STM32MP15 Flash mapping

Contents

STM32MP15 Flash mapping

Stable: 27.01.2020 - 10:40 / Revision: 23.01.2020 - 11:40

Contents

1 Supported Flash memory technologies

STM32MP15 boards support the following types of Flash memory:

- SD card on the SDMMC interface present on EVAL and DISCO boards
- *e*•MMC on the SDMMC interface present on EVAL board only
- Serial NOR Flash memory on the Dual QSPI interface present on EVAL board only
- NAND Flash memory on the FMC interface present on EVAL board only.

The next section lists all partitions used on STM32MP15 boards (size, name, and content), and the following sections show how they are mapped on the different types of Flash memory.

2 Flash partitions

The tables below list the partitions defined for STMP32MP15 boards.

2.1 Minimal

Size	Component	Comment
Remaining area	userfs	The user file system contains user data and examples
768 Mbytes	rootfs	Linux root file system contains all user space binaries (executable, libraries, and so on), and kernel modules

Size	Component	Comment
16 Mbytes	vendorfs	This partition is preferred to the rootfs for third-party proprietary binaries, and ensures that they are not contaminated by any open source licence, such as GPL v3
64 Mbytes	bootfs	The boot file system contains: • (option) the init RAM file system, which can be copied to the external RAM and used by Linux before mounting a fatter rootfs • Linux kernel device tree (can be in a Flattened Image Tree - FIT) • Linux kernel U-Boot image (can be in a Flattened Image Tree - FIT) • For all flashes except for NOR: the boot loader splash screen image, displayed by U-Boot • U-Boot distro config file *extlinux.conf* (can be in a Flattened Image Tree – FIT)
2 Mbytes	ssbl	The second stage boot loader (SSBL) is U-Boot, with its device tree blob (dtb) appended at the end
256 Kbytes to 512 Kbytes (*)	fsbl	The first stage boot loader is Arm Trusted Firmware (TF-A) or U-Boot Secondary Program Loader (SPL), with its device tree blob (dtb) appended at the end. At least two copies are embedded. Note: due to ROM code RAM needs, the FSBL payload is limited to 247 Kbytes.

(*): The partition size depends on the Flash technology, to be aligned to the block erase size of the Flash memory present on the board: NOR (256 Kbytes) / NAND (512 Kbytes).

Some boards can be equipped with multiple Flash devices, like the EVAL board, where all of the Flash devices can be programmed with STM32CubeProgrammer. However, caution must be taken for the serial NOR/NAND and SLC NAND because a **static bootable MTD partitioning** is defined in U-Boot include/configs/stm32mp1.h (look for *STM32MP_MTDPARTS*), with the consequence that up to 6 Mbytes of space is lost at the beginning of each such device, **even those which are not bootable**.

2.2 Optional

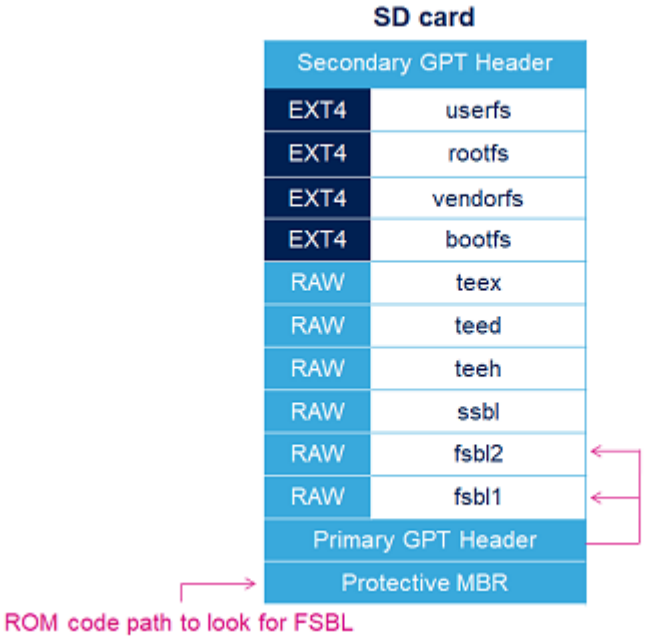
Size	Component	Comment
256 Kbytes (*)	env	This partition is used to store the U-Boot environment while booting from NOR Flash. For all other Flash devices, the U-Boot environment is stored either in an EXT4 bootfs partition (*e*•MMC, SD card), or UBI volumes (NAND).
256		

Size	Component	Comment
Kbytes to 512 Kbytes (*)	teeh	OP-TEE header
256 Kbytes to 512 Kbytes (*)	teed	OP-TEE pageable code and data
256 Kbytes to 512 Kbytes (*)	teex	OP-TEE pager

(*): The partition size depends on the Flash technology, as it should be aligned to the block erase size of the Flash device present on the board (256 Kbytes for NOR and 512 Kbytes for NAND).

