



RETRAM internal memory



Contents

1. RETRAM internal memory	3
2. ETZPC internal peripheral	6
3. How to assign an internal peripheral to a runtime context	10
4. Linux remoteproc framework overview	14
5. MCU SRAM internal memory	18
6. NVIC internal peripheral	22
7. OP-TEE overview	26
8. Power overview	30
9. Reserved memory	34
10. STM32CubeMP1 architecture	38
11. STM32CubeMX	42
12. STM32MP15 RAM mapping	46
13. STM32MP15 ROM code overview	50
14. STM32MP15 resources	54
15. STM32MPU Embedded Software architecture overview	58



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

Template:ArticleMainWriter Template:ArticleApprovedVersion

Contents

1 Peripheral overview	4
1.1 Features	4
1.2 Security support	4
2 Peripheral usage and associated software	5
2.1 Boot time	5
2.2 Runtime	5
2.2.1 Overview	5
2.2.2 Software frameworks	5
2.2.3 Peripheral configuration	5
2.2.4 Peripheral assignment	5



1 Peripheral overview

The **RETRAM** internal memory is 64 Kbytes wide and is physically near to the Arm[®]Cortex[®]-M4 for optimized performance from the core. It is located in the VSW power domain, allowing it to be supplied during Standby **low power mode**, and to retain retention firmware that can be executed very quickly by the Cortex-M4 on wake up from Standby mode.

1.1 Features

Refer to *STM32MP15 reference manuals* for the complete feature list, and to the software components introduced below to see which features are actually implemented.

1.2 Security support

The RETRAM is a **secure** peripheral (under ETZPC control).



2 Peripheral usage and associated software

2.1 Boot time

Linux[®] remoteproc framework (running on the Cortex-A7) loads the Cortex-M4 firmware to the RETRAM, starting at address 0x00000000. At least, it must load the part of the firmware containing the vector table, since the Cortex-M4 reset entry point is address 0x00000004. The rest of the firmware code is loaded into the MCU SRAM. The overall memory mapping is shown in the platform memory mapping section.

2.2 Runtime

2.2.1 Overview

The Cortex-M4 vector table is mapped from address 0x00000000 (so to the RETRAM) at reset, but it can be remapped by software to any other location by means of the vector table offset register (VTOR). Beyond the reset entry point (0x00000004), the exception table also contains the software entries table used by the NVIC to branch the software execution to the right interrupt service routine.

While going to Standby low power mode, the RETRAM can remain supplied, so it can preserve a (small) Cortex-M4 piece of retention firmware that is executed on wake up when the ROM code (running on Cortex-A7) restarts the Cortex-M4. All these constraints make the RETRAM the minimum (and default) choice for Cortex-M4 firmware.

RETRAM can be allocated to:

- the Cortex-A7 secure to be used under OP-TEE.

or

- the Cortex-A7 non-secure to be used under Linux as reserved memory.

or

- the Cortex-M4 for use with the STM32Cube MPU Package, either for **runtime firmware** that can be mapped in both RETRAM and MCU SRAM, or for **retention firmware** that only fits into the RETRAM, but could have some data in MCU SRAM (keeping in mind that these data are lost while entering Standby low power mode).

2.2.2 Software frameworks

Domain	Peripheral	Software components			Comment
OP-TEE	Linux	STM32Cube			
Core/RAM	RETRAM	OP-TEE overview	Linux reserved memory	STM32Cube	

2.2.3 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 (especially for external peripherals), according to the information given in the corresponding software framework article.

2.2.4 Peripheral assignment

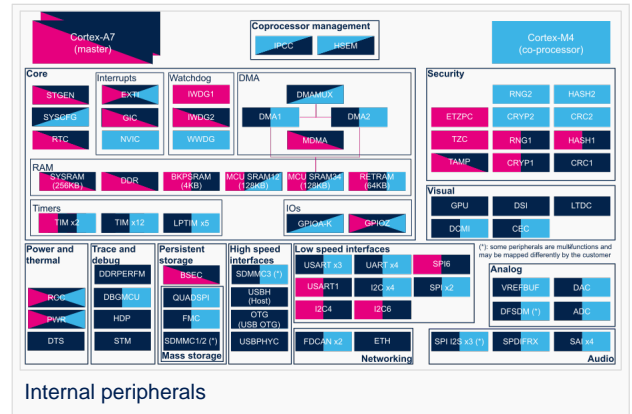


Check boxes illustrate the possible peripheral allocations supported by STM32 MPU Embedded Software:

- 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



Domain	Peripheral	Runtime allocation		Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)	
Core/RAM	RETRAM	RETRAM		Assignment (single choice)

Arm® is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



Cortex®

Linux® is a registered trademark of Linus Torvalds.

Microprocessor Unit

Open Portable Trusted Execution Environment

Random Access Memory (Early computer memories generally had serial 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 for volatile random-access semiconductor memories.)

Stable: 31.07.2020 - 14:57 / Revision: 31.07.2020 - 14:56

Template:ArticleMainWriter Template:ArticleApprovedVersion

Contents

1 Peripheral overview	8
1.1 Features	8
1.2 Security support	8
2 Peripheral usage and associated software	9
2.1 Boot time	9
2.2 Runtime	9
2.2.1 Overview	9



2.2.2 Software frameworks	9
2.2.3 Peripheral configuration	9
2.2.4 Peripheral assignment	9



1 Peripheral overview

The **RETRAM** internal memory is 64 Kbytes wide and is physically near to the Arm®Cortex®-M4 for optimized performance from the core. It is located in the VSW power domain, allowing it to be supplied during Standby **low power mode**, and to retain retention firmware that can be executed very quickly by the Cortex-M4 on wake up from Standby mode.

1.1 Features

Refer to *STM32MP15 reference manuals* for the complete feature list, and to the software components introduced below to see which features are actually implemented.

1.2 Security support

The RETRAM is a **secure** peripheral (under ETZPC control).



2 Peripheral usage and associated software

2.1 Boot time

Linux[®] remoteproc framework (running on the Cortex-A7) loads the Cortex-M4 firmware to the RETRAM, starting at address 0x00000000. At least, it must load the part of the firmware containing the vector table, since the Cortex-M4 reset entry point is address 0x00000004. The rest of the firmware code is loaded into the MCU SRAM. The overall memory mapping is shown in the platform memory mapping section.

2.2 Runtime

2.2.1 Overview

The Cortex-M4 vector table is mapped from address 0x00000000 (so to the RETRAM) at reset, but it can be remapped by software to any other location by means of the vector table offset register (VTOR). Beyond the reset entry point (0x00000004), the exception table also contains the software entries table used by the NVIC to branch the software execution to the right interrupt service routine.

While going to Standby low power mode, the RETRAM can remain supplied, so it can preserve a (small) Cortex-M4 piece of retention firmware that is executed on wake up when the ROM code (running on Cortex-A7) restarts the Cortex-M4. All these constraints make the RETRAM the minimum (and default) choice for Cortex-M4 firmware.

RETRAM can be allocated to:

- the Cortex-A7 secure to be used under OP-TEE.

or

- the Cortex-A7 non-secure to be used under Linux as reserved memory.

or

- the Cortex-M4 for use with the STM32Cube MPU Package, either for **runtime firmware** that can be mapped in both RETRAM and MCU SRAM, or for **retention firmware** that only fits into the RETRAM, but could have some data in MCU SRAM (keeping in mind that these data are lost while entering Standby low power mode).

2.2.2 Software frameworks

Domain	Peripheral	Software components			Comment
OP-TEE	Linux	STM32Cube			
Core/RAM	RETRAM	OP-TEE overview	Linux reserved memory	STM32Cube	

2.2.3 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 (especially for external peripherals), according to the information given in the corresponding software framework article.

