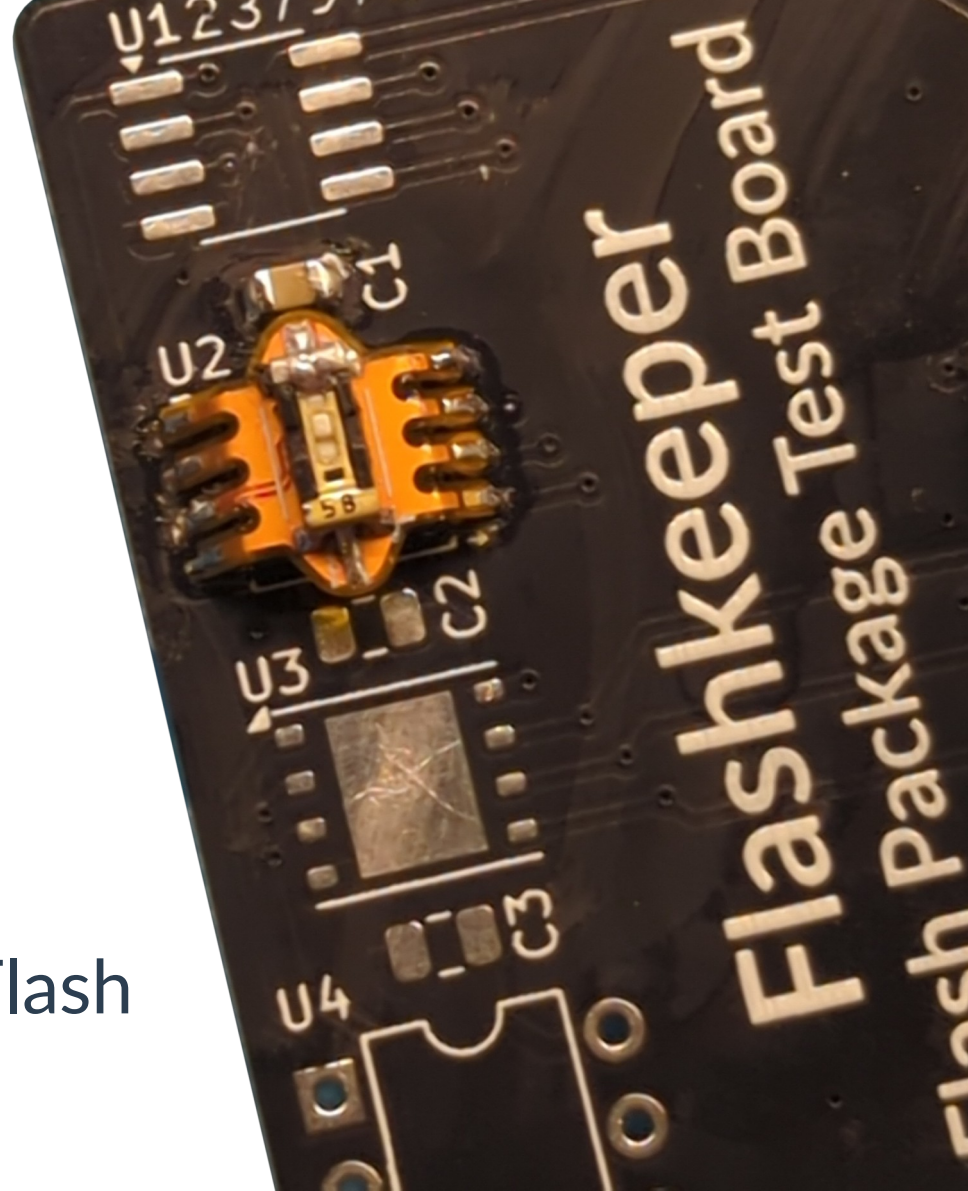


Flashkeeper

Manage and Protect your SPI Flash



We are...



Dodoid
(me)

- **Embedded developer, PCB designer, free-software sysadmin/system integrator**
- **Administrator of 15h.org**
 - Community of 100+ KGPE-D16 and other fam15h users
 - Maintainers of coreboot-15h
- **Occasional coreboot contributor**



tlaurion
of Insurgo

- **Platform Security Researcher**
- **Maintainer of linuxboot/Heads**
 - Heads available standalone and on partner systems
- **Long-time FOSS-firmware community leader**
- **Heads+Qubes system integrator (Insurgo PrivacyBeast X230)**
 - Qubes OS-certified in 2019

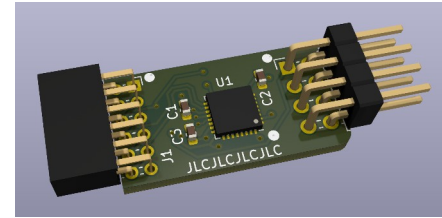
What if...

