



DAC internal peripheral

DAC internal peripheral



Contents

1. DAC internal peripheral	3
2. DAC Linux driver	8
3. DAC device tree configuration	13
4. DMA internal peripheral	18
5. EXTI internal peripheral	23
6. How to assign an internal peripheral to a runtime context	28
7. IIO overview	33
8. LPTIM internal peripheral	38
9. RCC internal peripheral	43
10. Regulator overview	48
11. STM32CubeMP1 architecture	53
12. STM32CubeMX	58
13. STM32MP15 resources	63
14. STM32MPU Embedded Software architecture overview	68
15. TIM internal peripheral	73
16. VREFBUF internal peripheral	78



A quality version of this page, approved on *12 February 2020*, was based off this revision.

Contents

1 Article purpose	4
2 Peripheral overview	5
2.1 Features	5
2.2 Security support	5
3 Peripheral usage and associated software	6
3.1 Boot time	6
3.2 Runtime	6
3.2.1 Overview	6
3.2.2 Software frameworks	6
3.2.3 Peripheral configuration	6
3.2.4 Peripheral assignment	6
4 References	8



1 Article purpose

The purpose of this article is to

- briefly introduce the DAC 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 DAC peripheral.



2 Peripheral overview

The **DAC** peripheral is a voltage output digital-to-analog converter.

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.

- It may be configured in **8- or 12-bit mode** (data could be left- or right-aligned)
- It has **two output channels**, each with its **own converter**
- The dual DAC channel mode could be done **independently or simultaneously**
- It has built-in noise and triangle **waveform generator** and supports **triggers** for conversions, using: TIM^[1], LPTIM^[2] or EXTI^[3]
- The DAC output buffer allows a high drive output current
- It can operate under **Normal mode** or low-power **Sample and Hold mode** (uses LSI clock, from RCC^[4]).
- It may be used in conjunction with the **DMA^[5]** controller (with underrun error detection)
- The **common reference voltage**, can be provided by either VREFBUF^[6] or any other external regulator^[7] wired to VREF+ pin.

2.2 Security support

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



3 Peripheral usage and associated software

3.1 Boot time

The DAC is not used at boot time.

3.2 Runtime

3.2.1 Overview

The DAC instances can be allocated to:

- the Arm®Cortex®-A7 non-secure core to be used under Linux® with the IIO framework
- or
- the Arm®Cortex®-M4 core to be used with STM32Cube Package with STM32Cube DAC driver.

The Peripheral assignment chapter describes which peripheral instance can be assigned to which context.

3.2.2 Software frameworks

Domain	Peripheral	Software components		Comment
OP-TEE	Linux	STM32Cube		
Analog	DAC	Linux IIO framework	STM32Cube DAC driver	

3.2.3 Peripheral configuration

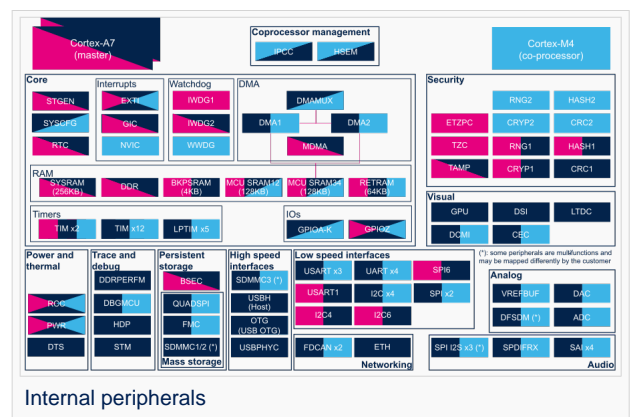
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via the STM32CubeMX tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information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

For the Linux kernel configuration, please refer to DAC device tree configuration and DAC Linux driver articles.

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.

Domain	Peripheral	Runtime allocation			Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)		
Analog	DAC	DAC			Assignment (single choice)



4 References

- TIM internal peripheral
- LPTIM internal peripheral
- EXTI internal peripheral
- RCC internal peripheral
- DMA internal peripheral
- VREFBUF internal peripheral
- Regulator overview

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

low-power timer (STM32 specific)

External Interrupt

Low Speed Internal oscillator (STM32 clock source)

Reset and Clock Control

Direct Memory Access

voltage reference buffer (STM32 specific)

Arm® is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



Cortex®

Linux® is a registered trademark of Linus Torvalds.

