Tyler Potyondy

website | linkedin | github

Education

UC San Diego Sep. 2022 – June 2027 (expected)

Ph.D. Computer Science and Engineering – advised by Pat Pannuto

UCLA Sep. 2018 – June 2022

B.S. Bioengineering, minor Bioinformatics

Research Publications

Abacus: Type-states for Increased Driver Correctness ASPLOS 2026

- Designed and built a Rust system to statically prevent driver device protocol violations—highest patched driver bug class in linux drivers (38%). Ported system to drivers in TockOS and RedoxOS (e.g., xHCI, 15.4 radio, uart).
- System incurs zero-cost overhead in binary code size and runtime performance. System detected 2 bugs in ported drivers. Provided +17.4% performance increase to RedoxOS xHCI driver.

Omniglot: Rust FFI Framework *(Best Paper Award)* OSDI 2025, RustConf 2025

- Built first Rust FFI framework that allows efficiently interacting with unsafe libraries (e.g., C code) while maintaining Rust safety and soundness. Introduces negligible overheads (0-3.4% for a given C library).
- Integrated libraries included in prototype: CryptoLib, LittleFS, LwIP, Brotli, libsodium, libpng.

TickTock: Verified Isolation in TockOS SOSP 2025

- Formally verified Tock's MPU implementation and process isolation using Flux (SMT based Rust verifier). Discovered 6 bugs breaking process isolation in Tock.

Experience

TockOS - Maintainer and Contributor April 2023 – Present

- 13th overall contributor to Tock, an embedded Rust OS that is deployed on millions of devices and used by Google, Microsoft, and OpenTitan. 31 merged PRs, contributed 4.2k LoC (net +3.2k)
- Maintain Tock's IEEE802.15.4 Networking and OpenThread Stack, cryptography working group lead, and member of the network working group.

Teaching Assistant - UCSD Fall 2023, Spring 2025, Winter 2026

- Taught undergrad intro. to embedded systems, computer architecture, and wireless networking courses.
- Instruct students in writing bare metal C drivers for STM32 and implementing wireless networking stacks.

Projects

OpenThread | TockOS Port April 2023 – June 2024

- Designed and implemented port of OpenThread (Google's open source Thread implementation, considered "WiFi" of low power IoT) to run as Tock userspace process. My work made Tock the first Rust project to provide OpenThread support.
- Implemented custom kernel ringbuffer to handle tx/rx timing and latencies arising to OpenThread running as a userspace process vs. standard OpenThread bare-metal platforms. Achieve 5KiBps goodput.

ENTS (Environment NeTworked Sensing) | Tock Migration June 2025 – Dec. 2025

- ENTS is an open source hw/sw platform used by environmental researchers (100+ actively deployed nodes).
- Designed and implemented port of bare-metal C ENTS firmware to Tock (9.5K LoC). Integrated RadioLib LoRaWAN as userspace process. Refactored ENTS software architecture to be mutually distrustful and robust.

Tock Cryptography Interface Redesign June 2025 – Current

- Drafted RFC for cryptography interface redesign (virtualizable and enforce configuration/mode via Rust type system).
- RFC contributed to creation of Tock cryptography working group; ongoing work redesigning interface in conjunction with Microsoft's PlutonOS team and ZeroRisc (downstream Tock users).

nrf-rpc: Rust implementation of Nordic RPC client Feb. 2026 – Current

- Built open source Rust client for nordic's nrf-rpc protocol (custom protocol to enable inter processor communication).
- Allows usage of Nordic's certified BLE stack without compromising Rust safety guarantees or facing BLE timing constraints.

Languages: Rust, C/C++, Python, Go

Skills: JLink, OpenOCD, Make, GDB, Kernel/OS development, RTOS, Zephyr, Embedded Rust (no_std)