

ARM Snowflakes

Stop chasing our tails without losing the fun?

Overview

- Background: LAVA - automation for Linaro & others.
- Hardware diversity problems for automation & CI
- kernelci on ARM vs SD card multiplexer hardware
- Is all diversity useful or helpful?
 - less of what we want to prove we can do ...
 - more of what we know we should do?

Multiplexing the SD card

- Long term goal of mass testing & automation.
- Primarily for UBoot devices.
- Assumes SD card as primary boot media.
- Control device media
- Independent of device power status.

Automation, CI & the ARM ecosystem

- ARM is mass market
- KernelCI: <http://kernelci.org/stats/>
 - ~**1,200** boots per day.
 - **639,462** boots in total
 - **154** unique boards
 - **176** unique defconfigs
- UBoot? UEFI? Fastboot? Other bootloaders? (*why!?*)
- Automation requirements

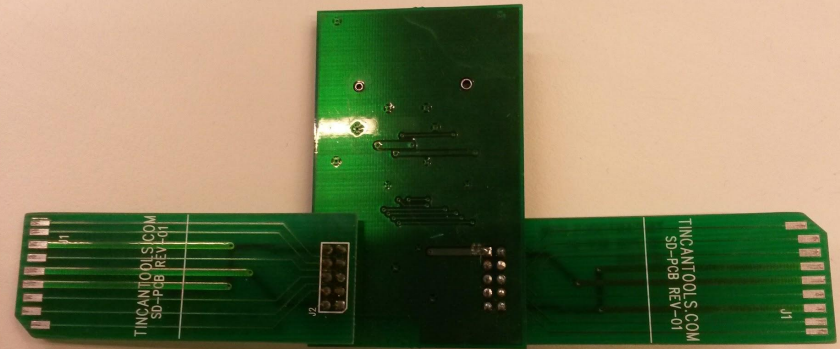
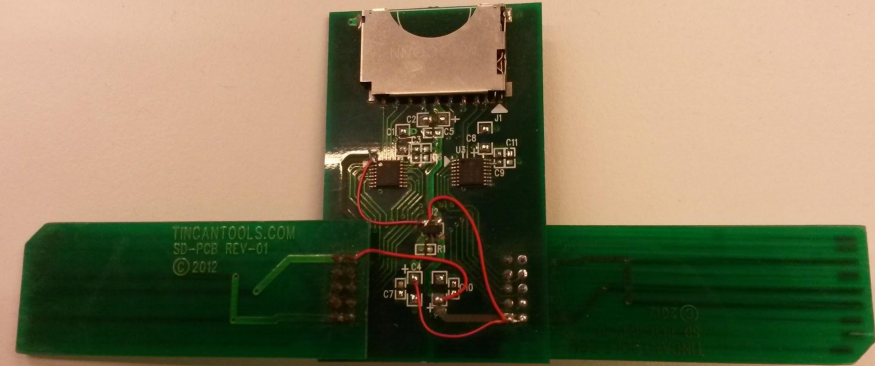
History and examples of SDMux

