👋 Dasharo User Group #8 🎉

Dasharo Open Source Firmware Validation Status

Agenda

- Introduction to Dasharo Open Source Firmware Validation (OSFV)
- Current state
- Recent improvements
- Work in progress - current priorities
- Q&A

Introduction to Dasharo OSFV

Introduction to Dasharo OSFV

- Main purpose
  - validation of (open-source) firmware
    - can be used for any firmware, really
  - mainly Dasharo with UEFI payload right now
- Using Robot Framework as a base

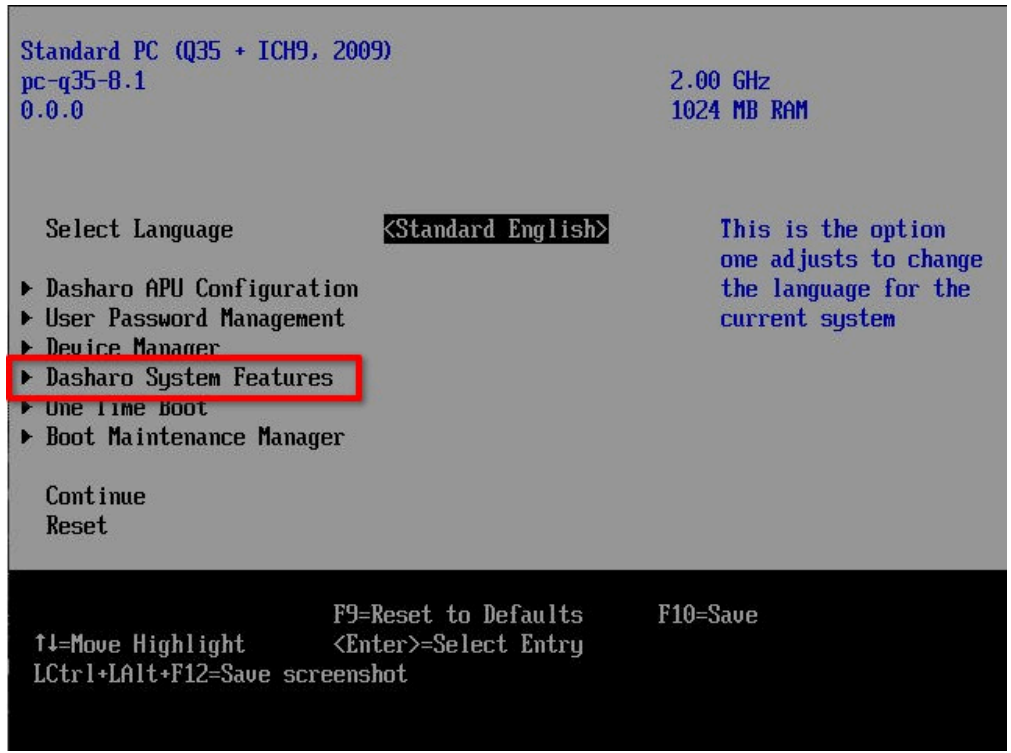
Introduction to Dasharo OSFV

- Use cases
  - testing Dasharo firmware releases
  - test-driven bug fixing (and adding new features)
  - regression testing
    - after introducing new features
    - after major changes (update base from upstream project)
  - validation of Dasharo-related tools (Dasharo Tools Suite, Dasharo Configuration Utility)
    - where possible, in QEMU

