

# Erin Hu

[e25hu@uwaterloo.ca](mailto:e25hu@uwaterloo.ca) | [github.com/erinjhu](https://github.com/erinjhu) | [linkedin.com/in/hu-erin](https://linkedin.com/in/hu-erin) | [erinhu.ca](https://erinhu.ca)

## Education

---

**University of Waterloo** - BAsC, Computer Engineering

2024 – 2029

**Relevant Courses:** Data Structures and Algorithms (C++), Computer Architecture (RISC-V), Digital Circuits (VHDL)

## Skills

---

**Programming:** C, C++, Python, Assembly (RISC-V)

**Tools:** Git, GitHub, Docker, Linux, WSL, ROS2, FreeRTOS, PuTTY, CMake, GTest, Oscilloscopes, AutoCAD

**Hardware:** ARM Cortex (STM32, RM46L852), Arduino, Raspberry Pi, UART, ADC, GPIO, PWM, ICs, I2C, Soldering, Sensors

**Interests:** Safety-Critical Embedded Systems, Drivers, Microcontrollers, Firmware, Robotics, DO-178C Certification

## Experience

---

**Embedded Software Engineer Intern** – *Wind River Systems*

Jan 2026 – Apr 2026 | Ottawa, ON

- Restructured thread priority and event signaling in **C** to avoid race condition in virtual machine manager
- Ensured regulatory **DO-178C** compliance standards for **safety-critical** airborne systems by documenting architecture specifications for 16+ **Helix Virtualization Platform hypervisor** and **VxWorks** modules in Polarion
- Resolved 18 customer-reported discrepancies from Jira between the VxWorks Board Support Package (**BSP**) integration guide documentation for PowerPC-based hardware by tracing kernel logic and APIs in Bitbucket source code
- Performed root cause analysis using assertions, stubs, and execution logs for **automated unit tests**, ensuring reliability

**Firmware Developer** – *UW Orbital*

May 2025 – Dec 2025 | Waterloo, ON

- Programmed modular **INA230** sensor driver in **C** to measure shunt/bus voltage and current for **RM46L852 MCU**
- Developed 14 **unit tests** using **C++** and Google Test (**GTest**); added and **validated** mock **I2C** register data bytes
- Developed a module with C and **FreeRTOS** to parse binary files of sun position data from NASA JPL's Horizons API
- Implemented a deterministic, statically-allocated thermal management task for **LM75BD** sensor telemetry data

**Autonomous Robotics Software Developer** – *WATonomous*

Nov 2024 – Apr 2025 | Waterloo, ON

- Programmed proportional-integral-derivative (**PID**) **control loop** for navigation of autonomous robot in **Gazebo**
- Programmed publisher/subscriber scripts in **C++** and **ROS2** inside **Docker** containers in a **Linux (WSL)** environment
- Implemented **real-time** emergency braking for autonomous racecar in F1Tenth Gym simulator; used **LiDAR** distance and speed odometry data to calculate time to collision and control vehicle linear velocity

**Systems Support Analyst Intern** – *Ministry of the Attorney General*

May 2025 – Aug 2025 | Toronto, ON

- Resolved and documented 50+ technical issues; coordinated client meetings using Remote Desktop and Teams
- Executed 120 organization-wide data backups with Backup Exec by controlling server data retention and deletion

**Transportation Infrastructure Design Intern** – *Stantec*

Jul 2024 – Aug 2024 | Markham, ON

- Collaborated with civil engineers on provincial transportation infrastructure for Metrolinx GO Transit LRT project
- Updated **AutoCAD** design specifications to represent the Bowmanville Rail Expansion in Oshawa, Ontario
- Revised linework and layers for curb and gutter removals to accurately represent the proposed construction

## Projects

---

**Voltage and Temperature Monitor** | C, FreeRTOS, STM32

[GitHub](#)

- Designed a **pre-emptive, multi-tasking, real-time** system using **C**, **FreeRTOS**, and **STM32** ARM Cortex-M4 MCU
- Implemented a mutex between voltage (**ADC sampling**) and status LED (**GPIO** with **PWM**) tasks for thread-safety
- Programmed modular **BMP180 I2C** temperature **sensor driver** for **calibration** and register bit manipulation
- Minimized latency using inter-task communication between button **ISR** and **UART queue** handler

**Boundary-Detecting Robot** | Python, Raspberry Pi, ICs

- Programmed **GPIO** peripherals in **Python** on **Raspberry Pi** to autonomously avoid environmental boundaries
- Controlled DC **motors** with L293D IC for responsive actuator feedback loops with photoresistors **sensors**

**Security Alarm System** | C++, Arduino

- Programmed an **Arduino** in **C++** to enable LED blinking and buzzer sounds until the correct passcode is inputted
- Applied Waterfall SDLC and documented requirements, design, and implementation in project proposal