- A standardized expansion port on every motherboard
- Provides power and SPI, on the same bus as the firmware SPI flash
- Standard form factor – expansion devices for it would work essentially everywhere
- Widely adopted by nearly every hardware vendor
- Open, well-documented standard, for anyone to use

Some things are close

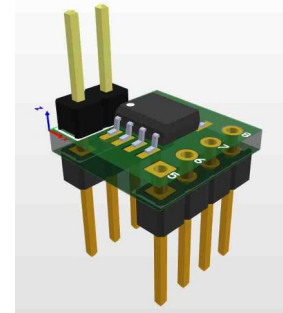
- **SPI headers, where they exist, are nonstandard.**

- SPI TPM headers are close, but vendor-dependent
- Some motherboards have nonstandard “factory” SPI headers, which may or may not be populated



- **Old motherboards used DIP flash**

- “Interposer-style” boards work here
- Rare today. DIP SPI flash actually hard to buy.
 - Ask a KGPE-D16 owner!

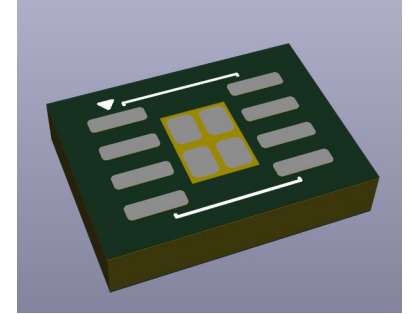
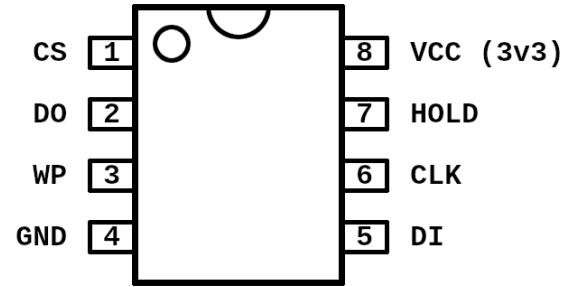


Except...



The 8-pin SPI Flash Footprint

- 8 pins, universal pinout
- 1.27mm pitch
- Width usually ~8 mm



- Most modern boards either WSON-8 8x6 or SOIC-8 208-mil
- WSON 6x5 and SOIC 150-mil exist but uncommon
- **Many boards have relatively flexible footprints**
 - Allows component substitution during manufacturing

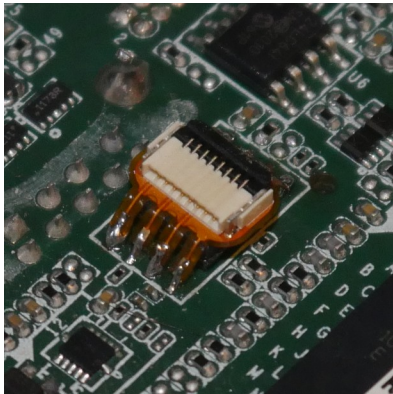
Unfortunately...

- **Flash packages footprints are ~standard, chip clips are not**
 - WSON and SOIC mostly share dimensions but not pin geometry
- **Chip clips themselves are not ideal**
 - Often too large to leave in a running system
 - Laptops, flash chips under heatsinks, etc
 - Not reliable long-term.
 - Sometimes not reliable short-term!
 - Sometimes don't fit at all (if other components are in the way)

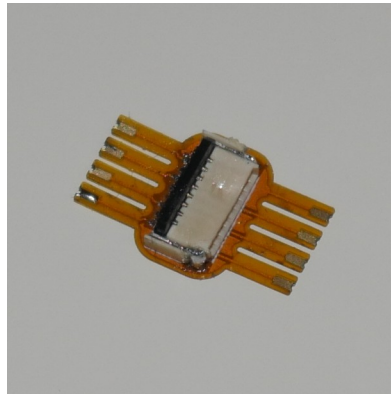
How do we add our own hardware
to existing SPI flash chips?

The Chip Adapter Family

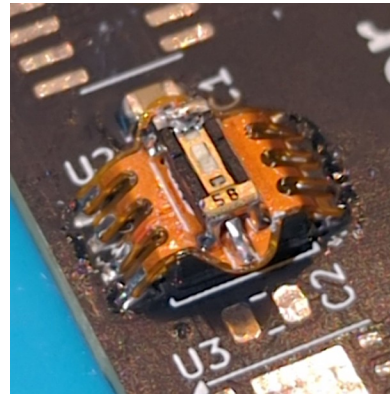
- 2-layer Flex PCB, soldered *on top of* host flash device
- 8 flexible legs reach down to pads



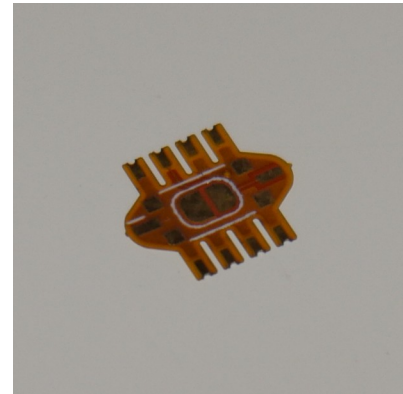
Breakout (SOIC)



Breakout (WSON)

