



---

## STM32MP15 device tree



---

## Contents

---



A quality version of this page, approved on 5 January 2021, was based off this revision.

## Contents

|  |    |
|--|----|
| 1 Purpose and scope .....  | 4  |
| 2 Device tree structure for Linux, U-Boot, TF-A and OP-TEE ..... | 5  |
| 2.1 STMicroelectronics software delivery .....                   | 5  |
| 2.2 Upstreamed device tree .....                                 | 5  |
| 2.2.1 Overview .....   | 5  |
| 2.2.2 Linux .....  | 6  |
| 2.2.3 U-Boot .....   | 7  |
| 2.2.4 TF-A .....   | 8  |
| 2.2.5 OP-TEE .....   | 9  |
| 2.3 STM32CubeMX generated device tree .....                      | 9  |
| 2.3.1 Overview .....   | 9  |
| 2.3.2 Linux .....  | 10 |
| 2.3.3 U-Boot .....   | 10 |
| 2.3.4 TF-A .....   | 11 |
| 2.3.5 OP-TEE .....   | 11 |



---

## 1 Purpose and scope

---

This article shows the structure of the device tree files as it is implemented in STM32MP15 for the following software components: Linux<sup>®</sup> kernel, U-Boot, TF-A and OP-TEE.

The purpose of this article is not to explain the [device tree](#) concept and the [device tree configuration](#) supported by STM32MP15. A good knowledge of the concept is a prerequisite before reading this article. The articles regarding the configuration can be read afterwards.



## 2 Device tree structure for Linux, U-Boot, TF-A and OP-TEE

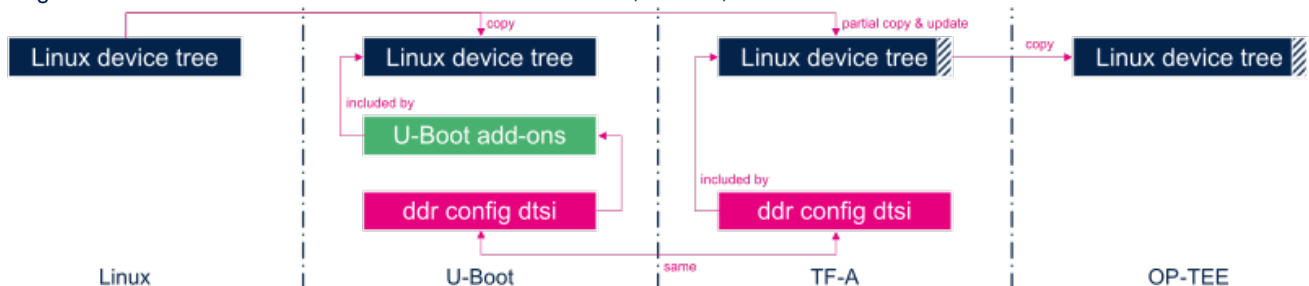
### 2.1 STMicroelectronics software delivery

The device tree files are used during **boot time** by the FSBL (TF-A), the secure OS (OP-TEE), the SSBL (U-Boot) and eventually by the Linux kernel.

These files are stored in different github repositories (one for each first level bullet below), but source code reuse is maximized from one repository to another:

- The initial device tree files supporting STM32MP15 are created in the **Linux** kernel.
  - The source code path is `arch/arm/boot/dts/stm32mp15*`.
- In **U-Boot**, Linux files are copied and overloaded with U-Boot properties (including DDR configuration).
  - The source code path is `arch/arm/dts/stm32mp15*`.
  - The U-Boot add-ons file, "`<dts-file-name>-u-boot.dtsi`", is automatically included at the end of "`<dts-file-name>.dts`" during device tree compilation (this is a generic U-Boot Makefile behavior).
- In **TF-A**, Linux files are partly copied (for device tree blob size optimization), completed with the security configuration (firewalling) and the DDR configuration (same files as U-Boot). **TF-A device tree bindings** reading is needed to understand which parts of the Linux device tree must be kept or can be deleted during the partial copy.
  - The source code path is `fdts/stm32mp15*`.
- In **OP-TEE**, TF-A files are reused, except the DDR configuration ones.
  - The source code path is `core/arch/arm/dts/stm32mp15*`.