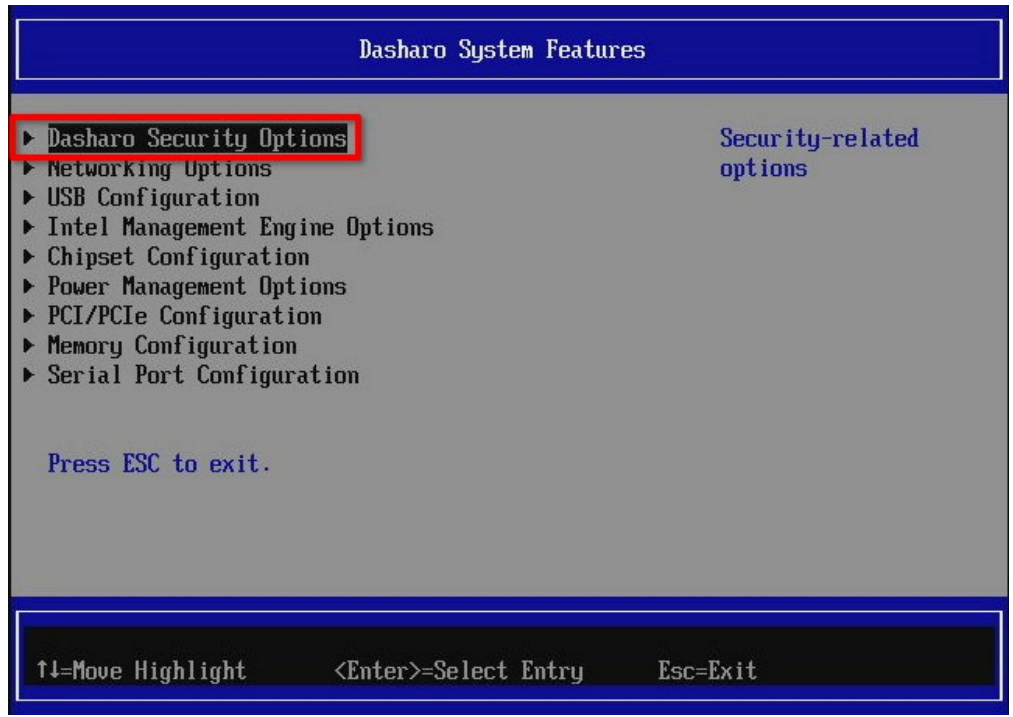
Typical setup

- Power control (power supply, power button)
- Serial console over telnet
- Move through firmware menus to switch certain options
- Check the result
  - In firmware
  - In OS
    - Dasharo Tools Suite - reference distribution for testing purposes
    - The latest Ubuntu LTS
    - The latest Windows 11
    - QubesOS for some tests

