USART internal peripheral

Contents

1. USART internal peripheral 3
2. Category:Getting started with STM32MP1 boards 9
3. DMA internal peripheral 17
4. ETZPC internal peripheral 23
5. How to assign an internal peripheral to a runtime context 29
6. OP-TEE overview 35
7. STM32CubeMP1 architecture 41
8. STM32CubeMX 47
9. STM32CubeProgrammer 53
10. STM32MP15 resources 59
11. STM32MPU Embedded Software architecture overview 65
12. Serial TTY device tree configuration 71
13. Serial TTY overview 77

Stable: 04.02.2020 - 16:08 / Revision: 04.02.2020 - 16:03

Stable: 04.02.2020 - 10:00 | Revision: 04.02.2020 - 10:00

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

Contents

1 Article purpose 4
2 Peripheral overview 5
2.1 Features 5
2.2 Security support 5
3 Peripheral usage and associated software 6
3.1 Boot time 6
3.2 Runtime 6
3.2.1 Overview 6
3.2.2 Software frameworks 6
3.2.3 Peripheral configuration 6
3.2.4 Peripheral assignment 6
4 How to go further 8
5 References 9

1 Article purpose

The purpose of this article is to:

- briefly introduce the USART 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 needed, how to configure the USART peripheral.

2 Peripheral overview

The **USART** peripheral is used to interconnect STM32 MPU devices with other systems, typically via RS232 or RS485 protocols. In addition, the USART supports the **Synchronous** mode that can be used for smartcard interfacing or SPI master /slave operation.

The **UART** peripheral is similar to the USART but does not support the Synchronous mode.

High-speed data communications can be achieved by using the DMA internal peripheral for multibuffer configuration.

2.1 Features

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

2.2 Security support

USART1 is a **secure** instance (under ETZPC control).
The other UARTs and USARTs are **non-secure** instances.

3 Peripheral usage and associated software

3.1 Boot time

All USART (except USART1) and UART instances are boot devices that support serial boot for Flash programming with STM32CubeProgrammer.

3.2 Runtime

3.2.1 Overview

The STM32 MPU devices feature four USART instances (supporting both Asynchronous and Synchronous modes), and four UART instances (supporting only Asynchronous mode).

USART1 can be allocated to:

- the Arm® Cortex®-A7 secure core to be used under OP-TEE with the USART OP-TEE driver, typically to communicate with a smartcard.

All USART and UART instances can be allocated to:

- the Arm® Cortex®-A7 non-secure core to be used under Linux® with the tty framework. However, the Linux® kernel supports only the UART Asynchronous mode (Synchronous mode not supported).

or

- the Arm® Cortex®-M4 to be used with STM32Cube MPU Package with USART HAL driver. Both USART Synchronous and Asynchronous modes are supported by the STM32Cube MPU Package.

Chapter Peripheral assignment describes which peripheral instance can be assigned to which context.

3.2.2 Software frameworks

Domain	Peripheral	Software frameworks			Comment
Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Low speed interface	USART	USART OP-TEE driver	Linux serial /tty framework	STM32Cube USART driver	

3.2.3 Peripheral configuration

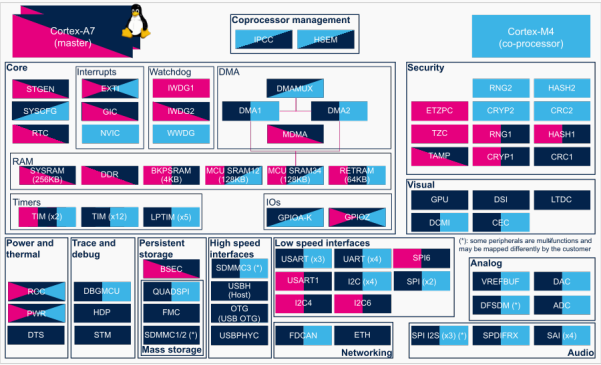
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (particularly for external peripherals) according to the information given in the corresponding software framework article or, for Linux in the Serial TTY device tree configuration 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)			
Low speed interface	USART	USART1				Assignment (single choice)
		USART2				Assignment (single choice)
		USART3				Assignment (single choice)
		UART4				Assignment (single choice). Used for Linux® serial console on ST boards.
		UART5				Assignment (single choice)
		USART6				Assignment (single choice)
		UART7				Assignment (single choice)
		UART8				Assignment (single choice)

4 How to go further

Additional documentation on USART peripheral is available on st.com:

- STM32 USART training [1] presents the STM32 Universal Synchronous/Asynchronous Receiver/Transmitter interface.
- STM32 USART automatic baud rate detection [2] presents STM32 USART automatic baud rate detection.

5 References

- Please refer to **stm32f7_peripheral_usart** document on st.com
- STM32 USART automatic baud rate detection application note (AN4908)

Universal Synchronous/Asynchronous Receiver/Transmitter

Microprocessor Unit

Serial Peripheral Interface

Universal Asynchronous Receiver/Transmitter

Open Portable Trusted Execution Environment

Stable: 17.06.2020 - 15:26 / Revision: 16.01.2020 - 09:30

Contents

1 Article purpose 10
2 Peripheral overview 11
2.1 Features 11
2.2 Security support 11
3 Peripheral usage and associated software 12
3.1 Boot time 12
3.2 Runtime 12
3.2.1 Overview 12
3.2.2 Software frameworks 12
3.2.3 Peripheral configuration 12
3.2.4 Peripheral assignment 12
4 How to go further 14
5 References 15

1 Article purpose

The purpose of this article is to:

- briefly introduce the USART 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 needed, how to configure the USART peripheral.

2 Peripheral overview

The **USART** peripheral is used to interconnect STM32 MPU devices with other systems, typically via RS232 or RS485 protocols. In addition, the USART supports the **Synchronous** mode that can be used for smartcard interfacing or SPI master /slave operation.

The **UART** peripheral is similar to the USART but does not support the Synchronous mode.

High-speed data communications can be achieved by using the DMA internal peripheral for multibuffer configuration.

2.1 Features

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

2.2 Security support

USART1 is a **secure** instance (under ETZPC control).
The other UARTs and USARTs are **non-secure** instances.

3 Peripheral usage and associated software

3.1 Boot time

All USART (except USART1) and UART instances are boot devices that support serial boot for Flash programming with STM32CubeProgrammer.

3.2 Runtime

3.2.1 Overview

The STM32 MPU devices feature four USART instances (supporting both Asynchronous and Synchronous modes), and four UART instances (supporting only Asynchronous mode).

USART1 can be allocated to:

- the Arm® Cortex®-A7 secure core to be used under OP-TEE with the USART OP-TEE driver, typically to communicate with a smartcard.

All USART and UART instances can be allocated to:

- the Arm® Cortex®-A7 non-secure core to be used under Linux® with the tty framework. However, the Linux® kernel supports only the UART Asynchronous mode (Synchronous mode not supported).

or

- the Arm® Cortex®-M4 to be used with STM32Cube MPU Package with USART HAL driver. Both USART Synchronous and Asynchronous modes are supported by the STM32Cube MPU Package.

Chapter Peripheral assignment describes which peripheral instance can be assigned to which context.

