STM internal peripheral

Contents

1. STM internal peripheral 3
2. STM32MP15 resources 6
3. Coresight overview 9
4. STM32CubeMX 13
5. STM32MPU Embedded Software architecture overview 16
6. How to assign an internal peripheral to a runtime context 19

STM internal peripheral

Stable: 26.09.2019 - 13:44 / Revision: 26.09.2019 - 13:43

A quality version of this page, accepted on *26 September 2019*, was based off this revision.

Contents

1 Article purpose 3
2 Peripheral overview 3
2.1 Features 3
2.2 Security support 4
3 Peripheral usage and associated software 4
3.1 Boot time 4
3.2 Runtime 4
3.2.1 Overview 4
3.2.2 Software frameworks 4
3.2.3 Peripheral configuration 5
3.2.4 Peripheral assignment 5
4 References 6

1 Article purpose

The purpose of this article is to:

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

2 Peripheral overview

The **STM** peripheral is used to log STM trace into the embedded trace FIFO (ETF). This trace can include hardware events (the list is given in the STM32MP15 reference manuals) or direct 'printf like' log from the Cortex®-A7. Once in the ETF buffer, the trace can directly be dumped from the Cortex®-A7 or to the trace port interface unit (TPIU), connected to an external probe able to decode it.

2.1 Features

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

2.2 Security support

The STM is a **non secure** peripheral.

3 Peripheral usage and associated software

3.1 Boot time

The STM is not used at boot time.

3.2 Runtime

3.2.1 Overview

The STM can be assigned to the Cortex®-A7 non-secure for using in Linux with coresight framework.
This driver allows to select the hardware events (listed in the STM32MP15 reference manuals) to log via the STM peripheral into the ETF and dump it in the Linux console for analysis.

3.2.2 Software frameworks

Do	Peri	Software frameworks			Comment
mai n Cor tex -A7 sec ure (O P- TE E)	Cor tex -A7 no n- sec ure (Li nux)	Cortex-M4 (STM32Cube)			
Tr ac e & De bu g	ST M		Linux Coresight framework		

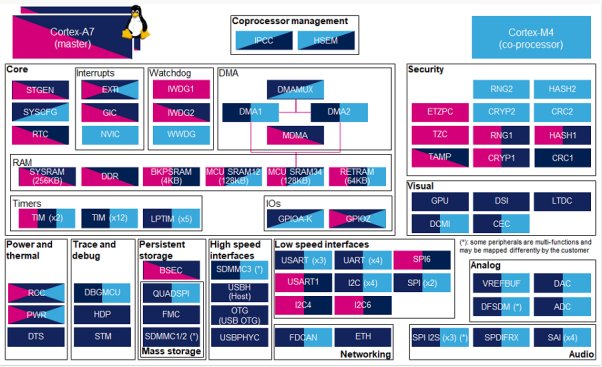
3.2.3 Peripheral configuration

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

3.2.4 Peripheral assignment

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

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



Internal peripherals

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.

Domain	Periph	Runtime allocation				Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Trace	S					

Domain & Debug	Peripheral	Runtime allocation				Comment
	STM					

4 References

System Trace Module

Open Portable Trusted Execution Environment

STM internal peripheral

Stable: 24.06.2020 - 17:52 / Revision: 24.06.2020 - 12:32

Contents

1 Article purpose 6
2 Peripheral overview 7
2.1 Features 7
2.2 Security support 7
3 Peripheral usage and associated software 7
3.1 Boot time 7
3.2 Runtime 7
3.2.1 Overview 7
3.2.2 Software frameworks 7
3.2.3 Peripheral configuration 8
3.2.4 Peripheral assignment 8
4 References 9

1 Article purpose

The purpose of this article is to:

- briefly introduce the STM peripheral and its main features
- indicate the level of security supported by this hardware block

- explain how each instance can be allocated to the three runtime contexts and linked to the corresponding software components
- explain, when necessary, how to configure the STM peripheral.

2 Peripheral overview

The **STM** peripheral is used to log STM trace into the embedded trace FIFO (ETF). This trace can include hardware events (the list is given in the STM32MP15 reference manuals) or direct 'printf like' log from the Cortex®-A7. Once in the ETF buffer, the trace can directly be dumped from the Cortex®-A7 or to the trace port interface unit (TPIU), connected to an external probe able to decode it.