2.2.4 Peripheral assignment

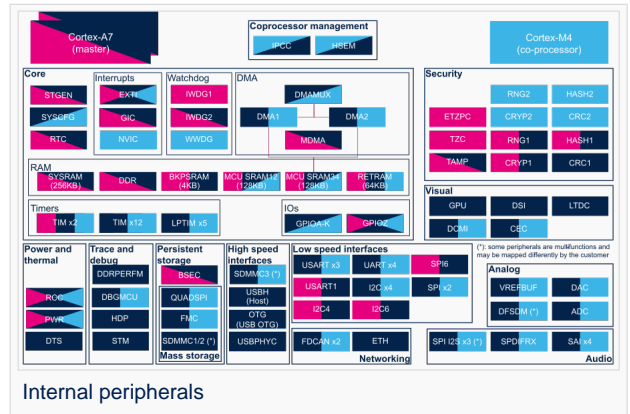


Check boxes illustrate the possible peripheral allocations supported by STM32 MPU Embedded Software:

- 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



Domain	Peripheral	Runtime allocation		Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)	
Core/RAM	RETRAM	RETRAM		Assignment (single choice)

Arm® is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



Cortex®

Linux® is a registered trademark of Linus Torvalds.

Microprocessor Unit

Open Portable Trusted Execution Environment

Random Access Memory (Early computer memories generally had serial 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 for volatile random-access semiconductor memories.)

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

Template:ArticleMainWriter Template:ArticleApprovedVersion

Contents

1 Peripheral overview	12
1.1 Features	12
1.2 Security support	12
2 Peripheral usage and associated software	13
2.1 Boot time	13
2.2 Runtime	13
2.2.1 Overview	13



2.2.2 Software frameworks	13
2.2.3 Peripheral configuration	13
2.2.4 Peripheral assignment	13



1 Peripheral overview

The **RETRAM** internal memory is 64 Kbytes wide and is physically near to the Arm[®]Cortex[®]-M4 for optimized performance from the core. It is located in the VSW power domain, allowing it to be supplied during Standby **low power mode**, and to retain retention firmware that can be executed very quickly by the Cortex-M4 on wake up from Standby mode.

1.1 Features

Refer to *STM32MP15 reference manuals* for the complete feature list, and to the software components introduced below to see which features are actually implemented.

1.2 Security support

The RETRAM is a **secure** peripheral (under ETZPC control).



2 Peripheral usage and associated software

2.1 Boot time

Linux[®] remoteproc framework (running on the Cortex-A7) loads the Cortex-M4 firmware to the RETRAM, starting at address 0x00000000. At least, it must load the part of the firmware containing the vector table, since the Cortex-M4 reset entry point is address 0x00000004. The rest of the firmware code is loaded into the MCU SRAM. The overall memory mapping is shown in the platform memory mapping section.

2.2 Runtime

2.2.1 Overview

The Cortex-M4 vector table is mapped from address 0x00000000 (so to the RETRAM) at reset, but it can be remapped by software to any other location by means of the vector table offset register (VTOR). Beyond the reset entry point (0x00000004), the exception table also contains the software entries table used by the NVIC to branch the software execution to the right interrupt service routine.

While going to Standby low power mode, the RETRAM can remain supplied, so it can preserve a (small) Cortex-M4 piece of retention firmware that is executed on wake up when the ROM code (running on Cortex-A7) restarts the Cortex-M4. All these constraints make the RETRAM the minimum (and default) choice for Cortex-M4 firmware.

RETRAM can be allocated to:

- the Cortex-A7 secure to be used under OP-TEE.

or

- the Cortex-A7 non-secure to be used under Linux as reserved memory.

or

- the Cortex-M4 for use with the STM32Cube MPU Package, either for **runtime firmware** that can be mapped in both RETRAM and MCU SRAM, or for **retention firmware** that only fits into the RETRAM, but could have some data in MCU SRAM (keeping in mind that these data are lost while entering Standby low power mode).

2.2.2 Software frameworks

Domain	Peripheral	Software components			Comment
OP-TEE	Linux	STM32Cube			
Core/RAM	RETRAM	OP-TEE overview	Linux reserved memory	STM32Cube	

2.2.3 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 (especially for external peripherals), according to the information given in the corresponding software framework article.

2.2.4 Peripheral assignment

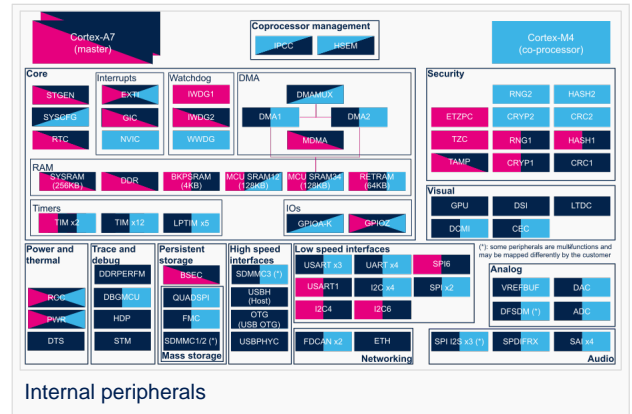


Check boxes illustrate the possible peripheral allocations supported by STM32 MPU Embedded Software:

- 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



Domain	Peripheral	Runtime allocation		Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)	
Core/RAM	RETRAM	RETRAM		Assignment (single choice)

Arm® is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



Cortex®

Linux® is a registered trademark of Linus Torvalds.

Microprocessor Unit

Open Portable Trusted Execution Environment

Random Access Memory (Early computer memories generally had serial 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 for volatile random-access semiconductor memories.)

Stable: 08.07.2021 - 08:19 / Revision: 08.07.2021 - 08:18

Template:ArticleMainWriter Template:ArticleApprovedVersion

Contents

1 Peripheral overview	16
1.1 Features	16
1.2 Security support	16
2 Peripheral usage and associated software	17
2.1 Boot time	17
2.2 Runtime	17
2.2.1 Overview	17



2.2.2 Software frameworks	17
2.2.3 Peripheral configuration	17
2.2.4 Peripheral assignment	17



1 Peripheral overview

The **RETRAM** internal memory is 64 Kbytes wide and is physically near to the Arm[®]Cortex[®]-M4 for optimized performance from the core. It is located in the VSW power domain, allowing it to be supplied during Standby **low power mode**, and to retain retention firmware that can be executed very quickly by the Cortex-M4 on wake up from Standby mode.

1.1 Features

Refer to *STM32MP15 reference manuals* for the complete feature list, and to the software components introduced below to see which features are actually implemented.

1.2 Security support

The RETRAM is a **secure** peripheral (under ETZPC control).



2 Peripheral usage and associated software

2.1 Boot time

Linux[®] remoteproc framework (running on the Cortex-A7) loads the Cortex-M4 firmware to the RETRAM, starting at address 0x00000000. At least, it must load the part of the firmware containing the vector table, since the Cortex-M4 reset entry point is address 0x00000004. The rest of the firmware code is loaded into the MCU SRAM. The overall memory mapping is shown in the platform memory mapping section.

2.2 Runtime

2.2.1 Overview

The Cortex-M4 vector table is mapped from address 0x00000000 (so to the RETRAM) at reset, but it can be remapped by software to any other location by means of the vector table offset register (VTOR). Beyond the reset entry point (0x00000004), the exception table also contains the software entries table used by the NVIC to branch the software execution to the right interrupt service routine.

While going to Standby low power mode, the RETRAM can remain supplied, so it can preserve a (small) Cortex-M4 piece of retention firmware that is executed on wake up when the ROM code (running on Cortex-A7) restarts the Cortex-M4. All these constraints make the RETRAM the minimum (and default) choice for Cortex-M4 firmware.

RETRAM can be allocated to:

- the Cortex-A7 secure to be used under OP-TEE.

or

- the Cortex-A7 non-secure to be used under Linux as reserved memory.

or

- the Cortex-M4 for use with the STM32Cube MPU Package, either for **runtime firmware** that can be mapped in both RETRAM and MCU SRAM, or for **retention firmware** that only fits into the RETRAM, but could have some data in MCU SRAM (keeping in mind that these data are lost while entering Standby low power mode).