3.2.2 Software frameworks

Domain	Peripheral	Software frameworks			Comment
Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Low speed interface	USART	USART OP-TEE driver	Linux serial /tty framework	STM32Cube USART driver	

3.2.3 Peripheral configuration

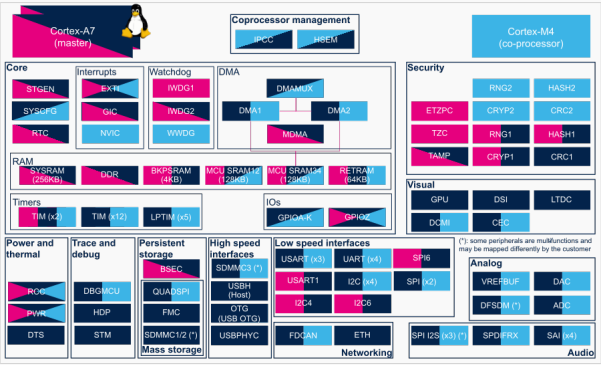
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (particularly for external peripherals) according to the information given in the corresponding software framework article or, for Linux in the Serial TTY device tree configuration 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)			
Low speed interface	USART	USART1				Assignment (single choice)
		USART2				Assignment (single choice)
		USART3				Assignment (single choice)
		UART4				Assignment (single choice). Used for Linux® serial console on ST boards.
		UART5				Assignment (single choice)
		USART6				Assignment (single choice)
		UART7				Assignment (single choice)
		UART8				Assignment (single choice)

4 How to go further

Additional documentation on USART peripheral is available on st.com:

- STM32 USART training [1] presents the STM32 Universal Synchronous/Asynchronous Receiver/Transmitter interface.
- STM32 USART automatic baud rate detection [2] presents STM32 USART automatic baud rate detection.

5 References

- Please refer to **stm32f7_peripheral_usart** document on st.com
- STM32 USART automatic baud rate detection application note (AN4908)

Universal Synchronous/Asynchronous Receiver/Transmitter

Microprocessor Unit

Serial Peripheral Interface

Universal Asynchronous Receiver/Transmitter

Open Portable Trusted Execution Environment

Subcategories

This category has the following 2 subcategories, out of 2 total.

S

- STM32MP15 Discovery kits (6 P)
- STM32MP15 Evaluation boards (9 P)

Pages in category "Getting started with STM32MP1 boards"

The following 5 pages are in this category, out of 5 total.

L

- LEDs and buttons on STM32 MPU boards

S

- STM32MP1 Developer Package
- STM32MP1 Developer Package for Android
- STM32MP1 Distribution Package
- STM32MP1 Distribution Package for Android

Stable: 13.10.2020 - 08:29 / Revision: 13.10.2020 - 08:29

Contents

1 Article purpose 18
2 Peripheral overview 19
2.1 Features 19
2.2 Security support 19
3 Peripheral usage and associated software 20
3.1 Boot time 20
3.2 Runtime 20
3.2.1 Overview 20
3.2.2 Software frameworks 20
3.2.3 Peripheral configuration 20
3.2.4 Peripheral assignment 20
4 How to go further 22
5 References 23

1 Article purpose

The purpose of this article is to:

- briefly introduce the USART 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 needed, how to configure the USART peripheral.

2 Peripheral overview

The **USART** peripheral is used to interconnect STM32 MPU devices with other systems, typically via RS232 or RS485 protocols. In addition, the USART supports the **Synchronous** mode that can be used for smartcard interfacing or SPI master /slave operation.

The **UART** peripheral is similar to the USART but does not support the Synchronous mode.

High-speed data communications can be achieved by using the DMA internal peripheral for multibuffer configuration.

2.1 Features

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

2.2 Security support

USART1 is a **secure** instance (under ETZPC control).
The other UARTs and USARTs are **non-secure** instances.

3 Peripheral usage and associated software

3.1 Boot time

All USART (except USART1) and UART instances are boot devices that support serial boot for Flash programming with STM32CubeProgrammer.

3.2 Runtime

3.2.1 Overview

The STM32 MPU devices feature four USART instances (supporting both Asynchronous and Synchronous modes), and four UART instances (supporting only Asynchronous mode).

USART1 can be allocated to:

- the Arm® Cortex®-A7 secure core to be used under OP-TEE with the USART OP-TEE driver, typically to communicate with a smartcard.

All USART and UART instances can be allocated to:

- the Arm® Cortex®-A7 non-secure core to be used under Linux® with the tty framework. However, the Linux® kernel supports only the UART Asynchronous mode (Synchronous mode not supported).

or

- the Arm® Cortex®-M4 to be used with STM32Cube MPU Package with USART HAL driver. Both USART Synchronous and Asynchronous modes are supported by the STM32Cube MPU Package.

Chapter Peripheral assignment describes which peripheral instance can be assigned to which context.

3.2.2 Software frameworks

Domain	Peripheral	Software frameworks			Comment
Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Low speed interface	USART	USART OP-TEE driver	Linux serial /tty framework	STM32Cube USART driver	

3.2.3 Peripheral configuration

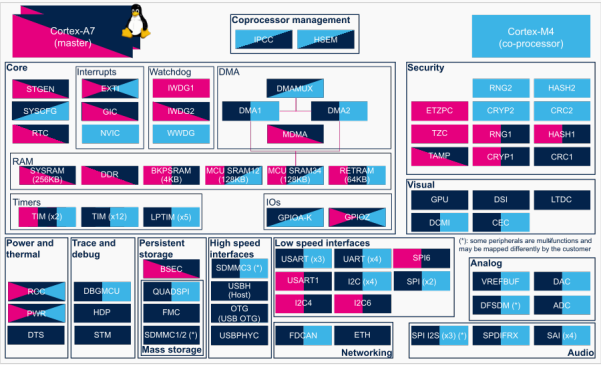
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (particularly for external peripherals) according to the information given in the corresponding software framework article or, for Linux in the Serial TTY device tree configuration 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)			
Low speed interface	USART	USART1				Assignment (single choice)
		USART2				Assignment (single choice)
		USART3				Assignment (single choice)
		UART4				Assignment (single choice). Used for Linux® serial console on ST boards.
		UART5				Assignment (single choice)
		USART6				Assignment (single choice)
		UART7				Assignment (single choice)
		UART8				Assignment (single choice)

4 How to go further

Additional documentation on USART peripheral is available on st.com:

- STM32 USART training [1] presents the STM32 Universal Synchronous/Asynchronous Receiver/Transmitter interface.
- STM32 USART automatic baud rate detection [2] presents STM32 USART automatic baud rate detection.

5 References

- Please refer to **stm32f7_peripheral_usart** document on st.com
- STM32 USART automatic baud rate detection application note (AN4908)

Universal Synchronous/Asynchronous Receiver/Transmitter

Microprocessor Unit

Serial Peripheral Interface

Universal Asynchronous Receiver/Transmitter

Open Portable Trusted Execution Environment

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

Contents

1 Article purpose 24
2 Peripheral overview 25
2.1 Features 25
2.2 Security support 25
3 Peripheral usage and associated software 26
3.1 Boot time 26
3.2 Runtime 26
3.2.1 Overview 26
3.2.2 Software frameworks 26
3.2.3 Peripheral configuration 26
3.2.4 Peripheral assignment 26
4 How to go further 28
5 References 29

1 Article purpose

The purpose of this article is to:

- briefly introduce the USART 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 needed, how to configure the USART peripheral.

