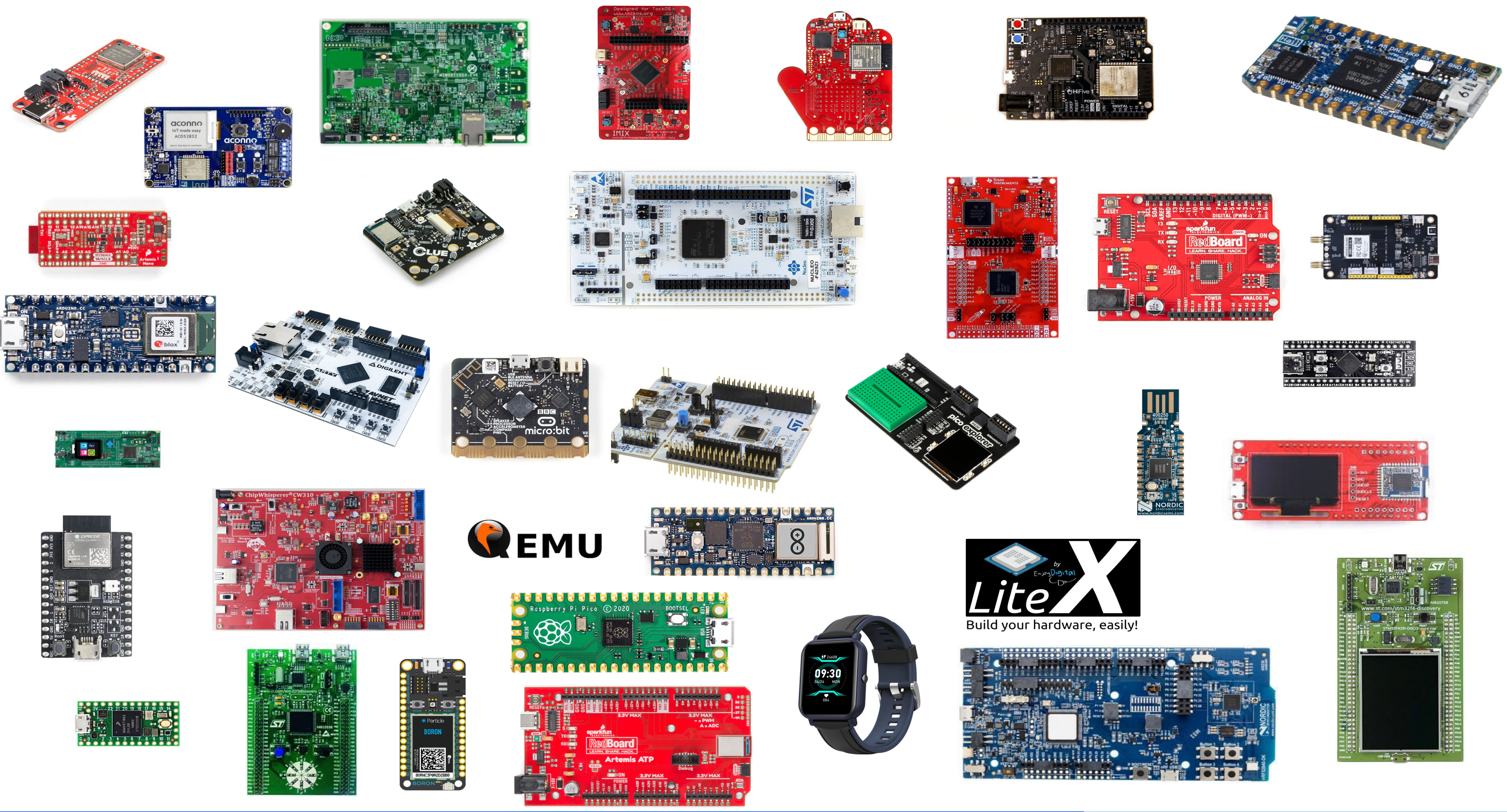
The Treadmill Distributed Hardware Testbed

