Qualcomm

SDCA and DeviceTrees

Srini Kandagatla

Senior Staff Engineer, Qualcomm Technologies International ,Ltd

Agenda

MIPI and SDCA – Intro

SDCA?

Problem Statement

Proposals

MIPI and SDCA – Intro

- MIPI Disco
  - Developed by: Software Working Group, An ACPI-based software architecture for discovering and enumerating devices that support MIPI Alliance protocol specifications on a system
- MIPI SoundWire®
  - Developed by: Audio Working Group , Introduced in 2014, consolidates many of the key attributes in mobile and PC audio interfaces, providing a common, comprehensive interface and scalable architecture that can be used to enable audio features and functions in multiple types of devices and across market segments.
- MIPI Disco SoundWire®
  - Developed by: Software Working Group, Enables SoundWire driver developers to easily discover and configure devices implementing MIPI SoundWire interfaces.
- MIPI SoundWire® Device Class for Audio.
  - Developed by: Software Working Group, Enables standardized mechanisms to interact with host-controllable audio devices connected via a MIPI SoundWire interface

Source: Interface Specifications for Mobile Products | MIPI Allliance

SoundWire® Device Class for Audio

- Enables standardized mechanisms to interact with host-controllable audio devices connected via a MIPI SoundWire interface
  - Facilitates the development of generic SoundWire® software that can provide audio functionality using the native operating system "out of-the-box" without being dependent upon the availability of device-specific or system-specific software. **Meaning audio using generic drivers and not needing vendor-specific drivers.**
  - Provide a framework for functionality of some common audio components (e.g., analog jacks, amplifiers, and microphones).
  - Optimizes integration of microphones, amplifiers and other audio devices into host platforms.

Source: Download MIPI SoundWire Device Class for Audio

Problem Statement

- SDCA Generic Drivers
  - MIPI SoundWire® Device Class for Audio(SDCA) Specifications is based on the ACPI by design, Generic drivers are implemented using fwnode APIs, On the other hand we have some level of ACPI and Device Tree unification at the kernel API level aka fwnode. Are we implying that generic drivers should work with device trees too?
  - Should Device Trees support Generic drivers?
- Device Tree Bindings.
  - SDCA aims at booting a system and deliver acceptable quality audio without needing vendor-specific drivers.
  - Should Device Tree support such specifications?

Proposals

- Option – 1: Should we add exactly same device tree bindings for SDCA?
  - Pros:
    - generic drivers work out of box.
  - Cons:
    - register level sequences part of the bindings.
- Option – 2: Ignore SDCA and add bindings as required
  - Pros:
    - Avoid some register sequences in dt bindings.
  - Cons:
    - Can not use generic drivers, Deviating from the Standard Interface specs
- Option – 3: Use Secondary fwnode
  - Pros:
    - Generic drivers work out of box which have access to ACPI tables.
  - Cons:
    - Platforms without ACPI tables, purely DT based might be left out.
    - Any concerns from DT Maintainers mixing DT and ACPI tables?

Thank you

Follow us on:

For more information, visit us at qualcomm.com & qualcomm.com/blog