

Alexander Bugar

(727) 710-7038 – me@alexanderbugar.com – alexanderbugar.com
github.com/cascade-interactive – linkedin.com/in/alexanderbugar

EDUCATION

University of South Florida

- B.S. in Computer Engineering – GPA: 3.78/4.00

Expected May 2028

Coursework: Data Structures, Computer Architecture (RISC-V), Digital Logic Design, Linear Systems

EXPERIENCE

Undergraduate Researcher – Bellini College REU

Aug 2026 – Present

University of South Florida – Advisor: Dr. Santosh Pandey

- Investigating workload placement for multimodal LLM inference across NVIDIA H200 and RTX 6000 GPUs, analyzing compute and communication contention bottlenecks in a three-student team

PROJECTS

N-Body Simulation – C++20, CUDA, Performance Optimization

GitHub

- Optimized 5,000-body physics integration by 10.1x (83.7 ms down to 8.3 ms) by migrating serial CPU workloads to parallelized CUDA kernels
- Implemented Velocity Verlet integration mapping one GPU thread per body to maximize hardware accelerator utilization on an RTX 4070 Ti SUPER

STM32 Flight Controller – C/C++, ARM Cortex, FreeRTOS, SPI, DMA

GitHub

- Engineered a 1 kHz FreeRTOS IMU pipeline on STM32G474, implementing interrupt-driven SPI DMA drivers to guarantee non-blocking sensor transfers
- Implemented a quaternion extended Kalman filter for attitude estimation; wrote 13 host-side GoogleTest cases validating rotation tracking, invalid sensor inputs, and long-run numerical stability
- Designed a custom STM32H743 flight controller board in Altium Designer, integrating 4 motor outputs, 5 UARTs, and hardware SD logging to overcome MCU peripheral limitations

Real-Time HIL Simulation Bridge – C++17, Unreal Engine, ESP32, UART, UDP

GitHub

- Developed a C++17 UDP-to-UART bridge to route simulated vehicle states to ESP32 firmware, enabling real-time hardware-in-the-loop (HIL) telemetry validation
- Built a basic Unreal Engine plugin to stream simulated vehicle telemetry over UDP, providing a visual front end
- Engineered UART fault recovery with CRC-32 validation, payload-length checks, and sync-byte scanning to systematically reject corrupted packets and restore stream alignment

Drone Physics Simulator – C++17, CMake, GitHub Actions, CTest

GitHub

- Built a 120 Hz C++17 drone physics engine modeling rigid-body dynamics, collisions, and simulated sensors
- Automated headless regression validation using CMake, CTest, and GitHub Actions; measured 2 mm altitude error over 10 s and a 0.12 ms maximum physics step under stress testing

SKILLS

Languages: C, C++ (C++17/20), CUDA, Python, MATLAB

Systems & Software: Linux, CMake, Git, GDB, PlatformIO, STM32CubeMX, Unreal Engine

Embedded: ARM Cortex, STM32, ESP32, FreeRTOS, Bare-Metal Drivers

Protocols & Interfacing: SPI, I2C, UART, USB CDC, DMA, Interrupt Handling

Hardware & Tools: Altium Designer, Oscilloscopes, Logic Analyzers, Hardware Debugging