TockWorld 7 – June 26, 2024
Leon Schuermann

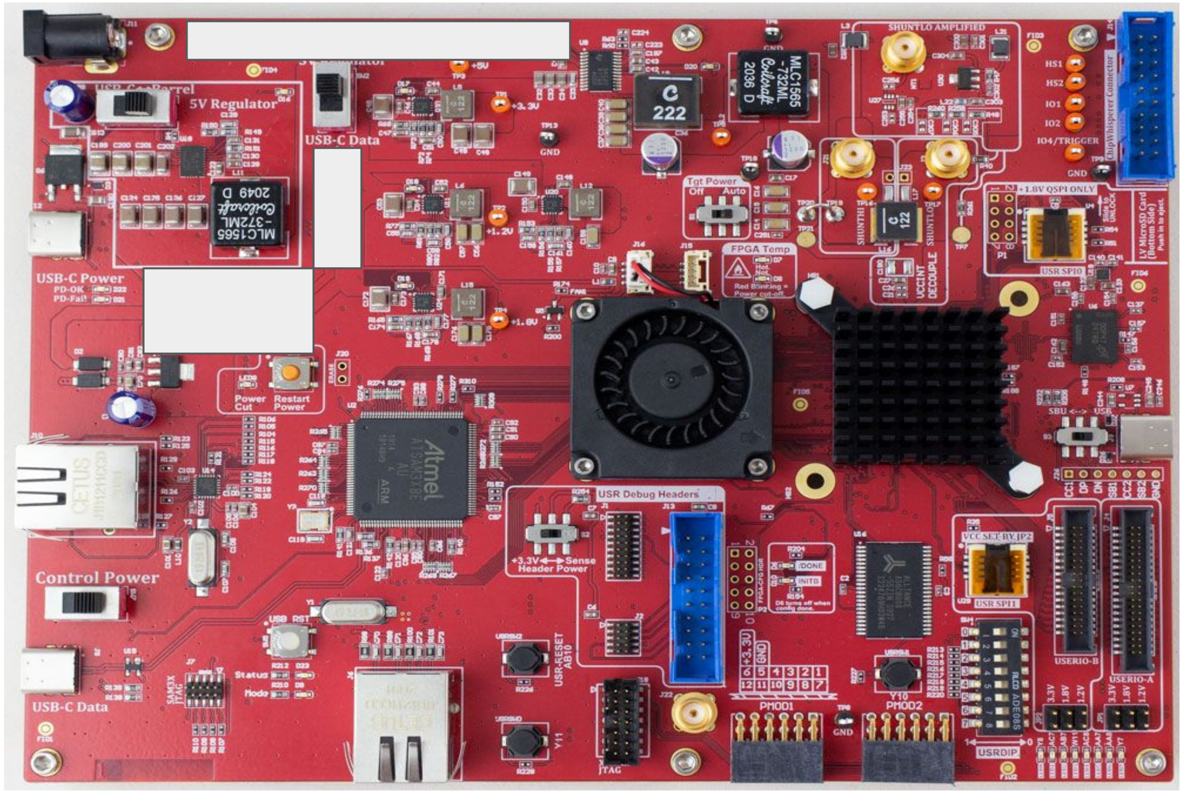
Tock lacks automated HW testing

- Today: low assurance that a change will not break boards / subsystems
  - HW tests require time + effort
  - No standardized test workflow:
    userspace examples, kernel unit tests, kernel integration tests
  - Interactions between hardware peripherals break isolated software components in subtle ways
- High testing effort for releases
  → Long delay between releases
  → Lots to test, hard to run them, knowledge around tests is lost

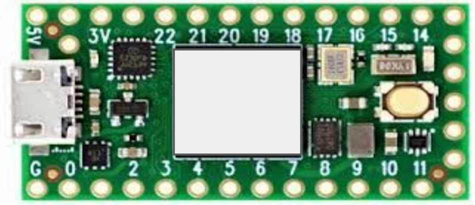
Tock supports *lots* of boards!


aconno
IMIX
HiFive1
CLUE
DIGILENT
ARTY
micro:bit
pico explorer
NUCLEO
RedBoard
Artemis Nano
NORDIC
ChipWhisperer CW310
QEMU
LiteX
Build your hardware, easily!
Raspberry Pi Pico
BORON
RedBoard Artemis ATP
09:30
www.st.com/stm32f4-discovery

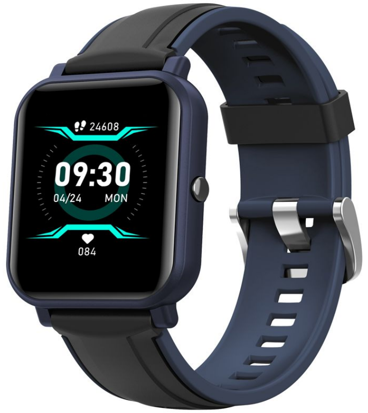
Name That Board!



