Stuart Yeadon

Low level Systems Engineer
C Rust Python ACPI

Email stuartyeadon@gmail.com
Telephone +44 7525 435 792
Website syeadon.info

Professional Summary

Apple Genius with 12 years experience diagnosing hardware and software across Apple's complete product and service ecosystem, combining first-class customer service with engineering skills in C, Rust, Python and ACPI-based systems.

Triages diagnostic reports, panic logs and user reported issues to troubleshoot faults across the full hardware-to-user software stack, isolating their root cause.

Brings a unique blend of product knowledge, engineering skills and customer interaction seeing first-hand how issues affect the customer experience.

Programming Skillset

C	Embedded systems, peripheral drivers and direct-to-register systems programming.
Rust	WGPU graphics, web_sys DOM JS interfacing, memory safety, ownership & lifetimes
Python	Local LLM tooling CPython, Flask, NumPy, nginx and API integration.
ACPI	Intel firmware analysis, DSDT/SSDT modification, hardware enumeration and injection
Debugging	Breakpoints, stepping, execution control, live expressions and runtime state inspection; familiar with LLDB/GDB debugging concepts and workflows.
Frameworks	XNU / Darwin, IOKit, Kernel Panics, ARM, x86-64, Ubuntu Server
CLI Toolchain	git, Clang, GCC, Cargo, Make, Crate, Bash, ftp, ssh

Education

2025	Certificate	**ARM & Embedded Systems**	Coursera | **View Certificate**
22-25	Certificate	**Computer Science**	Codecademy | **View Certificate**
2012	BSc (Hons)	**Commercial Music - 2:2**	Bath Spa University
07-09	A Levels	**Mathematics - B**	Bishop Vesey's Grammar School

Personal Interests

- **Travel & Landscape Photography** — Highlights include the Dolomites, Monkeys of Marrakesh and the Quiraing Circuit of Isle of Skye. See website for examples.
- **Grade 8 Guitarist & Classical Pianist** — Self-taught piano when 4 years old playing in school assembly. Achieved Grade 8 by Year 10. Highlights include yachting to a gig via Eric Clapton's yachtsman.
- **Charity fundraising** — Taking my espresso machine to store to connect with team members and raise money for important causes such as disaster appeals and local charities.

Technical Projects

macOS Nuvoton Super I/O Driver (Nuvoton NCT6796D)

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

A kernel extension driver and user-client tool for fan control and temperature readings of an Intel mac.

Features:

- Register-level communication with Nuvoton Super I/O hardware.
- User-client CLI tool for interaction and user settings control.
- Fan RPM measurement and MMIO modification for post boot BIOS modification.
- Custom fan control curves and speeds.
- Created a RAM debugging tool akin to WinHex for read access to system RAM.

Challenges:

- Learning Objective-C and C++ for the first time.
- Interpreting the cause of kernel panics by their specific caller, failure reason and frame pointer.
- Learning the IOKit boot hierarchy when kext injected failed to probe and target the hardware.

Solar-Powered Greenhouse Environmental Control System

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

An ARM Cortex autonomous smart greenhouse w/ lighting, heating and solar power.

Features:

- Bare-metal C direct-to-hardware firmware with direct register manipulation.
- Allocating pin I/O resources to drive multiple sensors whilst honouring hardware limitations.
- GPIO, ADC, I²C, RTC, Timers, PWM generation and 1-Wire protocol implementation.
- Hand soldered logic board, power supply and fused power distribution network.
- Hardware initialisation, fault diagnosis, and signal-level debugging.
- Heat-transfer applied physics to predict soil temperatures for optimal growth.
- A modular design for easier maintenance and installation.

Challenges:

- Diagnosing 1-Wire connection issues without dedicated debugging hardware.
- Manually timing heater power delivery due to chipset timer count constraints.
- Learning the STM32 development toolkit and IDE.
- Constructing the greenhouse using a variety of woodworking assembly techniques.

Future Tasks:

- Integrating matter-over-thread technology for HomeKit and remote monitoring support
- Full irrigation with gravity fed pumps and rain water reservoir for automatic watering.

GPU-Accelerated Terminal Rendering Engine

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

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

Features:

- Built an instanced rendering pipeline using indexed quad geometry and a GPU-sampled glyph atlas.
- Designed compact, alignment-safe data structures shared between Rust and WGSL.
- Implemented high-DPI rendering, dirty-region tracking and animation-frame scheduling.
- Managed long-lived WebGPU resources within Rust's ownership model.
- Maintained performance within a browser's single-threaded execution environment.
- Event-driven architecture using browser animation-frame scheduling and persistent Rust state.
- Modular rendering architecture designed to support text, shapes, images and post-processing effects.

Challenges:

- Learning the entire rendering framework from vertex to shader to buffer to layout to encoder to pass.
- Memory alignment and compression techniques to allow the system to scale efficiently.
- Debugging uncertain panics masked by mutable-access getters obscuring the caller's trace.
- Learning the cargo and wasm32 build requirements and packaging mechanisms

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

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

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

Features:

- A custom 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 and reconstruct coherent reading order.
- Automating extraction, categorisation and content assembly to reduce manual conversion work.

Challenges:

- Designing the pipeline for repeatable batch processing, validation and future optimisation.
- Detecting bottlenecks that caused spikes in CPU usage within a single thread.
- Reading about statistical learning, cluster analysis and categorical dissimilarity measures.