



# STM32MP15 Flash mapping for Android - v1.1.0



## Contents

---

1. STM32MP15 Flash mapping for Android - v1.1.0 .....	3
2. STM32MP15 Flash mapping for Android .....	5
3. STM32CubeProgrammer .....	5

# STM32MP15 Flash mapping for Android - v1.1.0

Stable: 15.04.2020 - 15:46 / Revision: 15.04.2020 - 15:45

This is a sub-article, please refer to the main one here after: [STM32MP15 Flash mapping for Android](#)

## 1 Flash partitions

The table below lists the partitions defined for STMP32MP15 boards.

Size	Partition	Comment
Remaining area	USERDATA	The USERDATA file system contains user data which are mainly filled during the first boot (including property files).
512KB	MISC	The MISC raw partition is a simple structure containing all data which can be shared between Android and the bootloader.
2GB	SUPER	The logical PRODUCT partition contains all applications and services which depends on the kind of product created.
		The logical SYSTEM partition contains all libraries, executables and configuration files used by the Android frameworks.
		The logical VENDOR partition contains all hardware dependent services
512KB	DT	The DT raw partition contains and Android dt image which includes the device tree blob associated to the boards (selected automatically during boot phase)
16MB	BOOT	The BOOT raw partition contains an Android boot image which includes the kernel image, the ramdisk and the kernel command line (within the header)
512KB	SPLASH	The SPLASH contains the splash screen image (bitmap)
3 x 512KB	TEE	The TEE raw partition contains the OP-TEE OS (separated in three partitions to insure capability to manage paging)
1MB	SSBL	The SSBL (Second Stage Boot Loader) raw partition contains the U-Boot image
512KB	FSBL	The FSBL (First Stage Boot Loader) raw partition contains the ARM Trusted Firmware (TF-A) image

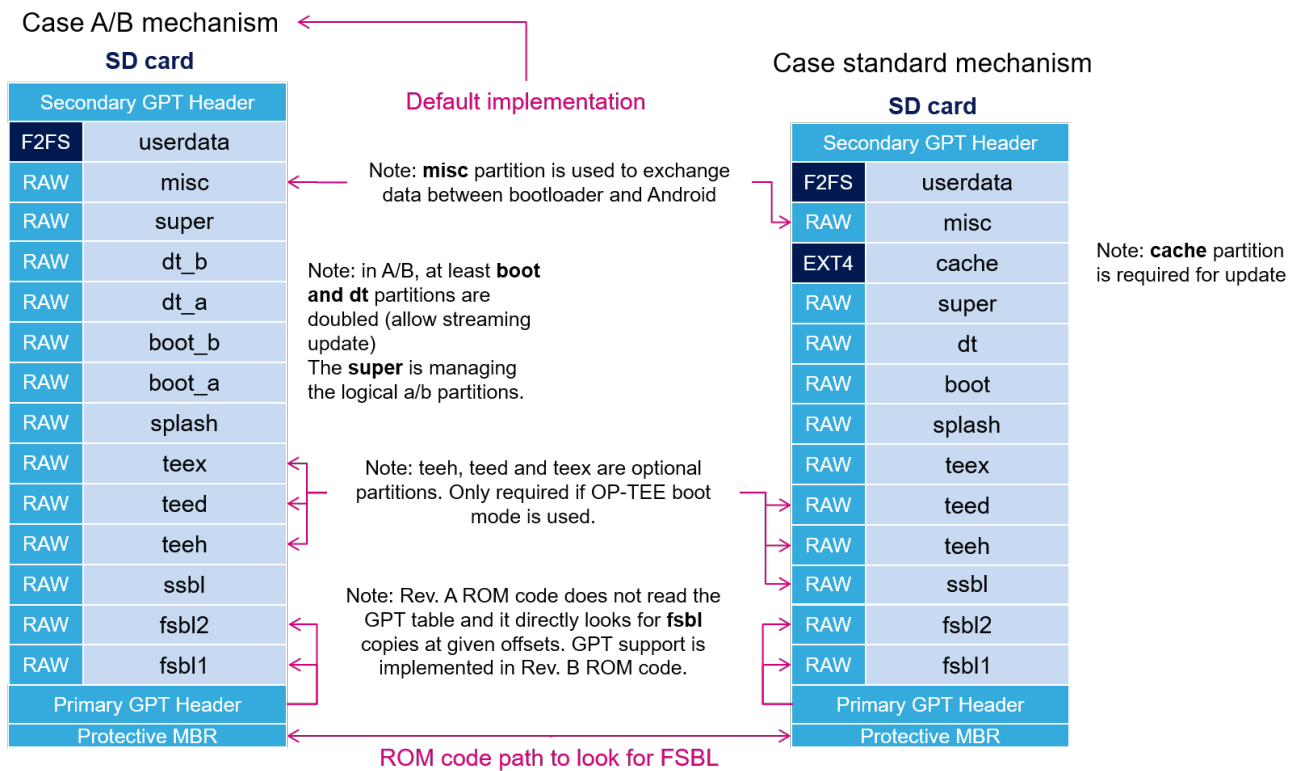
## 2 SD card memory mapping

The SD card has to be partitioned with GPT format to be recognized by STM32MP15 microcontrollers. The easiest way to achieve this is to use [STM32CubeProgrammer](#).