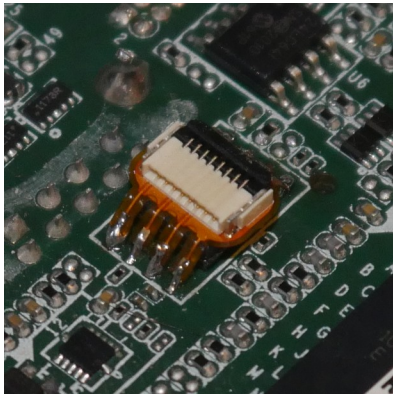


Write-Protect Switch

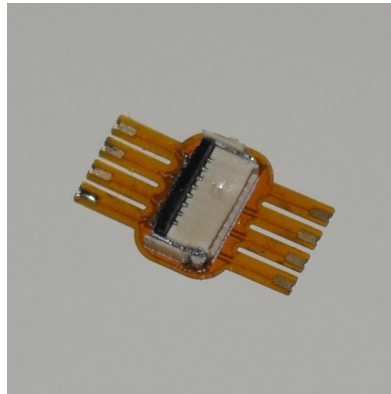


Write-Protect Low-Profile

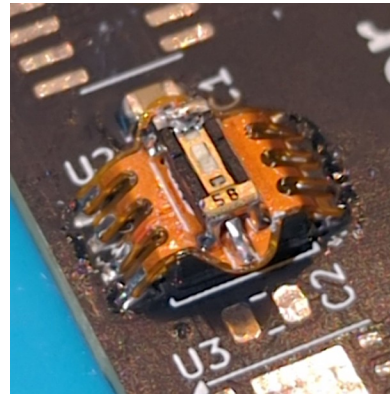
What does this allow us to do?



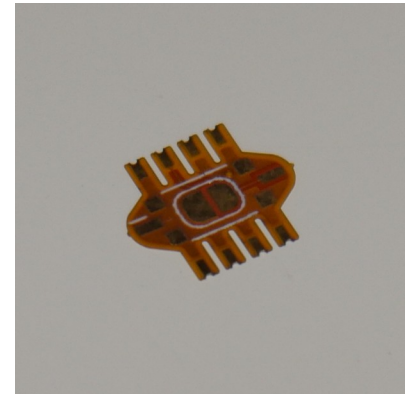
Breakout (SOIC)



Breakout (WSOIC)



