Khalil Burns

(343)-598-5856 | kjrburns@uwaterloo.ca| khalilburns.com | linkedin.com/in/khalil-burns | github.com/khalil-burns

Technical Skills

Languages: C++, C, Make, Assembly, VHDL, Java, Python
Developer Tools: Git, GitHub, Docker, VS Code, STM32CubeIDE, ArduinoIDE, VIM
Hardware: Raspberry Pi, Arduino, STM32, UART, CAN, Oscilloscope, Soldering, FPGA (Quartus Prime)

Experience

RTOS Developer Jan. 2026 - April 2026
Huawei Canada, iRTOS R&D Team *Ottawa, ON*

- Implemented a scheduler benefit analysis method in the kernel using **C** to quantify main thread startup time reductions from proposed priority acceleration changes, achieving **0.5ms** average accuracy validated against reference traces
- Embedded thread wakeup chain and run queue timestamp logging into the **C** scheduler for precise thread latency analysis
- Used **OpenHarmony HDC** and **Fastboot** to perform software feature validation and debugging directly on Huawei phones
- Developed **Python** scripts to parse, repair, and structure **50+** beta trace files on a per-thread basis, measuring benefits up to **15%**

Embedded Robotics Developer March 2026 - May 2026
Sielo Robotics *Ottawa, ON*

- Designed a low-cost object detection safety feature for the robotic arm, reducing additional hardware costs to under **$50**
- Programmed an **ESP32** in **C** to read multiple ultrasonic sensors and transmit distance data to the main module over **CAN**, enabling object detection within **20cm** with **100%** collision accuracy
- Researched a calibration issue delivered a report, driving the shift to high-accuracy components over unreliable workarounds

Firmware Developer May 2025 - Aug. 2025
Ford, BSP/Bootloader Team *Ottawa, ON*

- Developed and deployed firmware fixes in **C** for bootloader and NOR flash systems across multiple **Linux**-based ECUs
- Validated and debugged firmware fixes directly on hardware using **Fastboot MTD** utilities within a **Linux** environment
- Designed and implemented a reusable unit testing infrastructure in **C** with **Ceedling**, now used in the bootloader codebase
- Achieved **100%** test coverage on the block file system code, uncovering **10+** critical defects present in production code

Software Developer July 2023 - Aug. 2023
Nokia, IP Networks Division *Ottawa, ON*

- Developed a third-party **Python WebSocket** server to simulate communication between Nokia's NSP and external servers to test the speed of network interactions
- Containerized the WebSocket server with **Docker** for easier deployment and scalability during testing
- Integrated **Locust** for load testing to analyze the performance of the WebSocket server

Projects

Automatic Trumpet Playing Robot | *C, STM32, CAD/3D Printing, UART* | [GitHub]

- Designed and 3D printed a custom mechanical assembly in **Fusion** to mount DC motors onto a trumpet and actuate its valves
- Programmed an **STM32** microcontroller in **C** to drive motors via a motor driver IC, executing valve sequences with precise timing
- Implemented a **UART** interface between a host keyboard and the STM32 to allow real-time user-controlled note sequencing
- Integrated mechanical, electrical, and software subsystems to produce a functional automatic instrument

Feed-Forward Neural Network From Scratch | *Java, Spring Boot, JavaScript, HTML/CSS, WebSockets* | [GitHub]

- Created a custom feed-forward neural network in **Java**, including forward/backpropagation and weight/bias updates
- Coded the gradient descent algorithm with tunable hyperparameters (learning rate, batch size, etc.) for flexible model training
- Validated performance on the MNIST dataset, achieving **96%** accuracy without the use of any external ML libraries
- Developed a full-stack web app using a **Java Spring Boot** backend and **JavaScript** frontend to host and demo the project

Head Collision Detection System for Blind Users | *STM32, C, UART* | [GitHub]

- Programmed embedded **C** code on two **STM32** MCUs for a smart glasses obstacle-detection system to assist blind users
- Coded ultrasonic sensors to measure distance in real-time and provide **PWM**-based haptic feedback to alert users of obstacles
- Implemented custom **UART** communication protocols in **C** for data exchange between boards

Autonomous Tracking Car | *Raspberry Pi, Python, OpenCV, PWM* | [GitHub]

- Designed and coded a **Raspberry Pi**-based autonomous car with onboard motor driver and servo control circuitry
- Added real-time object tracking by mounting a camera on a servo, combining **OpenCV** with dynamic servo positioning
- Controlled steering and drive motors using **PWM** signals to enable responsive autonomous navigation

Education

University of Waterloo Sep. 2024 - Present
Bachelor of Applied Science in Computer Engineering – GPA: 90.18% *Waterloo, ON*
Coursework: **FPGA** & **VHDL** (Fundamentals of Digital Logic), **C++** (Data structures and algorithms)