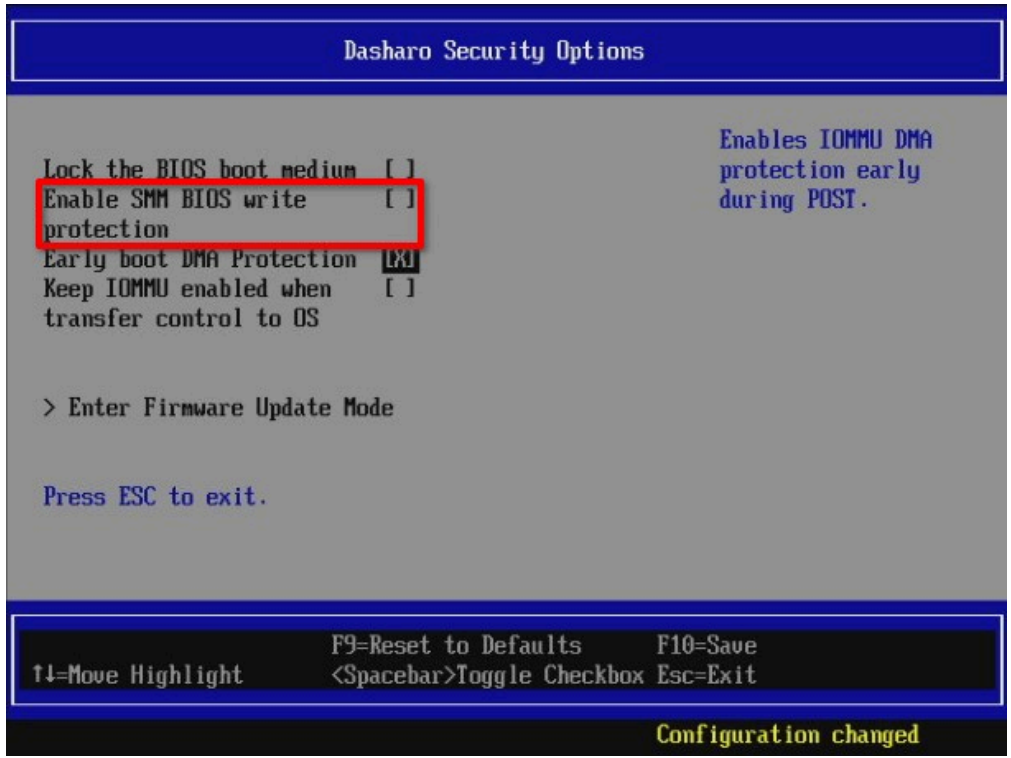
One test - step by step



One test - step by step



One test - step by step



One test - step by step

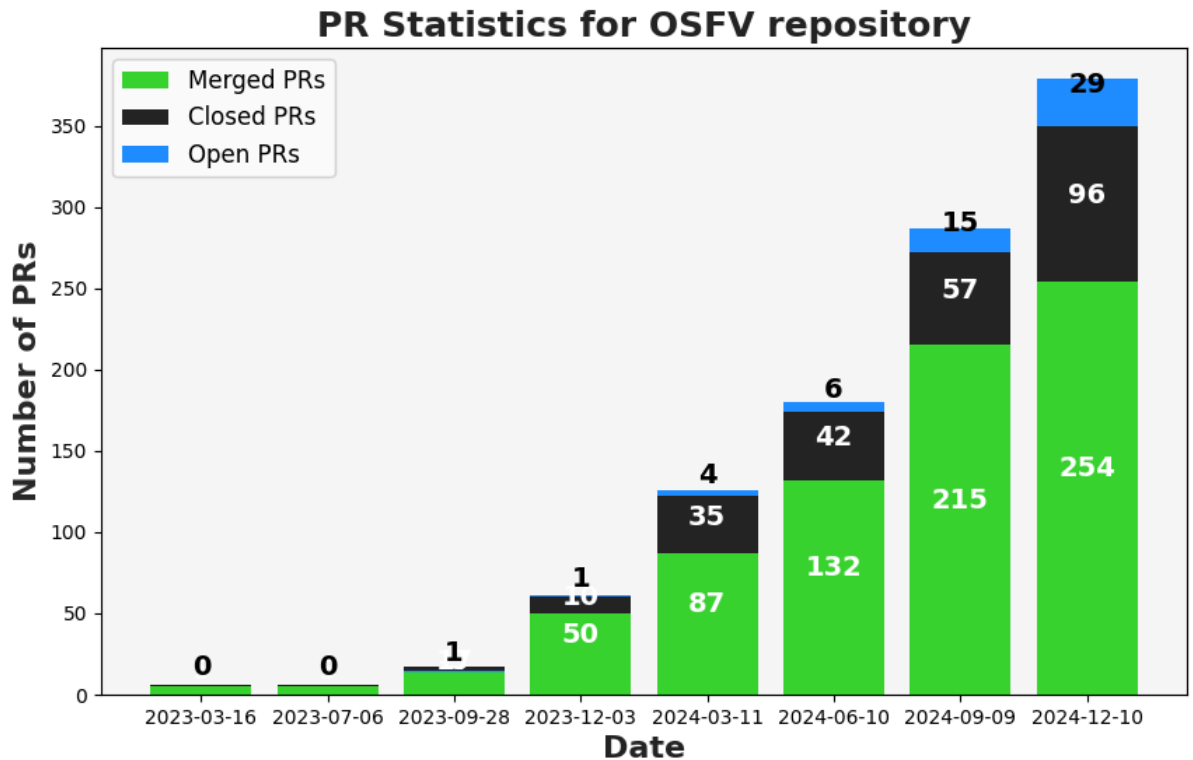
```
Power On
${setup_menu}=    Enter Setup Menu Tianocore And Return Construction
${dasharo_menu}=    Enter Dasharo System Features    ${setup_menu}
${network_menu}=    Enter Dasharo Submenu    ${dasharo_menu}    Dasharo Security Options
Set Option State    ${network_menu}    Enable SMM BIOS write    ${TRUE}
Save Changes And Reset
Boot System Or From Connected Disk    ubuntu
Login To Linux
Switch To Root User
Get Flashrom From Cloud
${out_flashrom}=    Execute Command In Terminal    flashrom -p internal
Should Contain    ${out_flashrom}    SMM protection is enabled
```

