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 9
4. Linux remoteproc framework overview 12
5. MCU SRAM internal memory 15
6. NVIC internal peripheral 18
7. OP-TEE overview 21
8. Power overview 24
9. Reserved memory 27
10. STM32CubeMP1 architecture 30
11. STM32CubeMX 33
12. STM32MP15 RAM mapping 36
13. STM32MP15 ROM code overview 39
14. STM32MP15 resources 42
15. STM32MPU Embedded Software architecture overview 45

Stable: 04.02.2020 - 15:59 / Revision: 04.02.2020 - 15:50

Stable: 04.02.2020 - 15:00 / Revision: 04.02.2020 - 15:00

A quality version of this page, approved on *4 February 2020*, was based off this revision.

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 ca 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

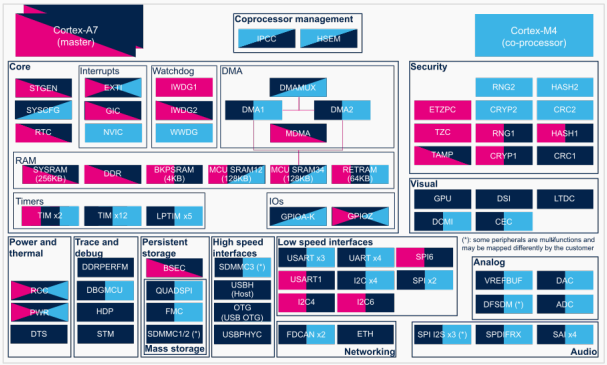
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (especially for external peripherals), according to the information given in 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 possiblities might be described in STM32MP15 reference manuals
.



Internal peripherals

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

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

Contents

1 Peripheral overview 7
1.1 Features 7
1.2 Security support 7
2 Peripheral usage and associated software 8
2.1 Boot time 8
2.2 Runtime 8
2.2.1 Overview 8
2.2.2 Software frameworks 8
2.2.3 Peripheral configuration 8
2.2.4 Peripheral assignment 8

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 ca 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

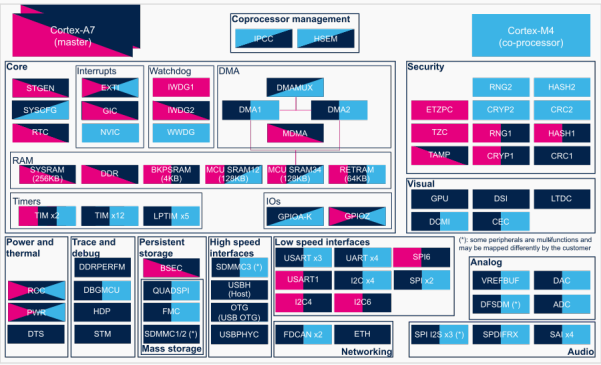
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (especially for external peripherals), according to the information given in 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 possiblities might be described in STM32MP15 reference manuals .



Internal peripherals

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

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

Contents

1 Peripheral overview 10
1.1 Features 10
1.2 Security support 10
2 Peripheral usage and associated software 11
2.1 Boot time 11
2.2 Runtime 11
2.2.1 Overview 11
2.2.2 Software frameworks 11
2.2.3 Peripheral configuration 11
2.2.4 Peripheral assignment 11

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 ca 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

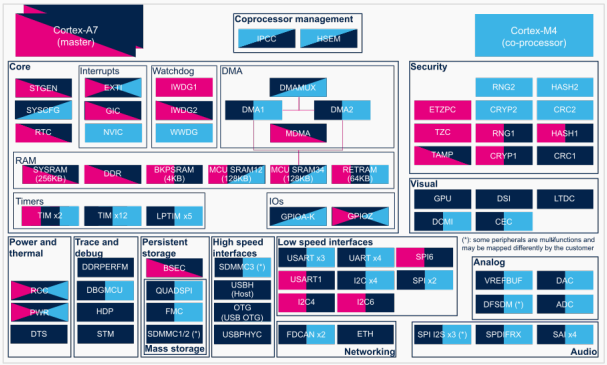
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (especially for external peripherals), according to the information given in 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 possiblities might be described in STM32MP15 reference manuals .