3 SD card memory mapping

The SD card has to be partitioned with GPT format in order to be recognized by the STM32MP15. The easiest way to achieve this is to use STM32CubeProgrammer.
The ROM code looks for the GPT entries whose name begins with "fsbl": fsbl1 and fsbl2 for example.
Note: The SD card can be unplugged from the board and inserted into a Linux host computer for direct partitioning with Linux utilities and access to the **bootfs**, **rootfs** and **userfs** partitions. The file system is Linux EXT4.



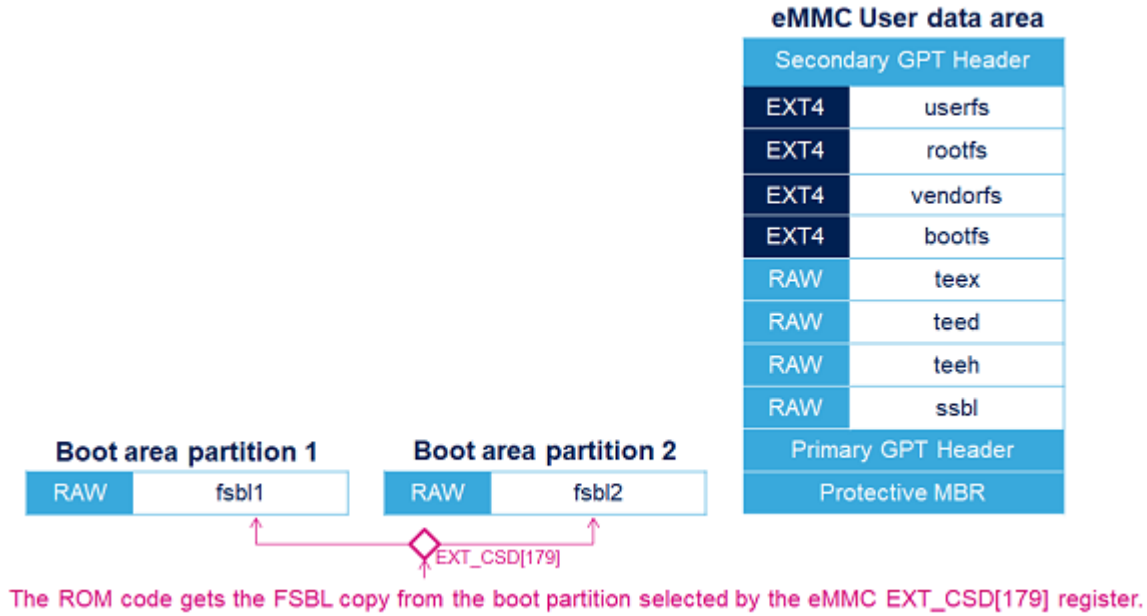
4 *e*•MMC memory mapping

The *e*•MMC embeds four physical partitions:

- Boot area partition 1: one copy of the FSBL
- Boot area partition 2: one copy of the FSBL
- User data area: formatted with GPT partitioning and used to store all remaining partitions
- Replay Protected Memory Block (RPMB): not shown in the figure below, since not involved in the current boot chain.

STM32CubeProgrammer has to be used to prepare the *e*•MMC with the layout shown below, and to populate each partition.

> The boot area partition used by the *e*•MMC boot sequence is selected via the EXT_CSD[179] register in the *e*•MMC. The STM32CubeProgrammer execution is concluded with the selection of the last written partition from the flashlayout file, typically partition 2. The other copy is never used as long as the user does not explicitly change the *e*•MMC EXT_CSD[179] register to select it.