Current state

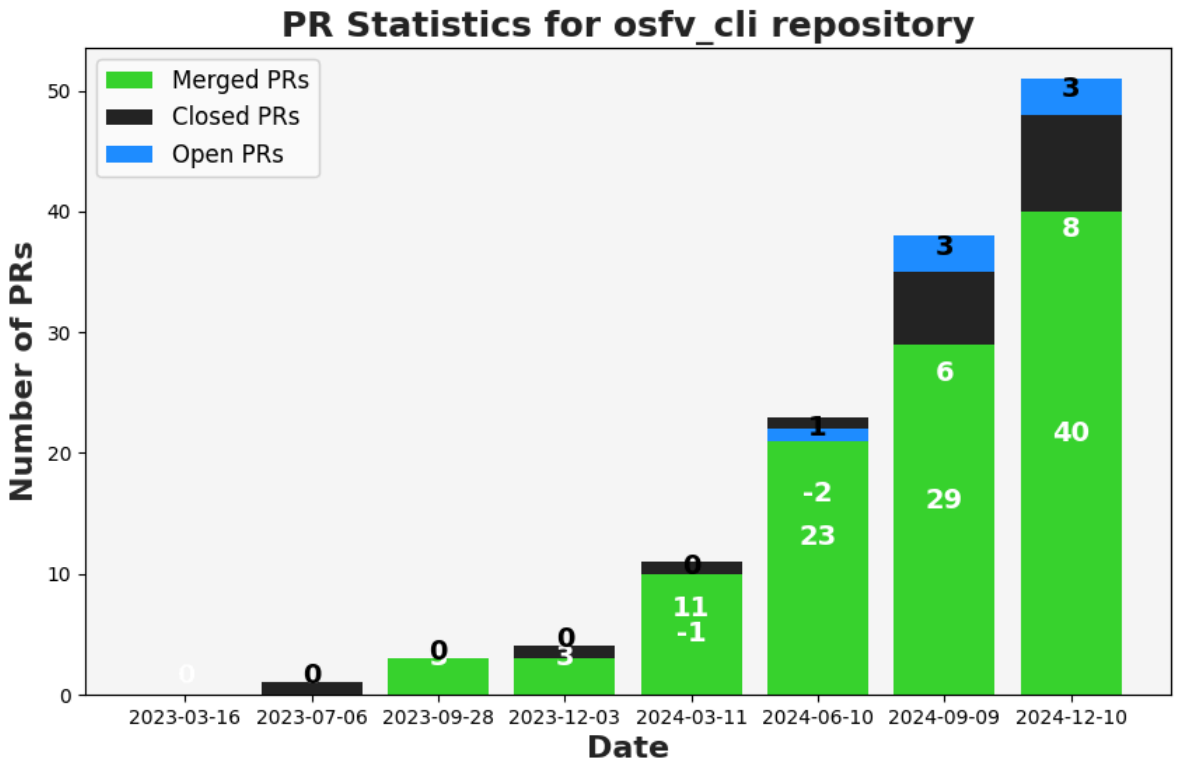
Current state

- 😞 No release 😞
- We are still failing to define (and reach) criteria for tagged release of test environment
- Right now it worked more like a `rolling release` in `develop` branch
- We wanted to merge `develop` into `main` once we are "ready" but it never happens
- It is confusing to see the default branch (`main`) with very little updates and outdated information
- Merged `develop` branch into `main` branch yesterday
  - we might use it as the default target branch until we figure it out

