Zain Naqvi

437-259-9006 | zainnaqvi.200621@gmail.com | Linkedin | GitHub | Portfolio

Education

McMaster University Hamilton, ON
Bachelors in Computer Engineering CO-OP - GPA: 3.82 *Sept. 2024 – May 2029*

Experience

Lead - Electrical Engineering Sub-Team September 2025 - Present
McMaster RoboSub *Hamilton, Ontario*

- Built a hardware-in-the-loop DVL emulator on a Teensy 4.1 using a custom C++ (OOP) library that replicates the Nortek Nucleus 1000's Ethernet interface for testing
- Designing the AUV's acoustics board in Altium and building DIY hydrophones from piezoelectric sensors to triangulate underwater acoustic signals for localization
- As Team Lead, rebuilt the team and secured $6,000 in Faculty of Engineering funding
- Developed a 12S - 5V at 3A Buck Converter on Altium (Schematic and PCB)

Projects

QEMU EDU Linux Kernel PCI Driver | *C, Linux Kernel, PCI Subsystem, MMIO, MSI, DMA, QEMU*

- Built a Linux kernel PCI driver for QEMU's EDU virtual device inside a QEMU guest. Implemented the pci_driver ID-table match and probe/remove lifecycle, mapped BAR 0 into kernel virtual address space with ioremap, and exposed the device to userspace as a misc character device (/dev/edu) with per-device state allocated by devm_kzalloc and recovered through container_of
- Converted the busy-poll read path to MSI interrupt-driven I/O using pci_alloc_irq_vectors and a struct completion
- Added 4 KB coherent DMA behind an ioctl interface with a fixed-width shared ABI header and a 28-bit DMA mask, then benchmarked programmed I/O against DMA across 11 transfer sizes using kernel-side ktime timestamps. Traced the flat 100 ms DMA floor to a hardcoded timer in QEMU's device model rather than reporting it as real transfer cost

Bare-Metal RTOS Kernel | *C, ARM Thumb Assembly, Cortex-M4*

- Built a preemptive RTOS from scratch on an ARM Cortex-M4 MSP432E401Y MCU with no HAL or vendor libraries. Context switcher written in ARM Thumb assembly using SysTick and PendSV exceptions, with isolated Process Stack Pointers and manually forged 16-word initial stack frames per task
- Implemented binary semaphores with priority-ordered wait queues and mutexes with priority inheritance from scratch. Debugged a real priority inversion scenario and a PendSV ordering race condition on live hardware using the Keil CMSIS-DAP debugger
- Wrote a first-fit heap allocator with block splitting and stack overflow detection

Multi-Node CAN Bus Network & Intrusion Detection System | *C, ARM Cortex-M4, CAN Bus, Embedded Security*

- Built a bare-metal multi-ECU automotive CAN network on an ARM Cortex-M4 (MSP432E401Y), driving both on-chip CAN controllers as independent nodes (Engine ECU and Body Control Module) across a twisted-pair bus with external SN65HVD230 transceivers. Configured every CAN register from scratch with manual bit-timing for 500 kbps
- Made a firmware-level Intrusion Detection System with five real-time detectors spanning over-range, too-fast, missing, in-range spoofing via rate-of-change, and rogue message IDs. Validated each detector on live hardware through an interactive UART fault-injection framework and a Python telemetry dashboard
- Quantified real-time performance with the Cortex-M4 DWT cycle counter: single-frame detection latency under 8 microseconds, a 2.8-microsecond worst-case interrupt handler, near-zero false positives across boundary traffic, and 98% measured CPU headroom, proving a deterministic, low-overhead pipeline

Technical Skills

Languages: C, C++, Assembly, Python, Verilog
Linux and Kernel: Linux kernel module development, PCI subsystem, MMIO register access, MSI interrupt handling, DMA API, ioctl interfaces, character devices, QEMU
Embedded Systems and Hardware Design: Bare-metal RTOS development, ARM Cortex-M4 architecture (exception model, NVIC, SysTick, PendSV), Embedded Systems (Arduino, ESP32, Pi), Keil MDK, SWD debugging, CMSIS-DAP, Verilog, PSpice, Schematic Design, PCB Design (Altium), Sensor Integration, I2C/SPI/UART/CAN Communication