Name That Board!



Name That “Board”!



Tock supports *lots* of boards

… some niche boards, used by only a few contributors

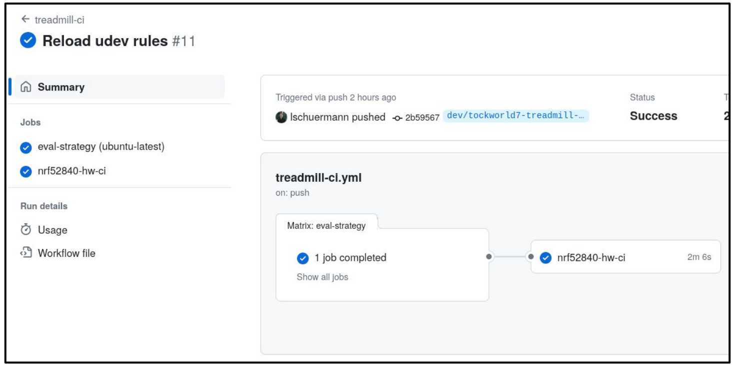
… some very expensive boards, infeasible to acquire (multiple of)

… some proprietary HW, which we'll not get our hands on

… some with heavy-weight / hard-to-use toolchains

- Difficulties getting these targets tested, e.g., for releases
- Maintenance- and refactoring-changes get merged without even basic testing

The Treadmill Distributed Hardware Testbed



← treadmill-ci

✓ **Reload udev rules** #11

Summary

Jobs

✓ eval-strategy (ubuntu-latest)