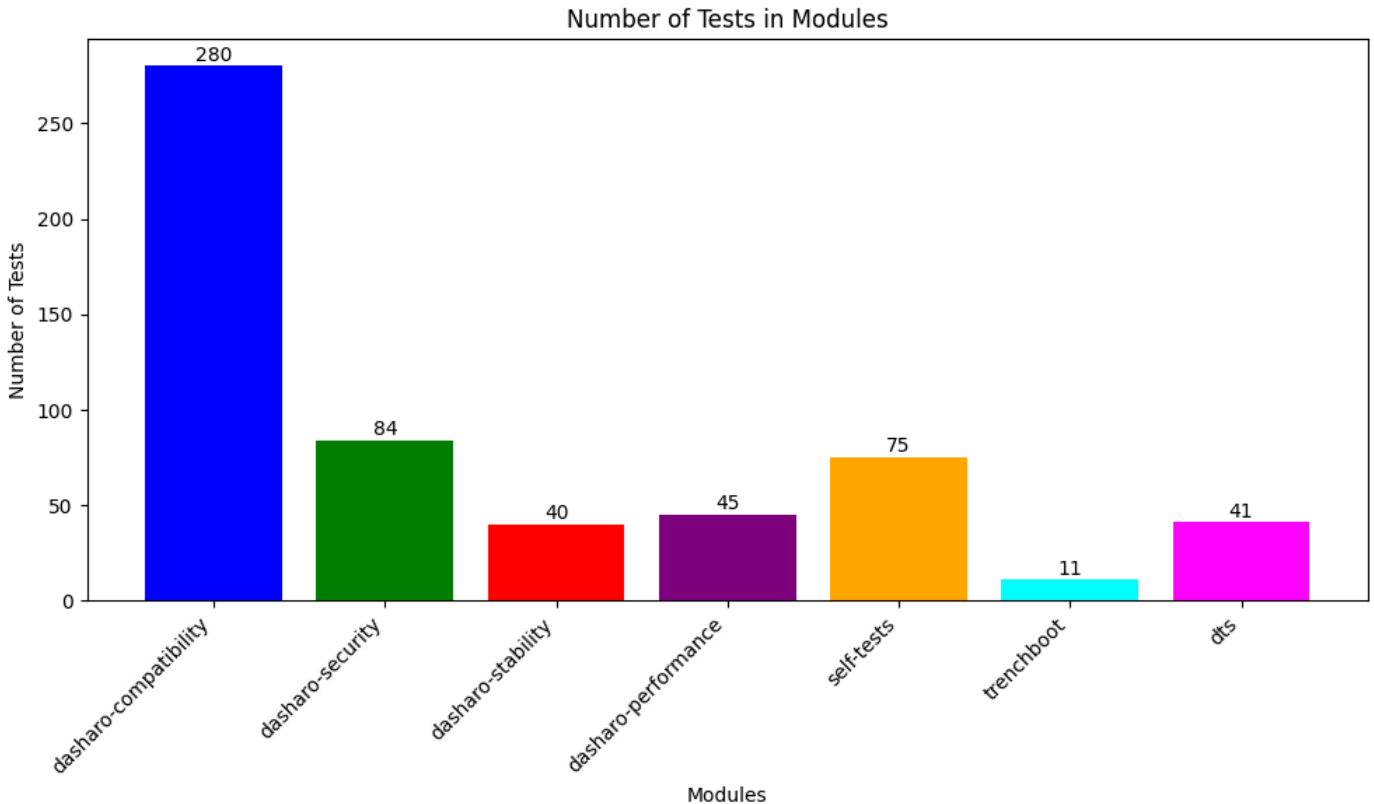
Statistics - Pull Requests



Statistics - Pull Requests



Statistics - tests



Recent improvements

New tests

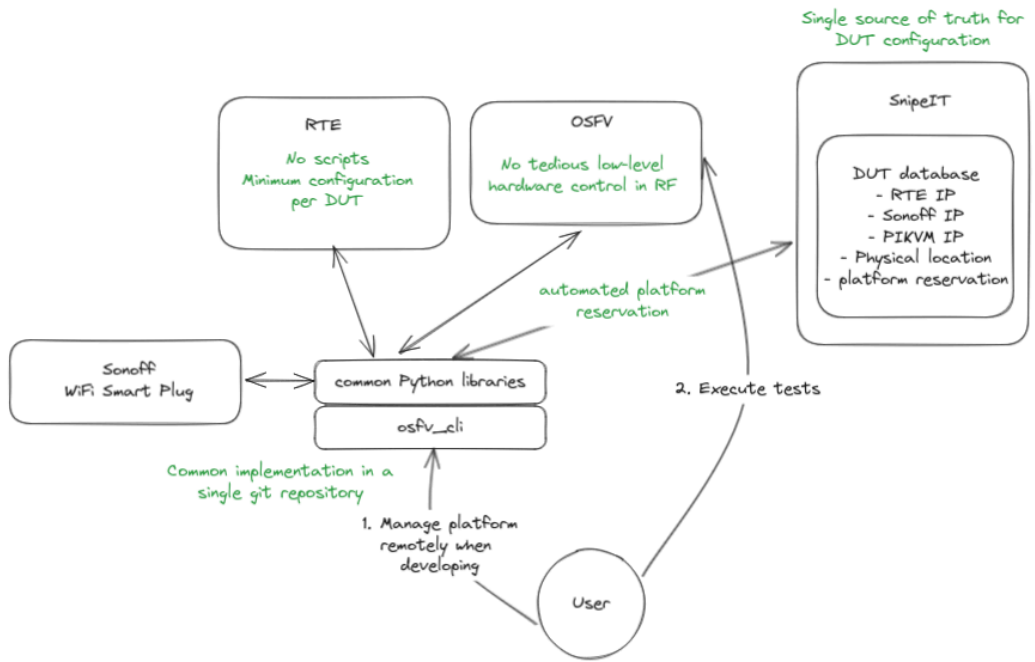
- Extended CPU suite (P/E cores, HT)
- Extended DCU tests
- Introduced more thorough testing of DTS in QEMU
- Suite for TrenchBoot development
  - Booting Linux and Xen from meta-trenchboot