2.2.2 Software frameworks

Domain	Peripheral	Software components			Comment
OP-TEE	Linux	STM32Cube			
Core/RAM	RETRAM	OP-TEE overview	Linux reserved memory	STM32Cube	

2.2.3 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 (especially for external peripherals), according to the information given in the corresponding software framework article.

2.2.4 Peripheral assignment

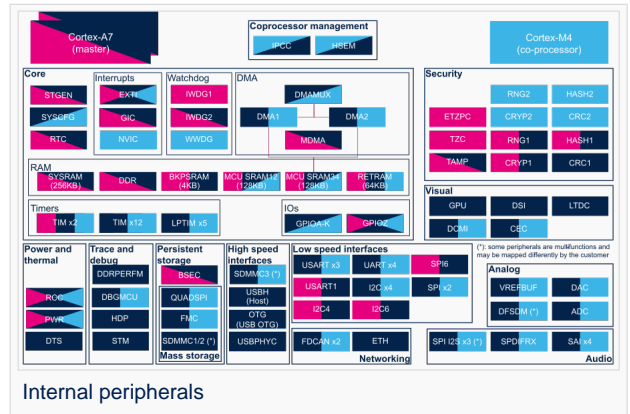


Check boxes illustrate the possible peripheral allocations supported by STM32 MPU Embedded Software:

- 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



Domain	Peripheral	Runtime allocation		Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)	
Core/RAM	RETRAM	RETRAM		Assignment (single choice)

Arm® is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



Cortex®

Linux® is a registered trademark of Linus Torvalds.

Microprocessor Unit

Open Portable Trusted Execution Environment

Random Access Memory (Early computer memories generally had serial 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 for volatile random-access semiconductor memories.)

Stable: 04.02.2020 - 15:59 / Revision: 04.02.2020 - 15:48

Template:ArticleMainWriter Template:ArticleApprovedVersion

Contents

1 Peripheral overview	20
1.1 Features	20
1.2 Security support	20
2 Peripheral usage and associated software	21
2.1 Boot time	21
2.2 Runtime	21
2.2.1 Overview	21



2.2.2 Software frameworks	21
2.2.3 Peripheral configuration	21
2.2.4 Peripheral assignment	21



1 Peripheral overview

The **RETRAM** internal memory is 64 Kbytes wide and is physically near to the Arm[®]Cortex[®]-M4 for optimized performance from the core. It is located in the VSW power domain, allowing it to be supplied during Standby **low power mode**, and to retain retention firmware that can be executed very quickly by the Cortex-M4 on wake up from Standby mode.

1.1 Features

Refer to *STM32MP15 reference manuals* for the complete feature list, and to the software components introduced below to see which features are actually implemented.

1.2 Security support

The RETRAM is a **secure** peripheral (under ETZPC control).



2 Peripheral usage and associated software

2.1 Boot time

Linux[®] remoteproc framework (running on the Cortex-A7) loads the Cortex-M4 firmware to the RETRAM, starting at address 0x00000000. At least, it must load the part of the firmware containing the vector table, since the Cortex-M4 reset entry point is address 0x00000004. The rest of the firmware code is loaded into the MCU SRAM. The overall memory mapping is shown in the platform memory mapping section.

2.2 Runtime

2.2.1 Overview

The Cortex-M4 vector table is mapped from address 0x00000000 (so to the RETRAM) at reset, but it can be remapped by software to any other location by means of the vector table offset register (VTOR). Beyond the reset entry point (0x00000004), the exception table also contains the software entries table used by the NVIC to branch the software execution to the right interrupt service routine.

While going to Standby low power mode, the RETRAM can remain supplied, so it can preserve a (small) Cortex-M4 piece of retention firmware that is executed on wake up when the ROM code (running on Cortex-A7) restarts the Cortex-M4. All these constraints make the RETRAM the minimum (and default) choice for Cortex-M4 firmware.

RETRAM can be allocated to:

- the Cortex-A7 secure to be used under OP-TEE.

or

- the Cortex-A7 non-secure to be used under Linux as reserved memory.

or

- the Cortex-M4 for use with the STM32Cube MPU Package, either for **runtime firmware** that can be mapped in both RETRAM and MCU SRAM, or for **retention firmware** that only fits into the RETRAM, but could have some data in MCU SRAM (keeping in mind that these data are lost while entering Standby low power mode).

2.2.2 Software frameworks

Domain	Peripheral	Software components			Comment
OP-TEE	Linux	STM32Cube			
Core/RAM	RETRAM	OP-TEE overview	Linux reserved memory	STM32Cube	

2.2.3 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 (especially for external peripherals), according to the information given in the corresponding software framework article.

2.2.4 Peripheral assignment

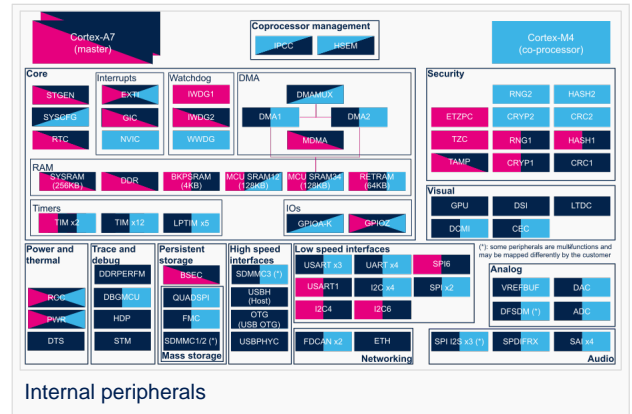


Check boxes illustrate the possible peripheral allocations supported by STM32 MPU Embedded Software:

- 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





2.2.2 Software frameworks	25
2.2.3 Peripheral configuration	25
2.2.4 Peripheral assignment	25



1 Peripheral overview

The **RETRAM** internal memory is 64 Kbytes wide and is physically near to the Arm[®]Cortex[®]-M4 for optimized performance from the core. It is located in the VSW power domain, allowing it to be supplied during Standby **low power mode**, and to retain retention firmware that can be executed very quickly by the Cortex-M4 on wake up from Standby mode.

1.1 Features

Refer to *STM32MP15 reference manuals* for the complete feature list, and to the software components introduced below to see which features are actually implemented.

1.2 Security support

The RETRAM is a **secure** peripheral (under ETZPC control).



2 Peripheral usage and associated software

2.1 Boot time

Linux[®] remoteproc framework (running on the Cortex-A7) loads the Cortex-M4 firmware to the RETRAM, starting at address 0x00000000. At least, it must load the part of the firmware containing the vector table, since the Cortex-M4 reset entry point is address 0x00000004. The rest of the firmware code is loaded into the MCU SRAM. The overall memory mapping is shown in the platform memory mapping section.

2.2 Runtime

2.2.1 Overview

The Cortex-M4 vector table is mapped from address 0x00000000 (so to the RETRAM) at reset, but it can be remapped by software to any other location by means of the vector table offset register (VTOR). Beyond the reset entry point (0x00000004), the exception table also contains the software entries table used by the NVIC to branch the software execution to the right interrupt service routine.

While going to Standby low power mode, the RETRAM can remain supplied, so it can preserve a (small) Cortex-M4 piece of retention firmware that is executed on wake up when the ROM code (running on Cortex-A7) restarts the Cortex-M4. All these constraints make the RETRAM the minimum (and default) choice for Cortex-M4 firmware.

RETRAM can be allocated to:

- the Cortex-A7 secure to be used under OP-TEE.

or

- the Cortex-A7 non-secure to be used under Linux as reserved memory.

or

- the Cortex-M4 for use with the STM32Cube MPU Package, either for **runtime firmware** that can be mapped in both RETRAM and MCU SRAM, or for **retention firmware** that only fits into the RETRAM, but could have some data in MCU SRAM (keeping in mind that these data are lost while entering Standby low power mode).

2.2.2 Software frameworks

Domain	Peripheral	Software components			Comment
OP-TEE	Linux	STM32Cube			
Core/RAM	RETRAM	OP-TEE overview	Linux reserved memory	STM32Cube	

2.2.3 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 (especially for external peripherals), according to the information given in the corresponding software framework article.