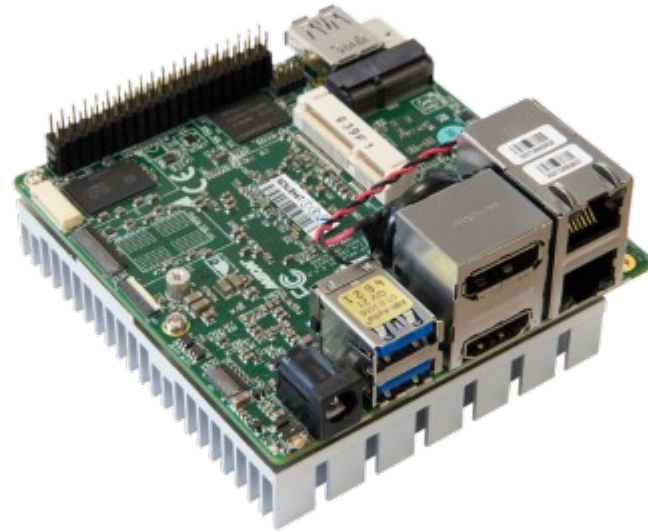
Write-Protect Switch



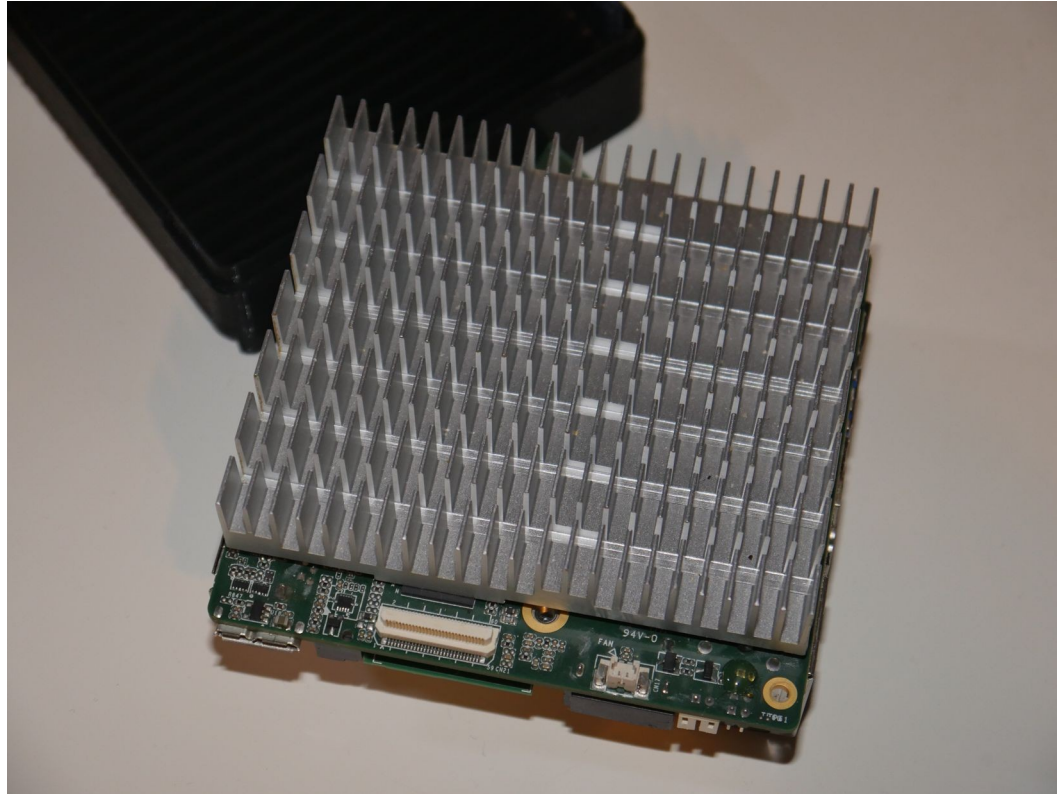
Write-Protect Low-Profile

Has this ever happened to you?

- **AAEON UP Squared**
- **Industrial SBC**
- **Intel Apollo Lake SoC**
 - e.g. Celeron N3350
- **Has a coreboot port**
- **Where's the SPI flash?**



Has this ever happened to you?



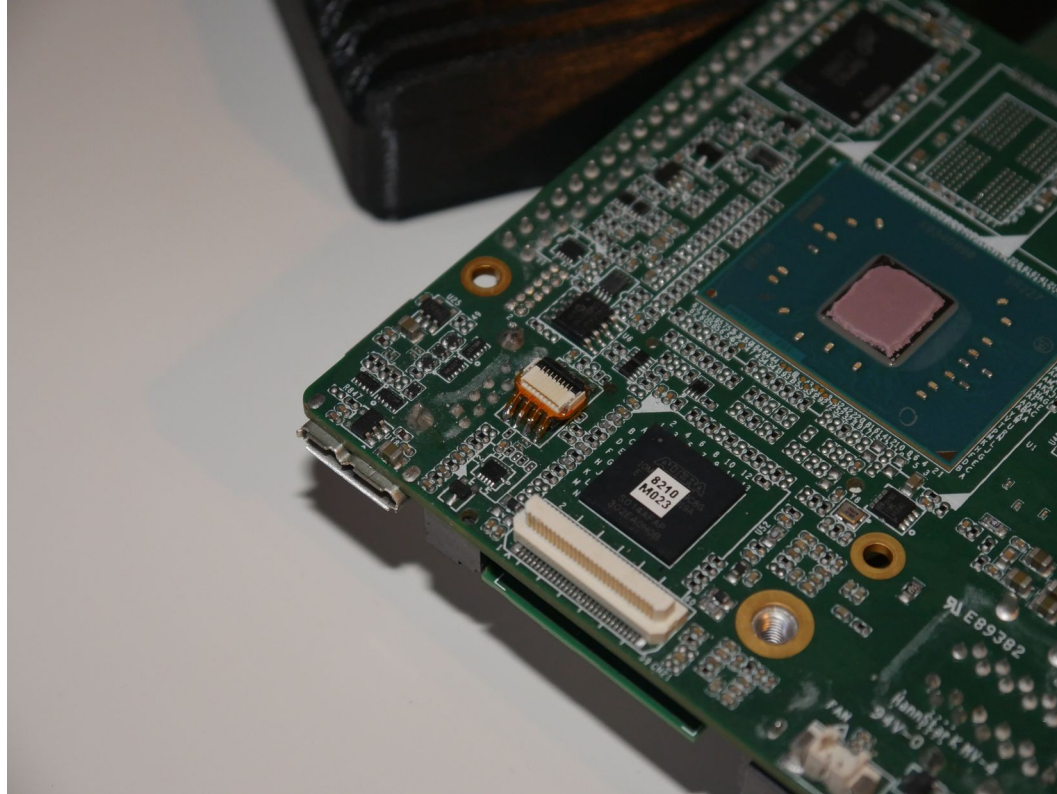
Has this ever happened to you?



Has this ever happened to you?



