



USART internal peripheral



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

Contents

1 Article purpose	3
2 Peripheral overview	4
2.1 Features	4
2.2 Security support	4
3 Peripheral usage and associated software	5
3.1 Boot time	5
3.2 Runtime	5
3.2.1 Overview	5
3.2.2 Software frameworks	5
3.2.3 Peripheral configuration	5
3.2.4 Peripheral assignment	5
4 How to go further	7
5 References	8



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

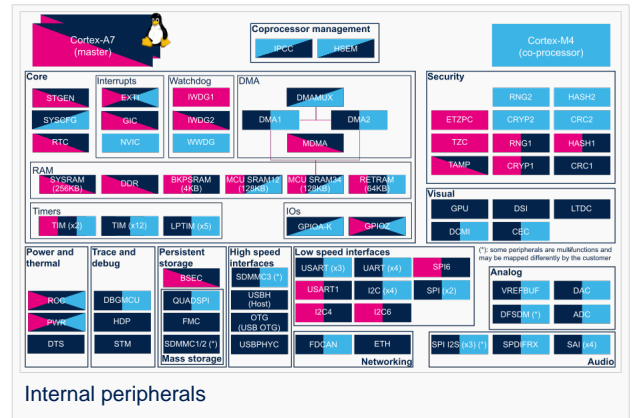


Check boxes illustrate the possible peripheral allocations supported by STM32 MPU Embedded Software:

- 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 possibilities might be described in STM32MP15 reference manuals



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