✓ nrf52840-hw-ci

Run details

Usage

Workflow file

Triggered via push 2 hours ago

lschuermann pushed -o- 2b59567 dev/tockworld7-treadmill-…

Status

Success

treadmill-ci.yml

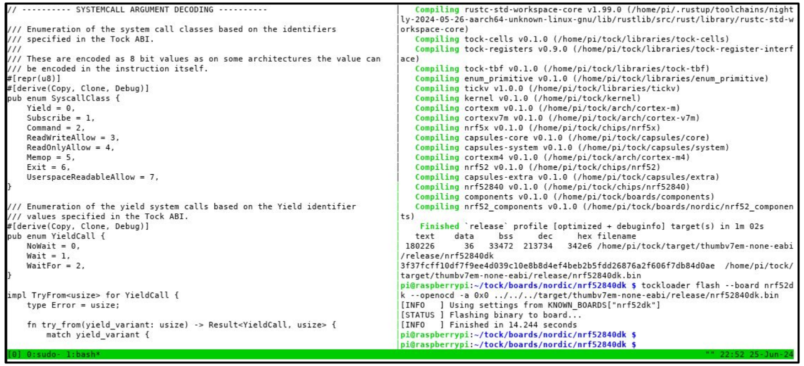
on: push

Matrix: eval-strategy

✓ 1 job completed

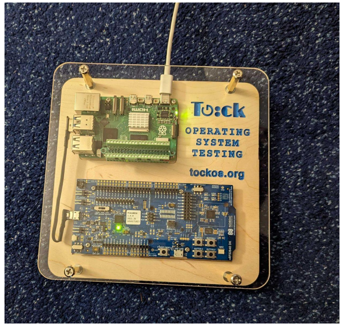
Show all jobs

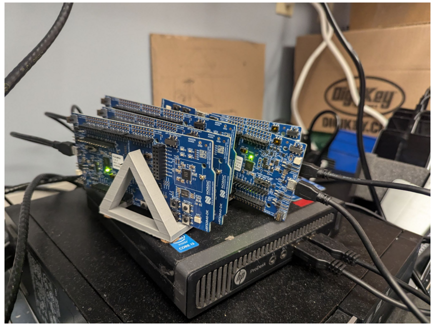
✓ nrf52840-hw-ci 2m 6s



```
// ---------- SYSTEMCALL ARGUMENT DECODING ----------

/// Enumeration of the system call classes based on the identifiers
/// specified in the Tock ABI.
///
/// These are encoded as 8 bit values as on some architectures the value can
/// be encoded in the instruction itself.
#[repr(u8)]
#[derive(Copy, Clone, Debug)]
pub enum SyscallClass {
    Yield = 0,
    Subscribe = 1,
    Command = 2,
    ReadWriteAllow = 3,
    ReadOnlyAllow = 4,
    Memop = 5,
    Exit = 6,
    UserspaceReadableAllow = 7,
}

/// Enumeration of the yield system calls based on the Yield identifier
/// values specified in the Tock ABI.
#[derive(Copy, Clone, Debug)]
pub enum YieldCall {
    NoWait = 0,
    Wait = 1,
    WaitFor = 2,
}

impl TryFrom<usize> for YieldCall {
    type Error = usize;

    fn try_from(yield_variant: usize) -> Result<YieldCall, usize> {
        match yield_variant {
[0] 0:sudo- 1:bash*
```