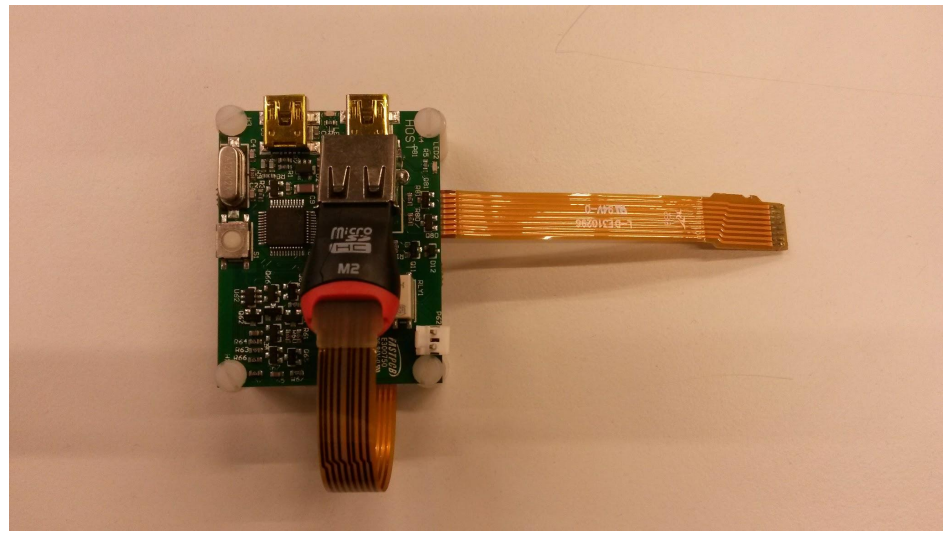
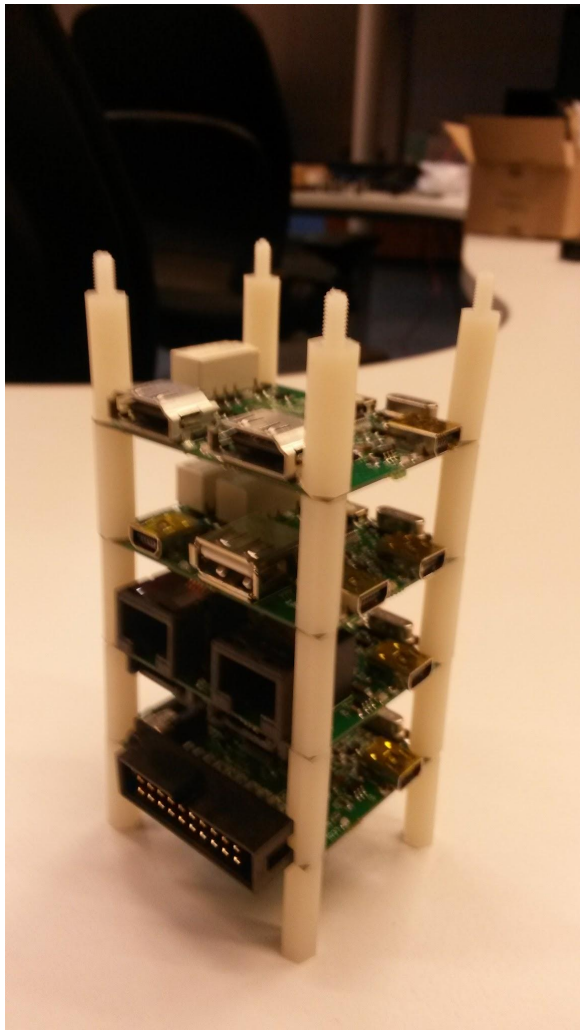
- LAVA is aware of **FOUR** attempts.
- All are independent designs, most by lone developers.
- None have succeeded in mass testing.
- Most repeat the same errors - unaware of others.

Original

SD size

- Fixed connections to devices
- Loose wires
- Fragile board connectors
- Large, bulky, not useful for microSD devices.
- Each connector at a different (fixed) height.





Feature Creep

Avoid adding GPIO, HDMI, Ethernet, USB.

Avoid needing **TWO** USB cables per PCB.