The figure below shows the device tree creation flow for Linux, U-Boot, TF-A and OP-TEE:



Color legend:

- **Linux** in dark blue
- **U-Boot** add-ons in green
- **DDR configuration** for TF-A and U-Boot in pink
- **STM32CubeMX generated parts** in light blue (not used above but visible in **STM32CubeMX** section down in this article)
- The **hashed** areas on Linux device tree files represent the fact that these files are modified, so partly copied and updated

The above color codes are used for all the figures included in this article.

## 2.2 Upstreamed device tree

### 2.2.1 Overview

As explained in the STM32 MPU Embedded Software Open Source Software (OSS) philosophy, STMicroelectronics maximizes the usage of open source software and the upstreaming to the communities.



So, the device tree files supporting the evaluation board and the discovery kit are **upstreamed** to their official respective repositories. But waiting for the upstream process completion, STMicroelectronics latest contributions are available in advance on github in Linux , U-Boot , TF-A and OP-TEE repositories.

**i Information**

The figures shown below are reflecting the **current implementation on github** and they may be somehow in advance compare to the **upstreamed** version for all parts that are still under review.

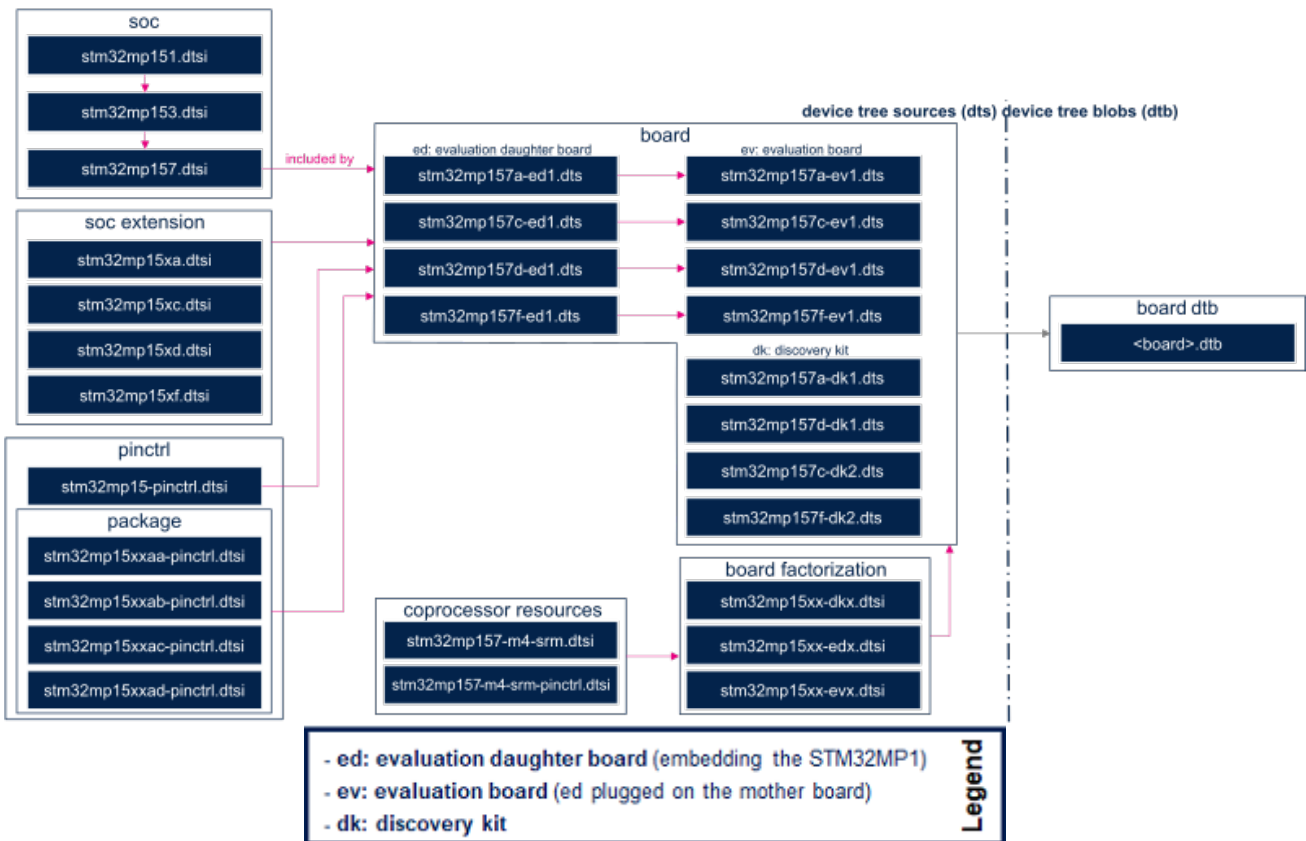
**i Information**

For the sake of simplicity, the figures below only show the **dtsi & dts files** structure, without showing the **h files** that are also included.

**i Information**

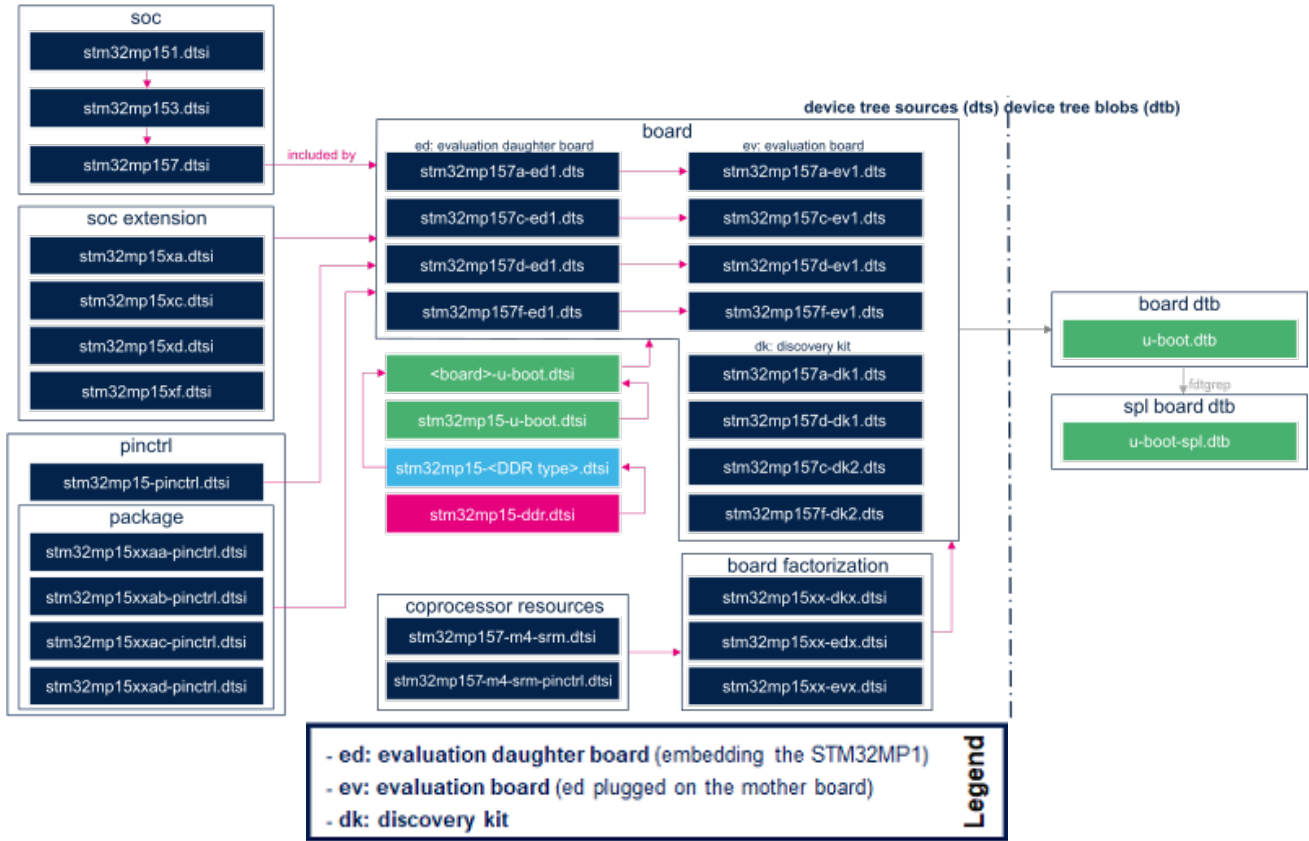
The **dtb files**, resulting from the compilation with the dtc compiler, may be named differently when the compilation is launched with the **Distribution Package**.