2 Peripheral overview

The **USART** peripheral is used to interconnect STM32 MPU devices with other systems, typically via RS232 or RS485 protocols. In addition, the USART supports the **Synchronous** mode that can be used for smartcard interfacing or SPI master /slave operation.

The **UART** peripheral is similar to the USART but does not support the Synchronous mode.

High-speed data communications can be achieved by using the DMA internal peripheral for multibuffer configuration.

2.1 Features

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

2.2 Security support

USART1 is a **secure** instance (under ETZPC control).
The other UARTs and USARTs are **non-secure** instances.

3 Peripheral usage and associated software

3.1 Boot time

All USART (except USART1) and UART instances are boot devices that support serial boot for Flash programming with STM32CubeProgrammer.

3.2 Runtime

3.2.1 Overview

The STM32 MPU devices feature four USART instances (supporting both Asynchronous and Synchronous modes), and four UART instances (supporting only Asynchronous mode).

USART1 can be allocated to:

- the Arm® Cortex®-A7 secure core to be used under OP-TEE with the USART OP-TEE driver, typically to communicate with a smartcard.

All USART and UART instances can be allocated to:

- the Arm® Cortex®-A7 non-secure core to be used under Linux® with the tty framework. However, the Linux® kernel supports only the UART Asynchronous mode (Synchronous mode not supported).

or

- the Arm® Cortex®-M4 to be used with STM32Cube MPU Package with USART HAL driver. Both USART Synchronous and Asynchronous modes are supported by the STM32Cube MPU Package.

Chapter Peripheral assignment describes which peripheral instance can be assigned to which context.

3.2.2 Software frameworks

Domain	Peripheral	Software frameworks			Comment
Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Low speed interface	USART	USART OP-TEE driver	Linux serial /tty framework	STM32Cube USART driver	

3.2.3 Peripheral configuration

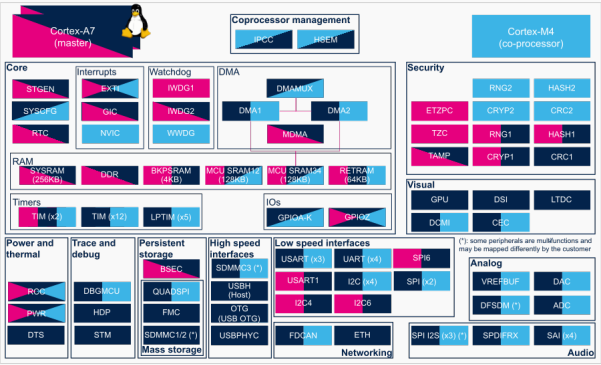
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (particularly for external peripherals) according to the information given in the corresponding software framework article or, for Linux in the Serial TTY device tree configuration 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)			
Low speed interface	USART	USART1				Assignment (single choice)
		USART2				Assignment (single choice)
		USART3				Assignment (single choice)
		UART4				Assignment (single choice). Used for Linux® serial console on ST boards.
		UART5				Assignment (single choice)
		USART6				Assignment (single choice)
		UART7				Assignment (single choice)
		UART8				Assignment (single choice)

4 How to go further

Additional documentation on USART peripheral is available on st.com:

- STM32 USART training [1] presents the STM32 Universal Synchronous/Asynchronous Receiver/Transmitter interface.
- STM32 USART automatic baud rate detection [2] presents STM32 USART automatic baud rate detection.

5 References

- Please refer to **stm32f7_peripheral_usart** document on st.com
- STM32 USART automatic baud rate detection application note (AN4908)

Universal Synchronous/Asynchronous Receiver/Transmitter

Microprocessor Unit

Serial Peripheral Interface

Universal Asynchronous Receiver/Transmitter

Open Portable Trusted Execution Environment

Stable: 16.02.2021 - 17:29 / Revision: 16.02.2021 - 17:11

Contents

1 Article purpose 30
2 Peripheral overview 31
2.1 Features 31
2.2 Security support 31
3 Peripheral usage and associated software 32
3.1 Boot time 32
3.2 Runtime 32
3.2.1 Overview 32
3.2.2 Software frameworks 32
3.2.3 Peripheral configuration 32
3.2.4 Peripheral assignment 32
4 How to go further 34
5 References 35

1 Article purpose

The purpose of this article is to:

- briefly introduce the USART 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 needed, how to configure the USART peripheral.

2 Peripheral overview

The **USART** peripheral is used to interconnect STM32 MPU devices with other systems, typically via RS232 or RS485 protocols. In addition, the USART supports the **Synchronous** mode that can be used for smartcard interfacing or SPI master /slave operation.

The **UART** peripheral is similar to the USART but does not support the Synchronous mode.

High-speed data communications can be achieved by using the DMA internal peripheral for multibuffer configuration.

2.1 Features

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

2.2 Security support

USART1 is a **secure** instance (under ETZPC control).
The other UARTs and USARTs are **non-secure** instances.

3 Peripheral usage and associated software

3.1 Boot time

All USART (except USART1) and UART instances are boot devices that support serial boot for Flash programming with STM32CubeProgrammer.

3.2 Runtime

3.2.1 Overview

The STM32 MPU devices feature four USART instances (supporting both Asynchronous and Synchronous modes), and four UART instances (supporting only Asynchronous mode).

USART1 can be allocated to:

- the Arm® Cortex®-A7 secure core to be used under OP-TEE with the USART OP-TEE driver, typically to communicate with a smartcard.

All USART and UART instances can be allocated to:

- the Arm® Cortex®-A7 non-secure core to be used under Linux® with the tty framework. However, the Linux® kernel supports only the UART Asynchronous mode (Synchronous mode not supported).

or

- the Arm® Cortex®-M4 to be used with STM32Cube MPU Package with USART HAL driver. Both USART Synchronous and Asynchronous modes are supported by the STM32Cube MPU Package.

Chapter Peripheral assignment describes which peripheral instance can be assigned to which context.

