# Darren Liu

d58liu@uwaterloo.ca Projects Linkedin Github

## Education

**University of Waterloo** **2024 – Present**

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

## Technical Skills

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

## Experience

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

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

- Developed firmware for **STM32** commands in **C** to forward **USART** packets for controlling a robotic arm, stepper motors, sensors, and actuators; implemented **Python** sequences to run commands on a multi-board **RS485** bus architecture with **STM32** nodes managed by a **Raspberry Pi**.
- Designed and constructed an industrial assembly line to automate the manufacturing of 3000+ microfluidic sensors, depositing polymers to a 0.1mm tolerance, using **Inventor** to design a two-stage press that interfaced with a conveyor belt and 4-axis robotic arm, increasing manufacturing speed by 15x and saving $10000+
- 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
- Modeled a two-piece aluminum mold for polymer casting in Autodesk Inventor and produced detailed manufacturing drawings, achieving cure tolerances of ±0.02 mm.

**Firmware Developer** **Sep 2024 – Present**

*Midnight Sun Solar Car Team* *Waterloo, ON*

- Developed a BMI323 IMU Driver on an **STM32** in **C**. Used **RTOS** to schedule high level tasks
- Configured **SPI** communication based on datasheets and schematics to set and fetch data from registers
- Developed a self-calibration function to correct sensor gain and offset, improving data precision and reliability.
- Implemented Cyclic Redundancy Check (CRC) algorithm in **C** to ensure reliable and error-free data communication for electric car's bootloader and telemetry systems

**Programming Lead** **Oct 2022 – May 2024**

*First Robotics Competition Team 7902* *Markham, ON*

- Controlled **12+** motors via **CAN** and **PWM** using PIDF loops, and command groups to execute autonomous routines.
- Deployed a **Raspberry Pi** based vision system that achieved **95%** accuracy in object detection and automated arm adjustments within **0.5°** of the target angle.
- Led a team of 6 programmers, teaching an OOP-based approach to robotics coding, delegated members to subsystems, managed and merged projects from multiple branches using **Git**. Worked on **integrating hardware and software**
- Won the Ontario District Championship and attended the FRC world championship as a mentor

## Projects

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

- Designed a drivetrain with 3 differential swerve modules in **Inventor**, driven by a custom **ESP32-based PCB** featuring a phase-enable H-Bridge interface and a two stage power regulator designed in **KiCad**
- Developed a motor driver in C++ with a PID feedback loop, achieving **±1%** accuracy in speed and position control
- Wrote higher-level, class-based code allowing for simultaneous 2-axis control of 3 modules, ensuring error between motors fell within a 1% margin, using an **ESP32 web server** to control modules and log test data

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

- Built an **ESP32-based** drone, using PIDF flight control based on feedback from an MPU6050 IMU using **I2C**
- Hosted an ESP32 web server for visualizing pitch and roll using a 3D model, as well as for controls and data logging.
- Designed an **STM32** based PCB flight controller, communicating to an IMU and radio module using a shared **SPI** bus

**Smart Trash Bin** | *C++, ESP32, OpenCV, Python*

- Incorporated ultrasonic and motion sensors along with a servo motor controlled by an **ESP32** with **C++** to sort waste
- Used a camera running **Python** with an OpenCV (computer vision system) **machine learning model** that processed 15 frames per second, using the serial library to send real-time detection results to a microcontroller with 98% accuracy