2.2.2 Linux





### 2.2.3 U-Boot

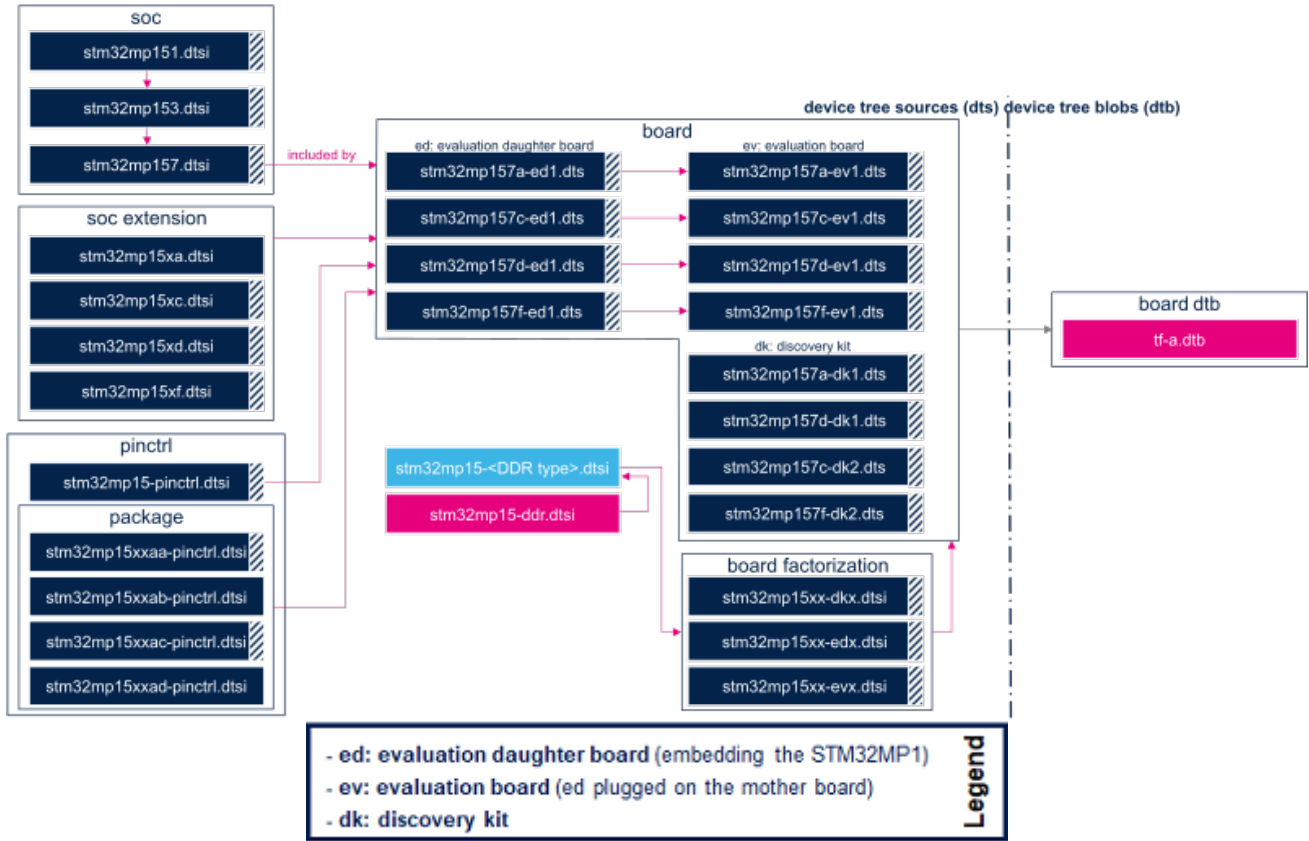


#### Information

'<board>-u-boot.dtsi' file inclusion at the end of '<board>.dts' file is not explicit (#include ...) but realized via an U-Boot generic mechanism (see [README](#) )