2.2.4 Peripheral assignment

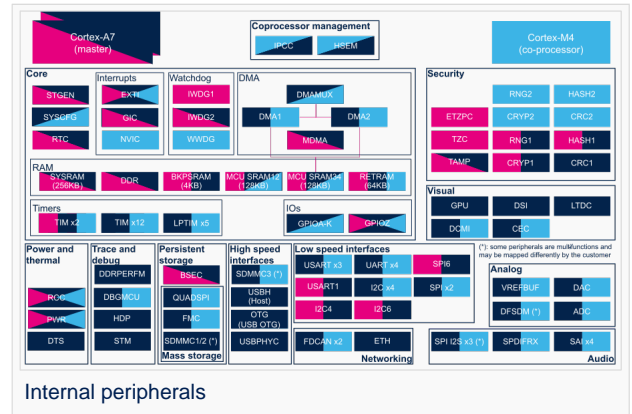


Check boxes illustrate the possible peripheral allocations supported by STM32 MPU Embedded Software:

- 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



Domain	Peripheral	Runtime allocation		Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)	
Core/RAM	RETRAM	RETRAM		Assignment (single choice)

Arm® is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



Cortex®

Linux® is a registered trademark of Linus Torvalds.

Microprocessor Unit

Open Portable Trusted Execution Environment

Random Access Memory (Early computer memories generally had serial 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 for volatile random-access semiconductor memories.)

Stable: 13.05.2020 - 08:56 / Revision: 13.05.2020 - 08:54

Template:ArticleMainWriter Template:ArticleApprovedVersion

Contents

1 Peripheral overview	28
1.1 Features	28
1.2 Security support	28
2 Peripheral usage and associated software	29
2.1 Boot time	29
2.2 Runtime	29
2.2.1 Overview	29



2.2.2 Software frameworks	29
2.2.3 Peripheral configuration	29
2.2.4 Peripheral assignment	29



1 Peripheral overview

The **RETRAM** internal memory is 64 Kbytes wide and is physically near to the Arm[®]Cortex[®]-M4 for optimized performance from the core. It is located in the VSW power domain, allowing it to be supplied during Standby **low power mode**, and to retain retention firmware that can be executed very quickly by the Cortex-M4 on wake up from Standby mode.

1.1 Features

Refer to *STM32MP15 reference manuals* for the complete feature list, and to the software components introduced below to see which features are actually implemented.

1.2 Security support

The RETRAM is a **secure** peripheral (under ETZPC control).



2 Peripheral usage and associated software

2.1 Boot time

Linux[®] remoteproc framework (running on the Cortex-A7) loads the Cortex-M4 firmware to the RETRAM, starting at address 0x00000000. At least, it must load the part of the firmware containing the vector table, since the Cortex-M4 reset entry point is address 0x00000004. The rest of the firmware code is loaded into the MCU SRAM. The overall memory mapping is shown in the platform memory mapping section.

2.2 Runtime

2.2.1 Overview

The Cortex-M4 vector table is mapped from address 0x00000000 (so to the RETRAM) at reset, but it can be remapped by software to any other location by means of the vector table offset register (VTOR). Beyond the reset entry point (0x00000004), the exception table also contains the software entries table used by the NVIC to branch the software execution to the right interrupt service routine.

While going to Standby low power mode, the RETRAM can remain supplied, so it can preserve a (small) Cortex-M4 piece of retention firmware that is executed on wake up when the ROM code (running on Cortex-A7) restarts the Cortex-M4. All these constraints make the RETRAM the minimum (and default) choice for Cortex-M4 firmware.

RETRAM can be allocated to:

- the Cortex-A7 secure to be used under OP-TEE.

or

- the Cortex-A7 non-secure to be used under Linux as reserved memory.

or

- the Cortex-M4 for use with the STM32Cube MPU Package, either for **runtime firmware** that can be mapped in both RETRAM and MCU SRAM, or for **retention firmware** that only fits into the RETRAM, but could have some data in MCU SRAM (keeping in mind that these data are lost while entering Standby low power mode).

2.2.2 Software frameworks

Domain	Peripheral	Software components			Comment
OP-TEE	Linux	STM32Cube			
Core/RAM	RETRAM	OP-TEE overview	Linux reserved memory	STM32Cube	

2.2.3 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 (especially for external peripherals), according to the information given in the corresponding software framework article.

2.2.4 Peripheral assignment

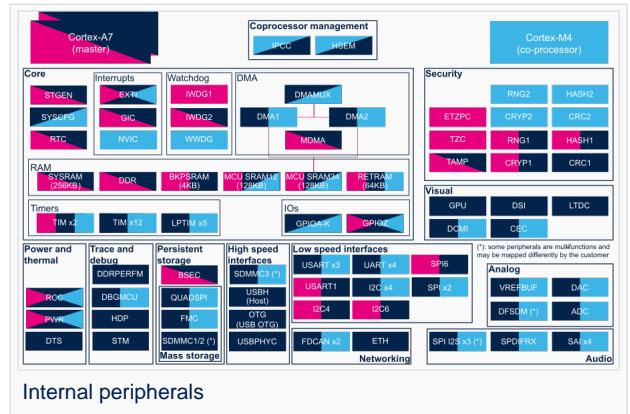


Check boxes illustrate the possible peripheral allocations supported by STM32 MPU Embedded Software:

- 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



Domain	Peripheral	Runtime allocation		Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)	
Core/RAM	RETRAM	RETRAM		Assignment (single choice)

Arm® is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



Cortex®

Linux® is a registered trademark of Linus Torvalds.

Microprocessor Unit

Open Portable Trusted Execution Environment

Random Access Memory (Early computer memories generally had serial 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 for volatile random-access semiconductor memories.)

Stable: 01.12.2020 - 10:35 / Revision: 01.12.2020 - 09:20

Template:ArticleMainWriter Template:ArticleApprovedVersion

Contents

1 Peripheral overview	32
1.1 Features	32
1.2 Security support	32
2 Peripheral usage and associated software	33
2.1 Boot time	33
2.2 Runtime	33
2.2.1 Overview	33



2.2.2 Software frameworks	33
2.2.3 Peripheral configuration	33
2.2.4 Peripheral assignment	33



1 Peripheral overview

The **RETRAM** internal memory is 64 Kbytes wide and is physically near to the Arm®Cortex®-M4 for optimized performance from the core. It is located in the VSW power domain, allowing it to be supplied during Standby *low power mode*, and to retain retention firmware that can be executed very quickly by the Cortex-M4 on wake up from Standby mode.

1.1 Features

Refer to *STM32MP15 reference manuals* for the complete feature list, and to the software components introduced below to see which features are actually implemented.

1.2 Security support

The RETRAM is a **secure** peripheral (under ETZPC control).



2 Peripheral usage and associated software

2.1 Boot time

Linux[®] remoteproc framework (running on the Cortex-A7) loads the Cortex-M4 firmware to the RETRAM, starting at address 0x00000000. At least, it must load the part of the firmware containing the vector table, since the Cortex-M4 reset entry point is address 0x00000004. The rest of the firmware code is loaded into the MCU SRAM. The overall memory mapping is shown in the platform memory mapping section.

2.2 Runtime

2.2.1 Overview

The Cortex-M4 vector table is mapped from address 0x00000000 (so to the RETRAM) at reset, but it can be remapped by software to any other location by means of the vector table offset register (VTOR). Beyond the reset entry point (0x00000004), the exception table also contains the software entries table used by the NVIC to branch the software execution to the right interrupt service routine.

While going to Standby low power mode, the RETRAM can remain supplied, so it can preserve a (small) Cortex-M4 piece of retention firmware that is executed on wake up when the ROM code (running on Cortex-A7) restarts the Cortex-M4. All these constraints make the RETRAM the minimum (and default) choice for Cortex-M4 firmware.