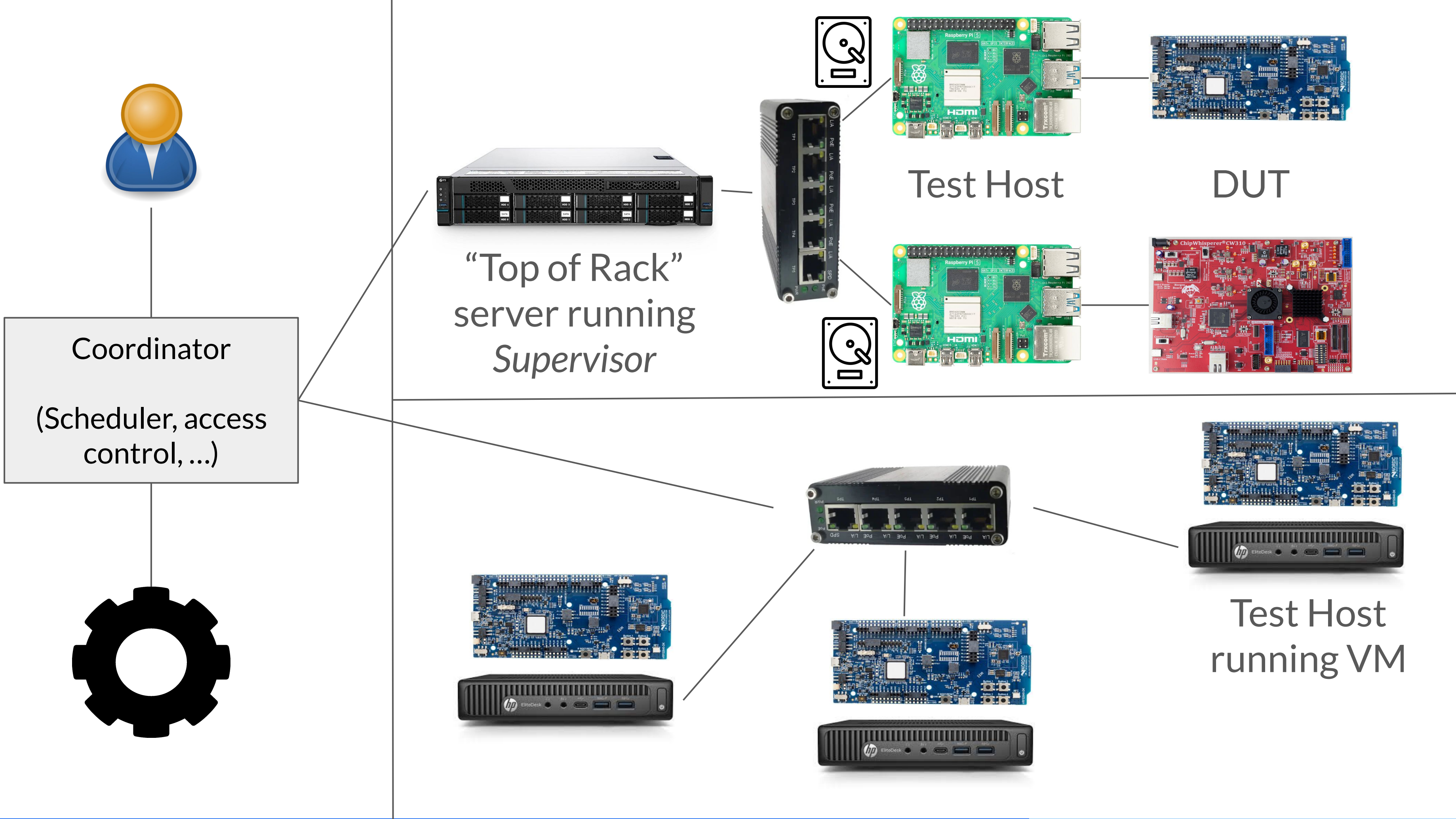
```
   Compiling rustc-std-workspace-core v1.99.0 (/home/pi/.rustup/toolchains/night
ly-2024-05-26-aarch64-unknown-linux-gnu/lib/rustlib/src/rust/library/rustc-std-w
orkspace-core)
   Compiling tock-cells v0.1.0 (/home/pi/tock/libraries/tock-cells)
   Compiling tock-registers v0.9.0 (/home/pi/tock/libraries/tock-register-interf
ace)
   Compiling tock-tbf v0.1.0 (/home/pi/tock/libraries/tock-tbf)
   Compiling enum_primitive v0.1.0 (/home/pi/tock/libraries/enum_primitive)
   Compiling tickv v1.0.0 (/home/pi/tock/libraries/tickv)
   Compiling kernel v0.1.0 (/home/pi/tock/kernel)
   Compiling cortexm v0.1.0 (/home/pi/tock/arch/cortex-m)
   Compiling cortexv7m v0.1.0 (/home/pi/tock/arch/cortex-v7m)
   Compiling nrf5x v0.1.0 (/home/pi/tock/chips/nrf5x)
   Compiling capsules-core v0.1.0 (/home/pi/tock/capsules/core)
   Compiling capsules-system v0.1.0 (/home/pi/tock/capsules/system)
   Compiling cortexm4 v0.1.0 (/home/pi/tock/arch/cortex-m4)
   Compiling nrf52 v0.1.0 (/home/pi/tock/chips/nrf52)
   Compiling capsules-extra v0.1.0 (/home/pi/tock/capsules/extra)
   Compiling nrf52840 v0.1.0 (/home/pi/tock/chips/nrf52840)
   Compiling components v0.1.0 (/home/pi/tock/boards/components)
   Compiling nrf52_components v0.1.0 (/home/pi/tock/boards/nordic/nrf52_componen
ts)
    Finished `release` profile [optimized + debuginfo] target(s) in 1m 02s
   text    data     bss     dec     hex filename
 180226      36   33472  213734   342e6 /home/pi/tock/target/thumbv7em-none-eabi
/release/nrf52840dk
3f37fcff10df7f9ee4d039c10e8b8d4ef4beb2b5fdd26876a2f606f7db84d0ae  /home/pi/tock/
target/thumbv7em-none-eabi/release/nrf52840dk.bin
pi@raspberrypi:~/tock/boards/nordic/nrf52840dk $ tockloader flash --board nrf52d
k --openocd -a 0x0 ../../../target/thumbv7em-none-eabi/release/nrf52840dk.bin
[INFO   ] Using settings from KNOWN_BOARDS["nrf52dk"]
[STATUS ] Flashing binary to board...
[INFO   ] Finished in 14.244 seconds
pi@raspberrypi:~/tock/boards/nordic/nrf52840dk $
pi@raspberrypi:~/tock/boards/nordic/nrf52840dk $
                                                           "" 22:52 25-Jun-24
```