2.2.4 TF-A

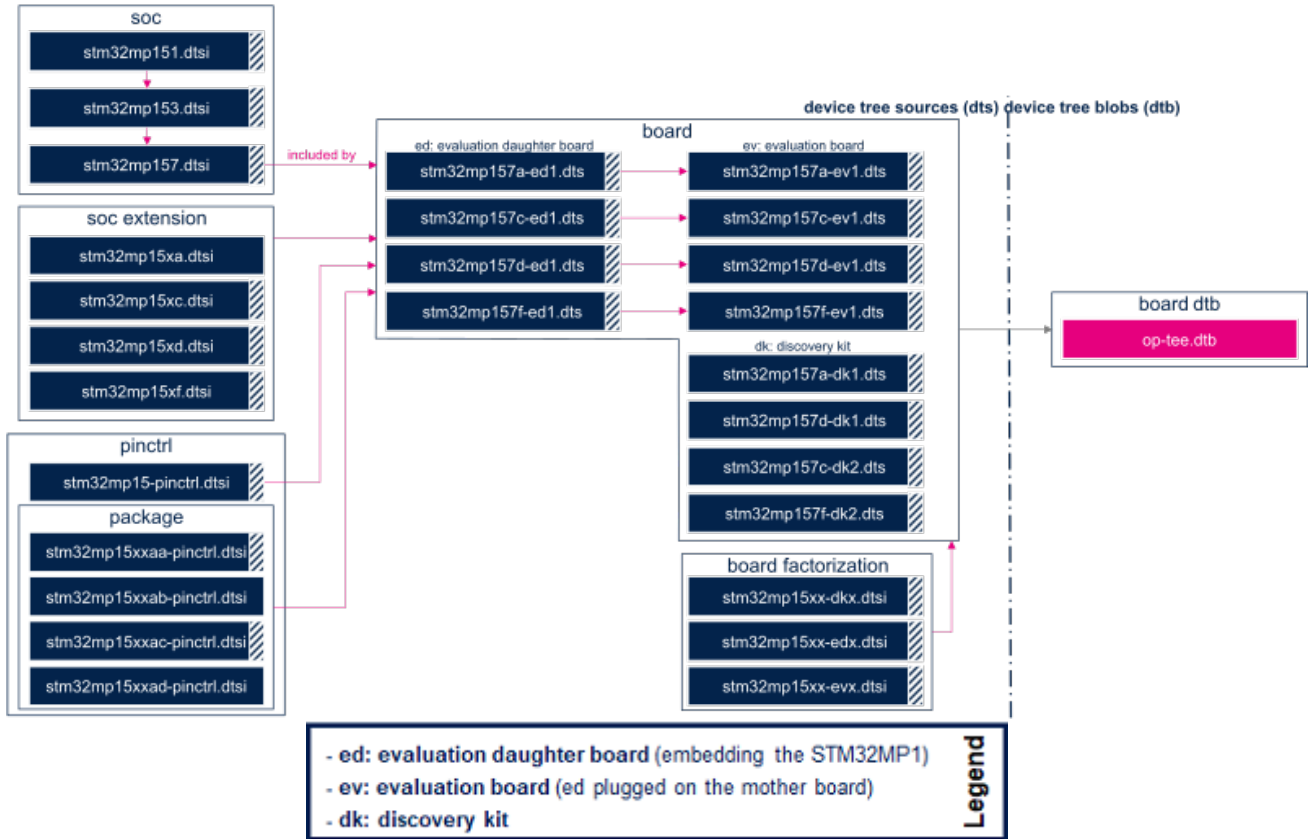


Note: The hashed areas on Linux device tree files represent the fact that these files are modified in TF-A.





## 2.2.5 OP-TEE



Note: The hashed areas on Linux device tree files represent the fact that these files are modified in OP-TEE.

