CRC internal peripheral

Stable: 24.09.2019 - 15:57 / Revision: 24.09.2019 - 15:56

Contents

1 Article purpose

The purpose of this article is to

- briefly introduce the CRC 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 how to configure the CRC peripheral.

2 Peripheral overview

The **CRC** peripheral is used to verify data transmission or storage integrity.

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 implemented.

2.2 Security support

CRC1 and CRC2 are **non secure** peripherals.

3 Peripheral usage and associated software

3.1 Boot time

CRC instances are not used at boot time.

3.2 Runtime

3.2.1 Overview

CRC instances can be allocated to:

- the Arm® Cortex®-A7 non-secure for using in Linux® with Linux Crypto framework

or

- the Arm® Cortex®-M4 for using in STM32Cube with STM32Cube CRC driver

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)	
Security	CRC		Linux Crypto framework	STM32Cube CRC driver	

3.2.3 Peripheral configuration

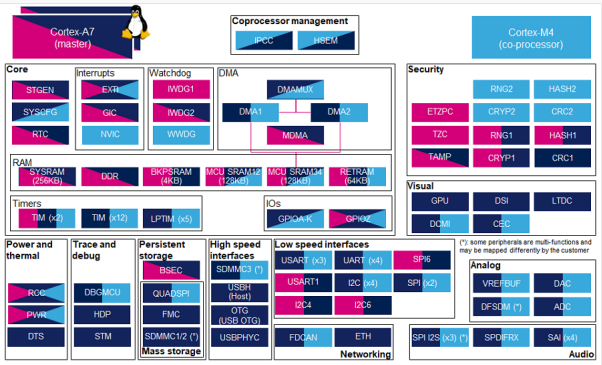
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (particularly for external peripherals), according to the information given in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

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)	
Security	CRC	CRC1		☐		
		CRC2			☐	

4 How to go further

Not applicable.

5 References

Cyclic redundancy check calculation unit

Open Portable Trusted Execution Environment