Open Portable Trusted Execution Environment

Stable: 16.01.2020 - 15:01 / Revision: 16.01.2020 - 14:56

Contents

1 Article purpose	9
2 Peripheral overview	10
2.1 Features	10
2.2 Security support	10
3 Peripheral usage and associated software	11
3.1 Boot time	11
3.2 Runtime	11
3.2.1 Overview	11
3.2.2 Software frameworks	11
3.2.3 Peripheral configuration	11
3.2.4 Peripheral assignment	11
4 References	13



1 Article purpose

The purpose of this article is to

- briefly introduce the DAC 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 DAC peripheral.



2 Peripheral overview

The **DAC** peripheral is a voltage output digital-to-analog converter.

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.

- It may be configured in **8- or 12-bit mode** (data could be left- or right-aligned)
- It has **two output channels**, each with its **own converter**
- The dual DAC channel mode could be done **independently or simultaneously**
- It has built-in noise and triangle **waveform generator** and supports **triggers** for conversions, using: TIM^[1], LPTIM^[2] or EXTI^[3]
- The DAC output buffer allows a high drive output current
- It can operate under **Normal mode** or low-power **Sample and Hold mode** (uses LSI clock, from RCC^[4]).
- It may be used in conjunction with the **DMA^[5]** controller (with underrun error detection)
- The **common reference voltage**, can be provided by either VREFBUF^[6] or any other external regulator^[7] wired to VREF+ pin.

2.2 Security support

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



3 Peripheral usage and associated software

3.1 Boot time

The DAC is not used at boot time.

3.2 Runtime

3.2.1 Overview

The DAC instances can be allocated to:

- the Arm®Cortex®-A7 non-secure core to be used under Linux® with the IIO framework
- or
- the Arm®Cortex®-M4 core to be used with STM32Cube Package with STM32Cube DAC driver.

The Peripheral assignment chapter describes which peripheral instance can be assigned to which context.

3.2.2 Software frameworks

Domain	Peripheral	Software components		Comment
OP-TEE	Linux	STM32Cube		
Analog	DAC	Linux IIO framework	STM32Cube DAC driver	

3.2.3 Peripheral configuration

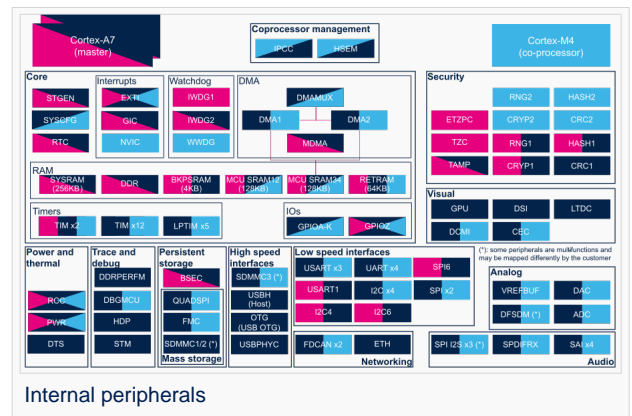
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via the STM32CubeMX tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information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

For the Linux kernel configuration, please refer to DAC device tree configuration and DAC Linux driver articles.

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.

Domain	Peripheral	Runtime allocation			Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)		
Analog	DAC	DAC			Assignment (single choice)



4 References

- TIM internal peripheral
- LPTIM internal peripheral
- EXTI internal peripheral
- RCC internal peripheral
- DMA internal peripheral
- VREFBUF internal peripheral
- Regulator overview

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

low-power timer (STM32 specific)

External Interrupt

Low Speed Internal oscillator (STM32 clock source)

Reset and Clock Control

Direct Memory Access

voltage reference buffer (STM32 specific)

Arm® is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



Cortex®

Linux® is a registered trademark of Linus Torvalds.

Open Portable Trusted Execution Environment

Stable: 26.03.2021 - 13:54 / Revision: 25.01.2021 - 10:05

Contents

1 Article purpose	14
2 Peripheral overview	15
2.1 Features	15
2.2 Security support	15
3 Peripheral usage and associated software	16
3.1 Boot time	16
3.2 Runtime	16
3.2.1 Overview	16
3.2.2 Software frameworks	16
3.2.3 Peripheral configuration	16
3.2.4 Peripheral assignment	16
4 References	18



1 Article purpose

The purpose of this article is to

- briefly introduce the DAC 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 DAC peripheral.



2 Peripheral overview

The **DAC** peripheral is a voltage output digital-to-analog converter.

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.

