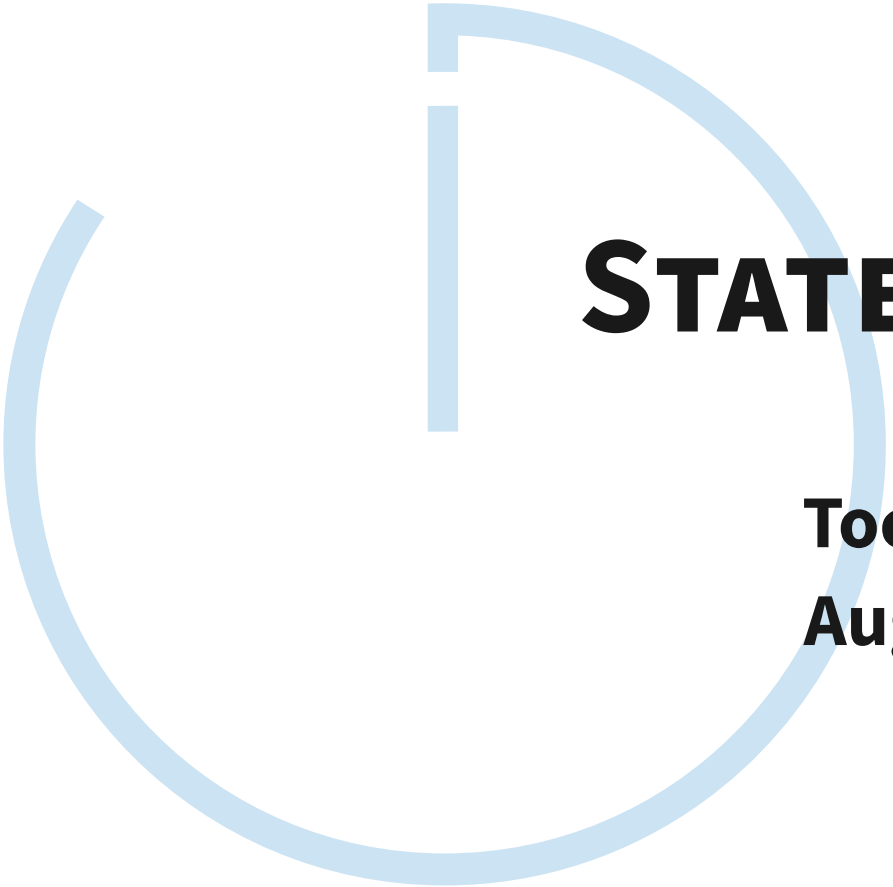
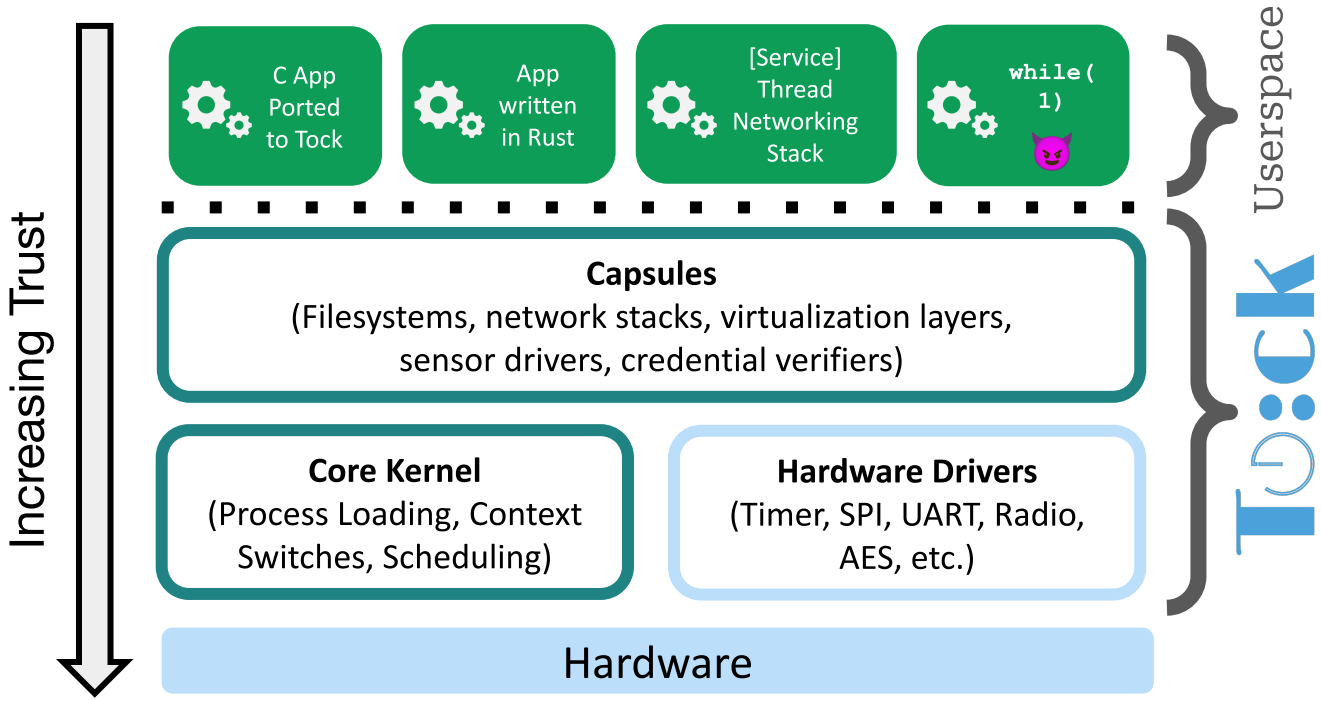