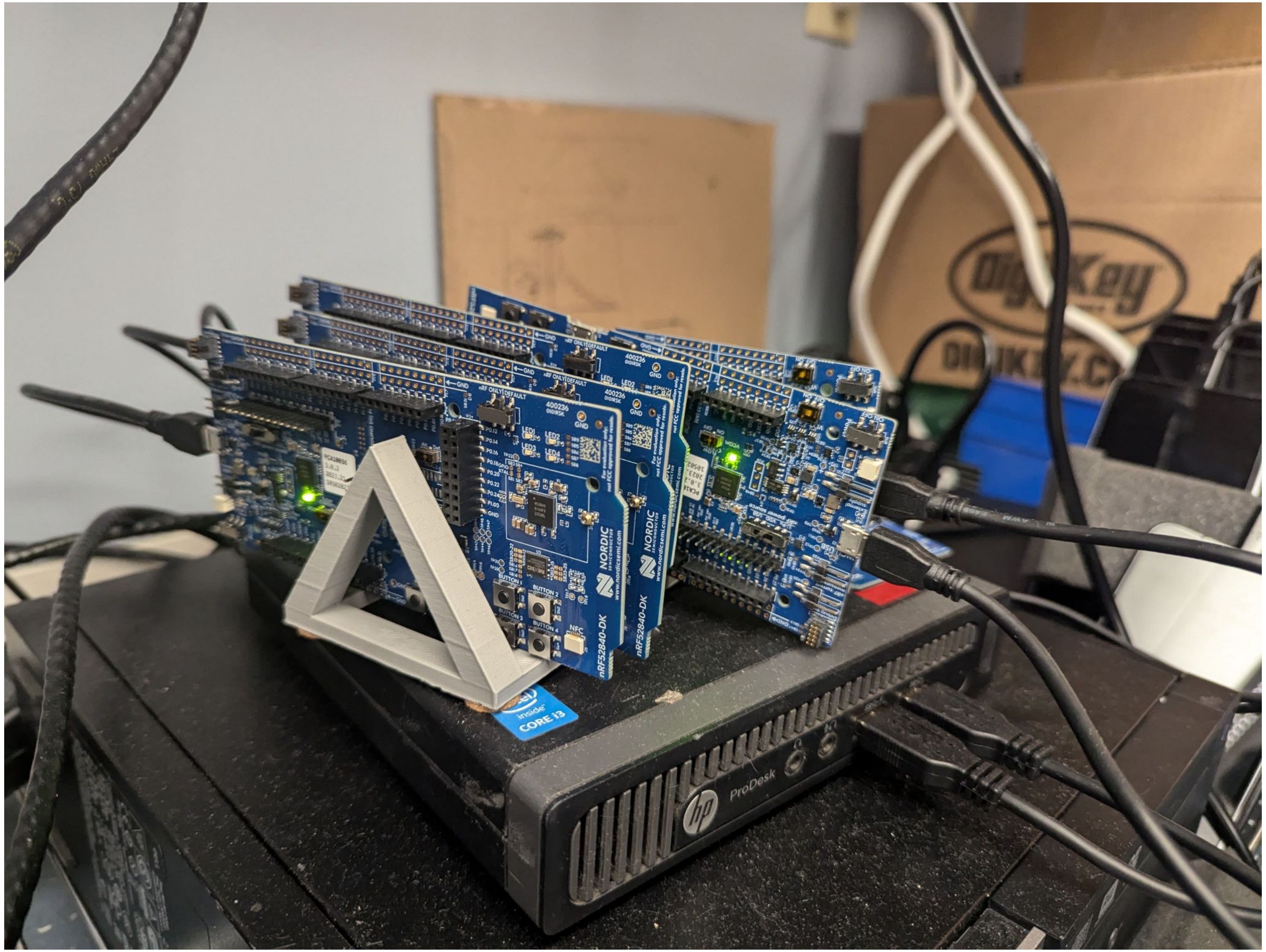
Goal: A Distributed, Reliable Testbed for Development + CI

