Alexander Zhan

+1 613-867-6268 | azhan@uwaterloo.ca | www.linkedin.com/in/azhan3 | https://github.com/azhan3

EDUCATION

University of Waterloo Waterloo, Ontario, Canada
Bachelor of Computer Engineering – Cumulative GPA: 90.83% *Sep 2024 – Apr 2029*

TECHNICAL SKILLS

Languages: C/C++, Python, Java, Typescript, JavaScript, HTML/CSS, SQLite, Arduino
Frameworks: Flask, Node.js, Express.js, Next.js, Angular, Django, React.js, Vue.js, Electron.js, React Native, Vert.x
Developer Tools: Linux, Vim, Jenkins, Git, Github, Docker, PostgreSQL, FastAPI, Firebase
Libraries: pandas, NumPy, PyTorch, Matplotlib, Scikit-learn, SciPy, OpenCV, Mediapipe, MongoDB, cURL

EXPERIENCE

Ford Motor Company Ottawa, ON
Software/Hardware Development Intern | Python, C++, Flask, Jenkins, Javascript, LTspice *May 2025 – Aug 2025*

- Developed an internal Bootstrap + Flask application to visualize real-time system thermistor temperatures as a heatmap on the Telematic Control Unit (TCU), addressing the need for efficient test validation, logging, and monitoring.
  - Reduced process time by 92% by eliminating the need for engineers to manually measure board temperatures using infrared cameras
  - Now used daily by 30+ engineers for streamlined diagnostics
- Developed a software-controlled 3-to-8 CAN multiplexer in C++ on Arduino Leonardo, automating a process that previously required engineers to manually reconnect CAN devices, reducing test setup time by 100% and eliminating human errors.
- Conducted Linux-based embedded system validation, resolving GPIO and board-level bugs to reduce deep sleep leakage current from 600μA to 20μA, and debugging UART, I2C, SPMI, and CAN signals with oscilloscope to ensure reliable communication.
- Identified design flaws in pre-production hardware with LTspice and worked with senior engineers to document and resolve them.

PROJECTS

Competitive Programming Debugger | *C++, cURL, Javascript, Electron.js, D3.js*
- Built a visual debugger for competitive programmers using C++, cURL, Electron, and D3.js, enabling real-time variable tracing through custom `dbg()` statements.
- Developed a system that automatically parses variables from C++ programs using a `dbg()` statement, enabling real-time visualization of data including arrays, integers, strings, graphs, trees, etc.
- Designed and implemented a dynamic user interface that presents variable data in a user-friendly manner by visualizing variables from each `dbg()` statement on separate pages

Fractal Visualizer in Zp Space | *Java, Vert.x, Node.js, React.js, Electron.js*
- Created a portable fractal visualization tool in Java to support research and analysis of number theory concepts in Z_p space.
- Implemented backend in Vert.x for real-time data communication between frontend and computation engine.
- Designed interactive UI using React.js and Electron.js for cross-platform usability.
- Developed Prime Race visualizations to compare the frequency of different prime classes (e.g., mod 4), with configurable settings.
- Utilized Node.js for server-side scripting, enabling seamless interaction between frontend and backend.
- Generated fractal images used in Canadian Undergraduate Mathematics Conference presentations.

Terreria | *Java, OpenGL, GSON, lwjgl, JBox2D*
- Developed a Java game similar to Terraria using the Lightweight Java Game Library and ImGui
- Serialized and deserialized map and character data using the Google gson library
- Used OpenGL to create a fragment shader and texture loading

Track-Pad | *Python, Flask, Mediapipe, SkLearn, OpenCV, VueJS, ElectronJS*
- TrackPad is a desktop application that allows users to control their computer's mouse through hand gestures, with the ability to customize gestures for various commands.
- Used Google's Mediapipe library and OpenCV for hand detection
- Implemented a machine learning model (KNN) to detect hand gestures via webcam.
- Implemented features such as scrolling, window switching, and voice input; UI built with Vue.js and Electron.
- Used Flask backend for gesture recognition and interaction logic.