



Contribution ID: 198

Type: **not specified**

Device Tree Metadata: How Bootloaders DTBs Pick and Apply Overlays

On many device tree based devices, the device tree blobs are commonly shipped with the kernel or OS image, not the firmware. If the image is meant to be generic, it would include multiple DTBs and possibly many DTBO combinations. The bootloader selects a DTB and optionally applies overlays matching the hardware. Known image “standards” include:

- FIT image: maps a compatible string to a kernel image, a DTB and zero or more DTBOs
- Android’s DTBO partition format: allows up to six 32-bit unsigned integers for DTB/DTBO matching, but the mechanism is entirely implementation defined

Elliot’s “Shipping Multiple Devicetrees: How to Identify Which DTB Is for My Board?” [1] talk from ELC 2024 touched on the need for some generic property(s) for identification. Armit proposed a generic “board-id” property [2] for this purpose.

Putting the board identifiers in the device tree itself only solves half the problem. Device tree overlays cannot reuse the same identifiers, as properties cannot be stacked or appended together. The overlays have extra selection criteria that are often tied to the firmware. The base DT needs to be identified. These bits of “metadata” are what is needed to put together a fully usable image for the firmware to load. The author’s own session “Second Source Component Probing on Device Tree Platforms” [3] at ELC 2024 briefly mentioned this.

This talk aims to start a discussion on what bits are needed for various bootloaders and images, and how to store that metadata close to if not within the source files.

[1] <https://sched.co/1aBFy>

[2] https://lore.kernel.org/all/1710418312-6559-1-git-send-email-quic_amrianan@quicinc.com/

[3] <https://sched.co/1aBGe>

Primary author: TSAI, Chen-Yu (Google, LLC)

Presenter: TSAI, Chen-Yu (Google, LLC)

Session Classification: Devicetree MC

Track Classification: Devicetree MC