

Po (Eric) Peng

Seattle, WA | (206)-234-2928 | ericpp.peng@gmail.com | [linkedin.com/in/po-peng](https://www.linkedin.com/in/po-peng) | github.com/ericpp-peng | [Website](#)

Education

University of Washington GPA 3.75 / 4	09/2025 – 06/2027
M.S. Electrical and Computer Engineering	Seattle, WA
• Coursework: High-Performance Computer Architecture, The Hardware/Software Interface, Computer Vision	
National Taiwan University of Science and Technology GPA 3.92 / 4.3	09/2018 – 08/2020
M.S. Electrical Engineering (Mobile Communication Specialization)	Taipei, Taiwan
Chang Gung University GPA 3.7 / 4.0	09/2014 – 06/2018
B.S. Electrical Engineering, Division of IC Design	Taoyuan, Taiwan

Skills

- **Programming Languages:** C/C++, Python, Java, Shell Scripts, JavaScript, Verilog
- **Computer Architecture & Systems:** RISC-V, Manycore Architecture, Memory Hierarchy, Network-on-Chip (NoC)
- **Embedded Systems:** Linux kernel, Zephyr, MCU, Bootloader, ESP32
- **Protocols:** TCP/IP, I2C, UART, LoRa, Ethernet, Zigbee, Modbus, CAN bus
- **Software Development:** GitLab CI/CD, Docker, Jira, Git, SQLite, Makefile, SDLC

Work Experience

University of Washington, Sensors, Energy, and Automation Laboratory (SEAL)	01/2026 – now
Embedded Software Developer	Seattle, WA
<i>Designed embedded firmware and sensing systems for an autonomous gas leak detection platform in industrial environments</i>	
• Architected an embedded edge-AI sensing system on ESP32, integrating on-node signal processing, ML inference, and LoRa communication, achieving 91% leak detection accuracy in industrial deployments	
Moxa - Global Leader in Industrial Connectivity (No. 1 in Serial Communication, Top 3 in Industrial Ethernet)	06/2021 – 10/2024
Embedded Software Engineer - R&D	Taipei, Taiwan
<i>Designed embedded Linux and bare-metal firmware, delivering robust, user-friendly, world-class industrial connectivity solutions</i>	
Protocol Gateways (Linux) - Achieved USD 3M/year revenue with +10% YoY growth	
• Led modularization of the IEC 60870-5-101/104 protocol stack for MGate 5192 , significantly reducing integration time for new products by over 50% through close collaboration with UI/UX, PM, and SQA teams	
• Implemented a proprietary CAN protocol module from scratch, covering main communication, backend infrastructure, data exchange, diagnostics features, and all related peripheral software modules	
• Improved the RESTful library for MGate 5000 series via IPC-based design, reducing API development time by 20%	
• Designed the SD card backup module for MGate 5000 series, independently resolving issues via Linux kernel analysis	
Media Converters (MCUs)	
• Led the full-cycle software development of Ethernet-to-fiber Converter , coordinating with cross-functional HW, PM, and SQA teams from project kickoff to successful market launch	
• Resolved communication issues for Japanese clients on CAN-to-fiber Converter by tracing MCU code with specifications	

Projects

High-Performance Parallel Computing on RISC-V Manycore Architecture	01/2026 – now
• Optimized parallel C++ kernels on a 2000-core RISC-V manycore system, achieving significant speedups through architecture-aware parallelization and latency hiding	
• Managed explicit data movement and synchronization in a software-managed memory hierarchy without cache coherence	
• Profiled and analyzed bottlenecks using cycle-accurate simulation, identifying compute, memory, and NoC limitations	
RTOS Implementation	09/2018 – 01/2019
• Modified the μ C/OS-II kernel scheduling to implement and evaluate various scheduling algorithms	

Extracurricular Activities

University of Washington	09/2025 – now
FIRST Robotics Mentor	Seattle, WA
• Mentored students in electronics, sensors, wiring, and software framework design, emphasizing PID control systems, Java programming, and computer vision	