State of Tock

TockWorld 8
August 2025

What is Tock?


C App Ported to Tock
App written in Rust
[Service] Thread Networking Stack
while(1)
Userspace
Increasing Trust
Capsules
(Filesystems, network stacks, virtualization layers, sensor drivers, credential verifiers)
Core Kernel
(Process Loading, Context Switches, Scheduling)
Hardware Drivers
(Timer, SPI, UART, Radio, AES, etc.)
Hardware
Tock

What is Tock?

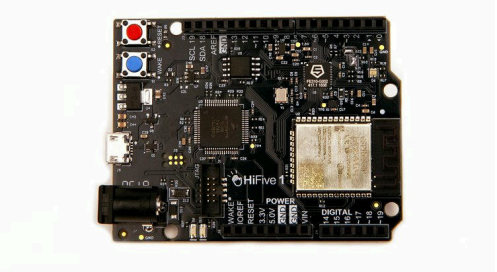
A hardware-root-of-trust OS

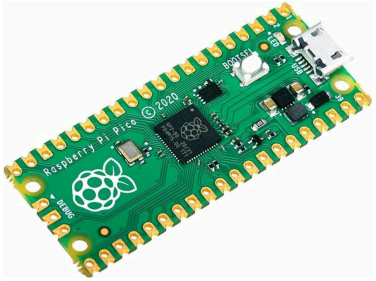
Chromebook

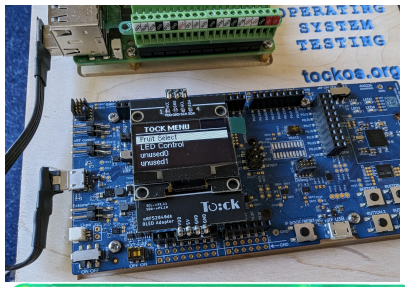
Pluton (on Copilot+ Laptops)

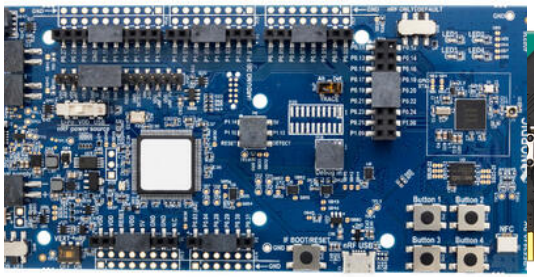
What is Tock?