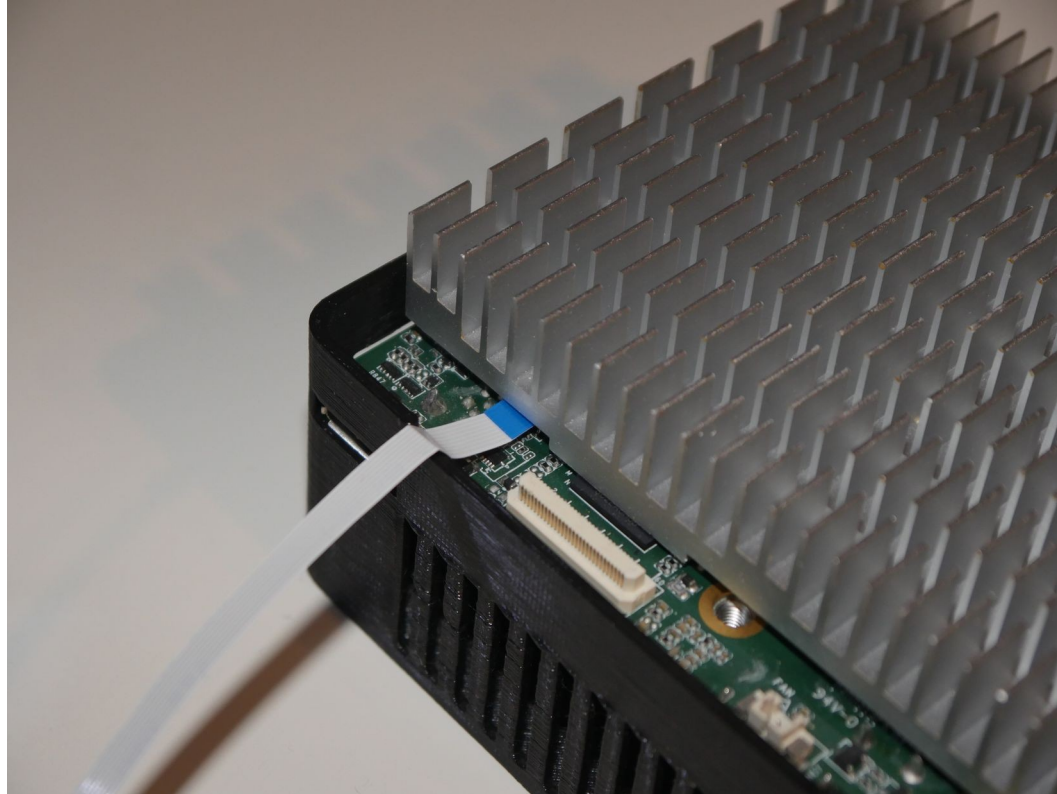
Breakout Chip Adapter



Breakout Chip Adapter



Breakout Chip Adapter



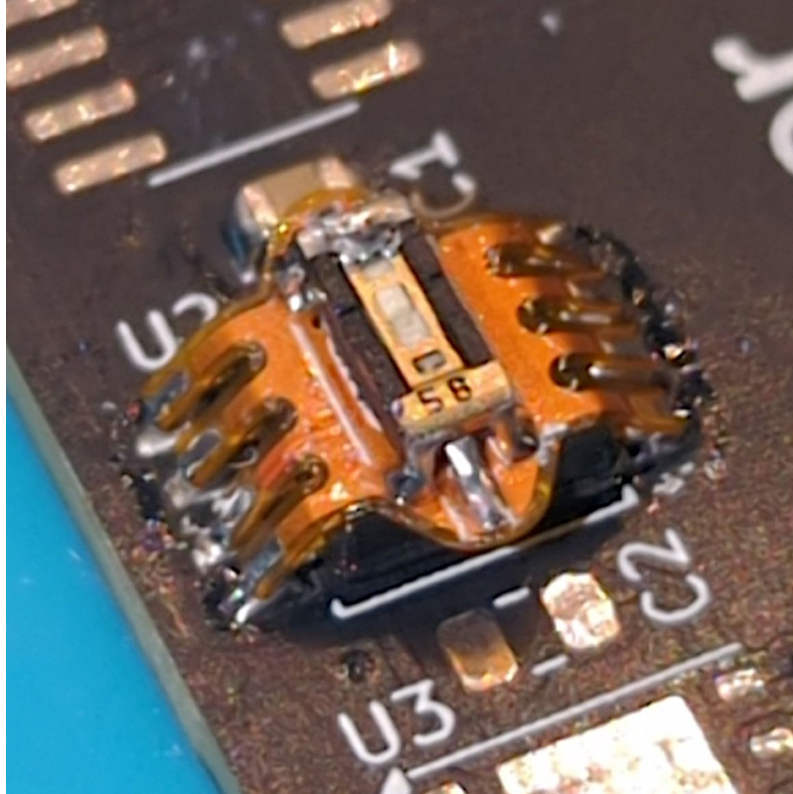
Solved!



Write Protection

- **Many SPI flash chips support partial write protection**
- **Can be used to protect the coreboot bootblock**
- **Needs to be configured in both hardware and software**
 - flashrom can configure WP behaviour in status registers
 - WP pin must be held low to enable
- **Many motherboards expose no WP control**
- **What if the chip just had a switch?**

The chip can just have a switch!



So, what else can we put
on top of an SPI flash chip?

You could...

- **Add a microcontroller**
- **Monitor data read from flash**
- **Debug SPI transactions directly**
- **Remotely read and write the flash chip**
 - And authenticate this!
- **Attest SPI flash contents**
- **An FPGA can do all of this (and basically anything else)**

Fitting an FPGA on top of a SOIC-8

- **Roughly 25 mm² of area**
- **Board must be self-contained**
 - Generate your own clocks
 - Regulate your own voltages
 - Store your own bitstream
 - Expose your own development port
- **Flexible PCBs require significantly larger feature sizes**
 - Make a System-on-Module, then design a chip adapter to carry it

The Flashkeeper SoM

- **Lattice iCE40UP5K FPGA**
 - Fully FOSS toolchain (yosys+nextpnr)
 - Highly-integrated (SPRAM, HFOSC, 3-ish voltage rails)
 - Can be one-time-programmed (NVCM)