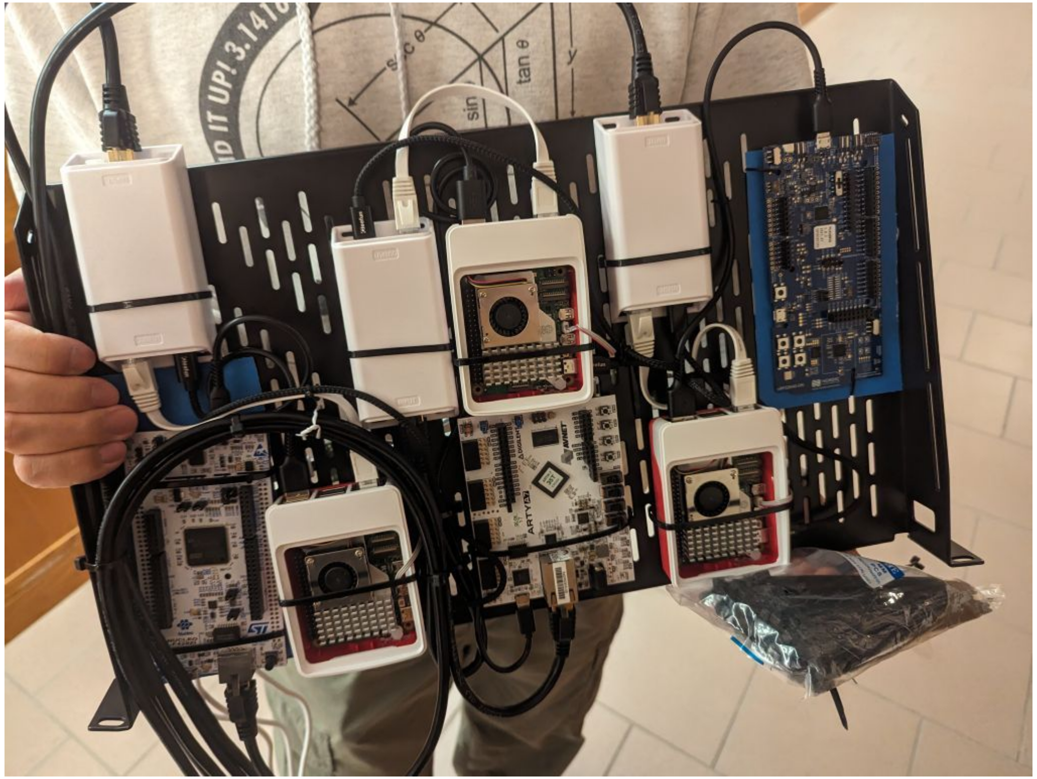
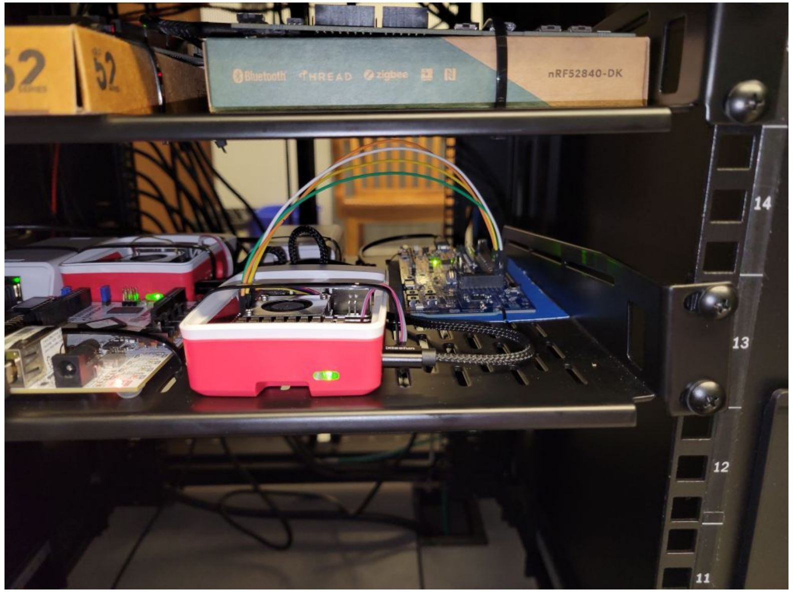
- Physically distributed across multiple different sites
  - Research institutions: UVA, UCSD, Princeton, …
  - At companies & downstream users; adding downstream targets into the upstream CI
- Reliable
  - Schedule among set of available boards
  - Retry on different HW in case of hardware failure, network outage, etc.
- Accommodate diverse testing workloads
  - Layer of abstraction: Linux environment with HW access
  - (Optional) access to hardware peripherals / GPIO
- Secure
  - Isolate different test jobs
  - Access control for individual boards (restrict type of workload & user access)

