Joshua Labasbas

Gainesville, FL

352-615-6010 jlabasbas24@gmail.com linkedin.com/in/joshua-labasbas github.com/hoshbosh hoshbosh.dev

Education

University of Florida – B.S. in Computer Science, Gainesville, FL May 2026

Technical Skills

Programming Languages: Python, C/C++, SQL, Rust, GLSL
Developer Tools: Linux, Docker, CMake, Git, gRPC, CAN bus/Vector CANape
Technologies/Frameworks: PyTorch, NCCL, FlashAttention, vLLM, OpenGL

Experience

Tesla **August 2025 – December 2025**
Software Engineering Intern *Palo Alto, California*

- Designed and built a containerized co-simulation stack (Docker, Rust IPC, simulated CAN) connecting 2 previously isolated validation domains, OS-level applications and ECU firmware, with hundreds of simulated signals and a shared interface designed to scale to 4 feature teams.
- Accelerated HIL test execution by 35% through developing a Python API within pytest to interface with can-bus signals via HIL gRPC channels, reducing overhead and expediting test development cadence via reduced compilation time
- Increased testing cadence for system-level features by 144x compared to testing on a real system by bringing up infrastructure for multiple components in a single SIL environment
- Reduced cost by 99.7% for testing system-level features by simulating environments around multiple components under test, removing the need for an expensive test system
- Expanded capabilities of SIL environments by simulating initial power off and power on, a commonly undertested portion of ECU lifecycle

Dream Team Engineering **January 2025 – Present**
Software Captain *Gainesville, Florida*

- Led software development of a Brownian Bridge Diffusion Model for CT-to-MRI brain synthesis, achieving SSIM 82.5% and PSNR 21.6 dB on SynthRAD2023 (~ 210 supervised pairs) by implementing a two-stage training strategy with MRI-to-MRI identity pretraining on $\sim 3,858$ volumes followed by CT$\rightarrow$MRI fine-tuning using SKC cross-attention and ISTA sampling
- Scaled training throughput to batch size 80 on 8 B200 GPUs on the HiPerGator HPC cluster by integrating mixed-precision training, FlashAttention via scaled dot product attention, and gradient clipping, reducing per-epoch time by 50% over the baseline
- Recovered a 10.5-point SSIM regression (72% $\rightarrow$ 82.5%) by diagnosing a staged-curriculum failure mode, reimplementing perceptual loss, and instrumenting W&B logging to isolate the degradation to a specific training phase

Dynon Avionics **May 2025 – August 2025**
Software Engineering Intern *Portland, Oregon*

- Contributed to avionics software development for experimental light aircraft, pushed production quality code onto thousands of aircraft
- Actualized pitot freeze warning systems, referencing outside air temp, ADARS status, and true airspeed to ground airspeed differentials to warn a user of dangerous flying conditions, increasing experimental aircraft pilot safety
- Upgraded flight simulation workflows to X Plane 12, utilizing Typescript and Node.JS to interface with avionics panels via RS-232 and UDP multicast packets for HIL testing

Machine Intelligence Lab UF **May 2024 – December 2024**
Software Team Member *Gainesville, Florida*

- Developed an underwater autonomous robot to compete in the international 2024 RoboSub competition
- Introduced simulated environments and obstacle maps mirroring RoboSub competition tasks to perform SIL testing of the AUV's navigation and vision systems
- Actualized 100% stability in 6 DoF tested in person with predetermined routes using PID tuning with HIL testing
- Trained a YOLOv7 computer vision model to identify competition obstacles using over 1,100 categorized images

Well Bilt Industries **August 2023 – September 2024**
Software Engineering Lead *Remote*

- Materialized a REST API using Express.js interfacing with MongoDB for storing project data for aircraft hangar doors
- Optimized estimate creation by a 75% reduction in production time by employing Electron.js and React.js technologies
- Generated engineering drawings with 1% error from hand-calculated values, and close to 1% error in financial estimates

Projects

Cloudstep | *Custom C++ Fighting Game Engine* **May 2025 - Present**

- Built a custom C++23 game engine using EnTT ECS and SDL3 GPU backend, with deterministic 60Hz fixed-timestep simulation, ozz-animation skeletal pipeline, and frame-perfect input buffering targeting competitive fighting-game requirements
- Implemented a frame-data-driven hit/hurtbox system and state-machine-based VFX pipeline (smear frames, morph targets, flipbook textures) in C++23, with all gameplay state replayable from a deterministic input log for testing and netcode preparations