



SDMMC internal peripheral



SDMMC internal peripheral

Stable: 14.05.2020 - 07:13 / Revision: 14.05.2020 - 07:12

Contents

1 Article purpose	2
2 Peripheral overview	2
2.1 Features	2
2.2 Security support	3
3 Peripheral usage and associated software	3
3.1 Boot time	3
3.2 Runtime	3
3.2.1 Overview	3
3.2.2 Software frameworks	4
3.2.3 Peripheral configuration	4
3.2.4 Peripheral assignment	4
4 How to go further	5
5 References	5

1 Article purpose

The purpose of this article is to

- briefly introduce the SDMMC peripheral and its main features
- indicate the level of security supported by this hardware block
- explain how each instance can be allocated to the three runtime contexts and linked to the corresponding software components
- explain, when necessary, how to configure the SDMMC peripheral.

2 Peripheral overview

The **SDMMC** peripheral is used to interconnect STM32 MPU to SD memory cards, SDIO and MMC devices.

2.1 Features

Refer to the [STM32MP15 reference manuals](#) for the complete list of features, and to the software components, introduced below, to see which features are implemented.

2.2 Security support

SDMMC1/2/3 instances are either **non-secure** or **secure** peripherals (under ETZPC control).



- When an SDMMC instance is **secure internal**, the DMA cannot be used to perform data transfers.
- STMicroelectronics does not provide secure MMC driver (see below chapter)

3 Peripheral usage and associated software

3.1 Boot time

SDMMC1/2 instances can be used to support memory boot on SD or MMC Flash devices.

The SDMMC3 is not used at boot time.

The SDMMC instances are ordered by address in the [device tree arch/arm/boot/dts/stm32mp151.dtsi](#) file:

```
sdmmc3: sdmmc@48004000 {
...
sdmmc1: sdmmc@58005000 {
...
sdmmc2: sdmmc@58007000 {
```



By default, in [OpenSTLinux](#) distribution, **sdmmc3 is disabled** so the sdmmc1 (SD card on Evaluation boards and Discovery kits) and sdmmc2 (eMMC on Evaluation boards and Wifi on Discovery kits) are respectively aliased to mmc0 and mmc1.

If you enable sdmmc3, it will take the mmc0 alias and the aliases above will shift, so don't forget to update the Linux kernel boot command accordingly!

For instance, 'root=/dev/mmcblk0p6' will become 'root=/dev/mmcblk1p6' to mount the rootfs from the sdmmc1 (SD card) when sdmmc3 is enabled.

3.2 Runtime

3.2.1 Overview

SDMMC1/2/3 instances can be allocated to:

- the Arm[®] Cortex[®]-A7 non-secure core to be controlled in Linux[®] by the MMC framework

or

- the Arm[®] Cortex[®]-M4 to be controlled in STM32Cube MPU Package by STM32Cube SDMMC driver

Chapter [#Peripheral assignment](#) describes which peripheral instance can be assigned to which context.

3.2.2 Software frameworks

Do	Peri	Software frameworks		Comment
main Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)		
Mass storage	SDMMC	Linux MMC framework	STM32Cube SDMMC driver	

3.2.3 Peripheral configuration

The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the [STM32CubeMX](#) tool for all internal peripherals, and then manually completed (particularly for external peripherals), according to the information given in the corresponding software framework article.

For Linux[®] kernel configuration, please refer to [SDMMC device tree configuration](#).

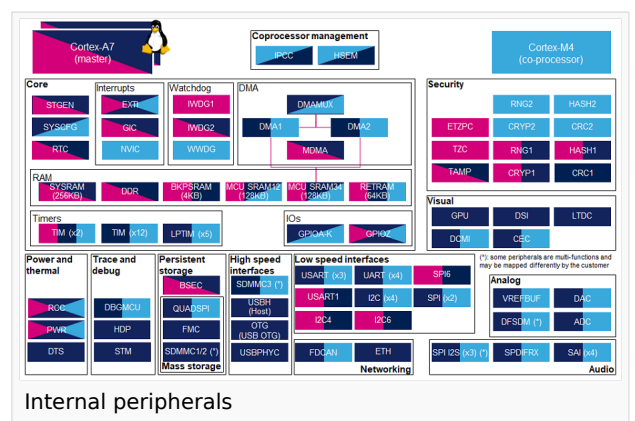
3.2.4 Peripheral assignment

Check boxes illustrate the possible peripheral allocations supported by STM32 MPU Embedded Software:

- means that the peripheral can be assigned () to the given runtime context.
- is used for system peripherals that cannot be unchecked because they are statically connected in the device.

Refer to [How to assign an internal peripheral to a runtime context](#) for more information on how to assign peripherals manually or via [STM32CubeMX](#).

The present chapter describes STMicroelectronics recommendations or choice of implementation. Additional possibilities might be described in [STM32MP15 reference manuals](#).



Do	Peri	Runtime allocation		Comment
main	insecure			



Do ma in st a nc e	Per in h era A 7 se cu re (O P T E E)	Runtime allocation				Comme nt
		Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
M a s s s t o r a g e	S D M M C	SDMMC1				
		SDMMC2				
		SDMMC3				Assign ment (singl e choic e)

4 How to go further

5 References

Microprocessor Unit

MultimediaCard

Direct Memory Access

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

former spelling for eMMC ('e' in italic)

Open Portable Trusted Execution Environment