3.2.2 Software frameworks

Domain	Peripheral	Software frameworks			Comment
Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Low speed interface	USART	USART OP-TEE driver	Linux serial /tty framework	STM32Cube USART driver	

3.2.3 Peripheral configuration

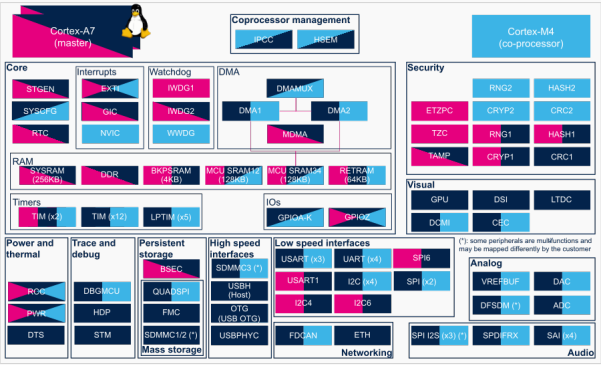
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (particularly for external peripherals) according to the information given in the corresponding software framework article or, for Linux in the Serial TTY device tree configuration 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)			
Low speed interface	USART	USART1				Assignment (single choice)
		USART2				Assignment (single choice)
		USART3				Assignment (single choice)
		UART4				Assignment (single choice). Used for Linux® serial console on ST boards.
		UART5				Assignment (single choice)
		USART6				Assignment (single choice)
		UART7				Assignment (single choice)
		UART8				Assignment (single choice)

4 How to go further

Additional documentation on USART peripheral is available on st.com:

- STM32 USART training [1] presents the STM32 Universal Synchronous/Asynchronous Receiver/Transmitter interface.
- STM32 USART automatic baud rate detection [2] presents STM32 USART automatic baud rate detection.

5 References

- Please refer to **stm32f7_peripheral_usart** document on st.com
- STM32 USART automatic baud rate detection application note (AN4908)

Universal Synchronous/Asynchronous Receiver/Transmitter

Microprocessor Unit

Serial Peripheral Interface

Universal Asynchronous Receiver/Transmitter

Open Portable Trusted Execution Environment

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

Contents

1 Article purpose 36
2 Peripheral overview 37
2.1 Features 37
2.2 Security support 37
3 Peripheral usage and associated software 38
3.1 Boot time 38
3.2 Runtime 38
3.2.1 Overview 38
3.2.2 Software frameworks 38
3.2.3 Peripheral configuration 38
3.2.4 Peripheral assignment 38
4 How to go further 40
5 References 41

1 Article purpose

The purpose of this article is to:

- briefly introduce the USART 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 needed, how to configure the USART peripheral.

2 Peripheral overview

The **USART** peripheral is used to interconnect STM32 MPU devices with other systems, typically via RS232 or RS485 protocols. In addition, the USART supports the **Synchronous** mode that can be used for smartcard interfacing or SPI master /slave operation.

The **UART** peripheral is similar to the USART but does not support the Synchronous mode.

High-speed data communications can be achieved by using the DMA internal peripheral for multibuffer configuration.

2.1 Features

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

2.2 Security support

USART1 is a **secure** instance (under ETZPC control).
The other UARTs and USARTs are **non-secure** instances.

3 Peripheral usage and associated software

3.1 Boot time

All USART (except USART1) and UART instances are boot devices that support serial boot for Flash programming with STM32CubeProgrammer.

3.2 Runtime

3.2.1 Overview

The STM32 MPU devices feature four USART instances (supporting both Asynchronous and Synchronous modes), and four UART instances (supporting only Asynchronous mode).

USART1 can be allocated to:

- the Arm® Cortex®-A7 secure core to be used under OP-TEE with the USART OP-TEE driver, typically to communicate with a smartcard.

All USART and UART instances can be allocated to:

- the Arm® Cortex®-A7 non-secure core to be used under Linux® with the tty framework. However, the Linux® kernel supports only the UART Asynchronous mode (Synchronous mode not supported).

or

- the Arm® Cortex®-M4 to be used with STM32Cube MPU Package with USART HAL driver. Both USART Synchronous and Asynchronous modes are supported by the STM32Cube MPU Package.

Chapter Peripheral assignment describes which peripheral instance can be assigned to which context.

3.2.2 Software frameworks

Domain	Peripheral	Software frameworks			Comment
Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Low speed interface	USART	USART OP-TEE driver	Linux serial /tty framework	STM32Cube USART driver	

3.2.3 Peripheral configuration

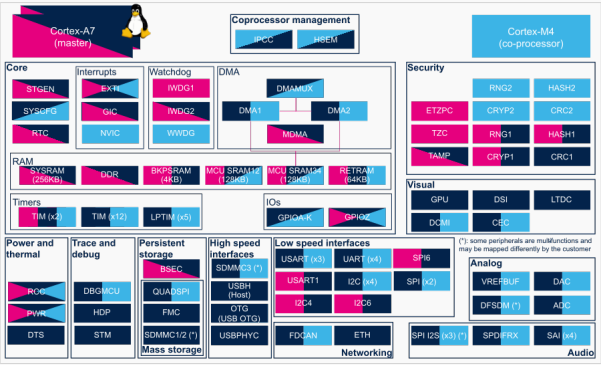
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (particularly for external peripherals) according to the information given in the corresponding software framework article or, for Linux in the Serial TTY device tree configuration 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)			
Low speed interface	USART	USART1				Assignment (single choice)
		USART2				Assignment (single choice)
		USART3				Assignment (single choice)
		UART4				Assignment (single choice). Used for Linux® serial console on ST boards.
		UART5				Assignment (single choice)
		USART6				Assignment (single choice)
		UART7				Assignment (single choice)
		UART8				Assignment (single choice)

4 How to go further

Additional documentation on USART peripheral is available on st.com:

- STM32 USART training [1] presents the STM32 Universal Synchronous/Asynchronous Receiver/Transmitter interface.
- STM32 USART automatic baud rate detection [2] presents STM32 USART automatic baud rate detection.

5 References

- Please refer to **stm32f7_peripheral_usart** document on st.com
- STM32 USART automatic baud rate detection application note (AN4908)

Universal Synchronous/Asynchronous Receiver/Transmitter

Microprocessor Unit

Serial Peripheral Interface

Universal Asynchronous Receiver/Transmitter

Open Portable Trusted Execution Environment

Stable: 17.11.2020 - 15:37 / Revision: 03.11.2020 - 13:18

Contents

1 Article purpose 42
2 Peripheral overview 43
2.1 Features 43
2.2 Security support 43
3 Peripheral usage and associated software 44
3.1 Boot time 44
3.2 Runtime 44
3.2.1 Overview 44
3.2.2 Software frameworks 44
3.2.3 Peripheral configuration 44
3.2.4 Peripheral assignment 44
4 How to go further 46
5 References 47

1 Article purpose

The purpose of this article is to:

- briefly introduce the USART 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 needed, how to configure the USART peripheral.

2 Peripheral overview

The **USART** peripheral is used to interconnect STM32 MPU devices with other systems, typically via RS232 or RS485 protocols. In addition, the USART supports the **Synchronous** mode that can be used for smartcard interfacing or SPI master /slave operation.

The **UART** peripheral is similar to the USART but does not support the Synchronous mode.

High-speed data communications can be achieved by using the DMA internal peripheral for multibuffer configuration.

2.1 Features

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

2.2 Security support

USART1 is a **secure** instance (under ETZPC control).
The other UARTs and USARTs are **non-secure** instances.

3 Peripheral usage and associated software

3.1 Boot time

All USART (except USART1) and UART instances are boot devices that support serial boot for Flash programming with STM32CubeProgrammer.

3.2 Runtime

3.2.1 Overview

The STM32 MPU devices feature four USART instances (supporting both Asynchronous and Synchronous modes), and four UART instances (supporting only Asynchronous mode).

USART1 can be allocated to:

- the Arm® Cortex®-A7 secure core to be used under OP-TEE with the USART OP-TEE driver, typically to communicate with a smartcard.