Internal peripherals

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

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

Contents

1 Peripheral overview 13
1.1 Features 13
1.2 Security support 13
2 Peripheral usage and associated software 14
2.1 Boot time 14
2.2 Runtime 14
2.2.1 Overview 14
2.2.2 Software frameworks 14
2.2.3 Peripheral configuration 14
2.2.4 Peripheral assignment 14

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 ca 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

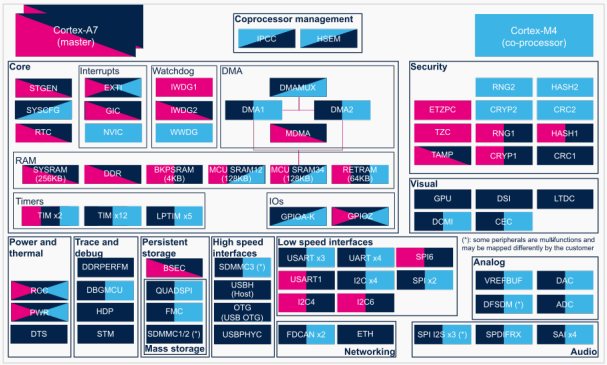
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (especially for external peripherals), according to the information given in 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 possiblities might be described in STM32MP15 reference manuals
.



Internal peripherals

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

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

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 ca 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

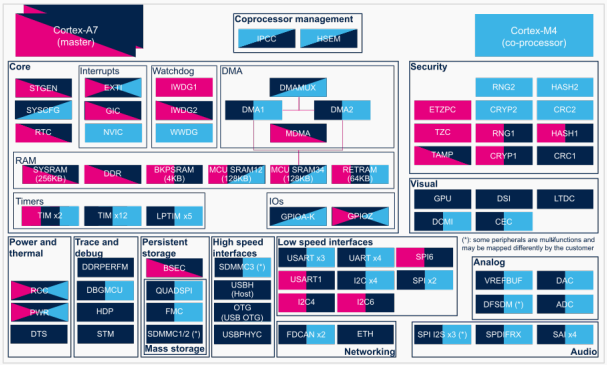
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (especially for external peripherals), according to the information given in 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 possiblities might be described in STM32MP15 reference manuals .



Internal peripherals

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

Stable: 25.03.2021 - 13:50 / Revision: 18.03.2021 - 17:29

Contents

1 Peripheral overview 19
1.1 Features 19
1.2 Security support 19
2 Peripheral usage and associated software 20
2.1 Boot time 20
2.2 Runtime 20
2.2.1 Overview 20
2.2.2 Software frameworks 20
2.2.3 Peripheral configuration 20
2.2.4 Peripheral assignment 20

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 ca 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

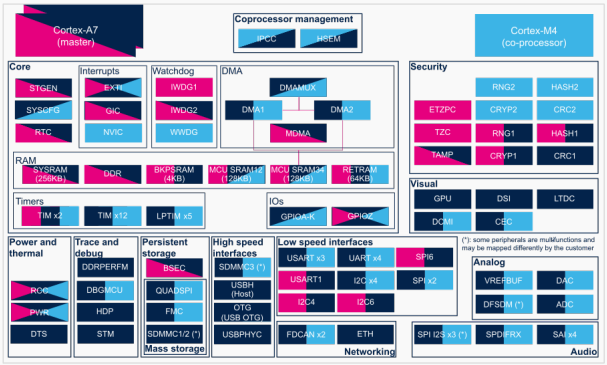
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (especially for external peripherals), according to the information given in 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 possiblities might be described in STM32MP15 reference manuals .



Internal peripherals

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

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

Contents

1 Peripheral overview 22
1.1 Features 22
1.2 Security support 22
2 Peripheral usage and associated software 23
2.1 Boot time 23
2.2 Runtime 23
2.2.1 Overview 23
2.2.2 Software frameworks 23
2.2.3 Peripheral configuration 23
2.2.4 Peripheral assignment 23

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 ca 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

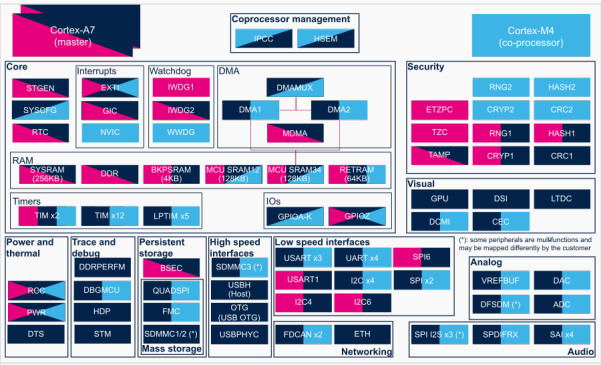
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (especially for external peripherals), according to the information given in 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 possiblities might be described in STM32MP15 reference manuals .