RETRAM can be allocated to:

- the Cortex-A7 secure to be used under OP-TEE.

or

- the Cortex-A7 non-secure to be used under Linux as reserved memory.

or

- the Cortex-M4 for use with the STM32Cube MPU Package, either for **runtime firmware** that can be mapped in both RETRAM and MCU SRAM, or for **retention firmware** that only fits into the RETRAM, but could have some data in MCU SRAM (keeping in mind that these data are lost while entering Standby low power mode).

2.2.2 Software frameworks

Domain	Peripheral	Software components			Comment
OP-TEE	Linux	STM32Cube			
Core/RAM	RETRAM	OP-TEE overview	Linux reserved memory	STM32Cube	

2.2.3 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 (especially for external peripherals), according to the information given in the corresponding software framework article.

2.2.4 Peripheral assignment

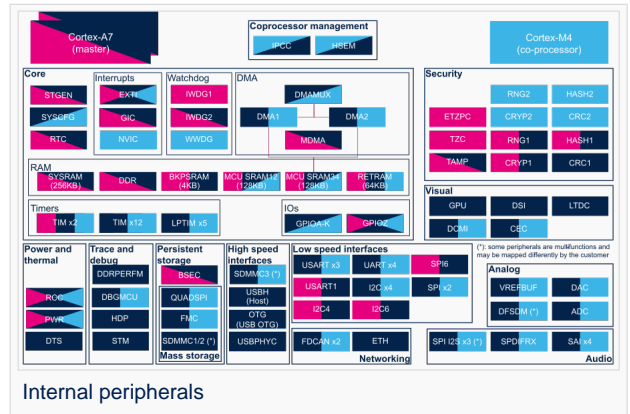


Check boxes illustrate the possible peripheral allocations supported by STM32 MPU Embedded Software:

- 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](#)



Domain	Peripheral	Runtime allocation		Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)	
Core/RAM	RETRAM	RETRAM		Assignment (single choice)

Arm® is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



Cortex®

Linux® is a registered trademark of Linus Torvalds.

Microprocessor Unit

Open Portable Trusted Execution Environment

Random Access Memory (Early computer memories generally had serial 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 for volatile random-access semiconductor memories.)

Stable: 25.03.2021 - 13:49 / Revision: 18.03.2021 - 14:55

Template:ArticleMainWriter Template:ArticleApprovedVersion

Contents

1 Peripheral overview	36
1.1 Features	36
1.2 Security support	36
2 Peripheral usage and associated software	37
2.1 Boot time	37
2.2 Runtime	37
2.2.1 Overview	37



2.2.2 Software frameworks	37
2.2.3 Peripheral configuration	37
2.2.4 Peripheral assignment	37



1 Peripheral overview

The **RETRAM** internal memory is 64 Kbytes wide and is physically near to the Arm[®]Cortex[®]-M4 for optimized performance from the core. It is located in the VSW power domain, allowing it to be supplied during Standby **low power mode**, and to retain retention firmware that can be executed very quickly by the Cortex-M4 on wake up from Standby mode.

1.1 Features

Refer to *STM32MP15 reference manuals* for the complete feature list, and to the software components introduced below to see which features are actually implemented.

1.2 Security support

The RETRAM is a **secure** peripheral (under ETZPC control).



2 Peripheral usage and associated software

2.1 Boot time

Linux[®] remoteproc framework (running on the Cortex-A7) loads the Cortex-M4 firmware to the RETRAM, starting at address 0x00000000. At least, it must load the part of the firmware containing the vector table, since the Cortex-M4 reset entry point is address 0x00000004. The rest of the firmware code is loaded into the MCU SRAM. The overall memory mapping is shown in the platform memory mapping section.

2.2 Runtime

2.2.1 Overview

The Cortex-M4 vector table is mapped from address 0x00000000 (so to the RETRAM) at reset, but it can be remapped by software to any other location by means of the vector table offset register (VTOR). Beyond the reset entry point (0x00000004), the exception table also contains the software entries table used by the NVIC to branch the software execution to the right interrupt service routine.

While going to Standby low power mode, the RETRAM can remain supplied, so it can preserve a (small) Cortex-M4 piece of retention firmware that is executed on wake up when the ROM code (running on Cortex-A7) restarts the Cortex-M4. All these constraints make the RETRAM the minimum (and default) choice for Cortex-M4 firmware.

RETRAM can be allocated to:

- the Cortex-A7 secure to be used under OP-TEE.

or

- the Cortex-A7 non-secure to be used under Linux as reserved memory.

or

- the Cortex-M4 for use with the STM32Cube MPU Package, either for **runtime firmware** that can be mapped in both RETRAM and MCU SRAM, or for **retention firmware** that only fits into the RETRAM, but could have some data in MCU SRAM (keeping in mind that these data are lost while entering Standby low power mode).

2.2.2 Software frameworks

Domain	Peripheral	Software components			Comment
OP-TEE	Linux	STM32Cube			
Core/RAM	RETRAM	OP-TEE overview	Linux reserved memory	STM32Cube	

2.2.3 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 (especially for external peripherals), according to the information given in the corresponding software framework article.

2.2.4 Peripheral assignment

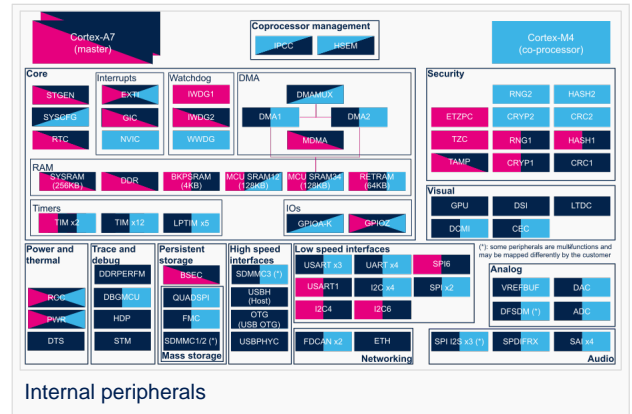


Check boxes illustrate the possible peripheral allocations supported by STM32 MPU Embedded Software:

- 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



Domain	Peripheral	Runtime allocation		Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)	
Core/RAM	RETRAM	RETRAM		Assignment (single choice)

Arm® is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



Cortex®

Linux® is a registered trademark of Linus Torvalds.

Microprocessor Unit

Open Portable Trusted Execution Environment

Random Access Memory (Early computer memories generally had serial 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 for volatile random-access semiconductor memories.)

Stable: 31.03.2021 - 11:58 / Revision: 23.03.2021 - 14:07

Template:ArticleMainWriter Template:ArticleApprovedVersion

Contents

1 Peripheral overview	40
1.1 Features	40
1.2 Security support	40
2 Peripheral usage and associated software	41
2.1 Boot time	41
2.2 Runtime	41
2.2.1 Overview	41



2.2.2 Software frameworks	41
2.2.3 Peripheral configuration	41
2.2.4 Peripheral assignment	41



1 Peripheral overview

The **RETRAM** internal memory is 64 Kbytes wide and is physically near to the Arm®Cortex®-M4 for optimized performance from the core. It is located in the VSW power domain, allowing it to be supplied during Standby *low power mode*, and to retain retention firmware that can be executed very quickly by the Cortex-M4 on wake up from Standby mode.

1.1 Features

Refer to *STM32MP15 reference manuals* for the complete feature list, and to the software components introduced below to see which features are actually implemented.

1.2 Security support

The RETRAM is a **secure** peripheral (under ETZPC control).



2 Peripheral usage and associated software

2.1 Boot time

Linux[®] remoteproc framework (running on the Cortex-A7) loads the Cortex-M4 firmware to the RETRAM, starting at address 0x00000000. At least, it must load the part of the firmware containing the vector table, since the Cortex-M4 reset entry point is address 0x00000004. The rest of the firmware code is loaded into the MCU SRAM. The overall memory mapping is shown in the platform memory mapping section.