The Flashkeeper SoM

- **Lattice iCE40UP5K FPGA**
 - Fully FOSS toolchain (yosys+nextpnr)
 - Highly-integrated (SPRAM, HFOSC, 3-ish voltage rails)
 - Can be one-time-programmed (NVCM)
- **TI TPS7A20 voltage regulator**

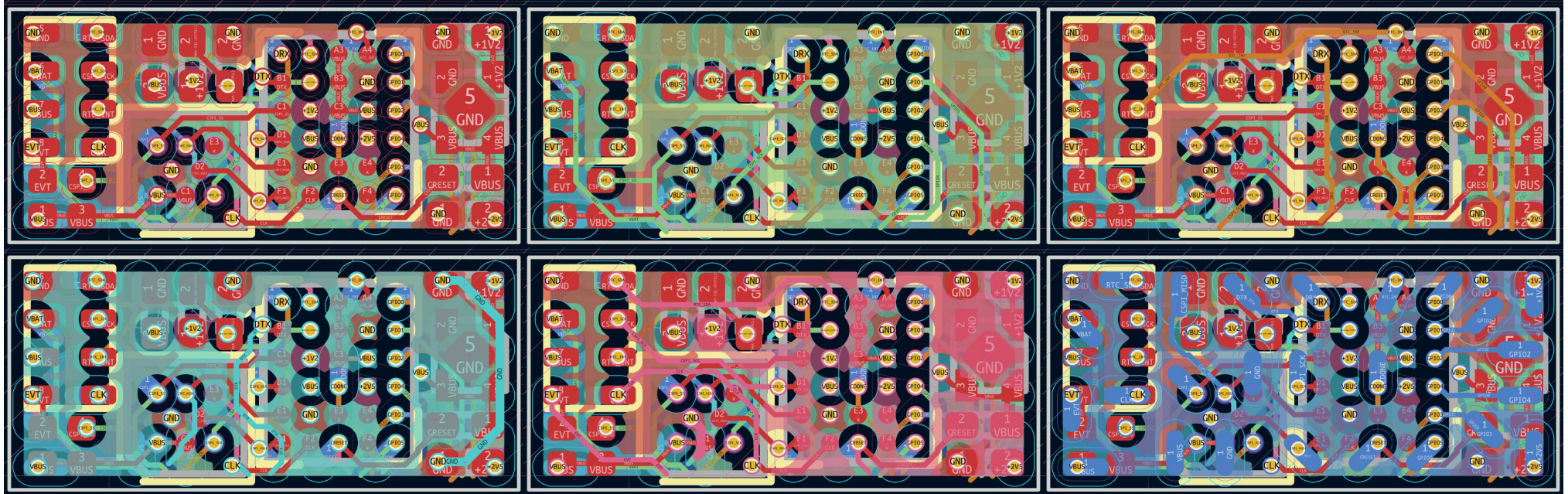
The Flashkeeper SoM

- **Lattice iCE40UP5K FPGA**
 - Fully FOSS toolchain (yosys+nextpnr)
 - Highly-integrated (SPRAM, HFOSC, 3-ish voltage rails)
 - Can be one-time-programmed (NVCM)
- **TI TPS7A20 voltage regulator**
- **Micro-Crystal RV-3028-C8 real-time clock**

The Flashkeeper SoM

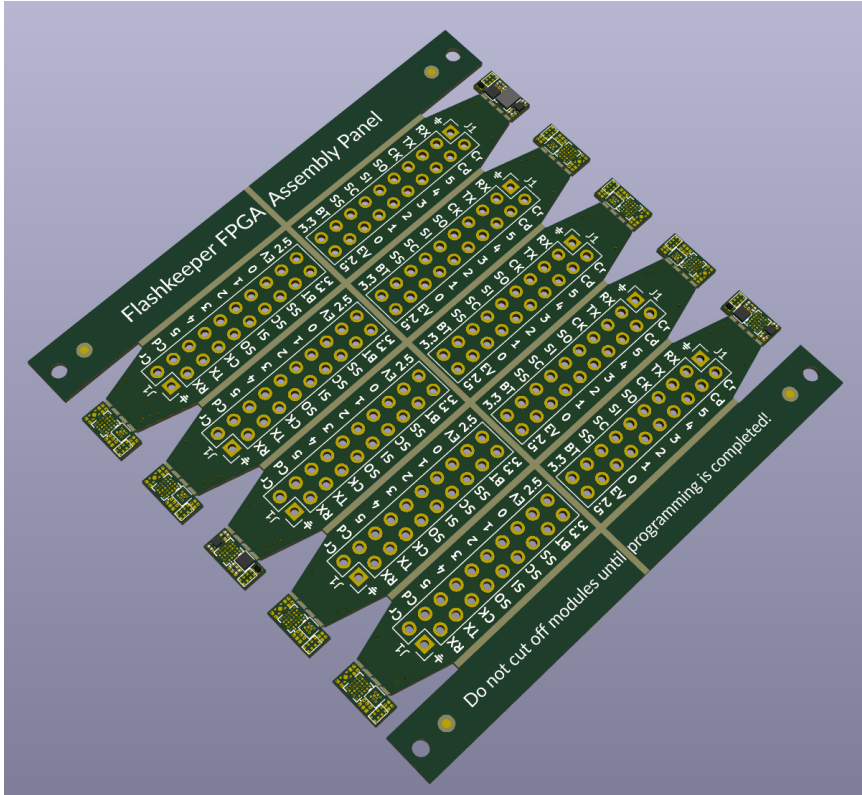
- **Lattice iCE40UP5K FPGA**
 - Fully FOSS toolchain (yosys+nextpnr)
 - Highly-integrated (SPRAM, HFOSC, 3-ish voltage rails)
 - Can be one-time-programmed (NVCM)
- **TI TPS7A20 voltage regulator**
- **Micro-Crystal RV-3028-C8 real-time clock**
- **Renesas AT25DF041B SPI Flash**

