Challenges in Device Tree Sync - kernel, Zephyr, U-boot, System DT

Date: 2023-11-15

Nishanth Menon
About us: TI Processors and Open source

Decades of contribution and collaboration

Ingrained culture to give back to the community

Upstream FIRST!

Focus on long term, sustainable and quality products

Upstream and opensource ecosystem in device architecture

Upstream FIRST mentality!
About me

Senior Member Technical Staff at Texas Instruments, Dallas.

- Community member
- U-Boot and kernel.org developer
- Maintainer, TI K3 Device tree
- Just a user of the sink, not the plumber
Disclaimers

- This is a technology presentation, not product-readiness or roadmap commitment.
- Opinions presented here are that of the speaker and may not reflect that of Texas Instruments Inc, DT maintainers or any of the software ecosystems.
- This presentation is not going to solve world DT problems.
Overview

- The Long-short Tale of DT
- Perspectives
- Likes and Dislikes
- Solution? Middle ground?
- Discussion/Q&A
Long short tale of Device Tree

• SPARC/PowerPC antecedents

• See this presentation by Neil Armstrong: https://elinux.org/images/0/06/ELCE_2019_DeviceTree_Past_Present_Future.pdf

• Device Tree and YAML debates
  • https://www.konsulko.com/yaml-and-device-tree
  • https://static.linaro.org/connect/lvc21f/presentations/LVC21F-315.pdf

• Device Trees and overlays
  • https://docs.kernel.org/devicetree/overlay-notes.html
And why exactly did I start this journey?

- Blink an LED from A53 from Linux and another LED from Zephyr running on M4F on AM625 based BeaglePlay
- Wanted to do DT just once! I expected:
  - board.dts
  - Board-A53.dts
  - Board-M4F.dts
- But the solution, apparently, isn’t that simple!
The Elephant – AM625

Main Domain
- A53
- I2C
- UART
- WDT
- R5F

MCU Domain
- M4F
- I2C
- UART

Wakeup Domain
- I2C
- UART
- WDT
- ...

Figure 1-1. Functional Block Diagram
https://www.ti.com/lit/pdf/spruv7
Three views of the Elephant

- Penguin People
- Submarine People
- Kite People
Usage in penguin people ecosystem

- **Tools:**
  - dtc - [https://git.kernel.org/pub/scm/utils/dtc/dtc.git/](https://git.kernel.org/pub/scm/utils/dtc/dtc.git/)
- libfdt, dtc is imported from git.kernel.org tree
- Dt-schema checkers are maintained separately.
  - Core bindings are maintained in dt-schema repository
  - Specific bindings maintained in kernel.org Documentation/devicetree/bindings/
- Basic rules:
  - Must be hardware description (in some cases, firmware description is allowed)
  - Shall adhere to device tree bindings (in yaml)
- Unstated rules: quite a few, some of which involves usage by Linux drivers.
- Licensing: GPL2
SoC View from the penguin people
DT View from penguin people

- How does the AM625 device tree structure look like?

```
board.dts  am625.dtsi
          am62.dtsi
          am62-wakeup.dtsi
          am62-mcu.dtsi
          am62-wakeup.dtsi

A53

bus

peripherals
```

board-overlay.dtso
Usage in submarine people ecosystem

• Tools:
  • libfdt, dtc - https://git.kernel.org/pub/scm/utils/dtc/dtc.git/
  • Tries to stay in sync with kernel.org
  • Also has it’s own additional bindings (doc/device-tree-bindings/)
    • Bootph binding – now part of dt-schema properties
    • Binman – in discussion, currently U-boot specific
  • Very similar to kernel.org (shares the same rules), import devicetree from kernel.org periodically to stay in sync, BUT..
    • Needs a few tweaks
    • Memory constraints
    • Has peripherals supported that kernel doesn't use.
  • Licensing: GPL2+
SoC View from the submarine people

Main Domain
- A53
- Main MPU
- SPL, U-Boot
- I2C
- UART
- WDT
- ...

MCU Domain
- R5F
- Boot processor
- SPL
- M4F
- I2C
- UART
- WDT
- ...

Wakeup Domain
- SPL
- Boot
- I2C
- UART
- WDT
- ...

U-Boot

Texas Instruments
How does the AM625 device tree structure look like?

