Piece of cake – testing remote embedded devices made easy using open-hardware MuxPi

Paweł Wieczorek
February 3, 2018
Samsung R&D Institute Poland
Agenda

1. Introduction
2. Previous efforts
3. Idea
4. Hardware
5. Software
6. Next steps
7. Conclusion
Introduction
Tizen use cases

Release engineering

- Continuous platform development
- QA step prior pulling new changes
- Package internal tests are not enough
Remote accessibility

- Easy to store in a secure manner
- Less effort than per developer
- Better utilized when shared
Previous efforts
- Linaro Automated Validation Architecture
- Automation system for deploying operating systems
- Virtual and physical hardware supported
- Allows running boot, bootloader and system level tests
Statistics

Since May 2014 we:

- Ran 13,432 jobs on 65 unique trees and 12,921 unique kernels.
- Performed 2,039,645 builds on 277 unique defconfigs.
- Performed 3,494,550 boots on 271 unique boards, across 3 architectures and 34 unique SoCs.

https://kernelci.org/stats/
SD MUX – open hardware

https://git.tizen.org/cgit/tools/testlab/sd-mux
Autohat board (SD MUX-based)

https://github.com/resin-io/autohat-board
$ dmesg | tail -12
[ 98.375599] usb 3-1: new full-speed USB device number 12 using xhci_hcd
[ 98.487663] usb 3-1: device descriptor read/64, error -71
[ 98.703656] usb 3-1: device descriptor read/64, error -71
[ 98.919658] usb 3-1: new full-speed USB device number 13 using xhci_hcd
[ 98.919969] usb 3-1: Device not responding to setup address.
[ 99.123998] usb 3-1: Device not responding to setup address.
[ 99.327681] usb 3-1: device not accepting address 13, error -71
[ 99.439718] usb 3-1: new full-speed USB device number 14 using xhci_hcd
[ 99.440049] usb 3-1: Device not responding to setup address.
[ 99.644028] usb 3-1: Device not responding to setup address.
[ 99.847719] usb 3-1: device not accepting address 14, error -71
[ 99.847819] usb usb3-port1: unable to enumerate USB device
Idea
Lessons learnt

STOP

11/32
Hardware
MuxPi
MuxPi components

- **NanoPi NEO**
  - USB / UART SWITCH & ID
  - 4-wire UART Level Shifter
  - Watchdog Timer

- **Power Control & Current Measurement**
- **μC Cortex-M0**
  - CTRL, ADC, DyPers, HDMI, GPIO, UI

- **SD-MUX**
  - 2-Port USB HUB
  - SD-READER

- **UI**
  - 4x LED, 2x Button, OLED Display

- **2x 2-channel DyPers**

- **Connectors**
  - ETH in
  - USB OTG
  - 5Pin USB
  - UART
  - 2x PWR
  - BARREL
  - USB ETH
  - 2x USB-A
  - μSD Adapter
  - HDMI
  - ADD-ONS
  - 4x DyPer

13/32
MuxPi components

- NanoPi NEO
  - USB / UART SWITCH & ID
  - 4-wire UART Level Shifter
  - Watchdog Timer
  - Power Control & Current Measurement
  - μC Cortex-M0
    - CTRL, ADC, DyPers, HDMI, GPIO, UI
  - Power Supply
  - UI
    - 4x LED, 2x Button, OLED Display
  - 2x 2-channel DyPers

- Connectors
  - ETH TS
  - USB OTG
  - 5Pin USB
  - UART
  - 2x PWR
  - BARREL

- SD-MUX
  - 2-Port USB HUB
  - SD-READER

- Connectors
  - ETH OUT
  - 2x USB-A
  - μSD Adapter
  - HDMI
  - ADD-ONS
  - 4x DyPer
MuxPi components

NanoPi NEO
- USB / UART SWITCH & ID
- 4-wire UART Level Shifter
- Watchdog Timer

2-Port USB HUB
- SD-READER

SD-MUX
- μC Cortex-M0
  - CTRL, ADC, DyPers, HDMI, GPIO, UI

Power Control & Current Measurement

Power Supply
- 4x LED, 2x Button, OLED Display

UI

2x 2-channel DyPers

Connectors
- ETH T5
- USB OTG
- 5Pin USB
- UART
- 2x PWR
- BARREL

Connectors
- ETH OUT
- 2x USB-A
- μSD Adapter
- HDMI
- ADD-ONS
- 4x DyPer
MuxPi components
MuxPi components
Indicators
Easy maintenance

--- BORUTA ---
192.168.0.2
Extensibility
Major improvements

- Independent (standalone)
- Aware of its state
- Easy to maintain
- Extensible from start
Building your own

NanoPi NEO ≈ $10
## Building your own

<table>
<thead>
<tr>
<th></th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>NanoPi NEO</td>
<td>$10</td>
</tr>
<tr>
<td>Parts</td>
<td>$80</td>
</tr>
</tbody>
</table>