SoM PCB Layout



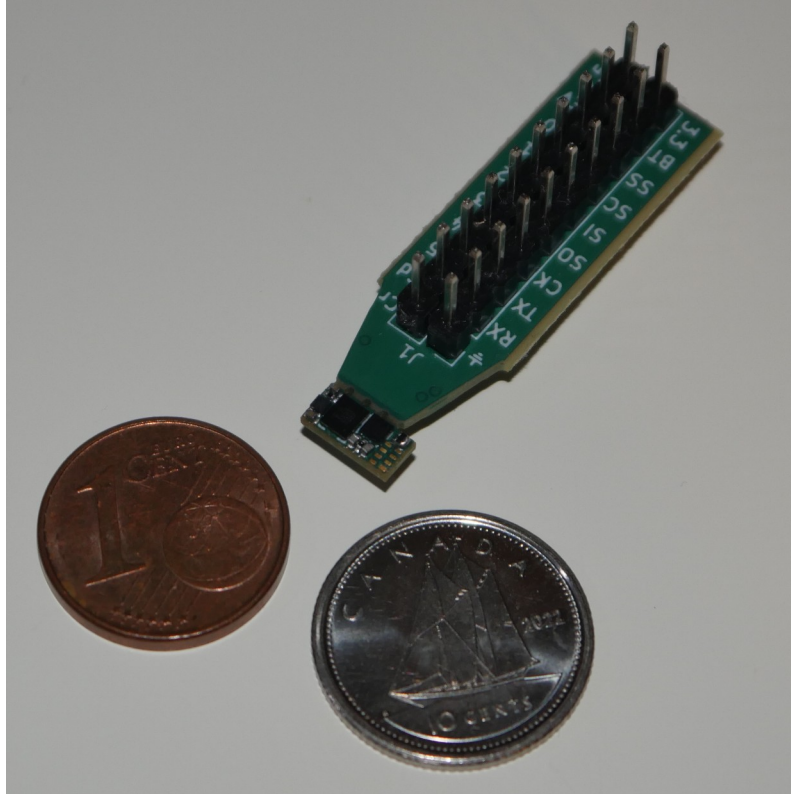
7.15 x 3.30 mm • Custom LGA-25 Footprint • Designed in KiCAD • 2 mil trace spacing
0.4mm BGA pitch • 0201 Passives • No blind/buried vias • Standard JLCPCB 6-layer process

It was too small for the fab...

