

Stefan deBruyn Spaceflight Systems Software Engineer

✉ stefanriverdebruyn@gmail.com  [/stefandebruyn](https://github.com/stefandebruyn)  [in/in/srdebruyn](https://www.linkedin.com/in/srdebruyn)  stefandebruyn.github.io

EXPERIENCE

SpaceX — Starship Software Engineer (July '22 - Present)

- Develop mission- and safety-critical systems software for the Starship space transportation system
- Own backend software for vehicle launchpads, test stands, and factory hardware performing embedded control
- Drive the full SDLC for launch and test infrastructure software that demands rapid iteration and 24/7 uptime
- Work regularly with embedded C++, real-time Linux, NI cRIO systems, and highly distributed systems

Texas Rocket Engineering Lab — Senior Avionics Software Engineer (May '19 - June '22; 3 yr)

- Worked as a lead developer on the software stack for a liquid bipropellant spaceshot rocket [1]
- Focused on development of the flight computer, flight software, and vehicle hardware-in-the-loop simulation
- FSW – worked with safety-critical embedded C/C++, real-time Linux embedded controllers, and Python
- HIL – worked with MSVC, NI VeriStand, NI PXI systems, and general embedded I/O (DIO, AIO, serial)
- Taught 30+ student engineers, managed a 100k+ LOC codebase, and wrote hundreds of pages of documentation

Blue Origin — Avionics Software Engineering Intern (May '21 - Aug '21; 3 mo)

- Worked on the common avionics software framework supporting Blue's launch vehicles and spacecraft
- Explored alternative middleware designs in support of 3rd party software interfaces and long-duration space missions
- Prototyped new middleware for flight computer state sharing that increases interoperability and runtime modularity
- Worked regularly with bare metal C++, JTAG debugging, and resource-constrained embedded controllers

CACI Intl. at NASA JSC — Flight Software Engineering Intern (May '20 - Aug '20; 3 mo)

- Developed embedded life support software for xEMU spacesuits in support of the NASA Artemis program
- Created a tool for automatic generation of life support fault detection code that saves months of dev work [2]
- Implemented new fault injection functionality in the Portable Life Support System astronaut training simulation
- Worked regularly with Python, safety-critical embedded C, and NASA's Trick and CFS frameworks

FAVORITE PROJECTS

[1] Halcyon — Liquid Bipropellant Spaceshot Rocket

texasrocketlab.com/halcyon

- Fully-recoverable, LOX-RP1 collegiate rocket with RCS roll stabilization and active attitude control designed to reach space
- Distributed real-time Linux flight computer controls all vehicle systems: propulsion, fluids, GNC, recovery, and telemetry
- Flight software validated by code review, CI/CD, unit testing, static analysis, and hardware-in-the-loop testing

[2] xEMU CWS — Spacesuit Caution & Warning System

nasa.gov/suitup

- Computer embedded within xEMU spacesuits which monitors for life support system malfunctions and misconfigurations
- Runs fault detection code automatically generated from requirements spec sheets by a lexical analyzer and metaprogrammer

[3] Surefire — Open-Source Flight Software Framework

github.com/stefandebruyn/surefire

- Highly configurable and portable C++ framework for advanced collegiate rocketry or other real-time control applications
- Provides task scheduling, software bus, state machines, and platform abstraction for both OSs and bare-metal platforms
- State machine DSL aids development of domain-specific control logic and runtime logic reconfiguration

SKILLS

C++ • C • Python • Java • Linux • GNU • RTOS • Networks • CI/CD • HIL • SIL • Simulation • Fault-Tolerant Software • Software V&V • FSW Architecture • Avionics • Control Theory

EDUCATION

The University of Texas at Austin

- BS Computer Science, 2022
- 3.82/4.00 GPA
- Thesis: *A Flight Software Framework and State Machine DSL for Advanced Collegiate Rockets* [3]

COURSEWORK

Operating Systems • Multicore Operating Systems • Computer Architecture • Programming for Correctness & High Performance • Program Verification & Debugging • Concurrency • Compilers • Algorithms & Complexity • Object-Oriented Programming • Autonomous Robotics I, II • Symbolic Programming • Data Structures • Discrete Math • Number Theory • Multivariable Calculus • Linear Algebra • Probability & Statistics