



Tinashe Masoka

140 Sagewood Blvd SW, Airdrie, AB | 587-966-9887 | masoka@ualberta.ca | [LinkedIn](#) | [GitHub](#)

ACADEMIC & CO-OP STATUS

Computer Engineering - Nanoscale System Design Option, Bsc Co-op

University of Alberta

Class of 2027

- Completed Academic Terms: 5
- Completed Work Terms: 1
- Available to begin employment(4-16 months): Jan 2025

TECHNICAL SKILLS

Programming Languages: C/C++, Python, Java, Assembly, Bash, HTML, JavaScript, VHDL

Frameworks and Tools: Docker, ROS, OpenCV, Git, Arduino IDE, LTspice

Operating Systems: Linux, Ubuntu, Windows

Hardware: Raspberry Pi Pico, STM32, Electrical Circuits

PROJECTS

Autonomous Robotic Vehicle Project(ARVP) UofA Club | *Python, ROS*

Since Jan 2024

- Embedded code for the robot's motion, ensuring precise navigation and control.
- Labeled and prepared 500+ data samples to improve computer vision, achieving 85% object detection accuracy.
- Contributed in securing the team's 7th-place finish among 28 universities at RoboSub 2024.

HackED Beta 2024 Hackathon – 2nd Place | *ChatGPT API, Spotify API, Python*

Nov 2024

- Collaborated with a team of 5 developers to create a music recommendation application tailored for integration with smart glasses, developed within 24 hours.
- Utilized outputs from a PyTorch-trained model on the COCO dataset (330,000+ annotated images with 80 object categories and 5 scene captions) to create context-aware music playlists.
- Designed and optimized prompts for the ChatGPT API to transform image-based text descriptions into relevant music genres, achieving a 90% genre-matching accuracy.
- Integrated the Spotify API to generate playlists within 5 seconds on average, ensuring seamless user experience.
- Presented the project to a panel of 6 judges, earning 2nd place out of 73 participants for its innovative approach to leveraging AI for enhanced user interaction.

Wordl (Interactive Word-Guessing Game) | *C*

Nov 2024

- Recreated the popular game Wordle in C, providing players 6 attempts to guess the correct word.
- Incorporated 3 colors for feedback using ANSI escape codes to visually indicate correct letters in the right position (green), correct letters in the wrong position (yellow) and incorrect letters(white).
- Utilized efficient string comparison and character-matching algorithms to evaluate player inputs and track letter occurrences.
- Developed logic to ensure accurate handling of repeated letters and edge cases, improving the user experience and robustness.

Morse Code Interpreter | *C++*

Aug 2024

- Developed a Morse code interpreter using an Arduino and circuit elements with real-time display output, enabling accurate translation of Morse code signals into characters on an 16x2 LCD screen.
- Designed a binary tree structure for efficient decoding of Morse code(26 letters & 2 for added functionality), ensuring swift and accurate mapping from signals to characters.
- Implemented a software-based debounced button press detection, distinguishing between short and long presses to reliably differentiate between dots(ranging from 10ms to 150 ms), dashes(ranging from 150ms to 500ms) and noise.
- Created a dynamic user interface on the 16x2 LCD, allowing users to see both their input in real-time and the corresponding decoded output.
- Managed resource utilization effectively by implementing memory management functions, including tree traversal and node deletion, to maintain optimal performance.