Darren Liu

647-468-3499 d58liu@uwaterloo.ca Projects Linkedin Github

Technical Skills

C, C++, Python, RTOS, GDB, STM32, RPi, Git, Altium, KiCad, Soldering, Oscilloscopes, DMM's, ESP32, ESP-IDF, Java, HTML/CSS, JavaScript, Git, CAD

Education

University of Waterloo

Candidate for BASc in Electrical Engineering, GPA: 3.97 — *Waterloo, ON*

Experience

Firmware Lead — **Sep 2024 – Present**

Midnight Sun Solar Car Team — *Waterloo, ON*

- Overseeing bring-up of the entire firmware-electrical system, featuring a multi-board **(STM32) CAN** network. Writing firmware and on-device tests for each component, including motor controls, current sensing, and LCD displays.
- Implemented embedded firmare using **C** and **RTOS** for the car's front and rear controller boards, using state machines to manage peripherals, sending & recieving data through inter-board **CAN** communication.
- Designed and implemented a custom, on-chip file system with block-based allocation, HAL memory access, nested directories, and path resolution, enabling persistent storage of config data to be used during system boot.
- Developed a **Python** telemetry client that retrieves and decodes **CAN** datagrams through RF, streaming real-time vehicle data to an InfluxDB + Grafana **Docker** stack, enabling live performance visualization and debugging.

Embedded Software Developer — **Sep 2025 – Dec 2025**

Indie Semiconductor — *Markham, ON*

- Designed an end-to-end embedded communication debugging pipeline; implemented firmware APIs in **C** to capture and log raw **MCU-host** byte streams over **UART**.
- Built a downstream decoding **Python** tool to extract 100+ unique struct definitions, generating JSON schemas to translate byte streams to readable data, improving error analysis and enabling offline decoding.
- Implemented unit tests using the **Unity** framework, ensuring full code coverage of companion CPU initialization modules. Additionally, restored deprecated on-device, dual-core unit tests and resolved associated build issues.
- Designed and implemented a GUI tool for editing and exporting JSON configuration files in text, binary, or C code format, with built-in error checking and comment parsing, simplifying custom config generation for overseas customers
- Streamlined the firmware build pipeline by implementing automatic exclusion of deprecated files, eliminating manual cleanup and reducing developer build failures.

Mechatronics Engineering Intern — **Jan 2025 – Apr 2025**

Kraken Sense (Triton Genomics) — *Oakville, ON*

- Architected embedded and control software for an automated industrial assembly line via a multi-board **RS-485** network; implemented command sequencing on a **Raspberry Pi** master in **Python** to manage STM32 nodes
- Developed **STM32** firmware to forward **USART** control packets to a 4-axis robotic arm, control motors, linear actuators, and sensors, enabling 0.1mm polymer deposition accuracy, and increasing manufacturing speeds by 15x.
- Serviced clients on-site at **MIT Lincoln Labs**, installing, repairing, and testing an automated DNA sampling device
- Designed and milled a driver **PCB** for an optical fluid sensor using **Altium**, increasing fluid priming consistency by 25x

Projects

Custom Hardware + Firmware Interface for Raspberry Pi CM4 | *C, Linux, Python, Make, KiCad*

- Implemented full software/hardware stack for a social robotics project, using audio, and sensors to interact with users.
- Wrote **C** firmware libraries for **I2C, I2S, PWM, and GPIO**, allowing for **multithreaded** sensor drivers, servos, and battery monitoring firmware; compiled with **Make** and run through **Python** applications on boot.
- Designed a PCB to interface with the RPi CM4, featuring two isolated buck converters, current sensing, I2C peripheral and I2S audio busses, and an impedance controlled camera connector.

Differential Swerve Drivetrain | *C++, ESP32, KiCad, Inventor*

- Designed a drivetrain with 3 differential swerve modules, driven by an **ESP32-based PCB** featuring phase-enable H-Bridge motor control, encoder inputs, and a two-stage power regulator designed in **KiCad**
- Developed a PID-based motor driver in C++, achieving **±1%** accuracy in speed and position control. Higher level code allows for simultaneous control of 3 modules, using Bluetooth to control modules and a web server to log test data

Custom Mini Drone | *C, RTOS, ESP32, KiCad, Inventor*

- Built an **ESP32-based** drone with a low-profile PCB supporting I2C communication to an IMU and four motor drivers
- Developed quadcopter firmware using **RTOS** tasks for IMU input, PWM motor control, PIDF-based flight control, and global data logging. Hosted a web server to recieve inputs, display telemetry data, and 3D flight visualization