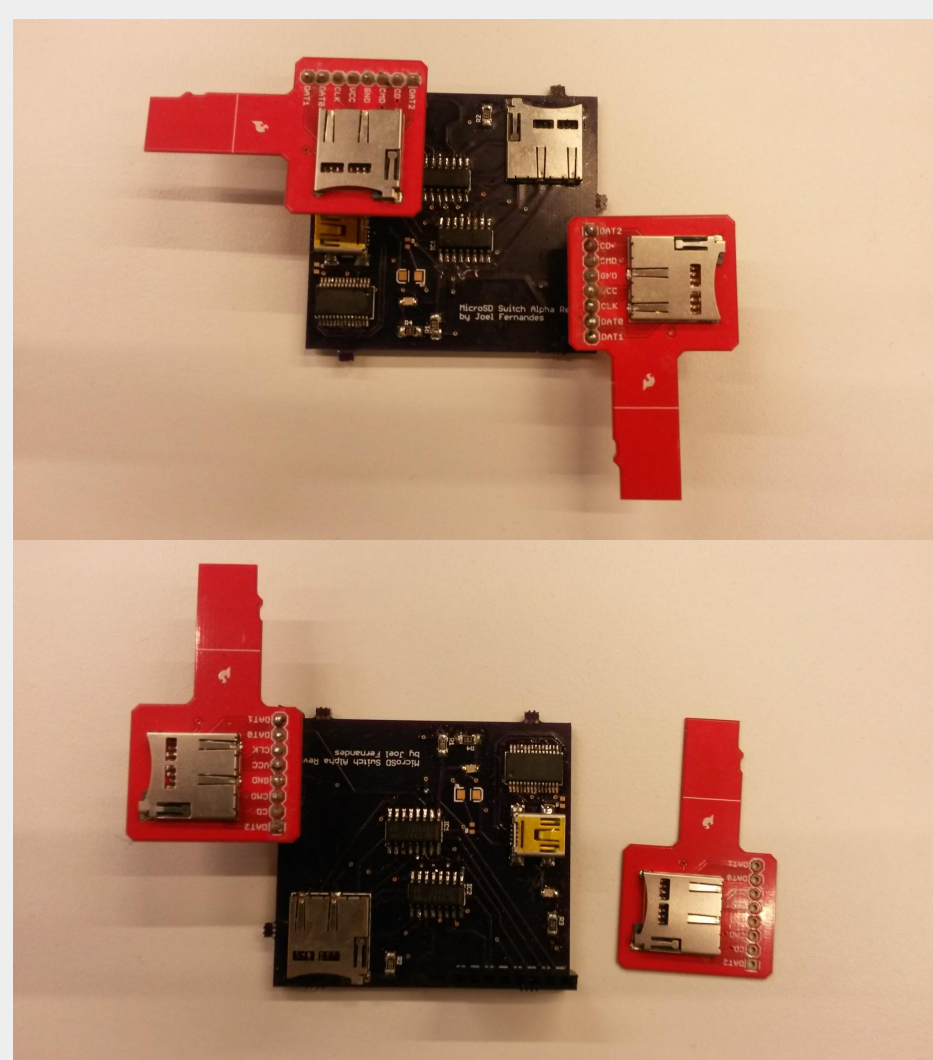
Avoid needing a Cortex-M & firmware.

Avoid mounting USB host at an angle of 30 degrees!

DO use cable connectors!

Oops!

Do not use fixed PCB
connections to the devices!
Do not use fragile daughter
board connections!



Scalable

Do not require an
external relay.

Do not require external
power control.



SD Multiplexer requirements for mass testing

<http://linux.codehelp.co.uk/?p=235>

1. **Ethernet**
 2. **No USB connections**
 3. **Removable media only**
 4. **Cable connections to device**
 5. **Serial numbers**
 6. **Status Interrogation**
 7. **Stable feature set**
 8. **Scalable**
 9. **Reliable power behaviour**
 10. **Software**
 11. **Support & Maintenance**
 12. **Mounting Holes**
- 20 devices per rack - network control
 - No hubs - no USB writes
 - Prototype boards could damage eMMC
 - Variable height / location connections
 - Unique, permanent, stable.
 - Verify state against reality
 - One job, do it well
 - KISS, single PCB
 - Boot from power up into safe mode
 - Queueing of requests
 - Hardware, firmware & software
 - Not dangling from cable.

Status

- Developing a new SD multiplexer will take significant time.
- ARMv7 UBoot development will start to reduce.
- How to test UEFI?
- Is mass testing only for kernels?
 - Is this why we can't have nice things?

What can we learn?

- Kernelci works because there is an automated recovery
 - **Kill power.**
- Linux is a single platform.
- One criterion from cold start:
 - Can we automate delivery of new content
 - **before** the next boot?

The lessons of SD MUX

- Any sufficiently complex component requires testing.
- All mass testing benefits from automation & recovery.
- Testing cannot be done with vapourware hardware.
- Hobbyist solutions do not scale to mass testing.
- Design for mass testing or it won't happen.

