JOVAN KOLEDIN

206.504.4184 | jovank@uw.edu | jovankoledin.github.io

SKILLS

Software & Firmware: Python, Matlab, C, C++, RTOS (SafeRTOS, FreeRTOS), ARM Cortex M
Communication Protocols & DSP: SPI, I2C, CAN, TCP/IP, UDP, BLE, QPSK, OFDM, FFT, CFAR
HDL & Hardware: SystemVerilog, Verilog, VHDL, Spectrum Analyzers, Logic Analyzers, Oscilloscopes, Multimeters, FPGAs
Tools & Build Systems: Bash, UART debugging, CMake, Make, SCons, CI/CD, OTA

PROFESSIONAL EXPERIENCE

Software Engineer Intern | **Tesla** | Palo Alto, CA | *Sept 2025 – Dec 2025*

- Designed and implemented a C++ thermal runaway detection algorithm using DSP low pass filtering on battery cell voltages, achieving a reduction in false positives from 10% to 0% across a fleet of one million+ Powerwalls.
- Architected a C-based cell physics simulation engine using a 2RC electrical model, enabling robust software-in-the-loop testing and validation of battery management algorithms without physical hardware.
- Developed and managed memory-safe C tasks in a SafeRTOS environment to execute high-frequency cell balancing operations on embedded ARM microcontrollers.

RF Firmware & Electrical Engineer | **Lockheed Martin** | Bothell, WA | *Aug 2023 – Aug 2025*

- Developed two low-level C Linux drivers for an FTDI USB-to-SPI RF signal generator and a custom PRBS board, directly programming TI RF chip registers (PLLs, mixers, attenuators) and validating all SPI transactions and RF outputs with logic and spectrum analyzers, saving $20K+ in recurring tool costs.
- Built real-time radar DSP engines in C++, implementing FFT and CA-CFAR algorithms to detect and classify RF targets and generating dual-antenna phase-modulated IQ radar data over UDP with configurable SNR, RCS, and Doppler parameters.
- Designed and verified FPGA communication IP in VHDL, creating an SPI interface for Xilinx FPGAs to control DACs and power amplifier modules.
- Produced hardware schematics and interconnect diagrams for RF optical signal generator systems, supporting board bring-up and system integration.

ACADEMIC RESEARCH

Graduate Research Assistant | **PN Computer Engineering Lab, University of Washington** | *Jan 2025 – Sept 2025*

- Developed a custom LLVM C++ compiler pass to translate LLVM Intermediate Representation (IR) into a custom Data Flow Graph (DFG) ISA for a novel reconfigurable dataflow hardware accelerator.
- Engineered a Python-based data-flow simulator with a Tkinter GUI to visualize DFG execution steps for graph based processors, enabling cycle-accurate verification of memory interactions and program flow.
- Designed and verified synthesizable SystemVerilog for reconfigurable vector and scalar processing elements executing variable-precision arithmetic, memory, and control flow operations.

PROJECTS

Light Phone – BLE Notification LED Display

- Programmed a C++ BLE driver stack for an ESP32 to act as a BLE peripheral and central device to enable subscription to Apple Notifications on an iPhone. The device ran two FreeRTOS tasks to update an LED matrix based on notifications.
- Created a CI/CD pipeline with GitHub Actions to automatically compile and deploy new firmware over Wi-Fi (OTA).

Raspberry Pi 5 LED Driver

- Wrote a low-level Linux C driver to control WS2812b LEDs, bit-banging the SPI MOSI GPIO pin to generate a precise, high-frequency signal required by the LED controller IC, restoring functionality broken by a hardware revision in the Pi 5.

Crossy Road ASIC Design

- Created a fully playable version of *Crossy Road* on an ASIC using Verilog, leveraging the SkyWater 130 nm toolchain and Yosys/OpenLane for GDS generation, with a scheduled tape-out in early 2026.
- Designed the core game logic, VGA monitor communication, and input handling, all in Verilog. Optimized original design with 5,000+ logic elements down to ~1,150 without removing any game features, verified on FPGA.

EDUCATION

University of Washington, Seattle | M.S. in Electrical and Computer Engineering | *Sept 2024 – June 2026*
Focus: Digital Signal Processing and Computer Architecture
Gonzaga University | B.S. in Electrical Engineering | *Aug 2019 – May 2023*
Minor: Computer Science