2.1 Features

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

2.2 Security support

The STM is a **non secure** peripheral.

3 Peripheral usage and associated software

3.1 Boot time

The STM is not used at boot time.

3.2 Runtime

3.2.1 Overview

The STM can be assigned to the Cortex®-A7 non-secure for using in Linux with coresight framework.
This driver allows to select the hardware events (listed in the STM32MP15 reference manuals) to log via the STM peripheral into the ETF and dump it in the Linux console for analysis.

3.2.2 Software frameworks

Do	Peri	Software frameworks	Comment
mai Cor tex	Cor tex		

Domain	Peripheral	Software frameworks			Comment
-A7 secure (OP-TEE)	-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Trace & Debug	STM		Linux Coresight framework		

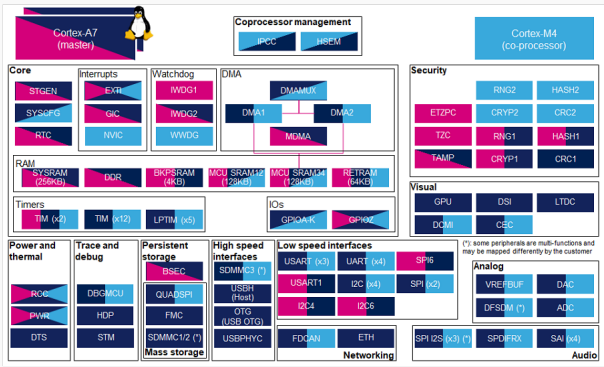
3.2.3 Peripheral configuration

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

3.2.4 Peripheral assignment

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

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



Internal peripherals

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.

Domain	Peripheral	Runtime allocation			Comment
	Cortex-A				

Domain	Peripheral	Runtime allocation				Comment
Instance	secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Trace&Debug	STM	STM				

4 References

System Trace Module

Open Portable Trusted Execution Environment

STM internal peripheral

Stable: ***Not stable*** */ Revision: 25.03.2020 - 15:22*

Contents

1 Article purpose 10
2 Peripheral overview 10
2.1 Features 10
2.2 Security support 10
3 Peripheral usage and associated software 10
3.1 Boot time 10

3.2 Runtime 11
3.2.1 Overview 11
3.2.2 Software frameworks 11
3.2.3 Peripheral configuration 11
3.2.4 Peripheral assignment 11
4 References 12

1 Article purpose

The purpose of this article is to:

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

2 Peripheral overview

The **STM** peripheral is used to log STM trace into the embedded trace FIFO (ETF). This trace can include hardware events (the list is given in the STM32MP15 reference manuals) or direct 'printf like' log from the Cortex®-A7. Once in the ETF buffer, the trace can directly be dumped from the Cortex®-A7 or to the trace port interface unit (TPIU), connected to an external probe able to decode it.

2.1 Features

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

2.2 Security support

The STM is a **non secure** peripheral.

3 Peripheral usage and associated software

3.1 Boot time

The STM is not used at boot time.

3.2 Runtime

3.2.1 Overview

The STM can be assigned to the Cortex®-A7 non-secure for using in Linux with coresight framework.
This driver allows to select the hardware events (listed in the STM32MP15 reference manuals) to log via the STM peripheral into the ETF and dump it in the Linux console for analysis.

3.2.2 Software frameworks

Do	Peri	Software frameworks			Comment
main Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Trace & Debug	STM		Linux Coresight framework		

3.2.3 Peripheral configuration

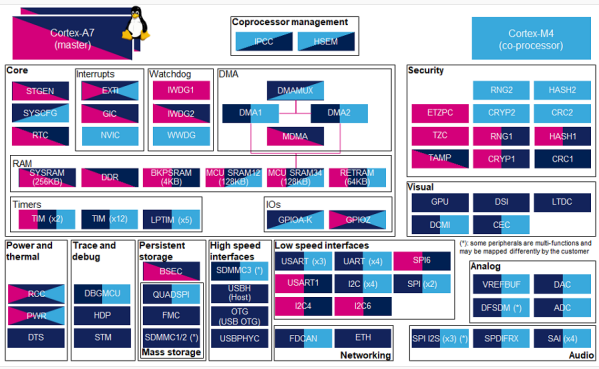
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (particularly for external peripherals), according to the information given in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

3.2.4 Peripheral assignment

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

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

Refer to How to assign an internal peripheral to a runtime context for more information on how to assign peripherals manually or via STM32CubeMX.
The present chapter describes STMicroelectronics recommendations or choice of implementation. Additional possiblities might be described in STM32MP15 reference manuals.