- [https://source.denx.de/u-boot/u-boot/-/tree/master/arch/arm/dts](https://source.denx.de/u-boot/u-boot/-/tree/master/arch/arm/dts)

```
R5-Board.dts
```

- `board.dts`
- `am65.dts`
- `am62.dts`
- `am62-wakeup.dts`
- `am62-mcu.dts`


**Uses bootph + binman + DDR dti**

From kernel.org
Usage in kite people ecosystem

- Has its own bindings and device tree model.
- Uses C headers as actual link to Zephyr.
- Mix of Native drivers and HAL drivers
- Resource constraints, runtime overheads (See [https://www.youtube.com/watch?v=w8GgP3h0M8M](https://www.youtube.com/watch?v=w8GgP3h0M8M))
- Licensing: Apache-2.0
SoC View from the kite people

Main Domain
- Main MPU
- A53
- I2C
- UART
- WDT
- R5F

MCU Domain
- I2C
- UART
- WDT

Wakeup Domain
- I2C
- UART
- WDT

M4F
DT View from kite people

- How does the AM625 device tree structure look like?
  https://github.com/zephyrproject-rtos/zephyr/tree/main/dts/arm64/ti

boards/arm64/am62x_a53/phycore_am62x_a53.dts

boards/arm/am62x_m4/am62x_m4_sk.dts

dts/arm64/ti/ti_am62x_a53.dtsi

dts/arm/ti/am62x_m4.dtsi

https://docs.zephyrproject.org/latest/build/dts/troubleshooting.html is your friend!
And… other people

- Jailhouse [https://github.com/siemens/jailhouse/tree/master/configs/arm64/dts](https://github.com/siemens/jailhouse/tree/master/configs/arm64/dts)
- Xen ……
- Trusted Firmware Cortex-A …

Likes

- Linux Kernel
  - Bindings are strict
  - Definitions of what can and cannot be integrated is clear
- U-Boot
  - Integration is clean
  - Language remains “same-ish” as kernel
- Zephyr
  - Focus on resource and performance
Dis-Likes

- Linux Kernel
  - Flexibility
  - Performance and Bloat (-EPROBEDEFER?)
- U-Boot
  - Still feels like a bunch of band-aids
- Zephyr
  - Device-tree language is the only common part with the rest of the ecosystem.
  - The actual dts, in reality, looks nothing like U-Boot or Linux Kernel
Solution? Middle Ground?

- System Device tree and Lopper? [https://github.com/devicetree-org/lopper](https://github.com/devicetree-org/lopper)

![Diagram](image)
Impediments

- Are we willing to adopt lopper in various s/w ecosystems?
- DT Licensing? GPL-2 Vs GPL-2+ Vs Apache-2 … (and others)
  - Solution might be dual or multiple licensing
- What will be canonical source for DT, domain hints and bindings?
  - Will kernel maintainers be willing to host DT, domain hints and bindings not belonging to kernel?
  - Can dt and bindings move out of kernel tree? (hasn't so far).. 
- Can lopper be integrated into all the s/w flows? (zephyr has potential, U-boot, kernel?)
Benefits

- Consolidated tooling
- DT itself is hard for new users. But, users shouldn’t need to learn new DT details based on which OS they are on.
- “Universal DT” OR putting “DT in ROM” is not what I expect any time in near future, but hoping to see lesser chaos.
Credits and Acknowledgement

- Texas Instruments Inc.
- The Linux Foundation.
- Linaro and OpenAMP members
- Vaishnav Achat, Christopher Friedt, Arnaud Pouliquen, Bill Mills, Tom Rini, Krzysztof Kozlowski, Rob Herring, and so many more..
Q&A

• Contact Information:
  – Nishanth Menon <nm@ti.com>

• Also on IRC NishanthMenon @ libera.chat #linux-ti

• Zephyr discord (#ti channel) https://discord.gg/nG2yqHRg

• BeagleBoard.org discord (#beagleplay channel) https://discord.gg/nUQjwnyw

Learn more about TI products

– https://www.ti.com/linux
– http://opensource.ti.com/
– https://www.ti.com/processors
– https://www.ti.com/edgeai

Why choose TI MCUs and processors?

✓ Scalability
  Our products offer scalable performance that can adapt and grow as the needs of your customers evolve.

✓ Efficiency
  We design products that extend battery life, maximize performance for every watt expended, and unlock the highest levels of system efficiency.

✓ Affordability
  We strive to make innovation accessible to all by creating cost-effective products that feature state-of-the-art technology and package designs.

✓ Availability
  Our investment in internal manufacturing capacity provides greater assurance of supply, supporting your growth for decades to come.