Daniel Lee

daniel.lee28011@gmail.com | 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, Computer Engineering **Waterloo, ON**

Technical Skills

Languages: C/C++, HTML, CSS, Python
Tools: Git/GitHub, VS Code, Google Sheets/Excel, FreeRTOS

Experience

Firmware Member **Sep 2025 - Present**
UW Orbital Design Team | *C/C++* | *FreeRTOS* | *Python* | *Linux (WSL)* **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
- Implemented commands to update safety-critical command handling using C and Python for the satellite's RM46 microcontroller to prevent false executions

Technical Director **Sep 2025 - Feb 2025**
Bishop O'Byrne FTC Robotics Team *CAD* | *Google Sheets* **Calgary, AB**

- Maintained the team's consistent progress by organizing weekly schedules and deadlines, keeping everyone on the same page and defining sub-team's goals, guiding the team's efforts
- Announced team goals and reminders to maintain productivity and cohesion among members to streamline component production
- Strengthened team morale during competition days by maintaining a calm and upbeat demeaner during time crunches, helping members focus on their tasks for increased team efficiency

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, initializing 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 during autonomous mode
- Implemented Tensorflow and OpenCV to detect unique game pieces that earned us 10 - 20 more points consistently per autonomous round

Projects

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

- Created a single-threaded HTTP server, made with C to handle TCP/IP protocols and packets which was later used to host my own chat messaging app via LAN