Internal peripherals

Domain	Peripheral	Runtime allocation				Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Trace&Debug	STM	STM				

4 References

System Trace Module

Open Portable Trusted Execution Environment

STM internal peripheral

*Stable: **Not stable** / Revision: 12.06.2020 - 13:25*

Contents

1 Article purpose 13
2 Peripheral overview 13
2.1 Features 14
2.2 Security support 14
3 Peripheral usage and associated software 14
3.1 Boot time 14
3.2 Runtime 14
3.2.1 Overview 14
3.2.2 Software frameworks 14
3.2.3 Peripheral configuration 15
3.2.4 Peripheral assignment 15
4 References 16

1 Article purpose

The purpose of this article is to:

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

2 Peripheral overview

The **STM** peripheral is used to log STM trace into the embedded trace FIFO (ETF). This trace can include hardware events (the list is given in the STM32MP15 reference manuals) or direct 'printf like' log from the Cortex®-A7. Once in the ETF buffer, the trace can directly be dumped from the Cortex®-A7 or to the trace port interface unit (TPIU), connected to an external probe able to decode it.

2.1 Features

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

2.2 Security support

The STM is a **non secure** peripheral.

3 Peripheral usage and associated software

3.1 Boot time

The STM is not used at boot time.

3.2 Runtime

3.2.1 Overview

The STM can be assigned to the Cortex®-A7 non-secure for using in Linux with coresight framework.
This driver allows to select the hardware events (listed in the STM32MP15 reference manuals) to log via the STM peripheral into the ETF and dump it in the Linux console for analysis.

3.2.2 Software frameworks

Do	Peri	Software frameworks			Comment
mai n Cor tex -A7 sec ure (O P- TE E)	Cor tex -A7 no n- sec ure (Li nux)	Cortex-M4 (STM32Cube)			
Tr ac e & De	ST		Linux Coresight		

Domain	Peripheral	Software frameworks			Comment
bug	M		framework		

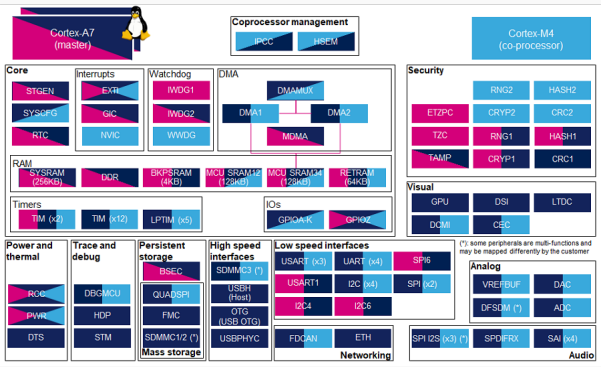
3.2.3 Peripheral configuration

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

3.2.4 Peripheral assignment

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

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



Internal peripherals

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.

Domain	Peripheral	Runtime allocation				Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Trac						

Domain & Debug	Peripheral	Runtime allocation				Comment
	STM	STM				

4 References

System Trace Module

Open Portable Trusted Execution Environment

STM internal peripheral

Stable: 15.10.2019 - 11:55 / Revision: 15.10.2019 - 11:55

Contents

1 Article purpose 16
2 Peripheral overview 17
2.1 Features 17
2.2 Security support 17
3 Peripheral usage and associated software 17
3.1 Boot time 17
3.2 Runtime 17
3.2.1 Overview 17
3.2.2 Software frameworks 17
3.2.3 Peripheral configuration 18
3.2.4 Peripheral assignment 18
4 References 19

1 Article purpose

The purpose of this article is to:

- briefly introduce the STM peripheral and its main features

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

2 Peripheral overview

The **STM** peripheral is used to log STM trace into the embedded trace FIFO (ETF). This trace can include hardware events (the list is given in the STM32MP15 reference manuals) or direct 'printf like' log from the Cortex®-A7. Once in the ETF buffer, the trace can directly be dumped from the Cortex®-A7 or to the trace port interface unit (TPIU), connected to an external probe able to decode it.

2.1 Features

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

2.2 Security support

The STM is a **non secure** peripheral.