All USART and UART instances can be allocated to:

- the Arm® Cortex®-A7 non-secure core to be used under Linux® with the tty framework. However, the Linux® kernel supports only the UART Asynchronous mode (Synchronous mode not supported).

or

- the Arm® Cortex®-M4 to be used with STM32Cube MPU Package with USART HAL driver. Both USART Synchronous and Asynchronous modes are supported by the STM32Cube MPU Package.

Chapter Peripheral assignment describes which peripheral instance can be assigned to which context.

3.2.2 Software frameworks

Domain	Peripheral	Software frameworks			Comment
Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Low speed interface	USART	USART OP-TEE driver	Linux serial /tty framework	STM32Cube USART driver	

3.2.3 Peripheral configuration

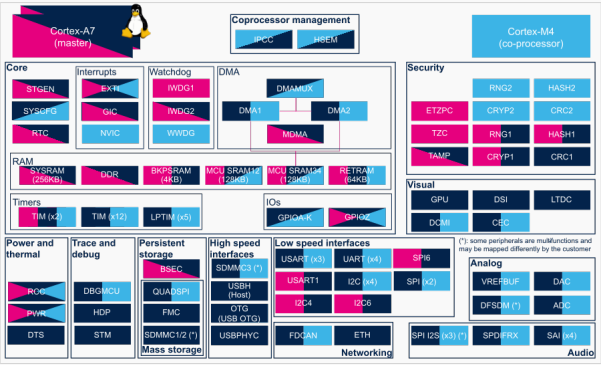
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (particularly for external peripherals) according to the information given in the corresponding software framework article or, for Linux in the Serial TTY device tree configuration 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)			
Low speed interface	USART	USART1				Assignment (single choice)
		USART2				Assignment (single choice)
		USART3				Assignment (single choice)
		UART4				Assignment (single choice). Used for Linux® serial console on ST boards.
		UART5				Assignment (single choice)
		USART6				Assignment (single choice)
		UART7				Assignment (single choice)
		UART8				Assignment (single choice)

4 How to go further

Additional documentation on USART peripheral is available on st.com:

- STM32 USART training [1] presents the STM32 Universal Synchronous/Asynchronous Receiver/Transmitter interface.
- STM32 USART automatic baud rate detection [2] presents STM32 USART automatic baud rate detection.

5 References

- Please refer to **stm32f7_peripheral_usart** document on st.com
- STM32 USART automatic baud rate detection application note (AN4908)

Universal Synchronous/Asynchronous Receiver/Transmitter

Microprocessor Unit

Serial Peripheral Interface

Universal Asynchronous Receiver/Transmitter

Open Portable Trusted Execution Environment

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

Contents

1 Article purpose 48
2 Peripheral overview 49
2.1 Features 49
2.2 Security support 49
3 Peripheral usage and associated software 50
3.1 Boot time 50
3.2 Runtime 50
3.2.1 Overview 50
3.2.2 Software frameworks 50
3.2.3 Peripheral configuration 50
3.2.4 Peripheral assignment 50
4 How to go further 52
5 References 53

1 Article purpose

The purpose of this article is to:

- briefly introduce the USART 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 needed, how to configure the USART peripheral.

2 Peripheral overview

The **USART** peripheral is used to interconnect STM32 MPU devices with other systems, typically via RS232 or RS485 protocols. In addition, the USART supports the **Synchronous** mode that can be used for smartcard interfacing or SPI master /slave operation.

The **UART** peripheral is similar to the USART but does not support the Synchronous mode.

High-speed data communications can be achieved by using the DMA internal peripheral for multibuffer configuration.

2.1 Features

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

2.2 Security support

USART1 is a **secure** instance (under ETZPC control).
The other UARTs and USARTs are **non-secure** instances.

3 Peripheral usage and associated software

3.1 Boot time

All USART (except USART1) and UART instances are boot devices that support serial boot for Flash programming with STM32CubeProgrammer.

3.2 Runtime

3.2.1 Overview

The STM32 MPU devices feature four USART instances (supporting both Asynchronous and Synchronous modes), and four UART instances (supporting only Asynchronous mode).

USART1 can be allocated to:

- the Arm® Cortex®-A7 secure core to be used under OP-TEE with the USART OP-TEE driver, typically to communicate with a smartcard.

All USART and UART instances can be allocated to:

- the Arm® Cortex®-A7 non-secure core to be used under Linux® with the tty framework. However, the Linux® kernel supports only the UART Asynchronous mode (Synchronous mode not supported).

or

- the Arm® Cortex®-M4 to be used with STM32Cube MPU Package with USART HAL driver. Both USART Synchronous and Asynchronous modes are supported by the STM32Cube MPU Package.

Chapter Peripheral assignment describes which peripheral instance can be assigned to which context.

3.2.2 Software frameworks

Domain	Peripheral	Software frameworks			Comment
Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Low speed interface	USART	USART OP-TEE driver	Linux serial /tty framework	STM32Cube USART driver	

3.2.3 Peripheral configuration

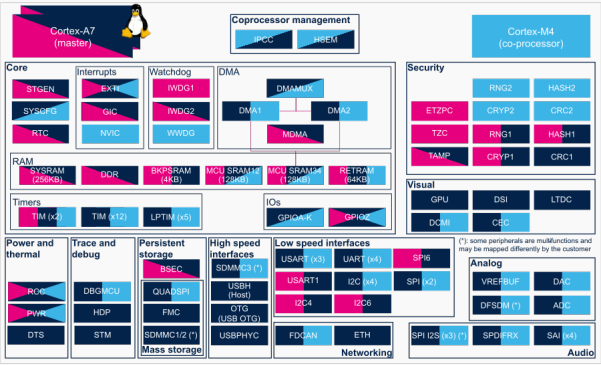
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (particularly for external peripherals) according to the information given in the corresponding software framework article or, for Linux in the Serial TTY device tree configuration 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)			
Low speed interface	USART	USART1				Assignment (single choice)
		USART2				Assignment (single choice)
		USART3				Assignment (single choice)
		UART4				Assignment (single choice). Used for Linux® serial console on ST boards.
		UART5				Assignment (single choice)
		USART6				Assignment (single choice)
		UART7				Assignment (single choice)
		UART8				Assignment (single choice)

4 How to go further

Additional documentation on USART peripheral is available on st.com:

- STM32 USART training [1] presents the STM32 Universal Synchronous/Asynchronous Receiver/Transmitter interface.
- STM32 USART automatic baud rate detection [2] presents STM32 USART automatic baud rate detection.

5 References

- Please refer to **stm32f7_peripheral_usart** document on st.com
- STM32 USART automatic baud rate detection application note (AN4908)

Universal Synchronous/Asynchronous Receiver/Transmitter

Microprocessor Unit

Serial Peripheral Interface

Universal Asynchronous Receiver/Transmitter

Open Portable Trusted Execution Environment

Stable: 20.11.2020 - 09:52 / Revision: 22.10.2020 - 12:46

Contents

1 Article purpose 54
2 Peripheral overview 55
2.1 Features 55
2.2 Security support 55
3 Peripheral usage and associated software 56
3.1 Boot time 56
3.2 Runtime 56
3.2.1 Overview 56
3.2.2 Software frameworks 56
3.2.3 Peripheral configuration 56
3.2.4 Peripheral assignment 56
4 How to go further 58
5 References 59

1 Article purpose

The purpose of this article is to:

- briefly introduce the USART 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 needed, how to configure the USART peripheral.

