

ZUHAITZ MÉNDEZ FERNÁNDEZ DE ARÁNGUIZ

Canary Islands, Spain (Open to relocation)

zuhaitz.dev zuhaitz.zechhub@gmail.com github.com/Zuhaitz-dev @zuhaitz_dev

OVERVIEW

Systems software engineer and technical educator specializing in compiler toolchains, low-level architecture, and bare-metal code generation. Deeply focused on Mechanical Sympathy, custom allocators, and hardware-oriented design. Having architected widely-used C compilers and foundational libraries utilizing strict memory safety and type-safe paradigms, I am currently applying my bare-metal systems expertise to high-performance Rust environments.

CORE ARCHITECTURE & OPEN SOURCE

Zen C Compiler | *Creator & Lead Architect (4,200+ GitHub Stars)* zenc-lang/zenc

- **Compiler Architecture:** Designed Zen C as a high-level systems protocol that transpiles directly to C, utilizing established backends for machine-code optimization while enforcing modern language safety.
- **Polyglot Interop:** Engineered native support for C++, Objective-C, and CUDA, ensuring seamless ABI-level interoperability and GPGPU computing capabilities within a single toolchain.
- **Safety Paradigms:** Architected a custom generics system and a rigorous 20+ job CI/CD pipeline enforcing MISRA C compliance, ASAN/UBSAN memory sanitization, and complexity limits.
- **Portability:** Integrated Cosmopolitan Libc to output Actually Portable Executables (APE), allowing native execution across Linux, macOS, and Windows.

z-libs (Classic C Libraries) | *Creator & Maintainer* github.com/z-libs

- **Adoption & Impact:** Engineered a suite of battle-tested, single-header C11 libraries (*zvec*, *zstr*, *zmap*, *zalloc*) utilized by industry leaders, including Tobi Lütke (CEO of Shopify).
- **C/C++ Interoperability:** Designed a seamless boundary using C11 `_Generic` macros alongside zero-overhead C++ RAII wrappers, eliminating `void*` type-erasure.
- **Allocator-Aware:** Built custom arena and pool allocators with built-in leak detection to ensure cache-friendly, high-performance memory management.

Bare-Metal Systems (b-compiler-x64 & forth-nasm-x64) | *Systems Programmer* GitHub Profile

- **Instruction Set Architecture:** Developed an x86-64 code-generating B compiler and a Forth system written in pure Assembly (NASM) to master hardware interfacing and stack-machine execution models.
- **Low-Level Execution:** Implemented direct system call handling, linear memory arrays, and indirect function dispatch tables without relying on standard library abstractions.

wasm32-tcc (Compiler-as-a-Library) | *Core Developer* [In Development / Unreleased]

- **Backend Engineering:** Engineered a custom WebAssembly backend patch for the TinyCC code generator, enabling fast, one-pass C compilation in sandboxed environments by bridging C memory semantics to the WASM stack machine.

TECHNICAL PRIMITIVES

- **Languages:** C (GNU C/MISRA), Zen C, C++, x86-64 Assembly, WebAssembly (WASM), CUDA, Lua, Rust.
- **Systems Concepts:** Memory-mapped I/O, Custom Allocators (Arenas/Pools), Data-Oriented Design (DOD), Pointer Aliasing, Type-Safe Macros, Instruction Set Architectures.
- **Tooling & Safety:** ASAN / UBSAN, Valgrind, GDB, Make, Actually Portable Executables.

EDUCATION

- **B.S. Computer Engineering (In Progress):** Universidad de La Laguna (ULL), Spain.
- **Dual High School Diplomas (USA & Spain):** Awarded a merit-based scholarship to concurrently earn an American High School diploma, ensuring native-level English proficiency for international engineering teams.