3 Peripheral usage and associated software

3.1 Boot time

The STM is not used at boot time.

3.2 Runtime

3.2.1 Overview

The STM can be assigned to the Cortex®-A7 non-secure for using in Linux with coresight framework.
This driver allows to select the hardware events (listed in the STM32MP15 reference manuals) to log via the STM peripheral into the ETF and dump it in the Linux console for analysis.

3.2.2 Software frameworks

Do	Peri	Software frameworks	Comment
mai Cor n	Cor		

Domain	Peri	Software frameworks			Comment
tex-A7 secure (OP-TEE)	tex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Trace & Debug	STM		Linux Coresight framework		

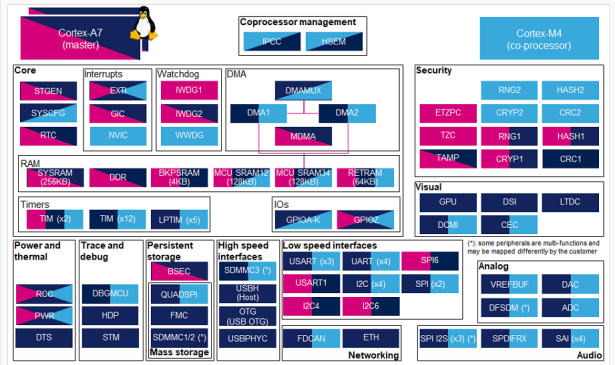
3.2.3 Peripheral configuration

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

3.2.4 Peripheral assignment

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

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



Internal peripherals

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.

Domain	Peripheral	Runtime allocation			Comment
	Cortex-				

Domain instance	Peripheral	Runtime allocation				Comment
	A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Trace&Debug	STM	STM				

4 References

System Trace Module

Open Portable Trusted Execution Environment

STM internal peripheral

Stable: ***Not stable*** */ Revision: 24.06.2020 - 11:43*

Contents

1 Article purpose 20
2 Peripheral overview 20
2.1 Features 20
2.2 Security support 20

3 Peripheral usage and associated software 21
3.1 Boot time 21
3.2 Runtime 21
3.2.1 Overview 21
3.2.2 Software frameworks 21
3.2.3 Peripheral configuration 21
3.2.4 Peripheral assignment 21
4 References 23

1 Article purpose

The purpose of this article is to:

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

2 Peripheral overview

The **STM** peripheral is used to log STM trace into the embedded trace FIFO (ETF). This trace can include hardware events (the list is given in the STM32MP15 reference manuals) or direct 'printf like' log from the Cortex®-A7. Once in the ETF buffer, the trace can directly be dumped from the Cortex®-A7 or to the trace port interface unit (TPIU), connected to an external probe able to decode it.

2.1 Features

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

2.2 Security support

The STM is a **non secure** peripheral.

3 Peripheral usage and associated software

3.1 Boot time

The STM is not used at boot time.

3.2 Runtime

3.2.1 Overview

The STM can be assigned to the Cortex®-A7 non-secure for using in Linux with coresight framework.
This driver allows to select the hardware events (listed in the STM32MP15 reference manuals) to log via the STM peripheral into the ETF and dump it in the Linux console for analysis.

3.2.2 Software frameworks

Domain	Peripheral	Software frameworks			Comment
Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Trace & Debug	STM		Linux Coresight framework		

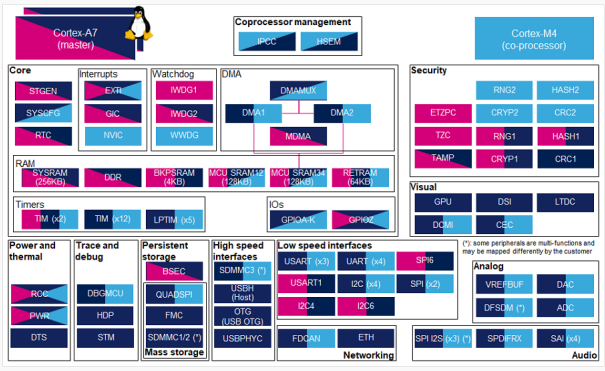
3.2.3 Peripheral configuration

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

3.2.4 Peripheral assignment

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

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



Internal peripherals

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.

Domain	Peripheral	Runtime allocation				Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Trace&Debug	STM	STM				

4 References

System Trace Module

Open Portable Trusted Execution Environment