

TRISTAN ALDERSON

Work Authorization: Dual US & Canadian Citizen

Surrey, BC • 604-828-1437

TALDERSON42@GMAIL.COM • linkedin.com/in/tristanalderson • tristanalderson.com

EDUCATION

Bachelor of Applied Science in Electrical and Computer Engineering
Queen's University • *Dean's Scholar*

Kingston, ON, Canada
05/2026

HARDWARE & EMBEDDED FOCUS

Hardware-aware embedded engineer focused on bringing up, debugging, and validating real systems where **PCBs, power rails, STM32/ESP32** firmware, **CAN** control networks, telemetry, and **Linux/FPGA** interfaces meet. Comfortable tracing failures from schematic and bench measurements through firmware, boot behavior, networking, and physical system behavior.

WORK EXPERIENCE

Avionics Lead

Queen's Rocket Engineering Team (QRET)

05/2025 – Present

Kingston, ON, Canada

- Lead a 10-person avionics team building hybrid-rocket control hardware across **STM32** valve nodes, **ESP32** telemetry, **CAN** networking, 24V power distribution, pressure sensing, GPS recovery, and launch-ops ground-station integration.
- Designed and brought up custom **STM32/ESP32** control and sensor PCBs with solenoid actuation, switched transducer power, CAN reporting, hardware fail-safes, KiCad schematics/layout, JLCPCB assembly, bench validation, and manual rework.
- Engineered a **24V avionics power module** with ORed ground/battery inputs, 3.3V/5V buck rails, PMOSFET reverse-polarity protection, and zener Vgs clamping; diagnosed an overheating shunt path during current-limited first power-on and reworked the board.
- Designed **AimNetwork**, a fixed-format **CAN** protocol with 64-bit payloads, strict node addressing, heartbeat health checks, time synchronization, bounded queues, and WiFi/LoRa forwarding so telemetry and commands remain predictable under fault conditions.

Embedded Software Engineer Intern

Evertz Microsystems

05/2024 – 08/2025

Burlington, ON, Canada

- Built **C++ Linux** control services for a constrained 1RU hardware platform, coordinating configuration, telemetry, status, and device control across **six ARM Linux nodes** using gRPC, REST APIs, and Redis-backed state.
- Developed ARM Linux monitor/application software for hardware-facing status paths including **SFP/QSFP** state, **NTP/PTP** synchronization, front-panel devices, remote configuration, event logging, and reboot-safe persistent device data.
- Debugged 1G/10G/100G **SFP/QSFP** links, multicast forwarding, TLS/OpenSSL failures, and Marvell switch behavior using vendor tooling, **GDB**, logs, board traces, and custom ACL rules.

AI Engineer & Avionics Developer

Queen's Rocket Engineering Team (QRET)

10/2023 – 08/2024

Kingston, ON, Canada

- Designed and flew a custom **SAM-M10Q GPS** daughter board with I2C breakout, physical separation from comms electronics, and enlarged ground plane to reduce RF interference and maintain GPS lock during flight/recovery.
- Reworked a handheld **ground-station receiver** for live telemetry decode and GPS recovery tracking, including a simplified 5V power path and 330mF supercapacitor backup to preserve GPS state through brief power loss.
- Secured **2nd Place at Launch Canada** by implementing YOLOv8 rocket tracking with dual-axis PID control and Matlab simulation; owned 18650 battery, altimeter, and GPS configuration during launch operations.

Programmer and Assembler

Radius Security

05/2023 – 09/2023

Richmond, BC, Canada

- Manufactured and tested remote-guarding surveillance units; earned internal recognition for wiring quality and contributed to **Lean** process changes that improved production efficiency by **20%**.

PROJECTS

LiteLM: FPGA-backed LLM Inference on ARM Linux

Queen's University

08/2025 – 03/2026

Kingston, ON, Canada

- Built a GPT-2 inference accelerator on an **AMD Xilinx Kria KV260**, using FPGA fabric for matrix operations and **ARM Cortex-A53 Linux** firmware for control, precision-sensitive operations, and system orchestration.
- Implemented the hardware/software boundary around **AXI4 DMA**, shared DDR memory maps, device-tree configuration, U-Boot boot files, overlay loading, and low-level C/C++ register and buffer interfaces between Linux and programmable logic.
- Debugged DMA-backed accelerator stability by tracing **CMA** sizing, **udmabuf** allocation, boot-time service memory use, device-tree configuration, and Linux-side transfer behavior.
- Earned **3rd Place for Top Computer Engineering Project** at the Queen's ECE Showcase.

SKILLS

Hardware & bring-up: KiCad, LTspice, schematic capture, PCB layout, JLCPCB assembly, manual rework, oscilloscopes, logic analyzers, current-limited supplies, board-level fault isolation

Embedded platforms: STM32, ESP32, Zynq UltraScale+, Kria KV260, ARM Cortex-A53, ARM Linux, x86 COM Express, Marvell switch ASICs

Power, RF & interfaces: DC-DC bucks, PMOSFET protection, zener clamps, 18650 packs, GPS/RF layout, ground planes, CAN, I2C, SPI, UART, LoRa, WiFi, SFP/QSFP

Firmware & software: C, C++, Python, Bash, Matlab, Verilog/SystemVerilog, STM32 HAL, ESP-IDF, fixed queues, heartbeat logic, telemetry protocols

Linux & debug: Embedded Linux, Buildroot, U-Boot, device trees, AXI4, AXI DMA, DDR memory maps, CMA, udmabuf, GDB, Git, logs, boot/service debugging