2.2 Runtime

2.2.1 Overview

The Cortex-M4 vector table is mapped from address 0x00000000 (so to the RETRAM) at reset, but it can be remapped by software to any other location by means of the vector table offset register (VTOR). Beyond the reset entry point (0x00000004), the exception table also contains the software entries table used by the NVIC to branch the software execution to the right interrupt service routine.

While going to Standby low power mode, the RETRAM can remain supplied, so it can preserve a (small) Cortex-M4 piece of retention firmware that is executed on wake up when the ROM code (running on Cortex-A7) restarts the Cortex-M4. All these constraints make the RETRAM the minimum (and default) choice for Cortex-M4 firmware.

RETRAM can be allocated to:

- the Cortex-A7 secure to be used under OP-TEE.

or

- the Cortex-A7 non-secure to be used under Linux as reserved memory.

or

- the Cortex-M4 for use with the STM32Cube MPU Package, either for **runtime firmware** that can be mapped in both RETRAM and MCU SRAM, or for **retention firmware** that only fits into the RETRAM, but could have some data in MCU SRAM (keeping in mind that these data are lost while entering Standby low power mode).

2.2.2 Software frameworks

Domain	Peripheral	Software components			Comment
OP-TEE	Linux	STM32Cube			
Core/RAM	RETRAM	OP-TEE overview	Linux reserved memory	STM32Cube	

2.2.3 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 (especially for external peripherals), according to the information given in the corresponding software framework article.

2.2.4 Peripheral assignment

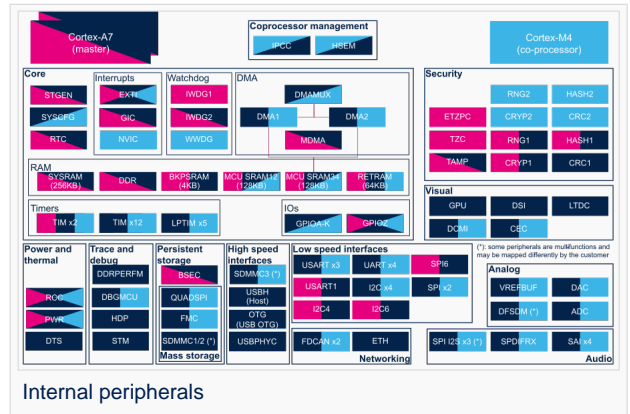


Check boxes illustrate the possible peripheral allocations supported by STM32 MPU Embedded Software:

- 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



Domain	Peripheral	Runtime allocation		Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)	
Core/RAM	RETRAM	RETRAM		Assignment (single choice)

Arm® is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



Cortex®

Linux® is a registered trademark of Linus Torvalds.

Microprocessor Unit

Open Portable Trusted Execution Environment

Random Access Memory (Early computer memories generally had serial 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 for volatile random-access semiconductor memories.)

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

Template:ArticleMainWriter Template:ArticleApprovedVersion

Contents

1 Peripheral overview	44
1.1 Features	44
1.2 Security support	44
2 Peripheral usage and associated software	45
2.1 Boot time	45
2.2 Runtime	45
2.2.1 Overview	45



2.2.2 Software frameworks	45
2.2.3 Peripheral configuration	45
2.2.4 Peripheral assignment	45



1 Peripheral overview

The **RETRAM** internal memory is 64 Kbytes wide and is physically near to the Arm[®]Cortex[®]-M4 for optimized performance from the core. It is located in the VSW power domain, allowing it to be supplied during Standby **low power mode**, and to retain retention firmware that can be executed very quickly by the Cortex-M4 on wake up from Standby mode.

1.1 Features

Refer to *STM32MP15 reference manuals* for the complete feature list, and to the software components introduced below to see which features are actually implemented.

1.2 Security support

The RETRAM is a **secure** peripheral (under ETZPC control).



2 Peripheral usage and associated software

2.1 Boot time

Linux[®] remoteproc framework (running on the Cortex-A7) loads the Cortex-M4 firmware to the RETRAM, starting at address 0x00000000. At least, it must load the part of the firmware containing the vector table, since the Cortex-M4 reset entry point is address 0x00000004. The rest of the firmware code is loaded into the MCU SRAM. The overall memory mapping is shown in the platform memory mapping section.

2.2 Runtime

2.2.1 Overview

The Cortex-M4 vector table is mapped from address 0x00000000 (so to the RETRAM) at reset, but it can be remapped by software to any other location by means of the vector table offset register (VTOR). Beyond the reset entry point (0x00000004), the exception table also contains the software entries table used by the NVIC to branch the software execution to the right interrupt service routine.

While going to Standby low power mode, the RETRAM can remain supplied, so it can preserve a (small) Cortex-M4 piece of retention firmware that is executed on wake up when the ROM code (running on Cortex-A7) restarts the Cortex-M4. All these constraints make the RETRAM the minimum (and default) choice for Cortex-M4 firmware.

RETRAM can be allocated to:

- the Cortex-A7 secure to be used under OP-TEE.

or

- the Cortex-A7 non-secure to be used under Linux as reserved memory.

or

- the Cortex-M4 for use with the STM32Cube MPU Package, either for **runtime firmware** that can be mapped in both RETRAM and MCU SRAM, or for **retention firmware** that only fits into the RETRAM, but could have some data in MCU SRAM (keeping in mind that these data are lost while entering Standby low power mode).

2.2.2 Software frameworks

Domain	Peripheral	Software components			Comment
OP-TEE	Linux	STM32Cube			
Core/RAM	RETRAM	OP-TEE overview	Linux reserved memory	STM32Cube	

2.2.3 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 (especially for external peripherals), according to the information given in the corresponding software framework article.

2.2.4 Peripheral assignment

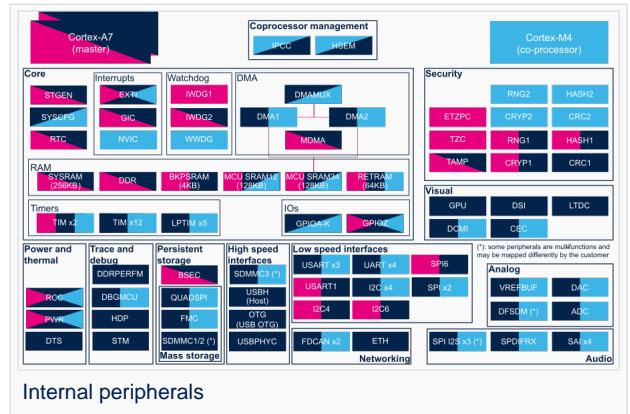


Check boxes illustrate the possible peripheral allocations supported by STM32 MPU Embedded Software:

- 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



Domain	Peripheral	Runtime allocation		Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)	
Core/RAM	RETRAM	RETRAM		Assignment (single choice)

Arm® is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



Cortex®

Linux® is a registered trademark of Linus Torvalds.

Microprocessor Unit

Open Portable Trusted Execution Environment

Random Access Memory (Early computer memories generally had serial 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 for volatile random-access semiconductor memories.)

Stable: 05.01.2021 - 17:13 / Revision: 05.01.2021 - 17:08

Template:ArticleMainWriter Template:ArticleApprovedVersion

Contents

1 Peripheral overview	48
1.1 Features	48
1.2 Security support	48
2 Peripheral usage and associated software	49
2.1 Boot time	49
2.2 Runtime	49
2.2.1 Overview	49



2.2.2 Software frameworks	49
2.2.3 Peripheral configuration	49
2.2.4 Peripheral assignment	49



1 Peripheral overview

The **RETRAM** internal memory is 64 Kbytes wide and is physically near to the Arm®Cortex®-M4 for optimized performance from the core. It is located in the VSW power domain, allowing it to be supplied during Standby **low power mode**, and to retain retention firmware that can be executed very quickly by the Cortex-M4 on wake up from Standby mode.

1.1 Features

Refer to *STM32MP15 reference manuals* for the complete feature list, and to the software components introduced below to see which features are actually implemented.

1.2 Security support

The RETRAM is a **secure** peripheral (under ETZPC control).