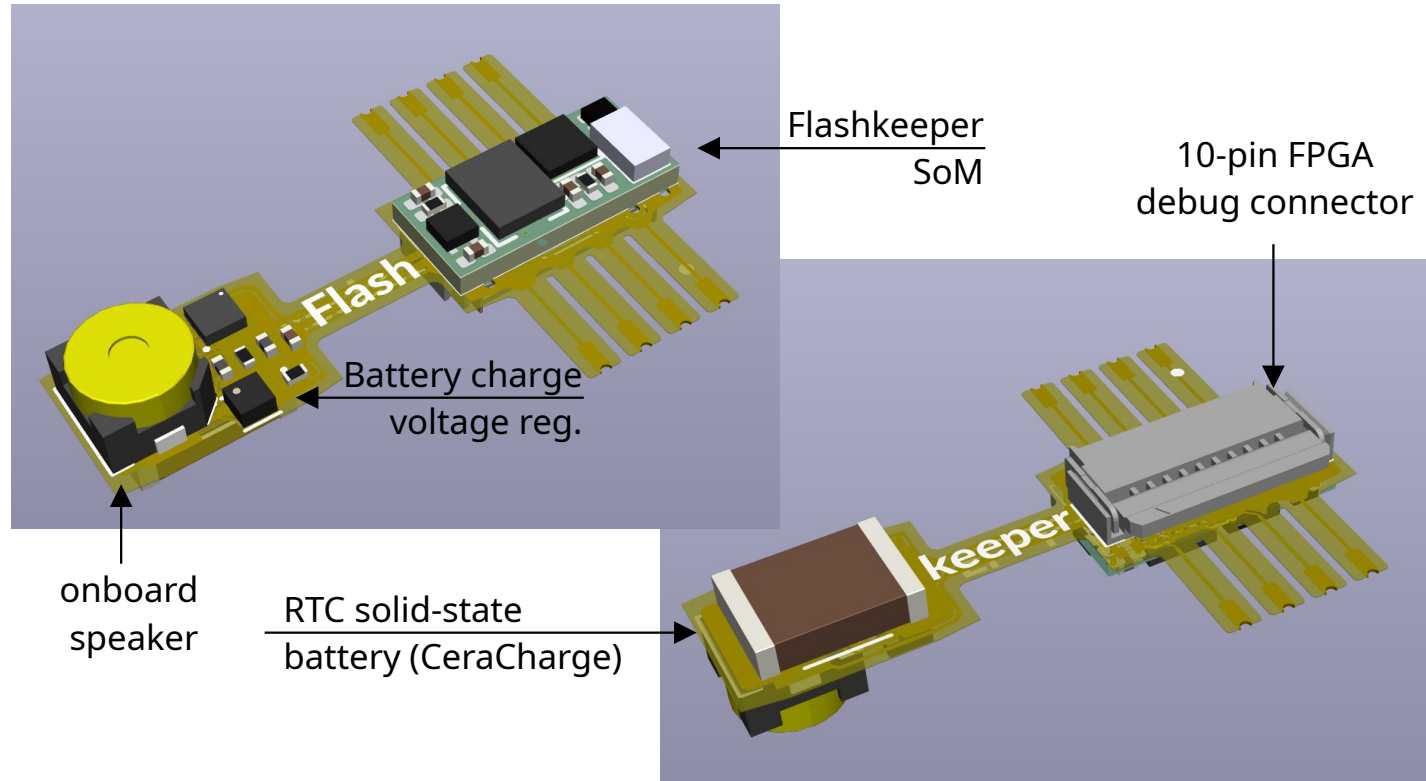
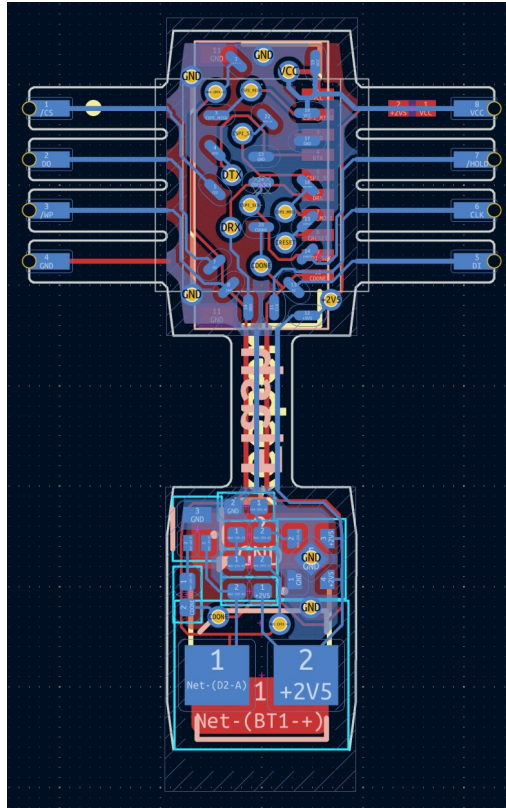


- Custom panel design
- Pick-and-place friendly
 - Fills minimum 70x70mm size
 - Fiducials, edge-rails
- 10 Flashkeeper SoMs
- Snap-off development breakout headers

The SoM and its snap-off header



The FPGA Chip Adapter



What does it run?

- **Modified PicoSoC**

- PicoRV32 RISC-V RV32Izmmul CPU
- 128 KiB SPRAM, 8 KiB bitstream boot ROM
- Ring-oscillator entropy source
- Memory-mapped access to onboard and host SPI flash

- **Flashkeeper FPGA Firmware**

- Read, write, and debug SPI flash
- TweetNaCl (lightweight cryptography)
- Extensible, Forth-like onboard shell (EDE)

What does it run?

Whatever you want!
It's an FPGA.

Possible Applications

- **Authenticated external flashing**
- **Flash image verification**
- **Verification of the coreboot bootblock (coming very soon)**
 - At runtime (SPI bus sniffing), the FPGA can see every byte your platform loads from flash
 - At rest (host powered off), it can read the flash directly
- **Log/debug SPI flash/TPM transactions?**
- **coreboot spiconsole decoder?**

Flash verification under user control

- **The Flashkeeper FPGA is a general-purpose computer**
 - User-controlled software+gateway
 - Fully free and open hardware, software, and gateway
- **Fully open toolchains for everything**
 - Develop and build with KiCAD, FreeCAD, yosys, nextpnr, GCC, etc.
- **“Fused” and “read-only” don’t have to mean proprietary**
 - iCE40 NVCM allows user-controlled one-time programming with your own gateway, boot ROM, firmware, and secrets

Live Demo

Any questions?

- Find us online: <https://flashkeeper.org>
- Matrix: <https://matrix.to/#/%23flashkeeper:matrix.org>
- Repo (15h.org): <https://git.15h.org/Flashkeeper/flashkeeper>
- Repo (Github): <https://github.com/linuxboot/flashkeeper>
- Contributions, questions, and testing welcome!



Matrix



Git (15h.org)



Git (Github)