



Internal peripherals assignment table template



Contents

1. Internal peripherals assignment table template	3
2. How to assign an internal peripheral to a runtime context	3
3. STM32CubeMX	4
4. STM32MP15 resources	4
5. STM32MPU Embedded Software architecture overview	5



A quality version of this page, approved on 15 February 2019, was based off this revision.

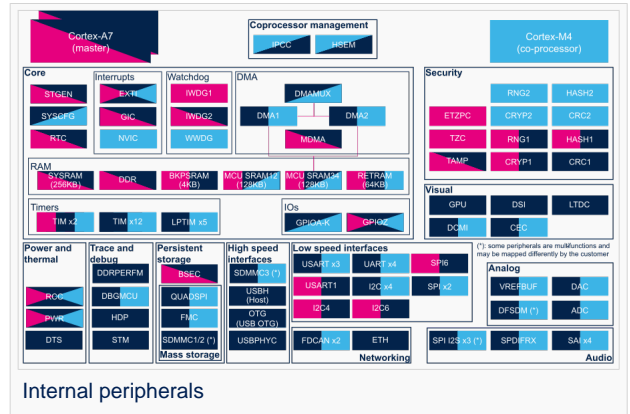
Template:ArticleMainWriter Template:ArticleApprovedVersion

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 main	Peripheral	Runtime allocation			Comment
		Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	

Cortex®

Open Portable Trusted Execution Environment

Linux® is a registered trademark of Linus Torvalds.

Stable: 08.03.2021 - 16:13 / Revision: 16.02.2021 - 17:11

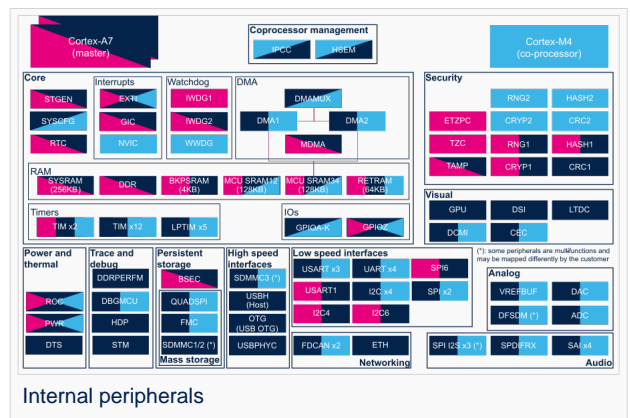
Template:ArticleMainWriter Template:ArticleApprovedVersion

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	Per	Runtime allocation			



main	peripheral	Runtime allocation			Comment
		Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	

Cortex®

Open Portable Trusted Execution Environment

Linux® is a registered trademark of Linus Torvalds.

Stable: 23.09.2020 - 13:22 / Revision: 12.06.2020 - 13:25

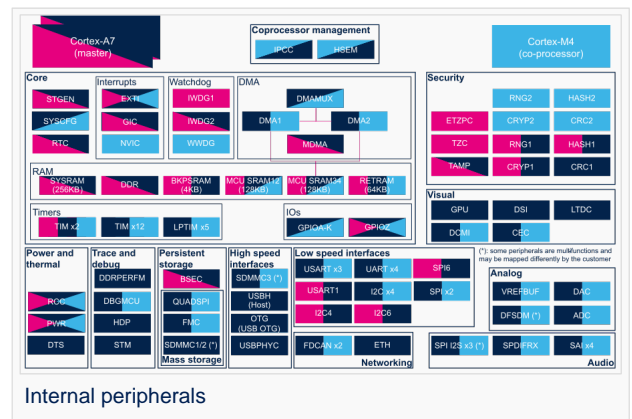
Template:ArticleMainWriter Template:ArticleApprovedVersion

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	Per	Runtime allocation			Comment
		Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	

Cortex®

Open Portable Trusted Execution Environment

Linux® is a registered trademark of Linus Torvalds.

Stable: 17.11.2021 - 16:41 / Revision: 17.11.2021 - 10:47

Template:ArticleMainWriter Template:ArticleApprovedVersion

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

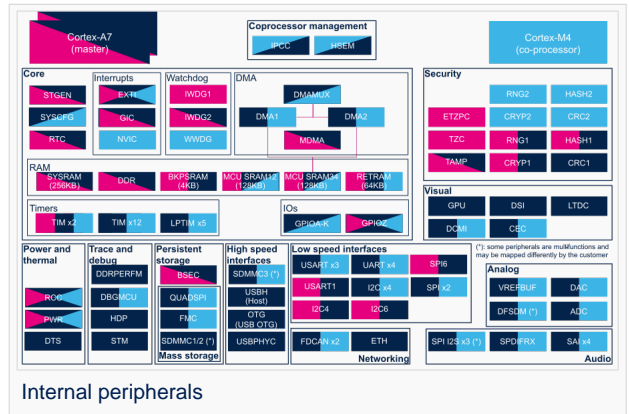
- means that the peripheral can be assigned () to the given runtime context.
- is used for system peripherals that cannot be unchecked because they are statically connected in the device.

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.



Internal peripherals assignment table template



Do main	Peripheral	Runtime allocation			Comment
		Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	

Cortex®

Open Portable Trusted Execution Environment

Linux® is a registered trademark of Linus Torvalds.

Stable: 26.03.2021 - 11:32 / Revision: 12.03.2021 - 11:07

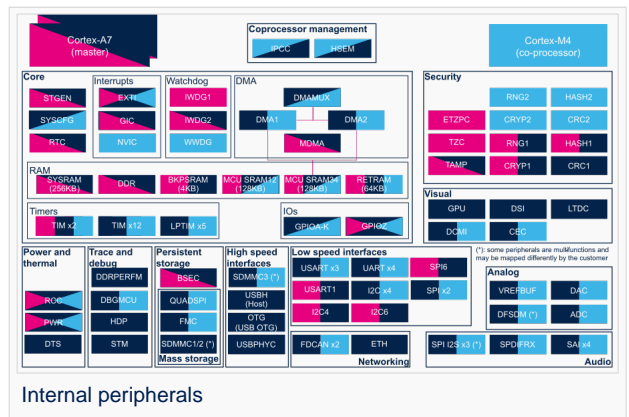
Template:ArticleMainWriter Template:ArticleApprovedVersion

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 main	Peripheral	Runtime allocation			Comment
		Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	

Cortex®

Open Portable Trusted Execution Environment



Linux® is a registered trademark of Linus Torvalds.