-ready multitasking to an ultra low power MCU using less than 3 KB of ROM
 - Led discussions and delivered presentations to drive the design of a hardware-agnostic non-volatile memory stack, establishing a unified API to support diverse storage technologies including EEPROM, OTPM, and ReRAM
 - Implemented a high-performance CRC library, optimized with lookup tables and assembly-level tuning, performing as few as 1.25 memory loads per byte of data
 - Researched the JEDEC SFDP spec to implement a flash controller with dynamic runtime detection of device capabilities
- Drone Software Project Manager** | *WARG, 90+ member student design team* Sept. 2024 – Present
- Managed a 9-member team developing software to power a distributed network of microcontrollers on a unified CAN bus, using a decentralized leader election algorithm to provide a plug-and-play experience for the drone's Flight Lighting System
 - Architected software to use embedded hardware acceleration techniques including DMA, interrupts, and hardware timers, to optimize performance
 - Debugged distributed software using GDB and network capture tools like logic analyzers and oscilloscopes
 - Led software development as project manager using project boards and PR-based workflows
- Software Engineering Intern** | *Endex* May 2026 – Aug. 2026
- Built financial agentic AI, backed by OpenAI
- Embedded Software Development Intern** | *Ford Motors Research & Development Centre* Jan. 2025 – Apr. 2025
- Built a fault-tolerant parser for a binary logging file format, bypassing corrupted payload information to ensure reliable testing while tracking down bugs in other teams' logging tools using Wireshark
 - Redesigned a concurrent logging library using state machines to ensure thread safety and clarity
 - Analyzed the architecture of a massive codebase to create specialized SELinux policy files and improve security
 - Developed an embedded C++ hardware abstraction layer to enable testing via gTest and gMock

PROJECTS

- Low Power Semantic Visual SLAM** | *Rust, Assembly, SIMD, ESP32* June 2026 – Present
- Develop handwritten image processing kernels in assembly, using SIMD and minimizing data dependencies to eliminate pipeline hazards and 4x throughput
 - Fine-tuned and quantized image embeddings model in TensorFlow, optimizing for high-throughput ESP-NN inference
 - Developed a PnP RANSAC pipeline for robust camera localization, rejecting dynamic outliers
- Closed Loop Stepper Motor ASIC** | *Verilog, TinyTapeout*
- Build a FSM-based SPI peripheral, using best practices such as cascading flip-flops to avoid metastability issues
 - Implement an in-hardware PI controller using accumulators and power-of-2 coefficients to eliminate unnecessary MAC units
 - Implement motion planning math using a FSM, saving 4-5x area over the naive MACs and divide units
- Robotic Path Planning Library** | *Trajectory Planning, Kalman Filtering, PID*
- Use RProp numerical optimization to generate optimal Hermite spline trajectories
 - Developed Kalman-filter based odometry and IMU-based sensor fusion to adaptively react to disturbances
- Point-of-Storage Telemetry Platform** | *C, STM32, GDB, Oscilloscope, DMM* Sept. 2024 – Dec. 2024
- Implemented a sensor driver using hardware timers and interrupts to minimize protocol overhead
 - Designed custom fault-tolerant embedded protocol using Hamming error correction and a reduced baud rate to enable robust communication over a 2-meter serial link
- Functional programming language compiler** | *Rust, LLVM* Dec. 2022 – Present
- Developed a to-LLVM compiler for a functional programming language using techniques including specialization, unboxing, and defunctionalization to maximize the language's performance

TECHNICAL SKILLS

Languages: C, Assembly, Verilog, C++, Rust, Zig, Python, Typescript, Go, Haskell, WGSL
Embedded: RTOS, DSP, Board Bring-up, FPGA/RTL debugging, ARM & RISC-V MCUs, Oscilloscopes & Logic Analyzers, DMMs, Altium, CAN/SPI/I2C/UART
Robotics: ROS, SLAM, State Estimation (EKF), Controls (PID, LQR), Trajectory Optimization (Splines, RRT)
Edge AI/ML: Model Architecture, SIMD/GPU acceleration, Computer Vision (CNNs, LSTMs), Quantization (QAT & PTQ)
Cloud: SQL DBs, REST APIs, ORMs, AWS (S3, EC2, ECS, Lambda, RDS, Athena, Kinesis, etc), Protobuf, Zod

EDUCATION

University of Waterloo | *Waterloo, ON* 2024 – Present
Bachelor of Applied Science in Computer Engineering — 3.98 GPA