2 Peripheral overview

The **USART** peripheral is used to interconnect STM32 MPU devices with other systems, typically via RS232 or RS485 protocols. In addition, the USART supports the **Synchronous** mode that can be used for smartcard interfacing or SPI master /slave operation.

The **UART** peripheral is similar to the USART but does not support the Synchronous mode.

High-speed data communications can be achieved by using the DMA internal peripheral for multibuffer configuration.

2.1 Features

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

2.2 Security support

USART1 is a **secure** instance (under ETZPC control).
The other UARTs and USARTs are **non-secure** instances.

3 Peripheral usage and associated software

3.1 Boot time

All USART (except USART1) and UART instances are boot devices that support serial boot for Flash programming with STM32CubeProgrammer.

3.2 Runtime

3.2.1 Overview

The STM32 MPU devices feature four USART instances (supporting both Asynchronous and Synchronous modes), and four UART instances (supporting only Asynchronous mode).

USART1 can be allocated to:

- the Arm® Cortex®-A7 secure core to be used under OP-TEE with the USART OP-TEE driver, typically to communicate with a smartcard.

All USART and UART instances can be allocated to:

- the Arm® Cortex®-A7 non-secure core to be used under Linux® with the tty framework. However, the Linux® kernel supports only the UART Asynchronous mode (Synchronous mode not supported).

or

- the Arm® Cortex®-M4 to be used with STM32Cube MPU Package with USART HAL driver. Both USART Synchronous and Asynchronous modes are supported by the STM32Cube MPU Package.

Chapter Peripheral assignment describes which peripheral instance can be assigned to which context.

3.2.2 Software frameworks

Domain	Peripheral	Software frameworks			Comment
Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Low speed interface	USART	USART OP-TEE driver	Linux serial /tty framework	STM32Cube USART driver	

3.2.3 Peripheral configuration

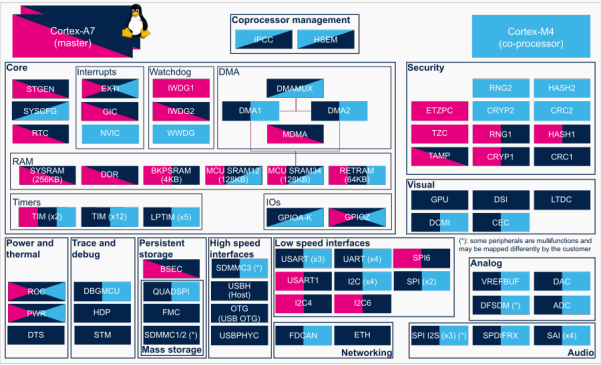
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (particularly for external peripherals) according to the information given in the corresponding software framework article or, for Linux in the Serial TTY device tree configuration 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)			
Low speed interface	USART	USART1				Assignment (single choice)
		USART2				Assignment (single choice)
		USART3				Assignment (single choice)
		UART4				Assignment (single choice). Used for Linux® serial console on ST boards.
		UART5				Assignment (single choice)
		USART6				Assignment (single choice)
		UART7				Assignment (single choice)
		UART8				Assignment (single choice)

4 How to go further

Additional documentation on USART peripheral is available on st.com:

- STM32 USART training [1] presents the STM32 Universal Synchronous/Asynchronous Receiver/Transmitter interface.
- STM32 USART automatic baud rate detection [2] presents STM32 USART automatic baud rate detection.

5 References

- Please refer to **stm32f7_peripheral_usart** document on st.com
- STM32 USART automatic baud rate detection application note (AN4908)

Universal Synchronous/Asynchronous Receiver/Transmitter

Microprocessor Unit

Serial Peripheral Interface

Universal Asynchronous Receiver/Transmitter

Open Portable Trusted Execution Environment

Stable: 17.11.2020 - 17:06 / Revision: 10.11.2020 - 07:49

Contents

1 Article purpose 60
2 Peripheral overview 61
2.1 Features 61
2.2 Security support 61
3 Peripheral usage and associated software 62
3.1 Boot time 62
3.2 Runtime 62
3.2.1 Overview 62
3.2.2 Software frameworks 62
3.2.3 Peripheral configuration 62
3.2.4 Peripheral assignment 62
4 How to go further 64
5 References 65

1 Article purpose

The purpose of this article is to:

- briefly introduce the USART 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 needed, how to configure the USART peripheral.

2 Peripheral overview

The **USART** peripheral is used to interconnect STM32 MPU devices with other systems, typically via RS232 or RS485 protocols. In addition, the USART supports the **Synchronous** mode that can be used for smartcard interfacing or SPI master /slave operation.

The **UART** peripheral is similar to the USART but does not support the Synchronous mode.

High-speed data communications can be achieved by using the DMA internal peripheral for multibuffer configuration.

2.1 Features

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

2.2 Security support

USART1 is a **secure** instance (under ETZPC control).
The other UARTs and USARTs are **non-secure** instances.

3 Peripheral usage and associated software

3.1 Boot time

All USART (except USART1) and UART instances are boot devices that support serial boot for Flash programming with STM32CubeProgrammer.

3.2 Runtime

3.2.1 Overview

The STM32 MPU devices feature four USART instances (supporting both Asynchronous and Synchronous modes), and four UART instances (supporting only Asynchronous mode).

USART1 can be allocated to:

- the Arm® Cortex®-A7 secure core to be used under OP-TEE with the USART OP-TEE driver, typically to communicate with a smartcard.

All USART and UART instances can be allocated to:

- the Arm® Cortex®-A7 non-secure core to be used under Linux® with the tty framework. However, the Linux® kernel supports only the UART Asynchronous mode (Synchronous mode not supported).

or

- the Arm® Cortex®-M4 to be used with STM32Cube MPU Package with USART HAL driver. Both USART Synchronous and Asynchronous modes are supported by the STM32Cube MPU Package.

Chapter Peripheral assignment describes which peripheral instance can be assigned to which context.

