

Towards common mainline device testing

Nícolas F. R. A. Prado

LPC 2024

Motivation

- Large fraction of code base comes from devices
- Device regressions affect system functionality
- Covering everything requires subsystem-specific testing
- Testing generic device framework is simpler and yet fruitful
- **Goal:** detect regressions through generic device layer
- Objectives:
 - Minimum (ideally none) false-positives
 - Minimum maintenance

Generic device testing status

- Areas that have had tests submitted for:
 - Probe
 - [DT kselftest](#), [ACPI kselftest](#), [Discoverable bus \(USB/PCI\) kselftest](#)
 - Device error logs
 - [Error log kselftest](#)
 - Existence in /sys/bus/ or /sys/class/
 - [Device existence kselftest](#)
- Other areas?

Device existence kselftest

- Currently an RFC on the list:
 - <https://lore.kernel.org/all/20240724-kselftest-dev-exist-v1-1-9bc21aa761b5@collabora.com/>
- Two steps:
 - Reference generation
 - Validation against reference

Reference generation

- Run with `--generate-reference` on **known-good** kernel
- Device snapshot will be generated to YAML file:
- Reference will be stored out-of-tree in [kernelci/platform-test-parameters](https://github.com/kernelci/platform-test-parameters)

```
bus:
...
nvmem:
  count: 2
  devices:
  - info:
    uevent: 'OF_NAME=efuse

    OF_FULLNAME=/soc/efuse@11c10000

    OF_COMPATIBLE_0=mediatek,mt8195-efuse

    OF_COMPATIBLE_1=mediatek,efuse

    OF_COMPATIBLE_N=2
    '
  - info:
    uevent: 'OF_NAME=flash

    OF_FULLNAME=/soc/spi@1132c000/flash@0

    OF_COMPATIBLE_0=jedec,spi-nor

    OF_COMPATIBLE_N=1
    '

```



Validation against ref.

- Run without
 - -generate-reference
- Missing devices will cause failure
 - Based on device count on bus/class
- Failure example:

```
# Missing devices for subsystem 'nvmem': 1 (Expected 2, found 1)
# =====
# Devices expected:
#
#   uevent:
#     OF_NAME=efuse
#     OF_FULLNAME=/soc/efuse@11c10000
#     OF_COMPATIBLE_0=mediatek,mt8195-efuse
#     OF_COMPATIBLE_1=mediatek,efuse
#     OF_COMPATIBLE_N=2
#
#   uevent:
#     OF_NAME=flash
#     OF_FULLNAME=/soc/spi@1132c000/flash@0
#     OF_COMPATIBLE_0=jedec,spi-nor
#     OF_COMPATIBLE_N=1
#
# -----
# Devices found:
#
#   uevent:
#     OF_NAME=efuse
#     OF_FULLNAME=/soc/efuse@11c10000
#     OF_COMPATIBLE_0=mediatek,mt8195-efuse
#     OF_COMPATIBLE_1=mediatek,efuse
#     OF_COMPATIBLE_N=2
#
# -----
# Devices missing (best guess):
#
#   uevent:
#     OF_NAME=flash
#     OF_FULLNAME=/soc/spi@1132c000/flash@0
#     OF_COMPATIBLE_0=jedec,spi-nor
#     OF_COMPATIBLE_N=1
#
# =====
not ok 19 bus.nvmem
```



Open questions / gather feedback

- Is there any device property that is safe to match? (ie stable)
- Should probe also be checked? (What if it changes?)
- More identifying properties?
 - Currently: uevent, device/uevent, firmware_node/uevent and name
- More buses/classes to ignore? Currently: devlink
- Better way to ignore non-devices in /sys/class than checking for symlink?

Open questions / gather feedback (2)

- How to best select appropriate reference file?
 - Currently: system ID (DT compatible or DMI). Possibilities: kernel version, kernel config
- Better format for the reference file? Currently: YAML





Thank you!



COLLABORA

Open First



We are hiring
col.la/careers



COLLABORA

Open First

Stability of device properties

- E.g.: ChromeOS EC backlight device
- Sysfs path (unstable):
 - `/sys/devices/platform/soc/1100a000.spi/spi_master/spi0/spi0.0/cros-ec-dev.2.auto/cros-keyboard-leds.13.auto/leds/chromeos\:\:kbd_backlight/`
- Uevent (unstable?):
 - `DEVTYPE=mfd_device`
 - `DRIVER=cros-keyboard-leds`
 - `MODALIAS=platform:cros-keyboard-leds`



Stability of device properties (2)

- Class path (stable-ish?):
 - `/sys/class/leds/chromeos\:\:kbd_backlight/`
- Lack of docs on device properties in Documentation/ABI
- What would break userspace?
 - `light` & `brightnessctl` use only name in class path
 - But suffix can be added in case of name clash