- It may be configured in **8- or 12-bit mode** (data could be left- or right-aligned)
- It has **two output channels**, each with its **own converter**
- The dual DAC channel mode could be done **independently or simultaneously**
- It has built-in noise and triangle **waveform generator** and supports **triggers** for conversions, using: TIM^[1], LPTIM^[2] or EXTI^[3]
- The DAC output buffer allows a high drive output current
- It can operate under **Normal mode** or low-power **Sample and Hold mode** (uses LSI clock, from RCC^[4]).
- It may be used in conjunction with the **DMA^[5]** controller (with underrun error detection)
- The **common reference voltage**, can be provided by either VREFBUF^[6] or any other external regulator^[7] wired to VREF+ pin.

2.2 Security support

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



3 Peripheral usage and associated software

3.1 Boot time

The DAC is not used at boot time.

3.2 Runtime

3.2.1 Overview

The DAC instances can be allocated to:

- the Arm®Cortex®-A7 non-secure core to be used under Linux® with the IIO framework
- or
- the Arm®Cortex®-M4 core to be used with STM32Cube Package with STM32Cube DAC driver.

The Peripheral assignment chapter describes which peripheral instance can be assigned to which context.

3.2.2 Software frameworks

Domain	Peripheral	Software components		Comment
OP-TEE	Linux	STM32Cube		
Analog	DAC	Linux IIO framework	STM32Cube DAC driver	

3.2.3 Peripheral configuration

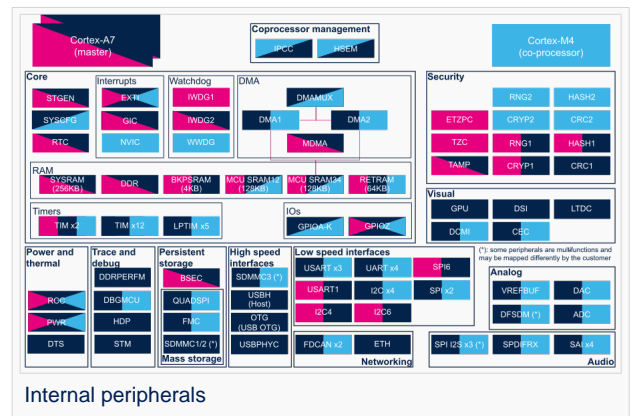
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via the STM32CubeMX tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information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

For the Linux kernel configuration, please refer to DAC device tree configuration and DAC Linux driver articles.

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.

Domain	Peripheral	Runtime allocation			Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)		
Analog	DAC	DAC			Assignment (single choice)



4 References

- TIM internal peripheral
- LPTIM internal peripheral
- EXTI internal peripheral
- RCC internal peripheral
- DMA internal peripheral
- VREFBUF internal peripheral
- Regulator overview

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

low-power timer (STM32 specific)

External Interrupt

Low Speed Internal oscillator (STM32 clock source)

Reset and Clock Control

Direct Memory Access

voltage reference buffer (STM32 specific)

Arm[®] is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



Cortex[®]

Linux[®] is a registered trademark of Linus Torvalds.

Open Portable Trusted Execution Environment

Stable: 13.10.2020 - 08:29 / Revision: 13.10.2020 - 08:29

Contents

1 Article purpose	19
2 Peripheral overview	20
2.1 Features	20
2.2 Security support	20
3 Peripheral usage and associated software	21
3.1 Boot time	21
3.2 Runtime	21
3.2.1 Overview	21
3.2.2 Software frameworks	21
3.2.3 Peripheral configuration	21
3.2.4 Peripheral assignment	21
4 References	23



1 Article purpose

The purpose of this article is to

- briefly introduce the DAC 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 DAC peripheral.



2 Peripheral overview

The **DAC** peripheral is a voltage output digital-to-analog converter.

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.

- It may be configured in **8- or 12-bit mode** (data could be left- or right-aligned)
- It has **two output channels**, each with its **own converter**
- The dual DAC channel mode could be done **independently or simultaneously**
- It has built-in noise and triangle **waveform generator** and supports **triggers** for conversions, using: TIM^[1], LPTIM^[2] or EXTI^[3]
- The DAC output buffer allows a high drive output current
- It can operate under **Normal mode** or low-power **Sample and Hold mode** (uses LSI clock, from RCC^[4]).
- It may be used in conjunction with the **DMA^[5]** controller (with underrun error detection)
- The **common reference voltage**, can be provided by either VREFBUF^[6] or any other external regulator^[7] wired to VREF+ pin.