2 Peripheral usage and associated software

2.1 Boot time

Linux[®] remoteproc framework (running on the Cortex-A7) loads the Cortex-M4 firmware to the RETRAM, starting at address 0x00000000. At least, it must load the part of the firmware containing the vector table, since the Cortex-M4 reset entry point is address 0x00000004. The rest of the firmware code is loaded into the MCU SRAM. The overall memory mapping is shown in the platform memory mapping section.

2.2 Runtime

2.2.1 Overview

The Cortex-M4 vector table is mapped from address 0x00000000 (so to the RETRAM) at reset, but it can be remapped by software to any other location by means of the vector table offset register (VTOR). Beyond the reset entry point (0x00000004), the exception table also contains the software entries table used by the NVIC to branch the software execution to the right interrupt service routine.

While going to Standby low power mode, the RETRAM can remain supplied, so it can preserve a (small) Cortex-M4 piece of retention firmware that is executed on wake up when the ROM code (running on Cortex-A7) restarts the Cortex-M4. All these constraints make the RETRAM the minimum (and default) choice for Cortex-M4 firmware.

RETRAM can be allocated to:

- the Cortex-A7 secure to be used under OP-TEE.

or

- the Cortex-A7 non-secure to be used under Linux as reserved memory.

or

- the Cortex-M4 for use with the STM32Cube MPU Package, either for **runtime firmware** that can be mapped in both RETRAM and MCU SRAM, or for **retention firmware** that only fits into the RETRAM, but could have some data in MCU SRAM (keeping in mind that these data are lost while entering Standby low power mode).

2.2.2 Software frameworks

Domain	Peripheral	Software components			Comment
OP-TEE	Linux	STM32Cube			
Core/RAM	RETRAM	OP-TEE overview	Linux reserved memory	STM32Cube	

2.2.3 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 (especially for external peripherals), according to the information given in the corresponding software framework article.

2.2.4 Peripheral assignment

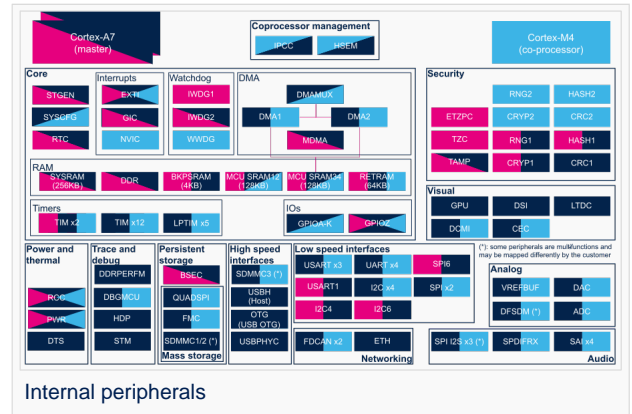


Check boxes illustrate the possible peripheral allocations supported by STM32 MPU Embedded Software:

- 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



Domain	Peripheral	Runtime allocation		Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)	
Core/RAM	RETRAM	RETRAM		Assignment (single choice)

Arm® is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



Cortex®

Linux® is a registered trademark of Linus Torvalds.

Microprocessor Unit

Open Portable Trusted Execution Environment

Random Access Memory (Early computer memories generally had serial 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 for volatile random-access semiconductor memories.)

Stable: 19.10.2021 - 14:25 / Revision: 19.10.2021 - 14:25

Template:ArticleMainWriter Template:ArticleApprovedVersion

Contents

1 Peripheral overview	52
1.1 Features	52
1.2 Security support	52
2 Peripheral usage and associated software	53
2.1 Boot time	53
2.2 Runtime	53
2.2.1 Overview	53



2.2.2 Software frameworks	53
2.2.3 Peripheral configuration	53
2.2.4 Peripheral assignment	53



1 Peripheral overview

The **RETRAM** internal memory is 64 Kbytes wide and is physically near to the Arm[®]Cortex[®]-M4 for optimized performance from the core. It is located in the VSW power domain, allowing it to be supplied during Standby **low power mode**, and to retain retention firmware that can be executed very quickly by the Cortex-M4 on wake up from Standby mode.

1.1 Features

Refer to *STM32MP15 reference manuals* for the complete feature list, and to the software components introduced below to see which features are actually implemented.

1.2 Security support

The RETRAM is a **secure** peripheral (under ETZPC control).



2 Peripheral usage and associated software

2.1 Boot time

Linux[®] remoteproc framework (running on the Cortex-A7) loads the Cortex-M4 firmware to the RETRAM, starting at address 0x00000000. At least, it must load the part of the firmware containing the vector table, since the Cortex-M4 reset entry point is address 0x00000004. The rest of the firmware code is loaded into the MCU SRAM. The overall memory mapping is shown in the platform memory mapping section.

2.2 Runtime

2.2.1 Overview

The Cortex-M4 vector table is mapped from address 0x00000000 (so to the RETRAM) at reset, but it can be remapped by software to any other location by means of the vector table offset register (VTOR). Beyond the reset entry point (0x00000004), the exception table also contains the software entries table used by the NVIC to branch the software execution to the right interrupt service routine.

While going to Standby low power mode, the RETRAM can remain supplied, so it can preserve a (small) Cortex-M4 piece of retention firmware that is executed on wake up when the ROM code (running on Cortex-A7) restarts the Cortex-M4. All these constraints make the RETRAM the minimum (and default) choice for Cortex-M4 firmware.

RETRAM can be allocated to:

- the Cortex-A7 secure to be used under OP-TEE.

or

- the Cortex-A7 non-secure to be used under Linux as reserved memory.

or

- the Cortex-M4 for use with the STM32Cube MPU Package, either for **runtime firmware** that can be mapped in both RETRAM and MCU SRAM, or for **retention firmware** that only fits into the RETRAM, but could have some data in MCU SRAM (keeping in mind that these data are lost while entering Standby low power mode).

2.2.2 Software frameworks

Domain	Peripheral	Software components			Comment
OP-TEE	Linux	STM32Cube			
Core/RAM	RETRAM	OP-TEE overview	Linux reserved memory	STM32Cube	

2.2.3 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 (especially for external peripherals), according to the information given in the corresponding software framework article.

2.2.4 Peripheral assignment

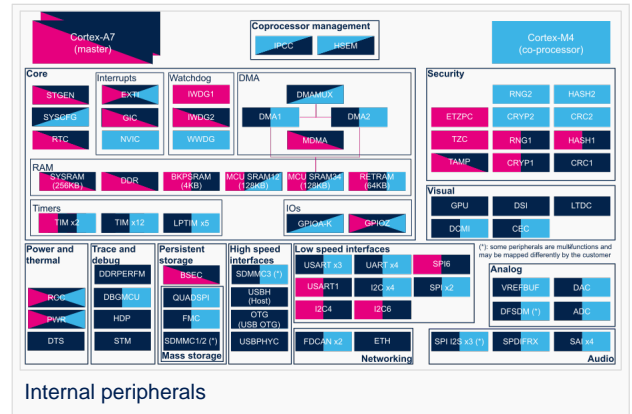


Check boxes illustrate the possible peripheral allocations supported by STM32 MPU Embedded Software:

- 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



Domain	Peripheral	Runtime allocation		Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)	
Core/RAM	RETRAM	RETRAM		Assignment (single choice)

Arm® is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



Cortex®

Linux® is a registered trademark of Linus Torvalds.

Microprocessor Unit

Open Portable Trusted Execution Environment

Random Access Memory (Early computer memories generally had serial 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 for volatile random-access semiconductor memories.)

Stable: 20.07.2021 - 09:02 / Revision: 11.03.2021 - 08:07

Template:ArticleMainWriter Template:ArticleApprovedVersion

Contents

1 Peripheral overview	56
1.1 Features	56
1.2 Security support	56
2 Peripheral usage and associated software	57
2.1 Boot time	57
2.2 Runtime	57
2.2.1 Overview	57



2.2.2 Software frameworks	57
2.2.3 Peripheral configuration	57
2.2.4 Peripheral assignment	57



1 Peripheral overview

The **RETRAM** internal memory is 64 Kbytes wide and is physically near to the Arm®Cortex®-M4 for optimized performance from the core. It is located in the VSW power domain, allowing it to be supplied during Standby **low power mode**, and to retain retention firmware that can be executed very quickly by the Cortex-M4 on wake up from Standby mode.