DT kselftest

- Skip compatibles that don't match any driver
- Fail on unprobed devices

```
1 TAP version 13
2 1..276
3 ok 1 / # SKIP
4 ok 2 /backlight-lcd0
5 ok 3 /cpus/cpu@0 # SKIP
6 ...
7 ok 221 /soc/spi@1100a000
8 ok 222 /soc/spi@1100a000/ec@0
9 ok 223 /soc/spi@1100a000/ec@0/i2c-tunnel
10 ok 224 /soc/spi@1100a000/ec@0/i2c-tunnel/sbs-battery@b
11 not ok 225 /soc/spi@1100a000/ec@0/keyboard-backlight
12 ok 226 /soc/spi@1100a000/ec@0/keyboard-controller
13 ok 227 /soc/spi@1100a000/ec@0/regulator@0
14 ok 228 /soc/spi@1100a000/ec@0/regulator@1
15 ok 229 /soc/spi@1100a000/ec@0/typec
16 ok 230 /soc/spi@1100a000/ec@0/typec/connector@0 # SKIP
17 ok 231 /soc/spi@1100a000/ec@0/typec/connector@1 # SKIP
18 ok 232 /soc/spi@1132c000
19 ok 233 /soc/spi@1132c000/flash@0
20 ...
21 # Totals: pass:203 fail:2 xfail:0 xpass:0 skip:71 error:0
```



Discoverable bus kselftest

- Reference files written by hand
 - Describes HW path to devices on USB or PCI bus
- Test checks device exists and probed
- Example `google, tomato.yaml`:

```
1 # SPDX-License-Identifier: GPL-2.0
2 # Copyright (c) 2024 Collabora Ltd
3 - type: usb-controller
4   dt-mmio: 112a0000
5   usb-version: 2
6   devices:
7     - path: 1
8       interfaces: [0, 1]
9       name: camera
10 - type: usb-controller
11   dt-mmio: 112b0000
12   usb-version: 2
13   devices:
14     - path: 1
15       interfaces: [0, 1]
16       name: bluetooth
17 - type: pci-controller
18   dt-mmio: 112f8000
19   devices:
20     - path: 0.0/0.0
21       name: wifi
```



Error Log kselftest

```
1 TAP version 13
2 1..10
3 # mtk-socinfo mtk-socinfo.0.auto: error -ENOENT: Failed to get socinfo data
4 # mtk-socinfo mtk-socinfo.0.auto: probe with driver mtk-socinfo failed with error -2
5 not ok 1 +platform:mtk-socinfo.0.auto
6 # mtk-scp 10500000.scp: error -EINVAL: invalid resource (null)
7 not ok 2 +platform:10500000.scp
8 # mt8183-pinctrl 10005000.pinctrl: pin GPIO7 already requested by 2-002c; cannot claim for 2-0015
9 # mt8183-pinctrl 10005000.pinctrl: error -EINVAL: pin-7 (2-0015)
10 # mt8183-pinctrl 10005000.pinctrl: error -EINVAL: could not request pin 7 (GPIO7) from group GPIO7 on device pinctrl_paris
11 not ok 3 +platform:10005000.pinctrl
12 # elan_i2c 2-0015: Error applying setting, reverse things back
13 not ok 4 +i2c:2-0015
14 # generic-adc-thermal thermal-sensor1: Thermal zone sensor register failed: -22
15 # generic-adc-thermal thermal-sensor1: probe with driver generic-adc-thermal failed with error -22
16 not ok 5 +platform:thermal-sensor1
17 # generic-adc-thermal thermal-sensor2: Thermal zone sensor register failed: -22
18 # generic-adc-thermal thermal-sensor2: probe with driver generic-adc-thermal failed with error -22
19 not ok 6 +platform:thermal-sensor2
20 # mt8183_mt6358_ts3a227 mt8183-sound: ASoC: driver name too long 'mt8183_mt6358_ts3a227_max98357' -> 'mt8183_mt6358_t'
21 not ok 7 +platform:mt8183-sound
22 # power_supply sbs-12-000b: driver failed to report `technology' property: -5
23 # power_supply sbs-12-000b: driver failed to report `technology' property: -5
24 # power_supply sbs-12-000b: driver failed to report `technology' property: -5
25 # power_supply sbs-12-000b: driver failed to report `technology' property: -5
26 # power_supply sbs-12-000b: driver failed to report `technology' property: -5
27 # power_supply sbs-12-000b: driver failed to report `technology' property: -5
28 # power_supply sbs-12-000b: driver failed to report `technology' property: -5
29 # power_supply sbs-12-000b: driver failed to report `technology' property: -5
30 # power_supply sbs-12-000b: driver failed to report `technology' property: -5
31 # power_supply sbs-12-000b: driver failed to report `technology' property: -5
32 # power_supply sbs-12-000b: driver failed to report `technology' property: -5
33 # power_supply sbs-12-000b: driver failed to report `technology' property: -5
34 # power_supply sbs-12-000b: uevent: failed to send synthetic uevent: -5
35 not ok 8 +power_supply:sbs-12-000b
36 # mtk-svs 1100bc00.svs: error -ENODEV: svs bank resource setup fail
37 not ok 9 +platform:1100bc00.svs
38 # mtk-mdp3 14001000.dma-controller0: can't get SCP node
39 not ok 10 +platform:14001000.dma-controller0
40 # Totals: pass:0 fail:10 xfail:0 xpass:0 skip:0 error:0
```