- Suite for Capsule Updates
- More

New platforms

- Protectli VP2430
- Protectli VP3210, VP3230
- Odroid H4
- Bring back Dell Optiplex - both UEFI and SeaBIOS

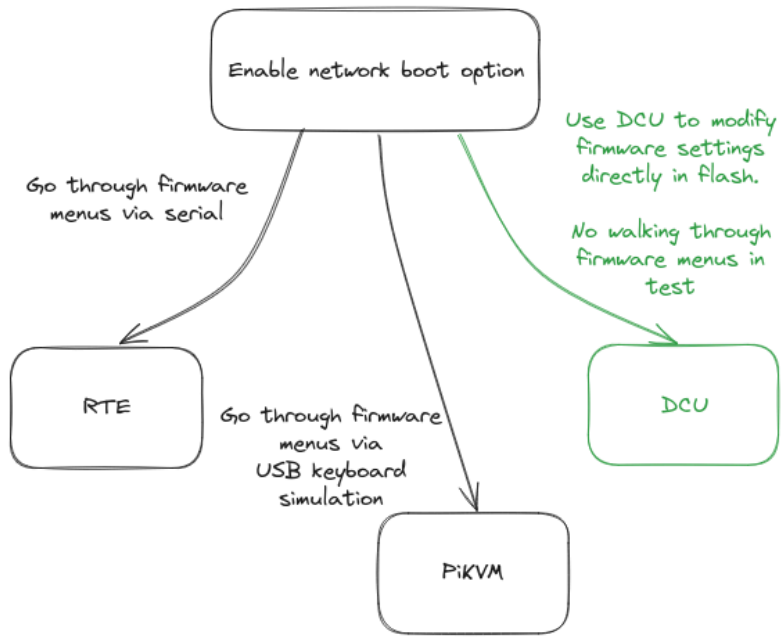
osfv_cli integration

- Integrate low-level hardware operations into Python libraries
- Reuse the same libraries by test framework and CLI tool