## 2.3 STM32CubeMX generated device tree

### 2.3.1 Overview

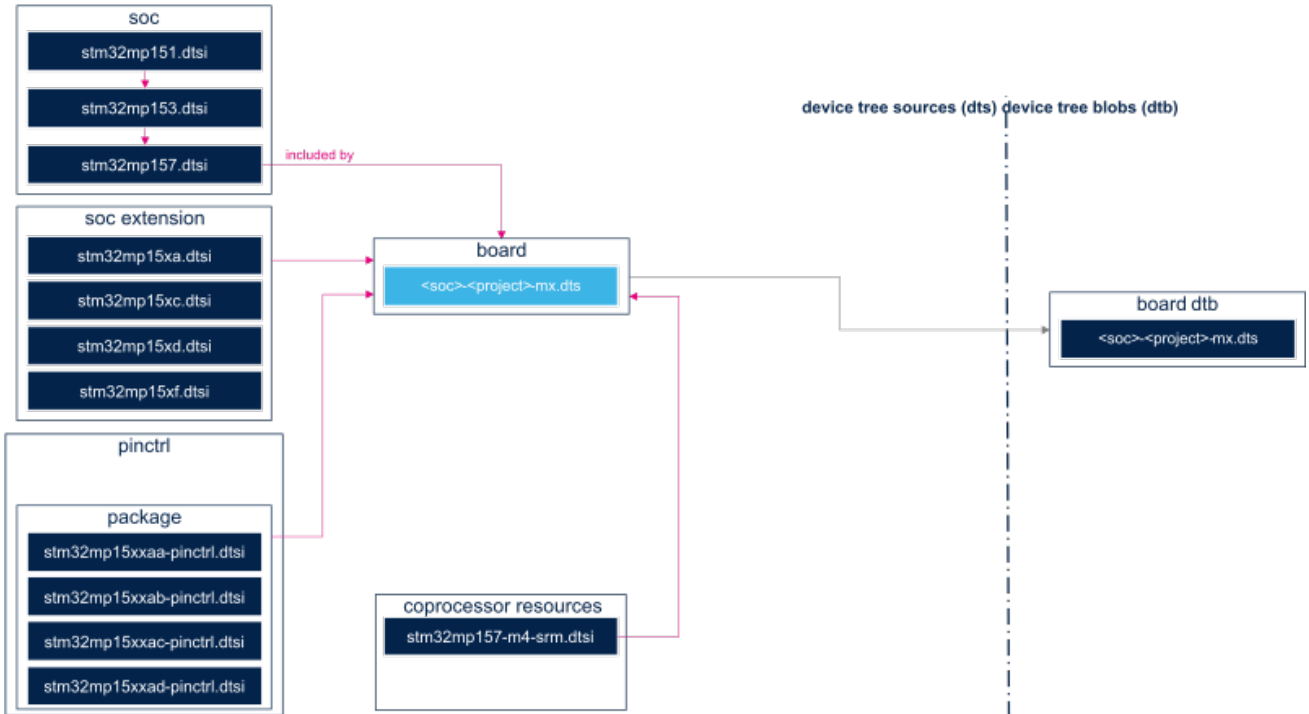
STM32CubeMX can be used to **generate** the device tree files for a given project. The tool generates the files **highlighted in light blue** in the following diagrams, including the upstreamed files shown in the previous chapter.

