What is a farm?
I’m guessing you wouldn’t be here if you didn’t already know.

- A board farm allows for the automated control of (typically) embedded devices such as reference boards (hardware agnostic)
- At a minimum allows you to power on/off the board and access a serial console
- Can provide the means to put newly built software onto the board and run a test suite
What is a farm?

More likely than not it’s a “pile of stuff” – [Collabora]
Witekio Farm

Also a sprawling mess

- Originally Built to support remote working and hardware sharing
- Rackmount with removable shelves that have common connectivity
- One single desktop PC that acts as the farm
- Access via SSH and a propriety set of ‘ebfarm’ scripts
- Now used as part of kernelci.org and to run automated tests for our customers
- Capabilities:
  - Control of power
  - USB relay for toggling DIP switches, pressing buttons
  - SDMux for SD card control
  - HDMI receiver for skype video access to boards [defunct]
  - Workspaces, and shared access control
  - Container isolation
Automatically build deploy software via customer SD card update mechanism
- Allows testing upgrades work and is power failure tolerant
- Executes a set of manual tests including a soak test which provides CPU idle information to track performance regressions
- Automated emails with test results and graphs showing historical changes.
Our Challenges

It hasn’t been easy

- USB issues (hubs disappearing, reappearing)
- Reliability and lack of notification when things go wrong
- Our SDMux is not perfect
- OpenVZ container has been an obstacle
- Maintaining the farm
- Scaling the farm – can we easily duplicate it to other sites? Probably not
- Automated testing with TCL/Expect – odd issues that only show up in test and difficult to debug
- Lack of support – not managed by IT managed by engineers.
Farms are a hot trend
Enabling the proliferation automated testing

- Loads of interest surrounding this topic, but why?
- Farms are enabling automated testing in a big way in the OSS community:
  - Kernelci.org
  - OSADL RT test lab (osadl.org)
  - Qualcomm ‘Boardfarm’ (OpenWRT) (github.com/qca/boardfarm)
  - Intel 0-day test bot (01.org/lkp)
  - Any others?
- Also used a lot in private
- Automated tests help prevent regression and maintain a level of quality
- Farms have other uses too – board sharing, remote development, etc
- We’re probably just touching the surface
The problem?

Too much diversity, not enough collaboration

- Very little collaboration:
  - no de-facto blueprint for creating a farm
  - no knowledge resource available
  - Limited collaboration on software

- Perhaps this is why every farm is completely unique in hardware and software

- We’re probably all facing the same challenges:
  - Making a scalable and reliable platform
  - Adding new capabilities

- We’re all in our own silos working on the basics
A missing piece?

- Do we need to work together on hardware and hardware abstraction?

CI
- LAVA, Jenkins, kernelci, test framework

Hardware abstraction
- Free Electrons ‘Lavabo’
- Our ‘ebfarm’
- What do you use?

Hardware
- PDUs, USB relays
- SDMux’s
- BayLibre’s capes/probes
The vision

A better solution

- We can achieve more if we do it together
- We can move on from the basics and focus on the stuff that generates value – often testing
- Reduce the friction when integrating with existing higher layers.
- At a minimum bring farmers together
Open Discussion
Let’s discuss

Open Discussion 1

- Who are we? What are we doing with farms? What problem are we trying to solve?
  - Do we have a common goal?
  - Learn from each other

- What challenges do you face?
  - Bad/cheap hardware? Lack of hardware skills? Scalability?

- What solutions already exist?
  - Are there technologies that we didn’t know were out there?
  - Lavabo, Tizen SD Mux, BayLibre power probes, LAVA, etc

- How can we collaborate?
  - Would a wiki be useful? A mailing list?
  - Leave your business card on the way out and we can create a list
Let’s discuss

Open Discussion 2

- What needs to be done?
  - Some open source farm framework? (apt-get install farm)
  - Collaborate on software support for relays, capes, sd-muxs, PDUs, etc?
  - Collaborate on hardware to create new capabilities (e.g. USB removal)
  - Produce a pool of knowledge on available hardware, where to buy it, how to wire up tricky boards?
  - Produce a blueprint/standard for a farm? From Linux distribution through to higher level interfaces?
  - Exclusive access? Board sharing? Documentation?
  - Self describing farm shelves (via USB sticks describing layout?)
  - Configuration files for existing boards?
  - Improve reliability? Infrastructure issues.

- What needs to happen next?

- Anything else?
Accountability & Commitment
Cutting Edge expertise
Agility & Scalability  Reliability
Local Teams  Worldwide presence
Global Actions
Compliance with customers teams

Thank you