The ROM code searches the GPT entries which name begins with "fsbl", such as fsbl1 and fsbl2.



The SD card can be unplugged from the board and inserted into a Linux<sup>®</sup> host computer for direct partitioning with Linux utilities.

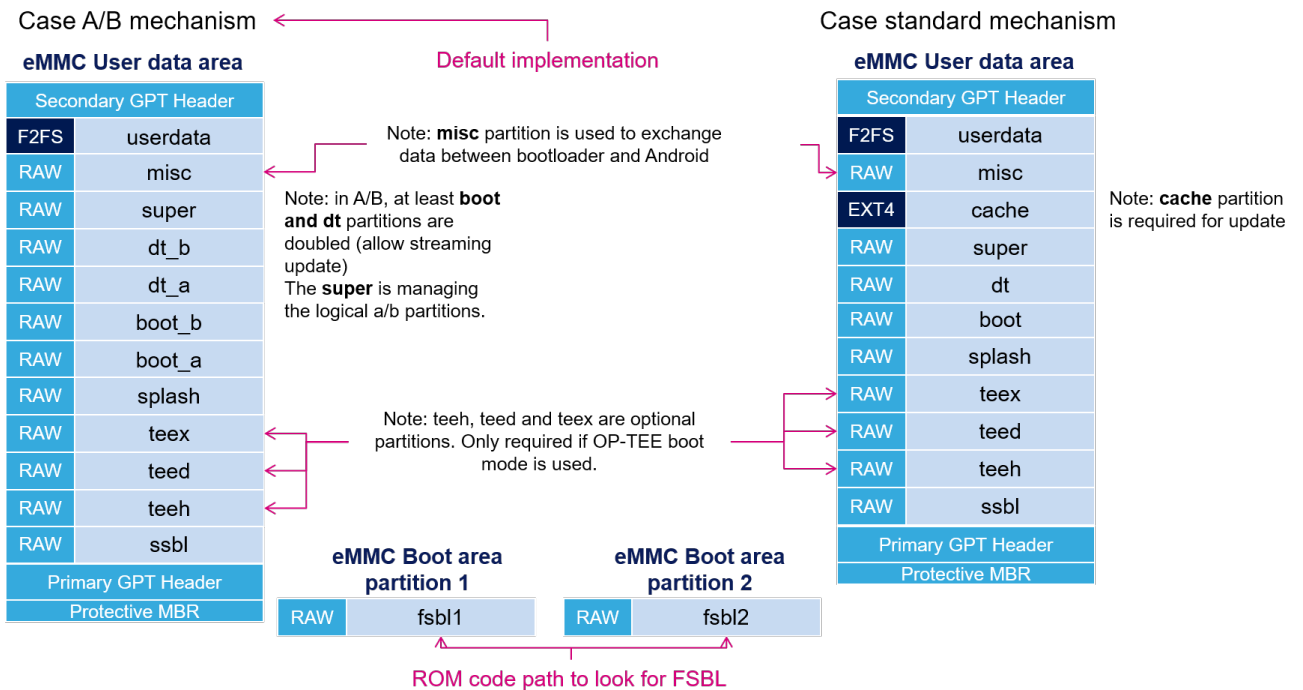


### 3 eMMC memory mapping

The eMMC embeds four physical partitions:

- Boot area partition 1: it is used to store the first copy of the FSBL.
- Boot area partition 2: it is used to store the second copy of the FSBL.
- User data area: it is formatted with GPT partitioning and used to store all remaining partitions.
- Replay protected memory block (RPMB): this partition is not shown in the figure below since it is not involved in the current boot chain.

STM32CubeProgrammer has to be used to prepare the eMMC with the layout shown below and to populate each partition.



SD memory card (<https://www.sdcard.org>)

GUID Partition Table

Read Only Memory

former spelling for eMMC ('e' in italic)

First Stage Boot Loader

## Permission error

Stable: 23.06.2020 - 13:56 / Revision: 12.06.2020 - 06:29

You do not have permission to read this page, for the following reason:

The action "Read pages" for the draft version of this page is only available for the groups ST\_editors, ST\_readers, Selected\_editors, sysop, reviewer

## Permission error

Stable: 27.07.2020 - 14:05 / Revision: 27.07.2020 - 14:05

You do not have permission to read this page, for the following reason:

The action "Read pages" for the draft version of this page is only available for the groups ST\_editors, ST\_readers, Selected\_editors, sysop, reviewer