

VREFBUF internal peripheral



# VREFBUF internal peripheral

Stable: 04.02.2020 - 15:59 / Revision: 04.02.2020 - 15:52

## Contents

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

## 1 Article purpose

The purpose of this article is to

- briefly introduce the VREFBUF 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 VREFBUF peripheral.

## 2 Peripheral overview

The **VREFBUF** peripheral is an internal voltage regulator.

### 2.1 Features

The VREFBUF is supplied via the VDDA pin. When enabled, it can provide a reference voltage in the range of: 1,5V, 1,8V, 2,048V or 2,5V.

The VREFBUF can be used to provide an analog voltage reference for:

- ADC internal peripheral<sup>[1]</sup>
- DAC internal peripheral<sup>[2]</sup>
- External components through the dedicated VREF+ pin.



The VREFBUF can be left unused. In this case, an external voltage regulator can provide reference voltage to VREF+ pin. Refer to the [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

The VREFBUF is a **non-secure** peripheral.

# 3 Peripheral usage and associated software

## 3.1 Boot time

The VREFBUF is usually not used at boot time. But it may be needed by the SSBL (see [Boot chains overview](#)), to supply the internal ADC<sup>[1]</sup> for example.

## 3.2 Runtime

### 3.2.1 Overview

The VREFBUF can be allocated to the Arm<sup>®</sup> Cortex<sup>®</sup>-A7 non-secure to be used under Linux<sup>®</sup> with regulator framework<sup>[3]</sup>.



The VREFBUF is a system resource<sup>[4]</sup> which needs to be also controlled by the resource manager<sup>[4]</sup> in case its consumers (e.g. ADC<sup>[1]</sup>, DAC<sup>[2]</sup> or an external device connected to VREF+ pin) are spread across:

- the Arm<sup>®</sup> Cortex<sup>®</sup>-A7 non-secure context
- the Arm<sup>®</sup> Cortex<sup>®</sup>-M4 context

For this reason, the direct control of VREFBUF from the Arm<sup>®</sup> Cortex<sup>®</sup>-M4 is not recommended in STM32Cube<sup>[5]</sup> by default.

It's recommended to implement it in STM32Cube **only if** all consumers and the VDDA supply pin are controlled in the Arm<sup>®</sup> Cortex<sup>®</sup>-M4 context.

The [Peripheral assignment](#) chapter describes which peripheral instance can be assigned to which context.

### 3.2.2 Software frameworks

Do	Peri	Software frameworks	Comment
main Cortex -A7	Cortex -A7 non-		

Do	Peri	Software frameworks		Comment
main secure (OPE) TE E)	secure (Linux)	Cortex-M4  (STM32Cube)		
Analog	VREFBUF		Linux regulator framework	

### 3.2.3 Peripheral configuration

The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via the [STM32CubeMX](#) tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information given in the corresponding software framework article.

- For the Linux kernel configuration, please refer to [device internal regulator](#). An example can be found also in [ADC DT configuration example](#)
- In case the control of VREFBUF consumers are spread across the various cores, see also [Resource manager for coprocessing](#)

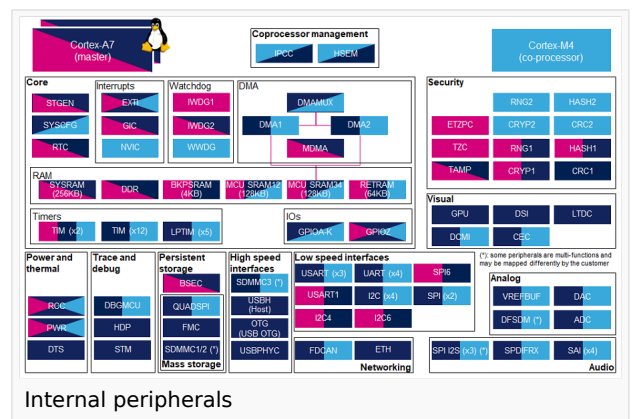
### 3.2.4 Peripheral assignment

**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	Peri	Runtime allocation		Comment
main secure (OPE) TE E)	secure (Linux)	Cortex-M4  (STM32Cube)		
Analog	VREFBUF		Linux regulator framework	



Do ma in st a nc e	Per in te r f a c e	Runtime allocation				Comme nt
		Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
A n a l o g	V R E F B U F	VREFBUF				Assig nment (singl e choice )

## 4 References

- 1.01.11.2 ADC internal peripheral
- 2.02.1 DAC internal peripheral
- Regulator overview, Linux® regulator framework overview
- 4.04.1 Resource manager for coprocessing, focus on system resources
- STM32CubeMP1 architecture

voltage reference buffer (STM32 specific)

Analog-to-digital converter. The process of converting a sampled analog signal to a digital code that represents the amplitude of the original signal sample.

Digital-to-analog converter (Electronic circuit that converts a binary number into a continuously varying value.)

Second Stage Boot Loader

Open Portable Trusted Execution Environment