The device tree files generated by STM32CubeMX can be **built** with the Developer Package or Distribution Package:

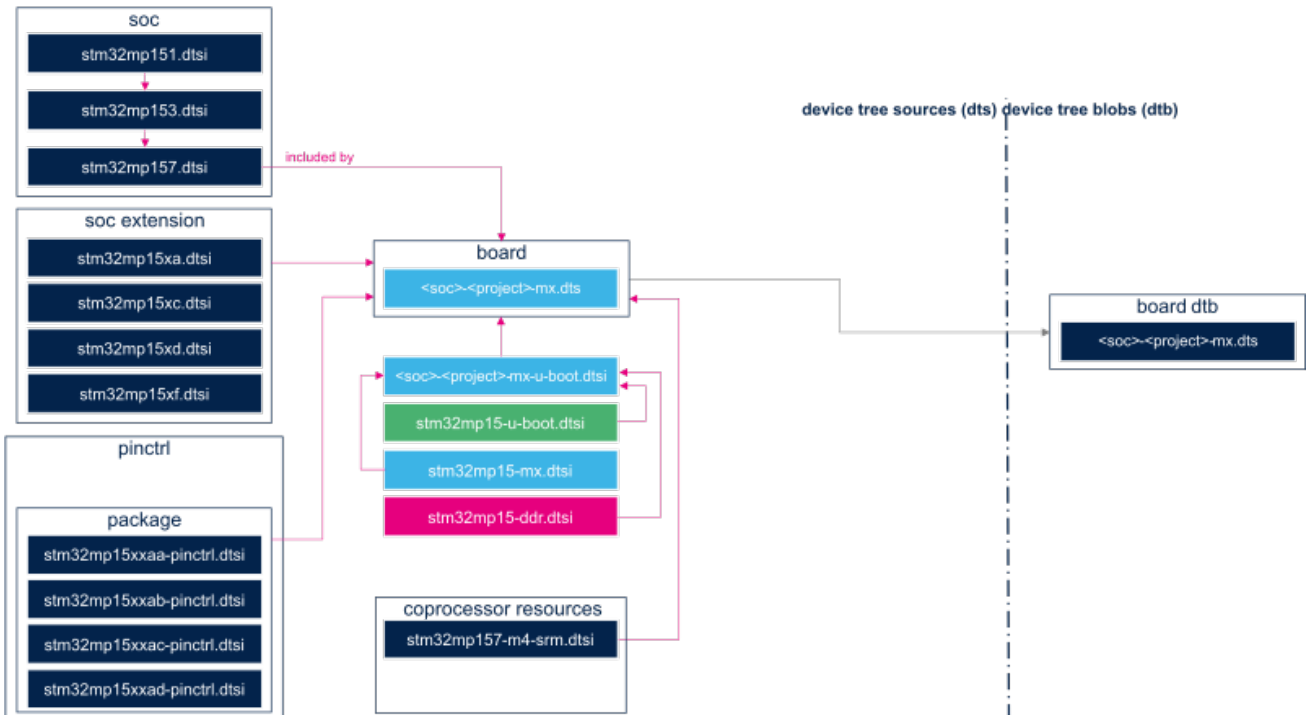
- Use the **Developer Package** to store the STM32CubeMX-generated files in the folder of your choice. Then, manually copy or link them into Linux, U-Boot and TF-A device tree folders (see introduction). Do not forget to add your device tree file name in `../Makefile.sdk` (for TF-A SDK and U-Boot SDK) and in `arch/arm/boot/dts/Makefile` (for Linux kernel SDK), before compiling them as explained in How to cross-compile with the Developer Package.
- Use the **Distribution Package** to store the STM32CubeMX-generated files in `meta-st/meta-st-stm32mp-addons/mx`, as explained in the How to create your own machine article. This article also describes how to build the complete image. Examples of Linux (device tree), U-Boot and TF-A individual compiling can be found in the How to cross-compile with the Distribution Package article.