[https://git.tizen.org/cgit/tools/muxpi/

---
Building your own

NanoPi NEO ≈ $10
Parts ≈ $80
Soldering skills High

https://git.tizen.org/cgit/tools/muxpi
Building your own

<table>
<thead>
<tr>
<th>Item</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>NanoPi NEO</td>
<td>$10</td>
</tr>
<tr>
<td>Parts</td>
<td>$80</td>
</tr>
<tr>
<td>Soldering skills</td>
<td>High</td>
</tr>
<tr>
<td>Patience</td>
<td>A LOT</td>
</tr>
</tbody>
</table>
**Building your own**

<table>
<thead>
<tr>
<th>NanoPi NEO</th>
<th>$10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parts</td>
<td>$80</td>
</tr>
<tr>
<td>Soldering skills</td>
<td>High</td>
</tr>
<tr>
<td>Patience</td>
<td>A LOT</td>
</tr>
</tbody>
</table>

https://git.tizen.org/cgit/tools/muxpi
Software
Multitier architecture

- “Do One Thing and Do It Well”
- RESTful HTTP APIs
- Homogeneous solution stack
Responsibilities

- Who knows what requires verification?
- Who knows which actions are necessary?
- Who knows where can it be done?
- Who knows how to do it?
Responsibilities

• Who knows what requires verification? **Perun**

• Who knows which actions are necessary?

• Who knows where can it be done?

• Who knows how to do it?
Responsibilities

• Who knows what requires verification? **Perun**

• Who knows which actions are necessary? **Weles**

• Who knows where can it be done?

• Who knows how to do it?
Responsibilities

- Who knows what requires verification? **Perun**
- Who knows which actions are necessary? **Weles**
- Who knows where can it be done? **Boruta**
- Who knows how to do it?
Responsibilities

• Who knows what requires verification? Perun
• Who knows which actions are necessary? Weles
• Who knows where can it be done? Boruta
• Who knows how to do it? MuxPi
MuxPi (farm)

- Manages single DUT
- Fully aware of its capabilities
- Requires only two interfaces
  - Power supply
  - Network connection (Ethernet)
• Dryad farm management system
• Schedules requests
  • Priority
  • Device groups
  • Delayed access
• Provides convenient access to selected Dryad
- Lightweight testing framework
- Provides LAVA-like interface
- YAML job definition ➔
  actions executed on DUT
  - Deploy
  - Boot
  - Test
  - Collect
• OS images testing system
• Schedules verification (per new set of OS images)
• Automates QA step of Release Engineering Duty
Keeping it simple
Keeping it simple (and decoupled)
Keeping it simple (and decoupled)
Keeping it simple (and decoupled)
Keeping it simple (and decoupled)
Next steps
Future plans

Hardware

• Audio I/O
• USB Type C investigation
• NanoPi serial console on USB

Software

• Web interfaces for current layers
• Service state management
• Release engineer's layer
Conclusion
Summary

- Quick setup
- Easy maintenance
- Responsibilities division
- Execution parallelization
- Environment unification
Questions?

• MuxPi
  
  https://wiki.tizen.org/MuxPi

• SD MUX (deprecated – lesson learnt)
  
  https://wiki.tizen.org/SD_MUX

• Mailing list
  
  general@lists.tizen.org

• #tizen on Freenode
  
  https://webchat.freenode.net/?channels=tizen
• *Metropolis* – simple, modern Beamer theme
Pictures used

- https://commons.wikimedia.org/wiki/File:Tux.svg
- https://commons.wikimedia.org/wiki/File:Wayland_Lo.png
- https://commons.wikimedia.org/wiki/File:Enlightenment_logo_black.png
- https://developer.tizen.org/sites/default/files/images/about_tizen_1.png
- https://commons.wikimedia.org/wiki/File:ColoredBlankMap-World-10E.svg
- https://commons.wikimedia.org/wiki/File:Nuvola_Korean_flag.svg
- https://commons.wikimedia.org/wiki/File:Nuvola_Polish_flag.svg
- https://commons.wikimedia.org/wiki/File:Nuvola_USA_flag.svg
- https://validation.linaro.org/static/docs/v2/_images/lava.svg
- https://wiki.linaro.org/Platform/LAB/LMP_in_practice
- https://forums.resin.io/uploads/resin/original/1X/88ab2e061cd644b18b95fa99ede9ce6b98adfa44.jpg
- https://commons.wikimedia.org/wiki/File:Italian_traffic_signs_-_fermarsi_e_dare_precedenza_-_stop.svg
- https://farm9.staticflickr.com/8263/28955874330_d1b1202ae8_k_d.jpg
- https://commons.wikimedia.org/wiki/File:PEO-smiley_smile.svg