

OWEN BRAKE

Waterloo Mechatronics Engineering · OwenBrake.com · (647)548-5493 · obrake@uwaterloo.ca

WORK EXPERIENCE

Embedded Firmware Engineer

Apple

Winter 2021

Remote

- Developed features for resource constrained embedded device in C and Python
- Designed and developed features in C and assembly for ARM Cortex-M0 system
- Specific features are redacted to preserve confidentiality

Software Engineer

Ford Motor Company

Summer 2020

Remote

- Worked on system to process vehicle core dump files into easily readable, accessible and shareable online formats using GDB and Java
- Rewrote permissions system to enable complex and nested conditions while maintaining performance on system with over 1 billion database records in Java and SQL
- Revamped vehicle file text editor to enable split screen, persistent session behaviour and numerous UX improvements in JS

Full Stack Developer

Groupdesk

Summer 2019

Toronto, ON

- Developed CRUD services, using Angular to remove user dependence on technicians
- Automated front end QA using Go, Docker and Chromedp to increase release efficiency and stability

Data Entry

Liberty Metrics

Fall 2016

Mississauga, ON

- Online data mining and compiling of hotel booking data

PROJECTS AND TEAMS

Waterloo Formula Electric Team (Head of Firmware)

September 2019 - Present

- Designed and implemented firmware for ARM Cortex-M7 and M0 boards in FreeRTOS and C which communicate on the CAN bus
- Developed sensor analytics platform on Python for Beaglebone to measure and visualize live vehicle performance remotely
- Worked on drivers for the various sensors and external boards on the car like: LTC6812, LTC6811, LTC4110, etc.

Flatten.ca, COVID-19 Analysis Platform

March 2020 - May 2020

- Developed and scaled web service to analyze geographic trends in COVID-19 across Ontario, handling 400,000+ users using the MERN stack
- Featured on various Canadian news outlets including: CBC, CTV, Globe and Mail, Toronto Star

Isidore, Custom Programming Language

December 2019 - July 2020

- Deployed JIT compiled, cross platform programming language built in LLVM using C++.
- Designed language to solve many of the runtime safety problems of C while retaining minimum overhead and lightning fast runtime performance

Self Driving Go Kart

June 2019 - August 2019

- Utilized Arduino, motor controllers and RC radio to allow remote control of Go Kart.
- Produced computer vision and control software in OpenCV and Python

SKILLS

Programming Languages: C, C++, Verilog, VHDL, ARM Assembly, Go, Python, Java, MERN/LAMP Stack
Software: FreeRTOS, LLVM, Git, STM32CubeMX, PID, GDB/LLDB, MATLAB, SQL
Electrical: Soldering, Circuit Design, Signal Analyzers, I²C, SPI, UART, CAN, ARM Cortex-M