### 2.3.2 Linux

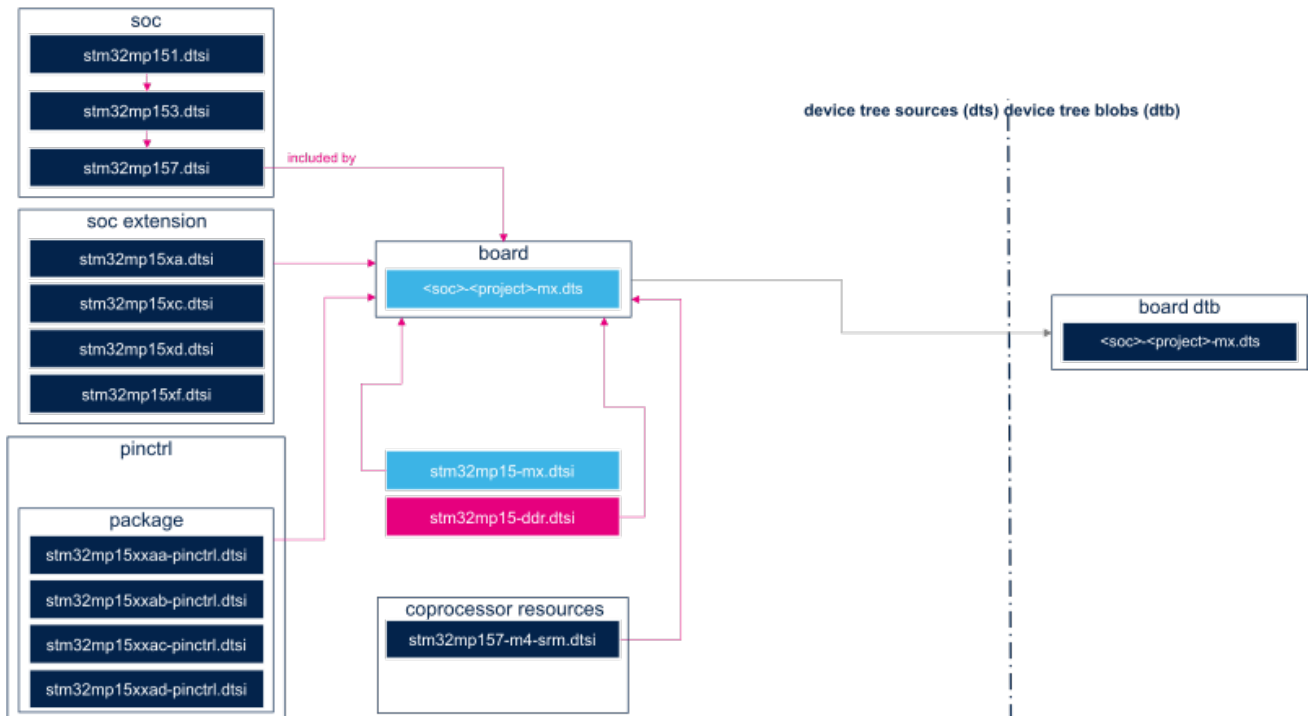


### 2.3.3 U-Boot





### 2.3.4 TF-A



### 2.3.5 OP-TEE

STM32CubeMX does not generate any device tree source files dedicated to OP-TEE, but these can be derived from the device tree source files generated for TF-A. In [OP-TEE github release](#), OP-TEE files are the very TF-A source files with the DDR dtsi files removed since fully unused. However, one can copy TF-A device tree source files `stm32mp15*` to `OP-TEE dts/` source directory and build OP-TEE with.

Linux<sup>®</sup> is a registered trademark of Linus Torvalds.

Das U-Boot -- the Universal Boot Loader (see [U-Boot\\_overview](#))

Trusted Firmware for Arm Cortex-A

Open Portable Trusted Execution Environment

First Stage Boot Loader

Operating System

Second Stage Boot Loader

Doubledata rate (memory domain)

Microprocessor Unit