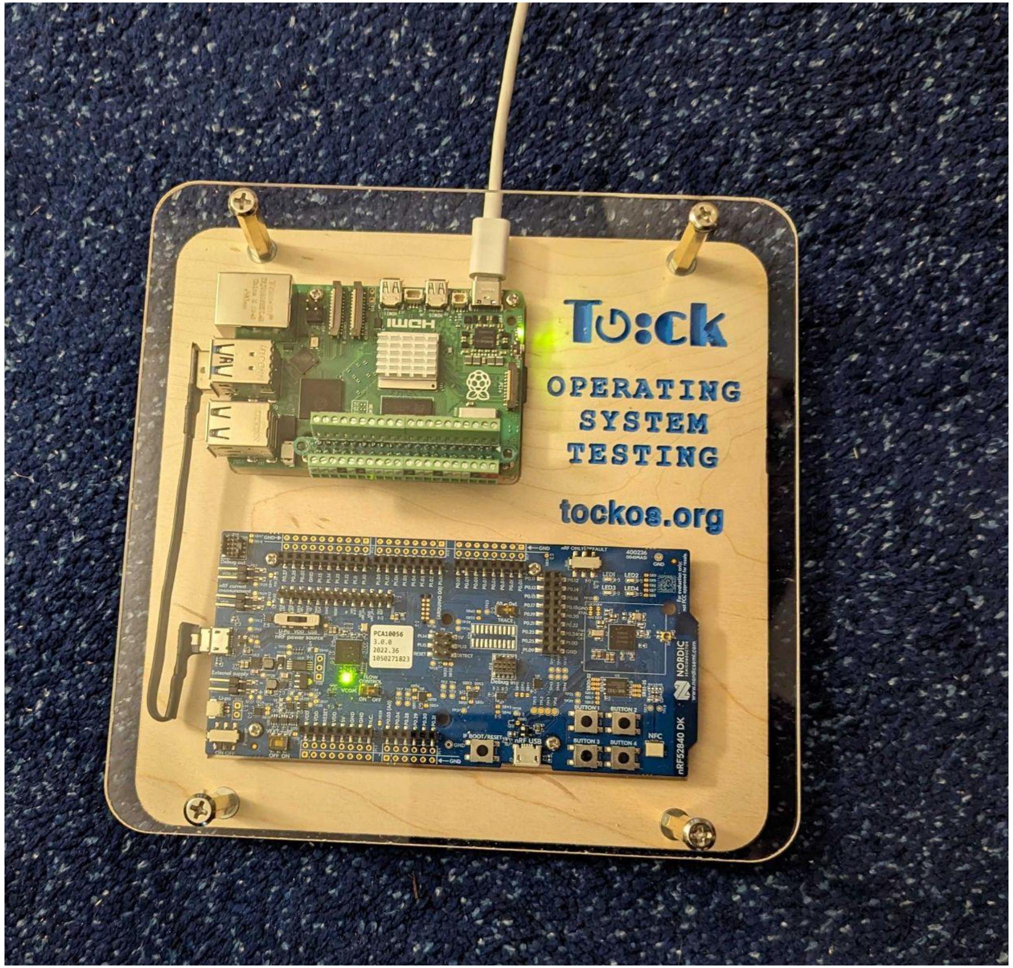

Coordinator
(Scheduler, access control, …)
“Top of Rack” server running *Supervisor*
Test Host
DUT
Test Host running VM

Current State

- Initial proof of concept working since ~January
  - Targeted Linux containers exclusively
  - Basic architecture seems decent
  - Coordinator written in Elixir + Phoenix → rewrite it in Rust!
- Rewrite started ~2-3 weeks ago
  - For now, focusing on low-level components & engineering
  - Taking in lessons learned from the first attempt
- Increasing momentum: 2-4 people working on this starting now!
- Interest from other communities as well – Rust Embedded, Embassy
- Hardware deployments “ready” at UCSD, Princeton, UVA(?)


nRF52840-DK
nRF52840-DK
NORDIC
NORDIC
ProDesk
CORE i3


ARTY A7
AVNET
DIGILENT
ST

Bluetooth
THREAD
zigbee
nRF52840-DK
14
13
12
11


Tock
OPERATING SYSTEM TESTING
tockos.org
nRF52840 DK
NORDIC

Demo