



CRC internal peripheral



## Contents

---

1. CRC internal peripheral .....	3
2. STM32MP15 resources .....	6
3. Crypto API overview .....	6
4. STM32CubeMP1 architecture .....	6
5. STM32CubeMX .....	6
6. STM32MPU Embedded Software architecture overview .....	7
7. How to assign an internal peripheral to a runtime context .....	7



# CRC internal peripheral

Stable: 12.02.2020 - 16:39 / Revision: 12.02.2020 - 16:37

## Contents

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

## 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

Do	Peri	Software frameworks			Comment
mai Cor tex -A7 sec ure (O P- TE E)	Cor tex -A7 no n- sec ure (Li nux )	Cortex-M4  (STM32Cube)			
			Linux Crypto framework	STM32Cube CRC driver	
Se cu rity	C R C				

### 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.

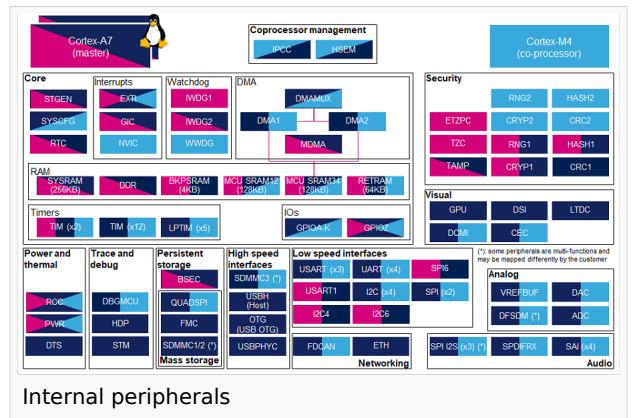
### 3.2.4 Peripheral assignment

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



Do	Per	Runtime allocation				Comme
S e c u r i t y	C R C	CRC1				
		CRC2				



---

## 4 How to go further

---

Not applicable.

---

## 5 References

---

Cyclic redundancy check calculation unit

Open Portable Trusted Execution Environment

### Permission error

---

*Stable: 24.06.2020 - 17:52 / Revision: 24.06.2020 - 12:32*

You do not have permission to read this page, for the following reason:

The action "Read pages" for the draft version of this page is only available for the groups ST\_editors, ST\_readers, Selected\_editors, sysop, reviewer

### Permission error

---

*Stable: 19.06.2020 - 08:24 / Revision: 19.06.2020 - 08:18*

You do not have permission to read this page, for the following reason:

The action "Read pages" for the draft version of this page is only available for the groups ST\_editors, ST\_readers, Selected\_editors, sysop, reviewer

### Permission error

---

*Stable: 21.02.2020 - 08:39 / Revision: 04.02.2020 - 15:22*

You do not have permission to read this page, for the following reason:

The action "Read pages" for the draft version of this page is only available for the groups ST\_editors, ST\_readers, Selected\_editors, sysop, reviewer

### Permission error

---

*Stable: 31.01.2020 - 13:04 / Revision: 31.01.2020 - 13:02*

You do not have permission to read this page, for the following reason:



The action "Read pages" for the draft version of this page is only available for the groups ST\_editors, ST\_readers, Selected\_editors, sysop, reviewer

## Permission error

---

*Stable: 15.10.2019 - 11:55 / Revision: 15.10.2019 - 11:55*

You do not have permission to read this page, for the following reason:

The action "Read pages" for the draft version of this page is only available for the groups ST\_editors, ST\_readers, Selected\_editors, sysop, reviewer

## Permission error

---

*Stable: 22.06.2020 - 09:50 / Revision: 22.06.2020 - 09:49*

You do not have permission to read this page, for the following reason:

The action "Read pages" for the draft version of this page is only available for the groups ST\_editors, ST\_readers, Selected\_editors, sysop, reviewer