1.1 Features

Refer to *STM32MP15 reference manuals* for the complete feature list, and to the software components introduced below to see which features are actually implemented.

1.2 Security support

The RETRAM is a **secure** peripheral (under ETZPC control).



2 Peripheral usage and associated software

2.1 Boot time

Linux[®] remoteproc framework (running on the Cortex-A7) loads the Cortex-M4 firmware to the RETRAM, starting at address 0x00000000. At least, it must load the part of the firmware containing the vector table, since the Cortex-M4 reset entry point is address 0x00000004. The rest of the firmware code is loaded into the MCU SRAM. The overall memory mapping is shown in the platform memory mapping section.

2.2 Runtime

2.2.1 Overview

The Cortex-M4 vector table is mapped from address 0x00000000 (so to the RETRAM) at reset, but it can be remapped by software to any other location by means of the vector table offset register (VTOR). Beyond the reset entry point (0x00000004), the exception table also contains the software entries table used by the NVIC to branch the software execution to the right interrupt service routine.

While going to Standby low power mode, the RETRAM can remain supplied, so it can preserve a (small) Cortex-M4 piece of retention firmware that is executed on wake up when the ROM code (running on Cortex-A7) restarts the Cortex-M4. All these constraints make the RETRAM the minimum (and default) choice for Cortex-M4 firmware.

RETRAM can be allocated to:

- the Cortex-A7 secure to be used under OP-TEE.

or

- the Cortex-A7 non-secure to be used under Linux as reserved memory.

or

- the Cortex-M4 for use with the STM32Cube MPU Package, either for **runtime firmware** that can be mapped in both RETRAM and MCU SRAM, or for **retention firmware** that only fits into the RETRAM, but could have some data in MCU SRAM (keeping in mind that these data are lost while entering Standby low power mode).

2.2.2 Software frameworks

Domain	Peripheral	Software components			Comment
OP-TEE	Linux	STM32Cube			
Core/RAM	RETRAM	OP-TEE overview	Linux reserved memory	STM32Cube	

2.2.3 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 (especially for external peripherals), according to the information given in the corresponding software framework article.

2.2.4 Peripheral assignment

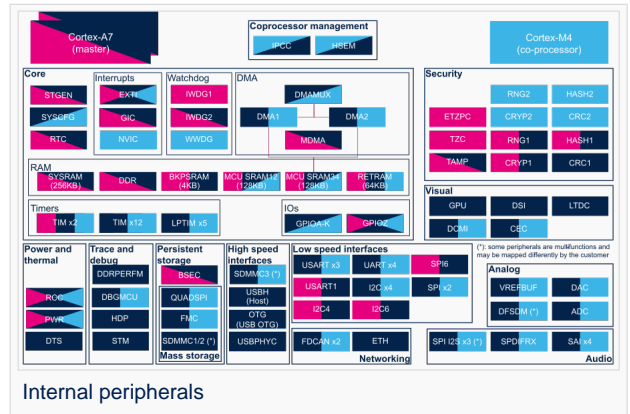


Check boxes illustrate the possible peripheral allocations supported by STM32 MPU Embedded Software:

- 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



Domain	Peripheral	Runtime allocation		Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)	
Core/RAM	RETRAM	RETRAM		Assignment (single choice)

Arm® is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



Cortex®

Linux® is a registered trademark of Linus Torvalds.

Microprocessor Unit

Open Portable Trusted Execution Environment

Random Access Memory (Early computer memories generally had serial 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 for volatile random-access semiconductor memories.)

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

Template:ArticleMainWriter Template:ArticleApprovedVersion

Contents

1 Peripheral overview	60
1.1 Features	60
1.2 Security support	60
2 Peripheral usage and associated software	61
2.1 Boot time	61
2.2 Runtime	61
2.2.1 Overview	61



2.2.2 Software frameworks	61
2.2.3 Peripheral configuration	61
2.2.4 Peripheral assignment	61



1 Peripheral overview

The **RETRAM** internal memory is 64 Kbytes wide and is physically near to the Arm[®]Cortex[®]-M4 for optimized performance from the core. It is located in the VSW power domain, allowing it to be supplied during Standby *low power mode*, and to retain retention firmware that can be executed very quickly by the Cortex-M4 on wake up from Standby mode.

1.1 Features

Refer to *STM32MP15 reference manuals* for the complete feature list, and to the software components introduced below to see which features are actually implemented.

1.2 Security support

The RETRAM is a **secure** peripheral (under ETZPC control).



2 Peripheral usage and associated software

2.1 Boot time

Linux[®] remoteproc framework (running on the Cortex-A7) loads the Cortex-M4 firmware to the RETRAM, starting at address 0x00000000. At least, it must load the part of the firmware containing the vector table, since the Cortex-M4 reset entry point is address 0x00000004. The rest of the firmware code is loaded into the MCU SRAM. The overall memory mapping is shown in the platform memory mapping section.

2.2 Runtime

2.2.1 Overview

The Cortex-M4 vector table is mapped from address 0x00000000 (so to the RETRAM) at reset, but it can be remapped by software to any other location by means of the vector table offset register (VTOR). Beyond the reset entry point (0x00000004), the exception table also contains the software entries table used by the NVIC to branch the software execution to the right interrupt service routine.

While going to Standby low power mode, the RETRAM can remain supplied, so it can preserve a (small) Cortex-M4 piece of retention firmware that is executed on wake up when the ROM code (running on Cortex-A7) restarts the Cortex-M4. All these constraints make the RETRAM the minimum (and default) choice for Cortex-M4 firmware.

RETRAM can be allocated to:

- the Cortex-A7 secure to be used under OP-TEE.

or

- the Cortex-A7 non-secure to be used under Linux as reserved memory.

or

- the Cortex-M4 for use with the STM32Cube MPU Package, either for **runtime firmware** that can be mapped in both RETRAM and MCU SRAM, or for **retention firmware** that only fits into the RETRAM, but could have some data in MCU SRAM (keeping in mind that these data are lost while entering Standby low power mode).

2.2.2 Software frameworks

Domain	Peripheral	Software components			Comment
OP-TEE	Linux	STM32Cube			
Core/RAM	RETRAM	OP-TEE overview	Linux reserved memory	STM32Cube	

2.2.3 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 (especially for external peripherals), according to the information given in the corresponding software framework article.

2.2.4 Peripheral assignment

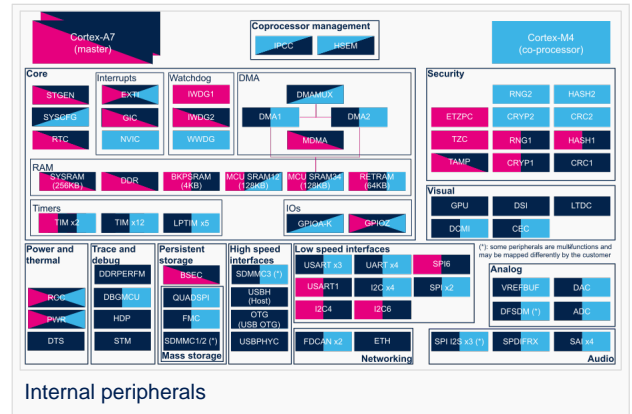


Check boxes illustrate the possible peripheral allocations supported by STM32 MPU Embedded Software:

- 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.



Domain	Peripheral	Runtime allocation		Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)	
Core/RAM	RETRAM	RETRAM		Assignment (single choice)

Arm® is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

Cortex®

Linux® is a registered trademark of Linus Torvalds.

Microprocessor Unit

Open Portable Trusted Execution Environment

Random Access Memory (Early computer memories generally had serial 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 for volatile random-access semiconductor memories.)