Contents

QUADSPI internal peripheral

Contents

1 Article purpose

The purpose of this article is to

- briefly introduce the QUADSPI 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 QUADSPI peripheral.

2 Peripheral overview

The Quad-SPI interface (**QUADSPI** peripheral) is used to interface the processor with serial NOR Flash and serial NAND Flash memories.

It supports:

- Single, dual or quad SPI Flash memories
- A dual-flash mode, allowing to agregate two Flash memories into a virtual single one
- Dual data rate and memory-mapped modes.

2.1 Features

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

2.2 Security support

QUADSPI is **non secure** peripheral.

3 Using the peripheral - associated software

3.1 Boot time

QUADSPI instance is boot device that support serial boot for Flash programming with STM32CubeProgrammer.

3.2 Runtime

3.2.1 Overview

The QUADSPI instances can be allocated to:

- the Arm® Cortex®-A7 non-secure core to be controlled in Linux® by the MTD framework

or

- the Arm® Cortex®-M4 to be controlled in STM32Cube MPU Package by QUADSPI HAL driver

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

3.2.2 Software frameworks

<table>
<tr><th>Dom</th><th>Peri</th><th colspan="3">Software frameworks</th><th>Comment</th></tr>
<tr><td>ain
Cortex-A7 secure (OP-TEE)</td><td>Cortex-A7 non-secure (Linux)</td><td>Cortex-M4

(STM32Cube)</td><td colspan="3"></td></tr>
<tr><td>Mass storage</td><td>QUADSPI</td><td></td><td>Linux MTD framework</td><td>STM32Cube QUADSPI driver</td><td></td></tr>
</table>

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.

For Linux kernel configuration, please refer to QUADSPI device tree configuration.

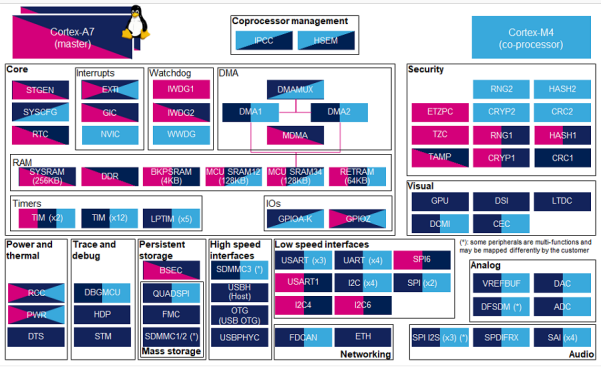
3.2.4 Peripheral assignment

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)			
Mass storage	QUADSPI	QUADSPI		☐	☐	Assignment (single choice)

4 How to go further

5 References

How to assign an internal peripheral to a runtime context

Invalid target: no reviewed revision corresponds to the given ID.

Return to How to assign an internal peripheral to a runtime context.

MTD overview

Invalid target: no reviewed revision corresponds to the given ID.

Return to MTD overview.

QUADSPI device tree configuration

Invalid target: no reviewed revision corresponds to the given ID.

Return to QUADSPI device tree configuration.

STM32CubeMP1 architecture

Invalid target: no reviewed revision corresponds to the given ID.

Return to STM32CubeMP1 architecture.

STM32CubeMX

Invalid target: no reviewed revision corresponds to the given ID.

Return to STM32CubeMX.

STM32CubeProgrammer

Invalid target: no reviewed revision corresponds to the given ID.

Return to STM32CubeProgrammer.

STM32MP15 resources

Invalid target: no reviewed revision corresponds to the given ID.

Return to STM32MP15 resources.

STM32MPU Embedded Software architecture overview

Invalid target: no reviewed revision corresponds to the given ID.

Return to STM32MPU Embedded Software architecture overview.