



Reviving early platform drivers

Bartosz Golaszewski

Senior Staff Engineer, Qualcomm

Prague 2026

Linux Plumbers Driver Core Microconference

Snapdragon and Qualcomm branded products are products of Qualcomm Technologies, Inc. and/or its subsidiaries.



Agenda



What is OF_DECLARE()?



Why is OF_DECLARE()?



Early Platform Drivers/Devices



Seamless Tree-wide Conversion?

What is OF_DECLARE()



What is OF_DECLARE()

- Places an `of_device_id` entry in a dedicated linker section
 - Sections defined in `include/asm-generic/vmlinux.lds.h`
- Subsystem code can iterate over the entries at any time and match OF nodes against the compatibles and potentially call associated callbacks
- ACPI has a similar thing: `ACPI_DECLARE_PROBE_ENTRY()`

Why is OF_DECLARE()



Why is OF_DECLARE()

- Driver model comes up pretty late, some devices need to be initialized very early
 - earlycon: OF_EARLYCON_DECLARE()
 - clocks: OF_CLK_DECLARE()
 - timers: TIMER_OF_DECLARE()
 - irqchips: IRQCHIP_DECLARE()
- The registered “early” entries are iterated over before initcalls, matched against OF nodes and the associated callbacks executed

Early Platform Drivers/Devices



Early Platform Drivers/Devices

- `OF_DECLARE()` ends up being just a function called for any matching OF node
- The same compatible may or may not later be matched for a true platform driver
 - If it is: it lacks any context of whatever initialization happened before
 - If it's not: it never becomes visible in the system as a regular device (unless `CLK_OF_DECLARE_DRIVER()`)
- Requires strict ordering at build-time as callbacks can't defer/use devlink
- `OF_POPULATED` inconsistencies
- Callbacks can't follow driver model good practices
- `aic_of_ic_init()` is close to 200 lines long but can't reuse any driver model APIs

Early Platform Drivers/Devices

- Different approach to early platform drivers existed on SH-Mobile and Blackfin
- Currently: legacy code confined to `arch/sh/` and replaced with `OF_DECLARE()`
- Previous implementation existed before devicetree, firmware nodes, device links, etc.

Early Platform Drivers/Devices

- Is it possible to fold OF_DECLARE () into the driver model?
- Potential for unification
 - Earlycon is registered really early (before unflattening) and supports console hangover
 - The rest is matched after the device_node tree is ready

Early Platform Drivers/Devices

- Patches: <https://lore.kernel.org/all/20180511162028.20616-1-brgl@bgdev.pl/>

```
On Fri, May 11, 2018 at 11:20 AM, Bartosz Golaszewski <brgl@bgdev.pl> wrote:
> This series is a follow-up to the RFC[1] posted a couple days ago.
>
> NOTE: this series applies on top of my recent patches[2] that move the previous
> implementation of early platform devices to arch/sh.
>
> Problem:
>
> Certain class of devices, such as timers, certain clock drivers and irq chip
> drivers need to be probed early in the boot sequence. The currently preferred
> approach is using one of the OF_DECLARE() macros. This however does not create
> a platform device which has many drawbacks - such as not being able to use
> devres routines, dev_log functions or no way of deferring the init OF function
> if some other resources are missing.
```

```
I skimmed though this and it doesn't look horrible (how's that for
positive feedback? ;) ). But before going into the details, I think
first there needs to be agreement this is the right direction.
```

Seamless Tree-wide Conversion?