Internal peripherals

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

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

Contents

1 Peripheral overview 25
1.1 Features 25
1.2 Security support 25
2 Peripheral usage and associated software 26
2.1 Boot time 26
2.2 Runtime 26
2.2.1 Overview 26
2.2.2 Software frameworks 26
2.2.3 Peripheral configuration 26
2.2.4 Peripheral assignment 26

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 ca 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

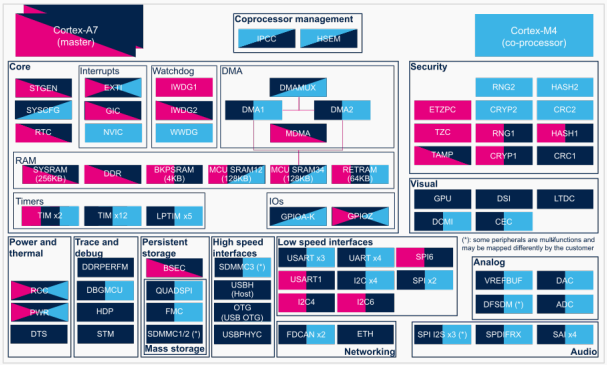
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (especially for external peripherals), according to the information given in 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 possiblities might be described in STM32MP15 reference manuals
.



Internal peripherals

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

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

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 ca 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

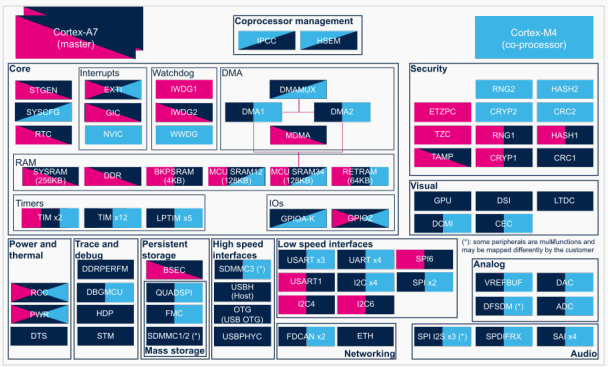
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (especially for external peripherals), according to the information given in 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 possiblities might be described in STM32MP15 reference manuals
.



Internal peripherals

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

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

Contents

1 Peripheral overview 31
1.1 Features 31
1.2 Security support 31
2 Peripheral usage and associated software 32
2.1 Boot time 32
2.2 Runtime 32
2.2.1 Overview 32
2.2.2 Software frameworks 32
2.2.3 Peripheral configuration 32
2.2.4 Peripheral assignment 32

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 ca 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

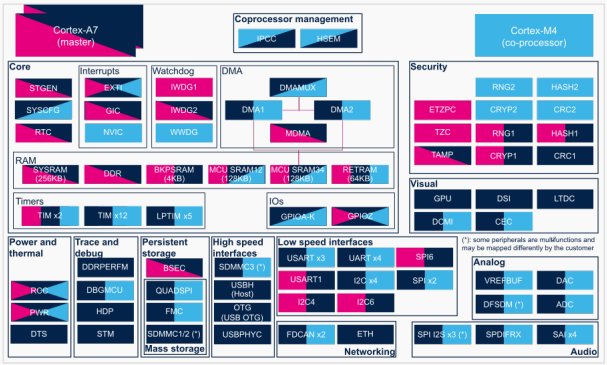
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (especially for external peripherals), according to the information given in 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 possiblities might be described in STM32MP15 reference manuals
.



Internal peripherals

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

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

Contents

1 Peripheral overview 34
1.1 Features 34
1.2 Security support 34
2 Peripheral usage and associated software 35
2.1 Boot time 35
2.2 Runtime 35
2.2.1 Overview 35
2.2.2 Software frameworks 35
2.2.3 Peripheral configuration 35
2.2.4 Peripheral assignment 35

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 ca 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

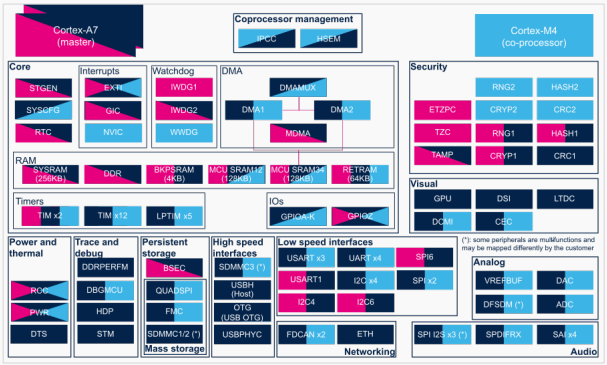
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (especially for external peripherals), according to the information given in 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 possiblities might be described in STM32MP15 reference manuals
.



