Caleb Andreano

calebandreano@gmail.com | www.linkedin.com/in/calebandreano | www.andreano.dev

Software Engineer at SpaceX and M.S. Machine Learning student at MSOE. Skills in embedded systems programming, software design for critical systems, FPGA and hardware design, data science and ML engineering.

Education

Milwaukee School of Engineering *Milwaukee, WI*
B.S. Computer Engineering | GPA: 3.95 — *May 2025*
M.S. Machine Learning | GPA: 4.00 — *Expected May 2026*

Experiences

SpaceX *Redmond, WA*
Software Engineer (*Starlink*) — *July 2025 – Present*

Milwaukee Tool *Brookfield, WI*
Firmware Engineering Intern: Battery Platform/NPD — *Jun 2024 – Aug 2024*

- Implemented Coulomb-count and open circuit voltage based state of charge algorithm for next-generation battery pack firmware using `c/c++` on TI and STM hardware, verifying with unit test suite and CSTAT static analysis.
- Developed data processing pipelines using `pandas`, `numpy`, `scipy` and `matplotlib` for interpretation and monitoring of circuit parameters, I^2C traffic and pack MCU serial communications.
- Used `golang`, `python`, mutlithreaded programming, analog circuit design principles and digital filters to build automated test fixtures and control equipment.
- Explored extended Kalman filters for modeling dynamical state transitions using MatLab.
- Result: Project firmware and tests will be integrated into next-generation M18 platform.

Gama Space *Paris, France* (*Remote*)
Guidance, Navigation, Controls Intern — *Feb 2023 – Oct 2023*

- Used `c` and FreeRTOS to design asynchronous UART and CAN drivers for an STM microcontroller.
- Reverse-engineered and redesigned Cubesat Space Protocol (`libcsp`) into a cross-plaform, memory and thread safe Rust library, providing and documenting an ergonomic API for integration into existing flight software for currently in-flight low Earth orbit satellite platform.
- Provided network interface drivers for UDP, TCP/IP, CAN, UART, and loopback.

Leadership & Co-curricular involvement

Mozee Motorsports (SAE Formula Hybrid)
Hardware Project Owner — *May 2024 – May 2025*

- Developed custom hardware and software for four-node fault-tolerant CAN2.0B drive-by-wire system.

Software Team Lead — *May 2023 – May 2024*

- Directed full system software rebuild for 2024 1st place world-champion hybrid vehicle.
- Implemented fully asychronous firmware using Rust and `embassy` for controlling multiple subsystems, including collecting and interpreting throttle input, controlling ICE throttle body (throttle-by-wire) and electric motor controller over CANFD, and validating safety system startup sequence, runtime monitoring, and fault detection.
- Attended 2024 Formula Hybrid + Electric competition in Loudon, NH, continuously developing and testing firmware during competition and passing all technical inspections.

Software Team Member — *Sep 2022 – May 2023*

Raider Center for Academic Success
Lab Assistant: Embedded Systems — *Jan 2024 – May 2025*
Tutor: Embedded Systems, Computer Architecture, Data Structures, Embedded Systems — *Jan 2024 – May 2025*
MSOE Jazz Band — *Feb 2022 – May 2025*
Tau Beta Pi: WI-Delta — *Aug 2023 – Present*

Selected Projects

FPGA Touchscreen/Servo Control

- Designed custom PWM servo driver hardware component in VHDL and a `c` API for control.
- Integrated touchscreen driver component into digital hardware system using Intel Platform Designer.
- Created a graphics library in `c` for drawing and rendering 3D graphics on the touchscreen.
- Deployed RTL design on Altera MAX 10 FPGA.
- Implemented application code to draw a shape on the touchscreen using a stylus or finger and and automatically trace the shape with a servo-controlled robotic arm and laser.

ARMv4 CPU on FPGA

- Designed and implemented a 5-stage single-cycle RISC ARMv4 CPU on an Altera MAX 10 FPGA.
- Implemented instruction fetch (program counter/memory), decode, execute (ALU, register file, shifter), write-back, and memory stages using VHDL and Intel Quartus, enabling a subset of the ARM ISA to run on the CPU.
- Created custom assembler and disassembler in `c++` to encode/decode from ARM to machine code.
- Verified design by implementing automated tests using ModelSim, validated by running programs on hardware interacting with memory-mapped peripherals and devices.

Network Interface Device

- Developed a CSMA/CD network device, implementing physical and data-link layers with a baremetal firmware design.
- Created Manchester Biphase II line encoder/decoder, interrupt driven, buffered, and variable length transmitter and receiver, CRC8 frame check sequence, asynchronous channel monitor and collision detection.
- Wrote repeatable test procedures, created design criteria and developed robust system requirements.
- Implemented common protocol standard enabling communication using unicast and broadcast addressing.

Tritone: Vector Calculator Expression Language/Interpreter

- Created context-free grammar and recursive descent expression parser in `c`.
- Developed abstract syntax tree data structure for expression evaluation and `dj2`-based dynamically-resizing linear probe hash table for variable assignment and retrieval.
- Implemented CLI interpreter for nested expression evaluation, implementing scalar and n-dimensional vector operations.

Coltrane: Round-Robin scheduler and synthesizer on STM32

- Developed a round-robin scheduling algorithm on an STM32 Arm platform using `c` and Arm Assembly.
- Implemented thread-level parallelism with custom synchronization primitives (mutex, semaphore).
- Enabled asynchronous user input through a serial console and UART driver, recording a song using an interrupt-driven keypad and song playback with multiple waveform types using a digital-to-analog converter and a piezo speaker.

Skills

Software: C/C++, Rust, Go, VHDL, Python, Java, HTML/CSS/JS, ARM Assembly, Nix, Bash
Hardware: PCB Design, Board Bringup, HIL/HITL testing, FPGA, RTL, Circuit Design, Digital Filters, Signal Processing
Technologies: Intel Quartus/ModelSim, NI MultiSim, Altium Designer/KiCad, TCP/IP