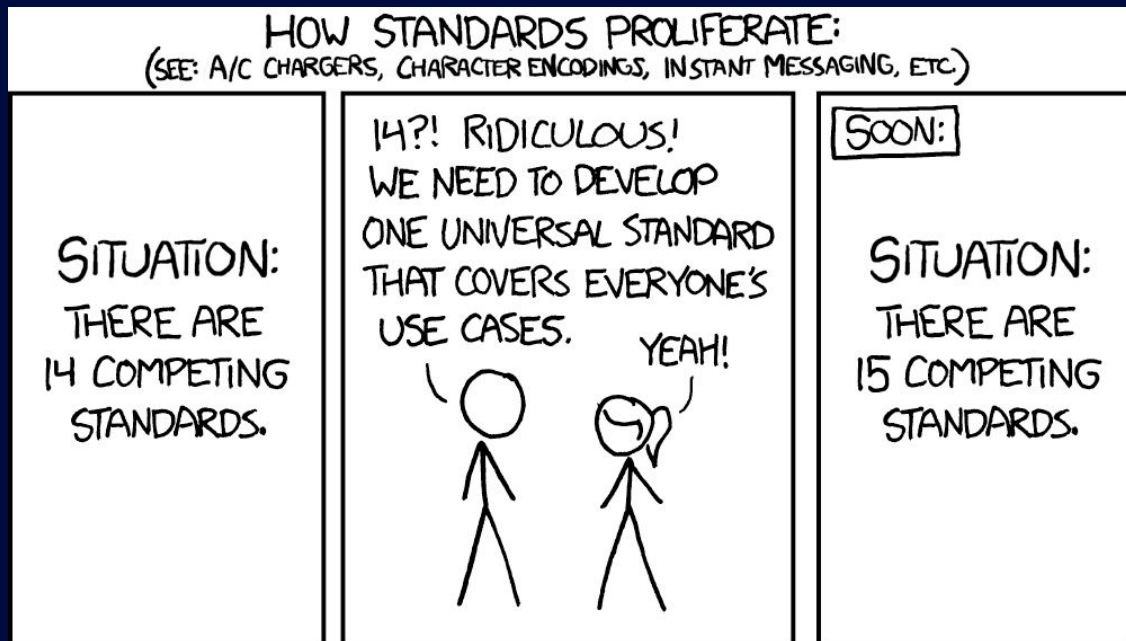


Seamless Tree-wide Conversion?

- Main concern from reviews:

Seamless Tree-wide Conversion?

- Main concern from reviews:



Seamless Tree-wide Conversion?

- Problem: we'd introduce early platform drivers but existing code would keep on using `OF_DECLARE( )` indefinitely

The Plan

1. Introduce Early Platform Driver core
 - a. `struct early_platform_device` embedding `struct platform_device`
 - b. `.early_probe()` callback
 - c. Subset of device APIs preferred in `.early_probe()`
 - d. Support for a subset of probe deferral and devlink
 - e. Promotion to “real” platform devices after driver model is up
2. Conversion of `OF_DECLARE()`
 - a. Single linker table (as with early drivers, ordering is no longer required)
 - b. `.early_probe()` wrapper around existing `OF_DECLARE()` callbacks
 - c. After the early platform bringup: iterate over registered entries and instantiate early platform devices
 - d. Early-devlink makes sure we satisfy dependencies at early probe

Thank you

Nothing in these materials is an offer to sell any of the components or devices referenced herein.

© Qualcomm Technologies, Inc. and/or its affiliated companies. All Rights Reserved.

Qualcomm and Snapdragon are trademarks or registered trademarks of Qualcomm Incorporated. Other products and brand names may be trademarks or registered trademarks of their respective owners.

References in this presentation to “Qualcomm” may mean Qualcomm Incorporated, Qualcomm Technologies, Inc., and/or other subsidiaries or business units within the Qualcomm corporate structure, as applicable. Qualcomm Incorporated includes our licensing business, QTL, and the vast majority of our patent portfolio. Qualcomm Technologies, Inc., a subsidiary of Qualcomm Incorporated, operates, along with its subsidiaries, substantially all of our engineering, research and development functions, and substantially all of our products and services businesses, including our QCT semiconductor business.

Snapdragon and Qualcomm branded products are products of Qualcomm Technologies, Inc. and/or its subsidiaries. Qualcomm patents are licensed by Qualcomm Incorporated.

Follow us on: [in](#) [X](#) [@](#) [v](#) [f](#)

For more information, visit us at [qualcomm.com](https://www.qualcomm.com) & [qualcomm.com/blog](https://www.qualcomm.com/blog)