3.2.2 Software frameworks

Domain	Peripheral	Software frameworks			Comment
Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Low speed interface	USART	USART OP-TEE driver	Linux serial /tty framework	STM32Cube USART driver	

3.2.3 Peripheral configuration

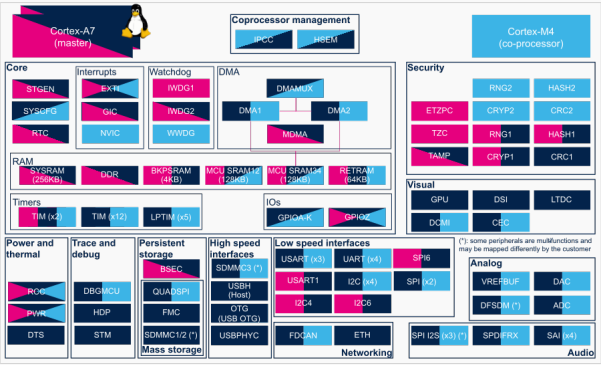
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (particularly for external peripherals) according to the information given in the corresponding software framework article or, for Linux in the Serial TTY device tree configuration 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)			
Low speed interface	USART	USART1				Assignment (single choice)
		USART2				Assignment (single choice)
		USART3				Assignment (single choice)
		UART4				Assignment (single choice). Used for Linux® serial console on ST boards.
		UART5				Assignment (single choice)
		USART6				Assignment (single choice)
		UART7				Assignment (single choice)
		UART8				Assignment (single choice)

4 How to go further

Additional documentation on USART peripheral is available on st.com:

- STM32 USART training [1] presents the STM32 Universal Synchronous/Asynchronous Receiver/Transmitter interface.
- STM32 USART automatic baud rate detection [2] presents STM32 USART automatic baud rate detection.

5 References

- Please refer to **stm32f7_peripheral_usart** document on st.com
- STM32 USART automatic baud rate detection application note (AN4908)

Universal Synchronous/Asynchronous Receiver/Transmitter

Microprocessor Unit

Serial Peripheral Interface

Universal Asynchronous Receiver/Transmitter

Open Portable Trusted Execution Environment

Stable: 25.09.2020 - 09:15 / Revision: 25.09.2020 - 09:13

Contents

1 Article purpose 66
2 Peripheral overview 67
2.1 Features 67
2.2 Security support 67
3 Peripheral usage and associated software 68
3.1 Boot time 68
3.2 Runtime 68
3.2.1 Overview 68
3.2.2 Software frameworks 68
3.2.3 Peripheral configuration 68
3.2.4 Peripheral assignment 68
4 How to go further 70
5 References 71

1 Article purpose

The purpose of this article is to:

- briefly introduce the USART 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 needed, how to configure the USART peripheral.

2 Peripheral overview

The **USART** peripheral is used to interconnect STM32 MPU devices with other systems, typically via RS232 or RS485 protocols. In addition, the USART supports the **Synchronous** mode that can be used for smartcard interfacing or SPI master /slave operation.

The **UART** peripheral is similar to the USART but does not support the Synchronous mode.

High-speed data communications can be achieved by using the DMA internal peripheral for multibuffer configuration.

2.1 Features

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

2.2 Security support

USART1 is a **secure** instance (under ETZPC control).
The other UARTs and USARTs are **non-secure** instances.

3 Peripheral usage and associated software

3.1 Boot time

All USART (except USART1) and UART instances are boot devices that support serial boot for Flash programming with STM32CubeProgrammer.

3.2 Runtime

3.2.1 Overview

The STM32 MPU devices feature four USART instances (supporting both Asynchronous and Synchronous modes), and four UART instances (supporting only Asynchronous mode).

USART1 can be allocated to:

- the Arm® Cortex®-A7 secure core to be used under OP-TEE with the USART OP-TEE driver, typically to communicate with a smartcard.

All USART and UART instances can be allocated to:

- the Arm® Cortex®-A7 non-secure core to be used under Linux® with the tty framework. However, the Linux® kernel supports only the UART Asynchronous mode (Synchronous mode not supported).

or

- the Arm® Cortex®-M4 to be used with STM32Cube MPU Package with USART HAL driver. Both USART Synchronous and Asynchronous modes are supported by the STM32Cube MPU Package.

Chapter Peripheral assignment describes which peripheral instance can be assigned to which context.

3.2.2 Software frameworks

Domain	Peripheral	Software frameworks			Comment
Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Low speed interface	USART	USART OP-TEE driver	Linux serial /tty framework	STM32Cube USART driver	

3.2.3 Peripheral configuration

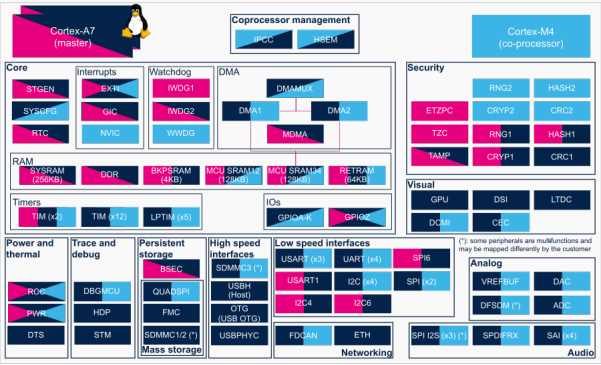
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (particularly for external peripherals) according to the information given in the corresponding software framework article or, for Linux in the Serial TTY device tree configuration 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)			
Low speed interface	USART	USART1				Assignment (single choice)
		USART2				Assignment (single choice)
		USART3				Assignment (single choice)
		UART4				Assignment (single choice). Used for Linux® serial console on ST boards.
		UART5				Assignment (single choice)
		USART6				Assignment (single choice)
		UART7				Assignment (single choice)
		UART8				Assignment (single choice)

4 How to go further

Additional documentation on USART peripheral is available on st.com:

- STM32 USART training [1] presents the STM32 Universal Synchronous/Asynchronous Receiver/Transmitter interface.
- STM32 USART automatic baud rate detection [2] presents STM32 USART automatic baud rate detection.

5 References

- Please refer to **stm32f7_peripheral_usart** document on st.com
- STM32 USART automatic baud rate detection application note (AN4908)

Universal Synchronous/Asynchronous Receiver/Transmitter

Microprocessor Unit

Serial Peripheral Interface

Universal Asynchronous Receiver/Transmitter

Open Portable Trusted Execution Environment

Stable: 02.11.2020 - 15:54 / Revision: 03.09.2020 - 13:39

Contents

1 Article purpose 72
2 Peripheral overview 73
2.1 Features 73
2.2 Security support 73
3 Peripheral usage and associated software 74
3.1 Boot time 74
3.2 Runtime 74
3.2.1 Overview 74
3.2.2 Software frameworks 74
3.2.3 Peripheral configuration 74
3.2.4 Peripheral assignment 74
4 How to go further 76
5 References 77

1 Article purpose

The purpose of this article is to:

- briefly introduce the USART 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 needed, how to configure the USART peripheral.

2 Peripheral overview

The **USART** peripheral is used to interconnect STM32 MPU devices with other systems, typically via RS232 or RS485 protocols. In addition, the USART supports the **Synchronous** mode that can be used for smartcard interfacing or SPI master /slave operation.

The **UART** peripheral is similar to the USART but does not support the Synchronous mode.

High-speed data communications can be achieved by using the DMA internal peripheral for multibuffer configuration.

2.1 Features

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

2.2 Security support

USART1 is a **secure** instance (under ETZPC control).
The other UARTs and USARTs are **non-secure** instances.

3 Peripheral usage and associated software

3.1 Boot time

All USART (except USART1) and UART instances are boot devices that support serial boot for Flash programming with STM32CubeProgrammer.

3.2 Runtime

3.2.1 Overview

The STM32 MPU devices feature four USART instances (supporting both Asynchronous and Synchronous modes), and four UART instances (supporting only Asynchronous mode).