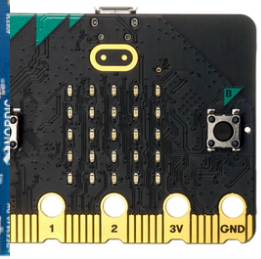
An operating system for embedded microcontrollers

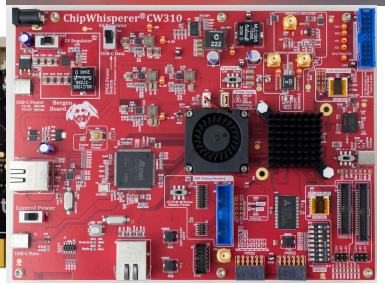


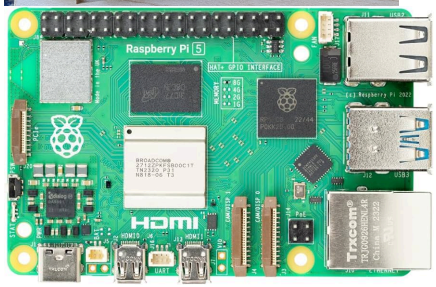


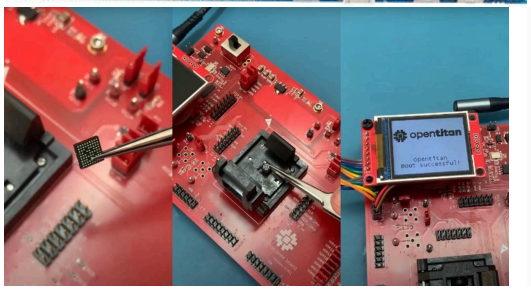












What is Tock?

A community of developers, users, practitioners, researchers with shared goals.

Extensibility at the core

- Userspace / kernel separation
- Capsules
- System call interface
- Memory isolation mechanisms
- Scheduler
- Access control
- …

Pragmatic use of formal and socio-technical tools

Formal

- Type-safety
- Hardware support for strong isolation
- Verification
- Careful reasoning about safety implications

Socio-Technical

- Separation of critical vs. non-critical
- Careful code review
- Rigorous testing
- Slow and steady design and progression

Co-development of

- Hardware
- Language
- Kernel
- Applications

Open source collaboration

- Practitioners
- Researchers
- Educators

A Year of Contributions

	Commits	PRs Merged	Contributors
Tock	1176	278	62
libtock-c	341	73	19
libtock-rs	27	11	7
tockloader	40	5	7
book	146	18	10

A long time ago, in a galaxy far away…

From: Philip Levis
Subject: [helena-project] SenSys poster/demo
To: helena-project@lists.stanford.edu
Date: Wed, 09 Jul 2014 13:15:12 -0700

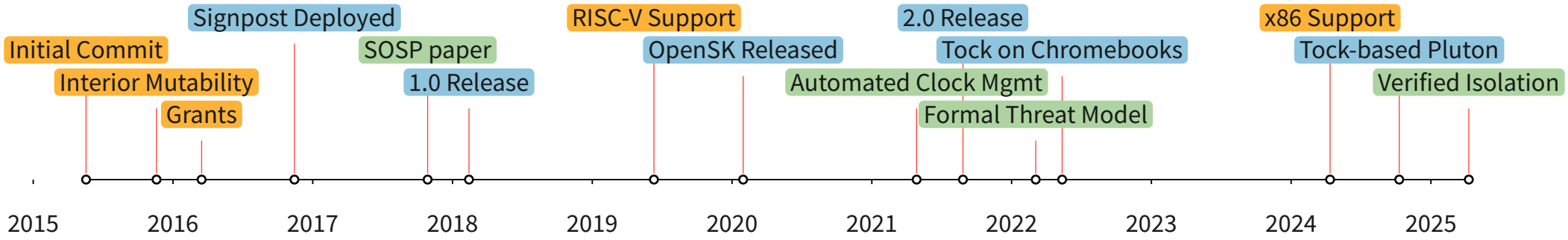
> ...
>
> Operating system: what should an operating system for such a device look like? Can we achieve something like the efficiency and dependability of TinyOS without being so difficult to extend and program?

Tock is 10 years old!

```
commit a14379b850bf47e89cd2945226cbf9bcbab5f43f
Author: Amit Aryeh Levy <amit@amitlevy.com>
Date: Tue May 19 15:29:44 2015

Initial commit
Barebones build system and boot to Rust on Storm
```

A Decade of Tock



2016: Dynamic userland code loading

2017: Tock training at RustConf, first deployment (Signpost)