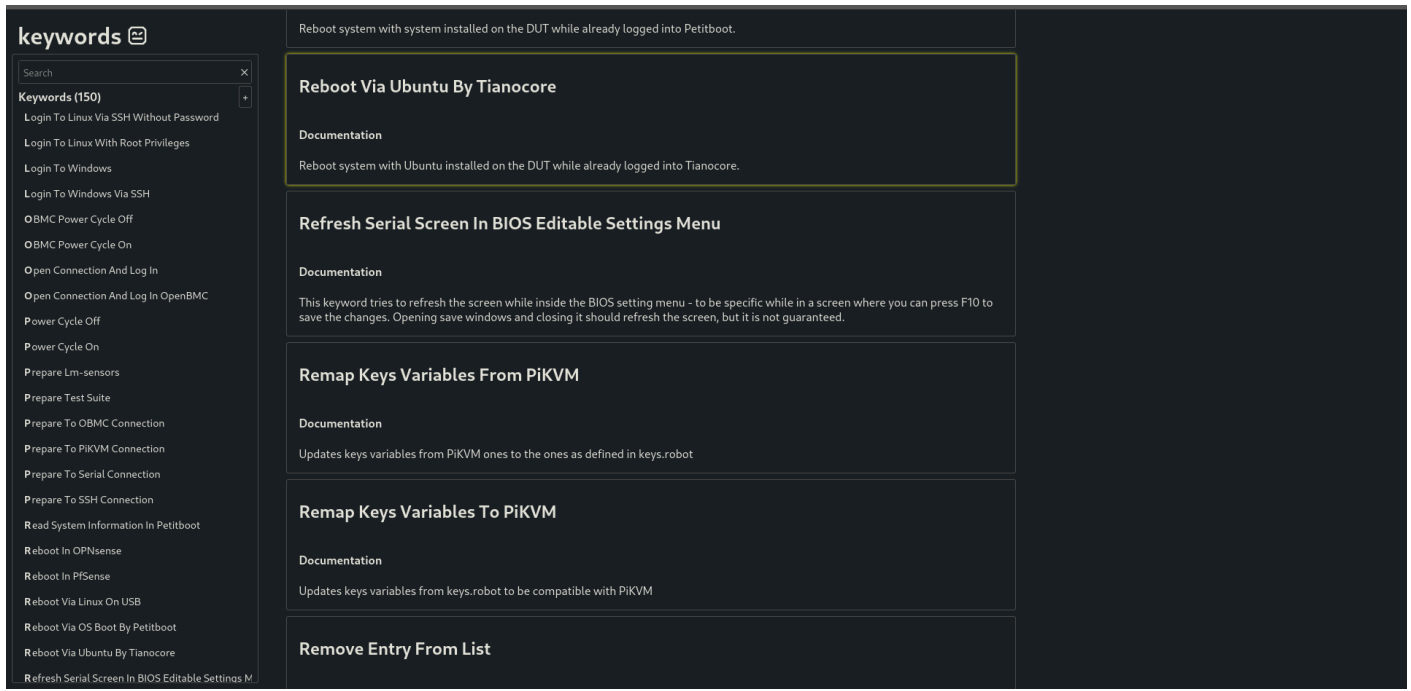


dcu integration

- Alternative interface for changing fw settings*
- Instead of manual steps, we can modify SMMSTORE variables directly



Keywords documentation



https://dasharo.github.io/open-source-firmware-validation/

Next steps

- Finalize SeaBIOS support
- Need to focus more on repeatability and reliability than adding new tests/features
  - resolve Power On reliability: https://github.com/Dasharo/open-source-firmware-validation/issues/607
  - identify other areas generating problems, implement stress testing of these areas
  - add some HW into CI loop next to the QEMU tests
    - serial + relay - one of the Protectli boards
    - serial + PiKVM + Sonoff - one of the MSI boards

Questions?