USART1 can be allocated to:

- the Arm® Cortex®-A7 secure core to be used under OP-TEE with the USART OP-TEE driver, typically to communicate with a smartcard.

All USART and UART instances can be allocated to:

- the Arm® Cortex®-A7 non-secure core to be used under Linux® with the tty framework. However, the Linux® kernel supports only the UART Asynchronous mode (Synchronous mode not supported).

or

- the Arm® Cortex®-M4 to be used with STM32Cube MPU Package with USART HAL driver. Both USART Synchronous and Asynchronous modes are supported by the STM32Cube MPU Package.

Chapter Peripheral assignment describes which peripheral instance can be assigned to which context.

3.2.2 Software frameworks

Domain	Peripheral	Software frameworks			Comment
Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Low speed interface	USART	USART OP-TEE driver	Linux serial /tty framework	STM32Cube USART driver	

3.2.3 Peripheral configuration

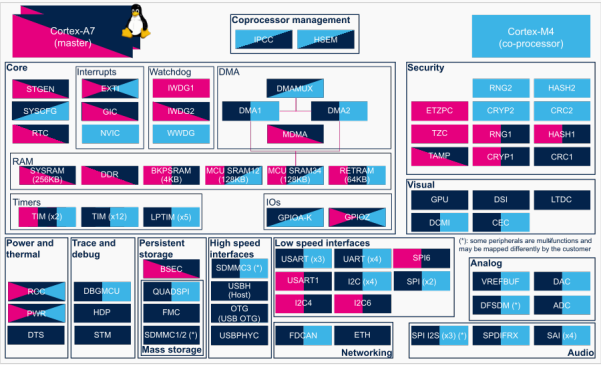
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (particularly for external peripherals) according to the information given in the corresponding software framework article or, for Linux in the Serial TTY device tree configuration 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)			
Low speed interface	USART	USART1				Assignment (single choice)
		USART2				Assignment (single choice)
		USART3				Assignment (single choice)
		UART4				Assignment (single choice). Used for Linux® serial console on ST boards.
		UART5				Assignment (single choice)
		USART6				Assignment (single choice)
		UART7				Assignment (single choice)
		UART8				Assignment (single choice)

4 How to go further

Additional documentation on USART peripheral is available on st.com:

- STM32 USART training [1] presents the STM32 Universal Synchronous/Asynchronous Receiver/Transmitter interface.
- STM32 USART automatic baud rate detection [2] presents STM32 USART automatic baud rate detection.

5 References

- Please refer to **stm32f7_peripheral_usart** document on st.com
- STM32 USART automatic baud rate detection application note (AN4908)

Universal Synchronous/Asynchronous Receiver/Transmitter

Microprocessor Unit

Serial Peripheral Interface

Universal Asynchronous Receiver/Transmitter

Open Portable Trusted Execution Environment

Stable: 17.03.2020 - 14:44 / Revision: 25.02.2020 - 14:49

Contents

1 Article purpose 78
2 Peripheral overview 79
2.1 Features 79
2.2 Security support 79
3 Peripheral usage and associated software 80
3.1 Boot time 80
3.2 Runtime 80
3.2.1 Overview 80
3.2.2 Software frameworks 80
3.2.3 Peripheral configuration 80
3.2.4 Peripheral assignment 80
4 How to go further 82
5 References 83

1 Article purpose

The purpose of this article is to:

- briefly introduce the USART 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 needed, how to configure the USART peripheral.

2 Peripheral overview

The **USART** peripheral is used to interconnect STM32 MPU devices with other systems, typically via RS232 or RS485 protocols. In addition, the USART supports the **Synchronous** mode that can be used for smartcard interfacing or SPI master /slave operation.

The **UART** peripheral is similar to the USART but does not support the Synchronous mode.

High-speed data communications can be achieved by using the DMA internal peripheral for multibuffer configuration.

2.1 Features

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

2.2 Security support

USART1 is a **secure** instance (under ETZPC control).
The other UARTs and USARTs are **non-secure** instances.

3 Peripheral usage and associated software

3.1 Boot time

All USART (except USART1) and UART instances are boot devices that support serial boot for Flash programming with STM32CubeProgrammer.

3.2 Runtime

3.2.1 Overview

The STM32 MPU devices feature four USART instances (supporting both Asynchronous and Synchronous modes), and four UART instances (supporting only Asynchronous mode).

USART1 can be allocated to:

- the Arm® Cortex®-A7 secure core to be used under OP-TEE with the USART OP-TEE driver, typically to communicate with a smartcard.

All USART and UART instances can be allocated to:

- the Arm® Cortex®-A7 non-secure core to be used under Linux® with the tty framework. However, the Linux® kernel supports only the UART Asynchronous mode (Synchronous mode not supported).

or

- the Arm® Cortex®-M4 to be used with STM32Cube MPU Package with USART HAL driver. Both USART Synchronous and Asynchronous modes are supported by the STM32Cube MPU Package.

Chapter Peripheral assignment describes which peripheral instance can be assigned to which context.

3.2.2 Software frameworks

Domain	Peripheral	Software frameworks			Comment
Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Low speed interface	USART	USART OP-TEE driver	Linux serial /tty framework	STM32Cube USART driver	

3.2.3 Peripheral configuration

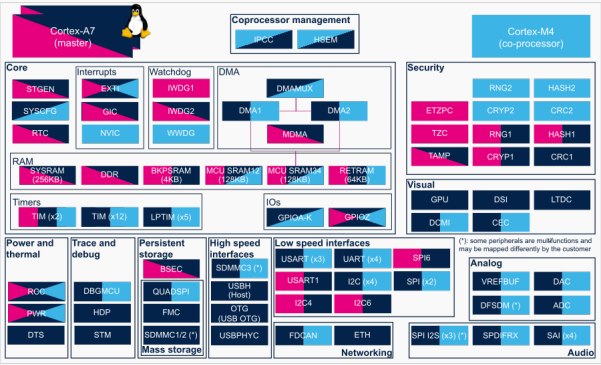
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (particularly for external peripherals) according to the information given in the corresponding software framework article or, for Linux in the Serial TTY device tree configuration 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)			
Low speed interface	USART	USART1				Assignment (single choice)
		USART2				Assignment (single choice)
		USART3				Assignment (single choice)
		UART4				Assignment (single choice). Used for Linux® serial console on ST boards.
		UART5				Assignment (single choice)
		USART6				Assignment (single choice)
		UART7				Assignment (single choice)
		UART8				Assignment (single choice)

4 How to go further

Additional documentation on USART peripheral is available on st.com:

- STM32 USART training [1] presents the STM32 Universal Synchronous/Asynchronous Receiver/Transmitter interface.
- STM32 USART automatic baud rate detection [2] presents STM32 USART automatic baud rate detection.

5 References

- Please refer to **stm32f7_peripheral_usart** document on st.com
- STM32 USART automatic baud rate detection application note (AN4908)

Universal Synchronous/Asynchronous Receiver/Transmitter

Microprocessor Unit

Serial Peripheral Interface

Universal Asynchronous Receiver/Transmitter

Open Portable Trusted Execution Environment