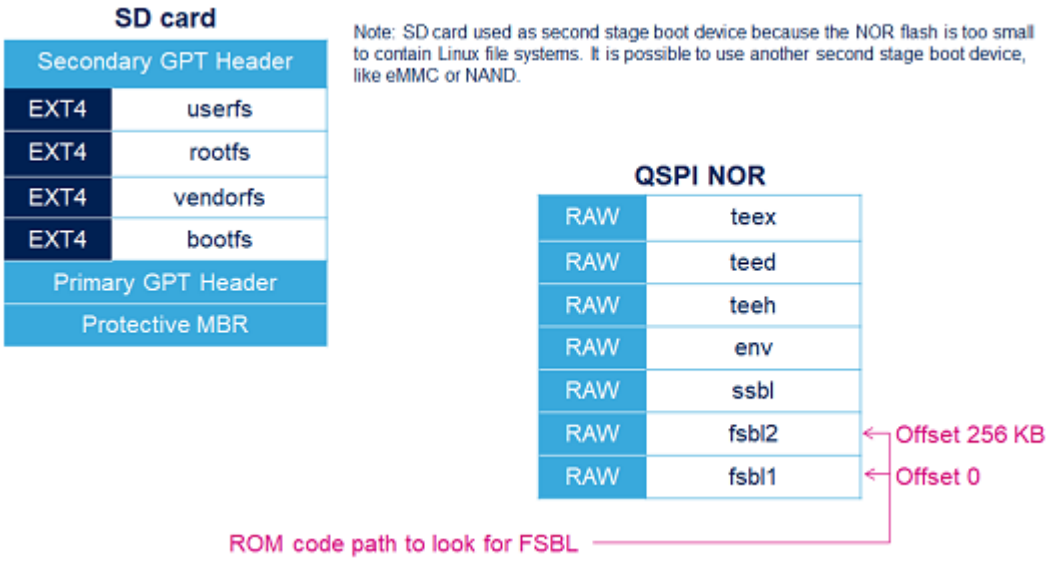


5 NOR memory mapping

As NOR Flash memory is expensive, its size is usually limited to the minimum needed to store only the bootloaders. The system files (bootfs, rootfs and userfs) are usually stored in another Flash memory, such as the SD card in OpenSTLinux distribution.
STM32CubeProgrammer must be used to prepare the NOR Flash and the SD card with the layout shown below, and to populate each partition.

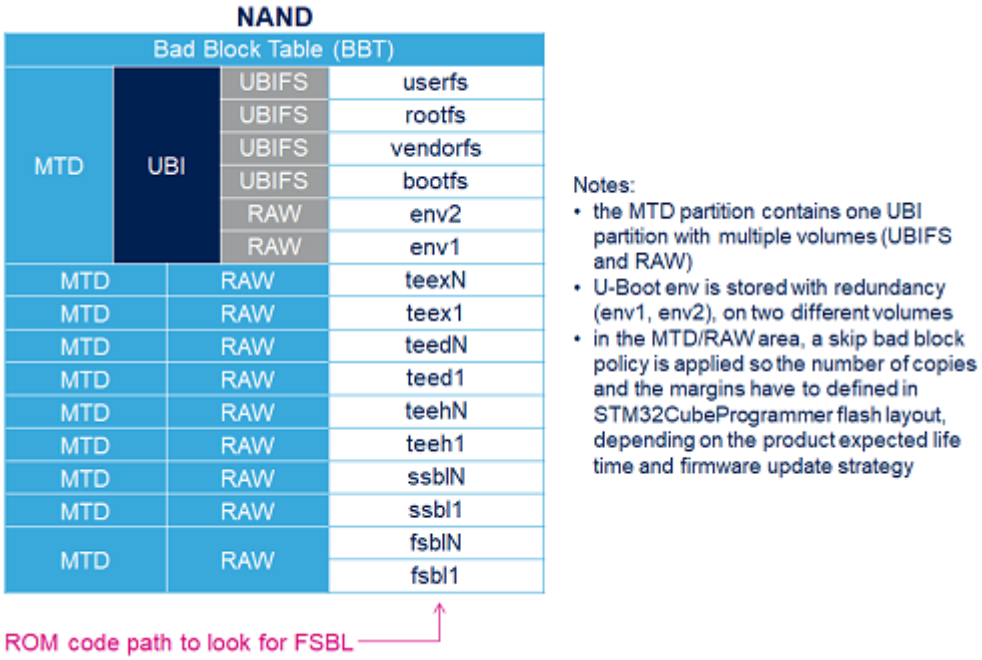
It is possible to use an *e*•MMC card or NAND as second-level Flash memory, rather than an SD card. This requires the following aspects to be changed:

- The Flash memory layout, using STM32CubeProgrammer in order to write the rootfs and userfs to the targeted Flash memory
- The Linux kernel parameters, using U-Boot, in order to indicate where the rootfs and userfs have to be mounted.



6 NAND memory mapping

STM32CubeProgrammer has to be used to prepare the NAND Flash memory with the layout shown below, and to populate each partition.



Flash memories combine high density and cost effectiveness of EPROMs with the electrical erasability of EEPROMs. For this reason, the Flash memory market is one of the most exciting areas of the semiconductor industry today and new applications requiring in system reprogramming, such as cellular telephones, automotive engine management systems, hard disk drives, PC BIOS software for Plug & Play, digital TV, set top boxes, fax and other modems, PC cards and multimedia CD-ROMs, offer the prospect of very high volume demand.

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

MultimediaCard

Random Access Memory (Early computer memories generally hadserial access. Memories where any given address can be accessed when desired were then called "random access" to distinguish them from the memories where contents can only be accessed in a fixed order. The term is used today forvolatile random-accesssemiconductor memories.)

Das U-Boot -- the Universal Boot Loader (see U-Boot_overview)

Second Stage Boot Loader

Trusted Firmware for Arm Cortex-A

Secondary Program Loader, *Also known as* ***U-Boot SPL***

Read Only Memory

First Stage Boot Loader

Single-Level Cell is a kind of NAND flash

Memory Technology Device

Flash memory shortened to gain space in titles, tables and block diagrams

Open Portable Trusted Execution Environment

GUID Partition Table

Permission error

Stable: 17.06.2020 - 15:26 / Revision: 16.01.2020 - 09:30

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: 21.02.2020 - 10:10 / Revision: 20.02.2020 - 10:16

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: 25.05.2020 - 07:32 / Revision: 25.05.2020 - 07:25

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