Arav Jaiman

301-368-4077 | aravjaiman@outlook.com | GitHub | LinkedIn

Active Secret Security Clearance

Education

University of Washington — Seattle, WA

Bachelor of Science in Computer Science, Paul G. Allen School — *GPA: 3.94*

Relevant Coursework: Discrete Math; Linear Algebra; Computer Architecture and Hardware-Software Systems; Probability, Statistics & Machine Learning Foundations

Experience

Software Engineering Intern — Lockheed Martin — Colorado Springs, CO

Rotary and Mission Systems, CommandIQ Product Line — *Summer 2026 – Present*

- Built REST and gRPC APIs in Rust for 1-2 microservices to derive and serve protobuf data-models for the CommandIQ product line
- Developed and deployed services from scratch on Kubernetes using Helm across Unix-based environments, resolving configuration and deployment issues within the team's CI/CD pipeline
- Wrote unit tests for all code changes and performed integration testing of the core threat evaluation system against a custom in-house simulator, validating end-to-end behavior prior to release
- Extended a core threat evaluation microservice, consumed by 5 downstream services, collaborating with backend and systems engineers to integrate it into the deployed CommandIQ product

Embedded Software Engineer — UW Formula Motorsports — Seattle, WA

Battery Management System Firmware Development — *Sept. 2025 – Aug. 2026*

- Wrote low-level SPI drivers in C++ on an RP2040 MCU from datasheet specs to interface with an ADBMS6830 analog front-end and MCP2515 CAN controller, configuring and triggering onboard ADC conversions to read per-cell voltages and GPIO-connected thermistor circuits for temperature sensing
- Engineered real-time voltage, temperature, and current monitoring across 7 cells using FreeRTOS task scheduling to run concurrent hardware-polling and fault-detection tasks without race conditions over shared SPI/CAN peripherals, preventing thermal runaway and over/under-voltage
- Developed passive cell balancing algorithm to maintain voltage deltas within 10mV, extending battery lifespan
- Validated firmware through hardware-in-the-loop (HIL) testing on a bench rig, cross-checking voltage and temperature readings against a voltmeter and ground-truth sensors, verifying CAN communication, and injecting fault conditions to confirm fault detection logic

AI Engineering Intern — Nēdl Labs — Remote

Health Policy Data Extraction — *Jun. 2025 – Aug. 2025*

- Architected a modular pipeline in Python to transform unstructured health insurance policy text into structured JSON outputs using large language models (LLMs), saving $10K in costs
- Integrated LangChain and Pydantic to enforce JSON schema validation, reducing malformed outputs by 100%
- Designed a two-stage compression pipeline that prompted the LLM to distill lengthy policy documents to their key information before running the main extraction prompt, overcoming context-length limits that previously prevented smaller LLMs from processing files up to 50+ pages

Research Intern — Texas A&M University — Remote

AI-Powered Pathfinding Algorithm — *May. 2024 – Oct. 2024*

- Researched and implemented a novel extraterrestrial rover localization method in Python that replaces satellite data with a Hugging Face monocular depth-estimation model and an onboard camera, retaining 100% pathfinding accuracy
- Adapted custom cost calculation functions from satellite-based pathfinding algorithm and applied A* search to point cloud data generated with Open3D, determining traversable and untraversable zones with 95% accuracy
- Accelerated image-to-map conversion by running PyTorch-based depth inference on a Jetson Orin Nano GPU (CUDA)

Projects

Autonomous 1:16 Scale Car — *In Progress*

- Building an autonomous driving platform on an RC car chassis with an NVIDIA Jetson Nano, using ROS2 Humble as the core software framework; containerized the full development environment in a custom Docker image for reproducible builds across machines
- Wrote an I2C driver from scratch for the PCA9685 PWM controller IC on embedded Linux (Jetson Nano), generating precise 50Hz PWM signals to drive a steering servo and throttle ESC, exposed as a ROS2 node for integration with the future autonomy stack
- Built two independent teleoperation systems for manual driving and data collection: a TCP-based client-server system streaming live camera feed from the Jetson to a laptop client at 30Hz while relaying driving commands back, and a gamepad-based system built on ROS2's `joy` driver package
- Currently collecting camera-based driving data to train a perception-driven behavior cloning model for autonomous navigation

Custom Overwatch Controller

- Transformed an Adventure Force Sentry X2 toy into a functional Overwatch controller using Arduino Nano, custom C++ firmware for real-time hardware input handling, paired with a custom Python driver for PC-side keyboard/mouse emulation
- Debugged a severe input-latency bug by instrumenting firmware and PC-side driver with timers, isolating a 500 msg/s to 9.9 msg/s processing bottleneck to a Python emulation library taking 100.66ms per call
- Resolved latency bug by switching to a system-level input library (1.19ms per call, 84x faster), achieving ¡3ms end-to-end latency from button press to in-game action
- Developed user-friendly aiming mechanism by integrating IMU angular velocity to derive precise orientation angles and mapping angle changes to pixel movement, enabling full 360° in-game turning through iterative calibration

Technical Skills

Languages: Python, C, C++, Rust, Java, JavaScript
Embedded & Systems: SPI, I2C, GPIO, ADC, PWM, FreeRTOS, Embedded Linux, Real-Time Systems, Low-Level Driver Development
Frameworks & Libraries: React, Node.js, Next.js, PyTorch, Hugging Face Transformers, LangChain, ROS2
Tools & Systems: Claude Code, Git, GitHub, GitLab, Kubernetes, Helm, Docker, REST, gRPC, Arduino, RP2040, Jetson Nano