Justin Ng

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

EDUCATION

University of British Columbia Expected Graduation 2025
Bachelor of Applied Science - Computer Engineering
Coursework: Operating Systems, Digital Logic Design, Microcomputer Systems, Networking, Data Structures & Algorithms

SKILLS

Languages: C, C++, Python, Verilog, SystemVerilog
Software Tools: GDB, Git, STM32CubeIDE, ModelSim, Quartus, Altium Designer, Android Studio

TECHNICAL EXPERIENCE

Firmware Development Engineer (Co-op) Jan. 2023 - Aug. 2023
Solidigm - SK hynix NAND Product Solutions Canada Ltd.

- 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 finite-state machines in C++ for managing Admin and I/O NVMe command contexts for DMA control paths and error handling flows for functional unit hardware queues
- Developed an event processing framework to handle firmware and functional unit interrupts across multiple cores

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

- Developed pytest and Selenium-based test frameworks to validate the SoC firmware, web UIs, manufacturing workflows, and VMS client/server applications for security IP camera products
- Implemented Python scripts to automate functional testing of network devices involving digital imaging and audio/video coding, as well as manual tests involving ethernet switches
- Constructed an IPv4/6 DHCP server for an isolated network in a Linux environment for test and development
- Debugged and modified firmware in C++ for use on camera devices under test

Firmware & Electrical Team Lead Sept. 2021 - Aug. 2023
UBC Mars Colony (Engineering Design Team) Sabatier Reactor Project

- Managed sub-team members and operations involving firmware, software, electrical development
- Developed the control system firmware with FreeRTOS on an Arduino ATmega2560 and an STM32 microcontroller to manage mass-flow controller valves, thermocouples, pressure transducers, and heat tape relays
- Wrote embedded I2C and SPI drivers for peripheral device sensors in C and C++
- Implemented PID and phase-angle triac control for power limiting applied to AC voltages to manage reaction heat

PROJECTS

FPGA Reduced Instruction Set Computer (RISC) Machine

- Developed a Turing complete CPU using finite state machine controllers, SRAM, and ALU datapaths in Verilog
- Simulated test bench modules using ModelSim, analyzing waveform diagrams of multi-bit signals for debugging
- Interfaced with a DE1-SoC development board using peripherals to read and display register contents, compiling and ensuring synthesizable hardware using Quartus

OS/161 Operating System Functionality

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

Web-Enabled Robot Alarm Clock Chaser

- Constructed an alarm robot using a Raspberry Pi 3 that drives away from you and forces you to get out of bed
- Developed an Express backend web service handling REST post requests with the Raspberry Pi, setting alarm audio and time and enabling onboard LCD, motor driver, imaging, and ultrasonic modules with Python multiprocessing