Daniel Lee

ds4lee@uwaterloo.ca | https://www.linkedin.com/in/daniel-s-l/ | https://github.com/GitWorkingTime |

Education

University of Waterloo **Sep 2025 - May 2030**
Candidate for Bachelor of Applied Science, Honours Computer Engineering **Waterloo, ON**

Technical Skills

Languages: C++, HTML, CSS, Javascript, SQL (SQLite)
Developer Tools: Git/GitHub, Linux (WSL), Google Workspace, Onshape
Frameworks: NodeJS, ElectronJS

Experience

Firmware Member *C* | *FreeRTOS* **Sep 2025 - Present**
UW Orbital Design Team **Waterloo, ON**

- Coded a custom driver in C for the LMB75B temperature sensor to get temperature values, utilizing Two's Complement to convert non-standard data types to integers as a simulation

Lead Programmer **Sep 2022 - Feb 2024**
Bishop O'Byrne FTC Robotics Team | *Java* | *OpenCV* | *Tensorflow* **Calgary, AB**

- Implemented basic odometry through the use of motor encoders and basic PID control, initiating the use of odometry for 2024-2025
- Coded April Tag detection and PID control of the robot's IMU to self correct the robot's heading as a proof of concept for consistent path trajectory
- Incorporated Tensorflow and OpenCV to detect unique game pieces that earned us 10 - 20 more points consistently per autonomous round

Projects

Delirium Detection Monitoring System | *NodeJS* | *SQLite* | *Javascript* | *HTML/CSS*

- Created an HTTP web server using NodeJS to receive sensor data reliably from an ESP32 via an HTTP POST request
- Designed and implemented the parsing and storing of ESP32 sensor data into a JSON file and into a simple SQL database using SQLite to retain the information even after refreshing the page
- Coded a dashboard using vanilla HTML, CSS, and Javascript and used Server-Side Events to live-update the dashboard whenever new data is received
- Utilized FreeRTOS to code a push button with an ESP32 on the ESP-IDF VSCode extension as a way to test HTTP connection between the ESP32 and the web server

Snackatron | *C++* | *PlatformIO*

- Worked alongside representatives from Human Computer Labs to attempt to build a robotic lamp called *LeLamp* as a base for our Hackathon project
- Build a basic robot using cardboard, ESP32, and various components to make it move with an IR remote and push out a cardboard tray in which the user can put treats.

Basic HTTP Web Server | *C* | *Javascript* | *HTML/CSS*

- Implemented a single-threaded HTTP web server in C to host my own custom messaging app, handling POST requests for both text messages and images, which are saved briefly on the server side
- Designed the user interface using vanilla HTML, CSS, and Javascript to allow users to send and receive text and images

HeaderSniffer | *Python*

- Coded a Python CLI tool that verifies the file format given a file path as a precaution to malware attacks
- Utilizes magic numbers within the beginning of a file to identify the file's true format to output a warning if the file's true format is different from what it appears in file explorer