JUSTIN NG

+1 (778) 384-1911, Vancouver, BC

ngjustin2002@gmail.com linkedin.com/in/ngjstn

github.com/ngjstn.com ngjstn.com

EXPERIENCE

Firmware Development Engineer (Co-op)
Jan. 2023 - Aug. 2023

Solidigm (formerly Intel NSG) *Vancouver, Canada*

- Implemented Flash Translation Layer (FTL) Direct Memory Access (DMA) functionality for fetching SGL and PRP host address descriptors and prepped data payloads in controller memory to/from NAND
- Designed hierarchical FSMs in C++ for managing admin and I/O NVMe command contexts for DMA control paths and error handling for functional unit hardware queues
- Developed event processing firmware to handle and process functional unit interrupts across multiple cores

Firmware Test Engineer (Co-op)
May 2022 - Dec. 2022

Motorola Solutions, Avigilon Division *Vancouver, Canada*

- Developed pytest and Selenium-based frameworks to validate the firmware for security network cameras
- Implemented Python scripts to automate functional testing of network devices involving digital imaging, audio/video coding, and user interface scraping
- Constructed an IPv4 DHCP server for an isolated subnet in Linux for test and development of cameras
- Debugged and modified firmware in C++ for use on camera devices under test

Firmware & Electrical Team Lead
Sep. 2021 - Aug. 2023

UBC Mars Colony Design Team, Sabatier Reactor *Vancouver, Canada*

- Managed sub-team members and operations involving firmware, software, and electrical development
- Developed the control system firmware running FreeRTOS on an STM32 microcontroller to manage mass-flow controllers, thermocouples, pressure transducers, and heat tape relays
- Programmed embedded I2C and SPI device drivers for peripheral device sensors in C and C++
- Implemented PID and phase-angle TRIAC control for limiting AC voltages to manage thermal reactions

PROJECTS

Motorola 68k Soft-Core Processor
- Interfaced the 68k microprocessor to add memory and hardware logic modules with the DE1-SoC FPGA
- Developed embedded drivers in C to communicate with off-board I2C EEPROM and SPI flash memory ICs
- Implemented the RTL designs for DRAM and set-associative cache controllers in Verilog and SystemVerilog

OS/161 Kernel Development
- Developed kernel features to support running on 32-bit MIPS systems, tested with System/161 simulation
- Implemented synchronization primitives, systems calls, virtual memory, and file system support in C

FPGA RISC Processor
- Developed the RTL designs of a simple CPU, implementing SRAM and ALU data paths in Verilog
- Interfaced with the DE1-SoC FPGA development board peripherals to read and display register contents
- Simulated test benches in ModelSim to run waveform analyses for hardware module debugging

EDUCATION

University of British Columbia
Expected Graduation 2025

Bachelors of Applied Science, Computer Engineering *Vancouver, Canada*

Coursework: Microcomputer Systems, Operating Systems, Digital Logic Design, Computer Networking

SKILLS

Languages	C, C++, Verilog/SystemVerilog, Python
Software	GNU Debugger (GDB), Git, STM32CubeIDE, ModelSim, Quartus, Altium Designer
Standards	I2C, SPI, CANBUS