2018: 1.0 release

2019: RISC-V support

2020: Pluggable scheduler

2021: 2.0 release, revised system call interface

2022: Subscribe & allow handled by kernel & read-only shared buffers

2022: Signed applications

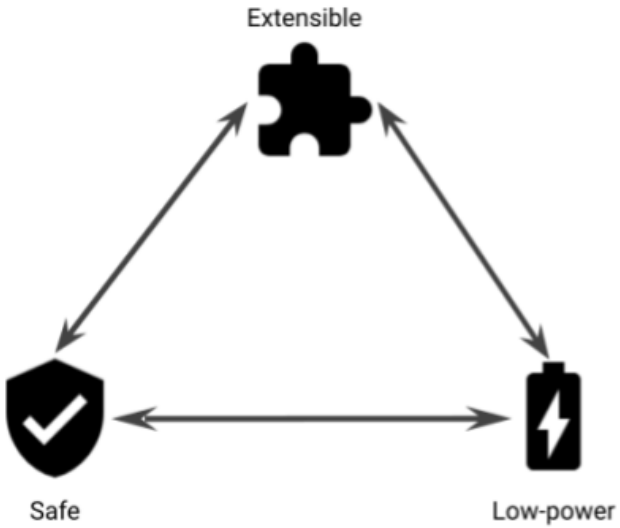
2024: Kernel compiles on stable Rust

2025: x86 Support

Tock at 10

Designed to be

- Safe & Secure
- Multi-programmable
- Resource Efficient



Based in Research

Built with Industry

- Google: OpenSK, Ti50, Pixel
- Microsoft: Pluton, Caliptra
- OxidOS Automotive
- zeroRISC
- HPE
- Infineon
- AMD

Tock in 2025 and Beyond

Exciting new frontiers

- Integrating non-Rust into the kernel
- Reusable userspace processes with better IPC
- New hardware capabilities
  - Virtual memory
  - CHERI
  - TrustZone
  - Multi-MCU / Multi-Core

Expanding Beyond Root-of-Trust Hardware

- Medical devices
  - Security in medical devices is a disaster
  - Newer regulations require better security practices
  - Loads of legacy C applications
  - Low resources, battery powered, networked
- Real sensor networks
- Bigger and beefier platforms
  - Raspberry PI-scale computers want “better” operating systems too

Technical Pain Points

- Code size still a challenge
  - Even more so with integrated chips without executable flash
- Security and trust
  - Streamlined vulnerability reporting and triage
  - Testing and reliability
- Zero-copy support without fate sharing

Ergonomic Pain Points

- Rust async and other programming patters
- Process relocation
- Dependency management

Tock Foundation

What is the Tock Foundation?

- A “new”[1] non-profit that supports education, research and development in secure operating systems
- Shepherds the Tock open source project:
  - Source code
  - Working groups
  - Events (like this one!)
  - Outreach
- Advocacy for *fundamentally* improving systems security

[1] Legally established in 2023, but just launched

The Tock Foundation:

Hires Engineers

- Safe MMIO Registers
- Rust userland
- HW-based continuous integration

The Tock Foundation:

Trains and Educates

Tock OS Book

This book introduces you to Tock, a secure embedded operating system for sensor networks and the Internet of Things. Tock is the first operating system to allow multiple untrusted applications to run concurrently on a microcontroller-based computer. The Tock kernel is written in Rust, a memory-safe systems language that does not rely on a garbage collector. Userspace applications are run in single-threaded processes that can be written in any language.

Getting Started

The book includes a quick start guide.

The Tock Foundation:

Broadens Tock's use cases

- Medical devices
- High-resilience sensing
- Payment, identity, authentication "everywhere"

The Tock Foundation:

Secures the Open Source Ecosystem

- Tooling for securing software supply chain
  - Dependency auditing
  - Mitigating Rust soundness holes
- Defense in depth
  - Hardening system call ABI
  - Code “Trust Tiers”
- Systematized code review for security
  - Detect modifications to sensitive code
  - Automating “good” patterns in Tock

The Tock Foundation:

Research & Development

- Verifying safety beyond Rust semantics
- Low-bandwidth/high-latency OTA updates
- Code-size reduction:
  - Panic-free kernel, vtable optimization, compiler improvements, …
- Automatically Translating C-to-Rust

TockWorld 2025

Session I - 10:15–12:00

Battling & leveraging Rust types

Lunch Break

Session II - 12:45–2:15

Compilers, hardware, and cores, oh my!

Break

Session III - 2:45–4:30

Hardened, better, faster, stronger

https://tockworld8.sessionize.com

WiFi: Microsoft Guest
Event Code: “TockWorld”

Chat (Matrix): #tockworld8:tockos.org
https://matrix.to/#/#tockworld8:tockos.org