

# [Alberto] [Damo] | Kernel & Systems Software Developer

[Oderzo, Italy]

✉ [alberto.damo@proton.me] • 🌐 github.com/mrmonopoly-cyber  
in alberto-damo-62b81b23a • Master's Student in Computer Science

## Profile

---

Systems and embedded software developer specializing in **low-level C programming**, **operating system internals**, and **bare-metal development**. Currently developing a **UEFI-compliant OS from scratch**, including a custom bootloader, kernel, and disk image generation utility for QEMU (Tianocore). Passionate about Linux, kernel programming, and open-source collaboration. Seeking to contribute to IBM's Linux kernel and virtualization initiatives while deepening expertise in systems architecture and performance.

## Technical Skills

---

**Programming:** C (expert), Assembly (x86\_64 / TriCore basics), Shell scripting

**Systems:** Linux internals, bootloaders, UEFI, memory management, kernel architecture

**Embedded:** AURIX TC375 (TriCore), tricorn-elf-gcc, linker scripts, atomic operations, GPIO & CAN control

**Tools:** Git, GCC, GDB, QEMU, OVMF (Tianocore), libgpiod, SocketCAN, CI/CD

**Concepts:** Concurrency, interrupts, synchronization, atomic operations, virtualization fundamentals

**Open Source:** GitHub project maintainer; enforces branch protection and CI for integrity

## Education

---

[Your University Name]

Master's Degree in Computer Engineering

[Expected Graduation Year]

Focus areas: Operating Systems, Embedded Systems, Computer Architecture.

## Selected Projects

---

**UEFI-Compliant Operating System from Scratch**

*Custom OS*

2025–Present

- Designed and implemented a **UEFI bootloader** and **monolithic kernel** entirely in C and Assembly.
- Built a UEFI-compliant **disk image generation utility** compatible with QEMU (OVMF / Tianocore).
- Developed early boot stages including **memory initialization**, console I/O, and process management foundations.

**Linux-Based Multi-Component System Simulator**

*ControlUnitLogicOperator*

2023–Present

- Built a modular simulation environment of 46 emulated components communicating via shared CAN buses and GPIOs.
- Implemented a **runtime trace stack** for error diagnostics using `__FILE__` and `__LINE__`.
- Designed a **Safety Critical System (SCS)** with atomic synchronization across threads.

**Command-Line Debugging Interface**

*DPS (Debug Peripheral System)*

2023–Present

- Created a **CLI-based debugging tool** supporting live monitoring and introspection of system state.
- Provides modular runtime telemetry for GPIO and CAN buses.

**Bare-Metal Safety-Critical Software**

*AURIX TC375 Firmware*

2024

- Developed firmware for Infineon AURIX TC375 using custom linker scripts and atomic synchronization.
- Designed deterministic exception-safe handling without standard libraries.

## Experience

---

### *Independent Systems Developer*

**Self-directed**

*2023–Present*

- Designed architectures for both embedded and OS-level systems with a focus on portability and maintainability.
- Built custom debugging infrastructure with traceable runtime analysis.
- Mentored peers on sustainable system design and CI/CD integration.

## Achievements

---

- Currently building a fully bootable UEFI-compliant OS from scratch.
- Published open-source systems demonstrating atomic synchronization and debugging architecture.
- Built a custom 32-bit Linux environment for emulation and testing.

## Additional Information

---

**Languages:** Fluent in English

**Interests:** Kernel engineering, virtualization, systems security, open-source development

**Goal:** Contribute to IBM's Linux kernel and confidential container initiatives