

## Contents

1. QUADSPI internal peripheral .....	2
2. How to assign an internal peripheral to a runtime context .....	9
3. MTD overview .....	10
4. QUADSPI device tree configuration .....	11
5. STM32CubeMP1 architecture .....	12
6. STM32CubeMX .....	13
7. STM32CubeProgrammer .....	14
8. STM32MP15 resources .....	15
9. STM32MPU Embedded Software architecture overview .....	16

## QUADSPI internal peripheral

**Contents**

<b>1</b>	<b>Article purpose</b> .....	<b>3</b>
<b>2</b>	<b>Peripheral overview</b> .....	<b>4</b>
2.1	Features .....	4
2.2	Security support .....	4
<b>3</b>	<b>Using the peripheral - associated software</b> .....	<b>5</b>
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
<b>4</b>	<b>How to go further</b> .....	<b>7</b>
<b>5</b>	<b>References</b> .....	<b>8</b>

## 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 aggregate 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

Dom	Peri	Software frameworks		Comment
Main Cortex-A7 non-secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)		
		Linux MTD framework	STM32Cube QUADSPI driver	
Mass storage	QUADSPI			

##### 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](#).

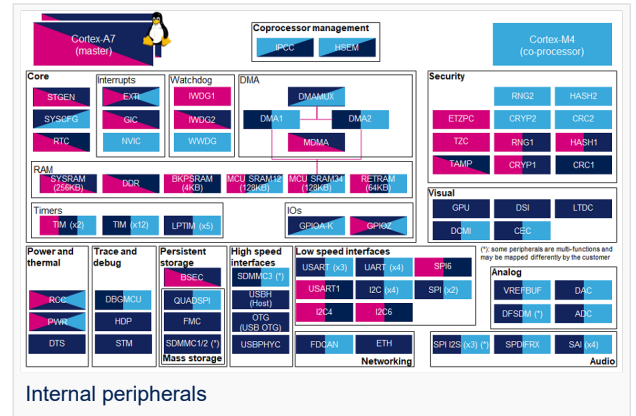
##### 3.2.4 Peripheral assignment

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 possibilities might be described in [STM32MP15 reference manuals](#).



Do	Per	Runtime allocation			Commen
mai n	igh Cort era ex- A7	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)		
Inst anc e	sec ure (OP - TEE )				
Ma ss stor age	QU AD SPI	QUADSPI		<input type="checkbox"/>	Assignme nt (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.](#)



## IMPORTANT NOTICE – READ CAREFULLY

STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST's terms and conditions of sale in place at the time of order acknowledgment.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of purchasers' products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, refer to [www.st.com/trademarks](http://www.st.com/trademarks). All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2023 STMicroelectronics – All rights reserved