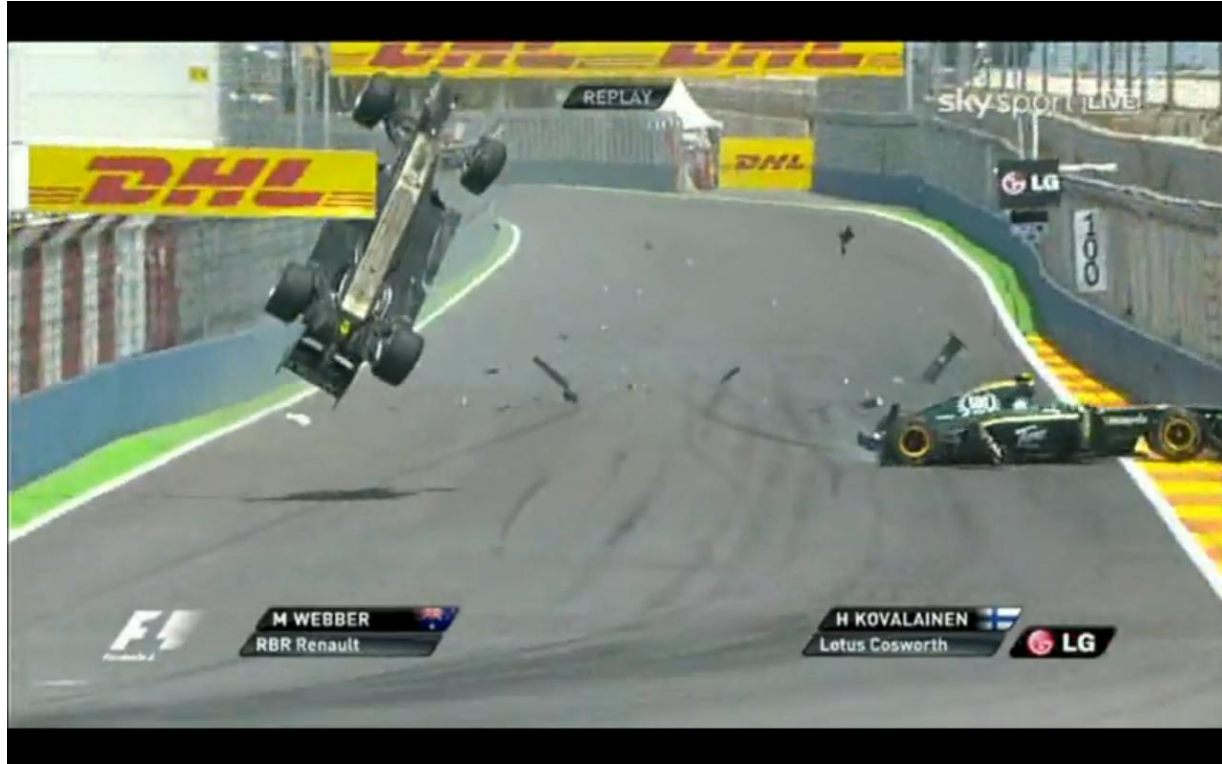
Mass testing

- Testing absolutely requires recovery.
- Matrix of options and failure modes requires scale.
- Keep production hardware "pure"?
- The benefits are sufficiently clear to everyone else
 - if your snowflake makes it hard, adapt or lose out.

Things will go wrong ...



... and much too expensive to change it later ...



Kernels vs bootloaders, UEFI & firmware

- KernelCI tests a single kernel.
- Similar matrix of boards, SoCs, configs.
- Automated testing needs to cover multiple methods.
- UEFI & UBoot each have endless variations in config, requirements & upstreams.
- Same infrastructure can be used
- Identify the benefits of mass testing of the entire boot

Quick thought experiment

- 2 boards, 1 running a vendor kernel, 1 running mainline
- Admins at a conference on a different continent.
- Which board(s) do you upgrade remotely?
- Does that change with 20 boards or 200 boards?
- Now swap “kernel” for “bootloader” in each case...
 - What if the distribution bootloader build had a process of mass testing?

Problem of snowflakes

- Diversity leads to multiple solutions.
- Little thought given to automated testing.
 - Bootloader testing needs more than a software solution.
- Kernel community has mechanisms to control diversity
- Lack of appetite for a **platform** model?
- Lack of cooperation at the hardware level?
- Lack of software tools hinders creation of hardware tools?

Just because I can

- *... does that mean I should?*
- ARM on Enterprise - aiming for consistency
- KernelCI works with diverse devices, a single community.
- Hardware mistakes are expensive
- Can we start making **platforms** instead of snowflakes?
- Mass testing of kernels with a single platform - kernelci.
- Mass testing of platforms with a single kernel is **blocked**.

Resisting more snowflakes

- Why is this one different?
- Implications of snowflake approach
 - Distribution support, mainline & upstream support.
 - Community involvement.
 - Snowflakes all the way down.
- Diversity can be good
 - Diversion to merely be a snowflake is harmful

Q&A