

Stuart Yeadon

LOW LEVEL SYSTEMS ENGINEER

C RUST PYTHON

Email stuartyeaddon@gmail.com

Telephone +44 7525 435 792

Website syeadon.info



PROFESSIONAL SUMMARY

Apple Genius with 12 years triaging hardware and software across the entire product and service ecosystem. Holds certificates in Computer Science and Embedded ARM Systems - writing in C, Rust & Python.

Brings first-hand insight on how technology meets the user, combining engineering understanding with practical experience of how systems behave, fail and ultimately shape the customer experience.

EDUCATION

2025	Certificate	ARM & Embedded Systems	Coursera View Certificate
22 - 25	Certificate	Computer Science	Codecademy View Certificate
09 - 12	BSc (Hons)	Commercial Music - 2:2	Bath Spa University
07 - 09	A Levels	Mathematics - B	Bishop Vesey's Grammar School

EMPLOYMENT HISTORY

2014 - Present	Apple Genius	Apple Store Touchwood
2012 - 2014	Musician	Bath & Bristol

INTERESTS

Landscape Photography - Highlights include Lake Garda & The Dolomites, Monkeys of Marrakesh and the Quiraing Circuit in Isle of Skye. Check out the Gallery on syeadon.info

Lead Guitar & Classical Pianist - Played the Piano from 4 years old. Achieved Guitar Grade 8 distinction at 15. Highlights include sailing to a gig with Eric Clapton's yachtman.

Charity fundraising - I take my espresso machine into store to connect with team members and raise money for important causes such as disaster appeals and local charities.

PROGRAMMING STORY

C / C++ Learned C following my studies by designing an ARM Cortex-M4 greenhouse controller to develop a deeper understanding of low-level systems and performance-critical code. Currently studying XNU, BSD and Mach codebases to understand the macOS kernel.

Rust Developed syeadon.info using my own Rust engine compiled to WebAssembly, with WebGPU powering rendering, shaders and post-processing while JavaScript and HTML serve only as a thin browser integration layer.

Python Started my coding journey in Python becoming addicted to solving problems on Codewars. Progressed to the top Kyu 1 level, implementing Trevino Pollock's algorithm that solves lagranges sum of four squares for numbers as large as 2^{1024} within 12 seconds. [View Paper](#)

Tooling	• Cargo	• CMake	• wasm-pack	• Conda
	• Git	• LLVM / Clang	• HomeBrew	• Venv

PROJECT TECHNICALS

Arm Cortex-M4 Solar-Powered Autonomous Greenhouse

STM32WB55 | Bare-Metal C | GPIO | ADC | I²C | 1-Wire | Low-Power Systems

An autonomous smart greenhouse controller w/ lighting, heating and solar power

- Direct-to-hardware firmware in C favouring direct register manipulation over HAL libraries.
- GPIO, ADC, I²C, RTC, Timers, PWM generation and 1-Wire protocol implementation.
- Hand soldered logic board, power supply and fused power distribution network.
- Signal debugging in the STM32 environment using live expressions and breakpoints.
- Modular product design for easier maintenance and installation.

macOS Nuvoton NCT6796D Super I/O Driver

IOKit | Objective-C | C++ | Xcode | Kernel Development | Hardware Monitoring

An IOKit kext and user-client tool for fan control and temperature readings of an Intel mac.

- Translated hardware datasheets into register maps, bank-selection logic and I/O operations.
- Built a user–kernel communication layer using IOUserClient and external method dispatch.
- Implemented temperature monitoring, fan RPM readings and manual speed control.
- Created a live terminal interface with continuously refreshed hardware data.
- Used multithreading and atomic state management for responsive monitoring.
- Added safeguards to prevent writes to reserved hardware registers.

WebAssembly Rendering Engine in Rust (runs syeadon.info)

Rust | WebAssembly | WebGPU/WGPU | WGSL | GPU Rendering | Systems Architecture

A high-performance terminal rendering engine for the browser, built in Rust and WebAssembly.

- GPU-instanced text rendering using packed glyph atlas and dedicated element pipelines
- Custom 1.4 million particle real-time smoke simulation without perceptible lag.
- Responsive high-DPI layout, image-aware text flow with keyboard, pointer and touch controls.
- Optimised performance via dirty-driven rendering, retained GPU resources, memory-conscious image decoding and automated WGSL validation.

AI-Assisted PDF-to-eBook Conversion Tooling — In Progress

Python | CPython | AI/ML | ARM NEON | SIMD | Document Processing

Python tooling to reconstruct PDF content into structured, eBook-ready documents.

- Automated extraction, categorisation and content assembly
- CPython extension using ARM NEON SIMD for hardware-accelerated content analysis.
- Dissimilarity-based categorisation for context aware extraction of complex PDF layouts.
- AI/ML-assisted processing to identify document structure for coherent reading order.