Internal peripherals

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

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

Contents

1 Peripheral overview 37
1.1 Features 37
1.2 Security support 37
2 Peripheral usage and associated software 38
2.1 Boot time 38
2.2 Runtime 38
2.2.1 Overview 38
2.2.2 Software frameworks 38
2.2.3 Peripheral configuration 38
2.2.4 Peripheral assignment 38

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 ca 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

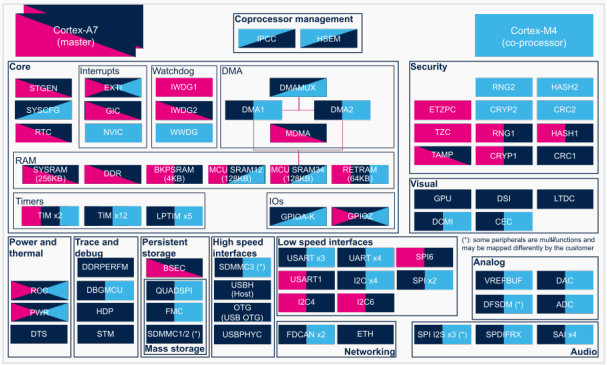
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (especially for external peripherals), according to the information given in 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 possiblities might be described in STM32MP15 reference manuals
.



Internal peripherals

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

Stable: 17.11.2021 - 13:32 / Revision: 17.11.2021 - 10:37

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 ca 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

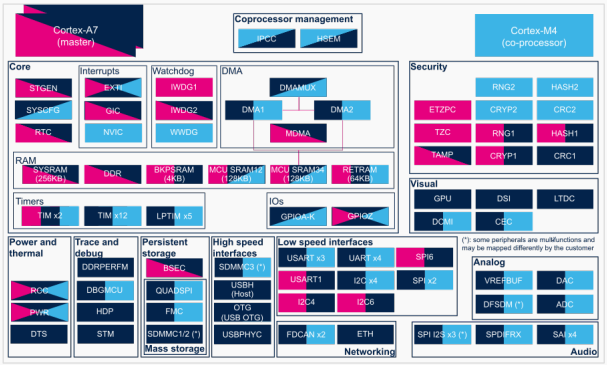
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (especially for external peripherals), according to the information given in 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 possiblities might be described in STM32MP15 reference manuals
.



Internal peripherals

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

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

Contents

1 Peripheral overview 43
1.1 Features 43
1.2 Security support 43
2 Peripheral usage and associated software 44
2.1 Boot time 44
2.2 Runtime 44
2.2.1 Overview 44
2.2.2 Software frameworks 44
2.2.3 Peripheral configuration 44
2.2.4 Peripheral assignment 44

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 ca 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

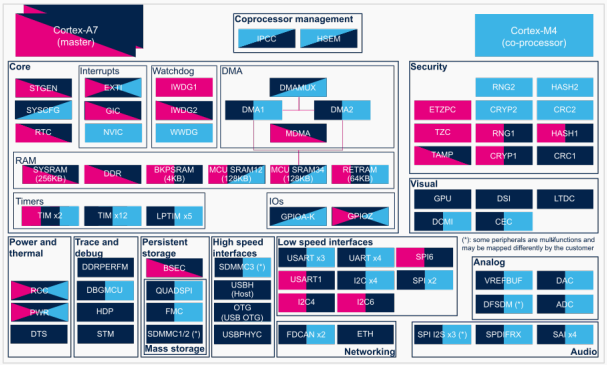
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (especially for external peripherals), according to the information given in 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 possiblities might be described in STM32MP15 reference manuals
.



Internal peripherals

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

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

Contents

1 Peripheral overview 46
1.1 Features 46
1.2 Security support 46
2 Peripheral usage and associated software 47
2.1 Boot time 47
2.2 Runtime 47
2.2.1 Overview 47
2.2.2 Software frameworks 47
2.2.3 Peripheral configuration 47
2.2.4 Peripheral assignment 47

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 ca 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

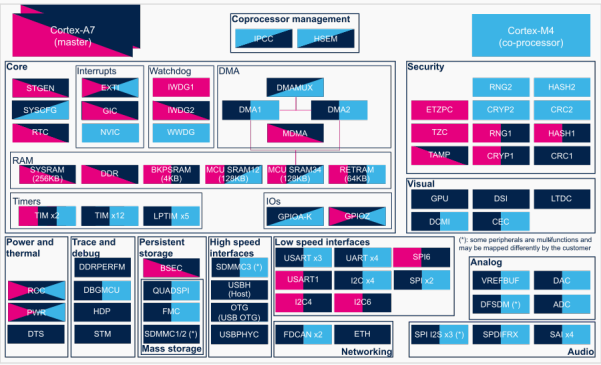
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (especially for external peripherals), according to the information given in 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 possiblities might be described in STM32MP15 reference manuals
.



Internal peripherals

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