2.2 Security support

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



3 Peripheral usage and associated software

3.1 Boot time

The DAC is not used at boot time.

3.2 Runtime

3.2.1 Overview

The DAC instances can be allocated to:

- the Arm®Cortex®-A7 non-secure core to be used under Linux® with the IIO framework
- or
- the Arm®Cortex®-M4 core to be used with STM32Cube Package with STM32Cube DAC driver.

The Peripheral assignment chapter describes which peripheral instance can be assigned to which context.

3.2.2 Software frameworks

Domain	Peripheral	Software components		Comment
OP-TEE	Linux	STM32Cube		
Analog	DAC	Linux IIO framework	STM32Cube DAC driver	

3.2.3 Peripheral configuration

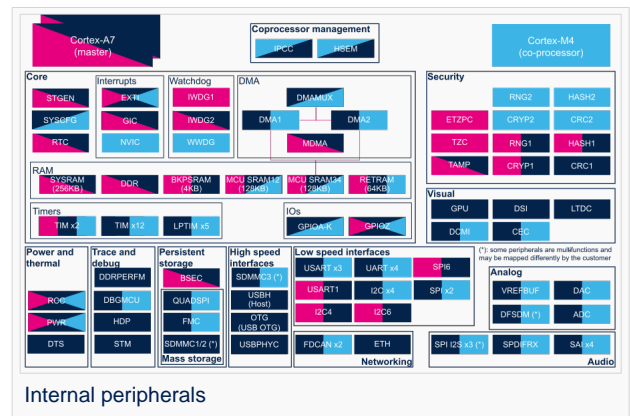
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via the STM32CubeMX tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information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

For the Linux kernel configuration, please refer to DAC device tree configuration and DAC Linux driver articles.

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.

Domain	Peripheral	Runtime allocation			Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)		
Analog	DAC	DAC			Assignment (single choice)



4 References

- TIM internal peripheral
- LPTIM internal peripheral
- EXTI internal peripheral
- RCC internal peripheral
- DMA internal peripheral
- VREFBUF internal peripheral
- Regulator overview

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

low-power timer (STM32 specific)

External Interrupt

Low Speed Internal oscillator (STM32 clock source)

Reset and Clock Control

Direct Memory Access

voltage reference buffer (STM32 specific)

Arm[®] is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



Cortex[®]

Linux[®] is a registered trademark of Linus Torvalds.

Open Portable Trusted Execution Environment

Stable: 26.03.2021 - 13:10 / Revision: 18.03.2021 - 14:45

Contents

1 Article purpose	24
2 Peripheral overview	25
2.1 Features	25
2.2 Security support	25
3 Peripheral usage and associated software	26
3.1 Boot time	26
3.2 Runtime	26
3.2.1 Overview	26
3.2.2 Software frameworks	26
3.2.3 Peripheral configuration	26
3.2.4 Peripheral assignment	26
4 References	28



1 Article purpose

The purpose of this article is to

- briefly introduce the DAC 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 DAC peripheral.



2 Peripheral overview

The **DAC** peripheral is a voltage output digital-to-analog converter.

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.

- It may be configured in **8- or 12-bit mode** (data could be left- or right-aligned)
- It has **two output channels**, each with its **own converter**
- The dual DAC channel mode could be done **independently or simultaneously**
- It has built-in noise and triangle **waveform generator** and supports **triggers** for conversions, using: TIM^[1], LPTIM^[2] or EXTI^[3]
- The DAC output buffer allows a high drive output current
- It can operate under **Normal mode** or low-power **Sample and Hold mode** (uses LSI clock, from RCC^[4]).
- It may be used in conjunction with the **DMA^[5]** controller (with underrun error detection)
- The **common reference voltage**, can be provided by either VREFBUF^[6] or any other external regulator^[7] wired to VREF+ pin.

2.2 Security support

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



3 Peripheral usage and associated software

3.1 Boot time

The DAC is not used at boot time.

3.2 Runtime

3.2.1 Overview

The DAC instances can be allocated to:

- the Arm®Cortex®-A7 non-secure core to be used under Linux® with the IIO framework
- or
- the Arm®Cortex®-M4 core to be used with STM32Cube Package with STM32Cube DAC driver.

The Peripheral assignment chapter describes which peripheral instance can be assigned to which context.

3.2.2 Software frameworks

Domain	Peripheral	Software components		Comment
OP-TEE	Linux	STM32Cube		
Analog	DAC		Linux IIO framework	STM32Cube DAC driver

3.2.3 Peripheral configuration

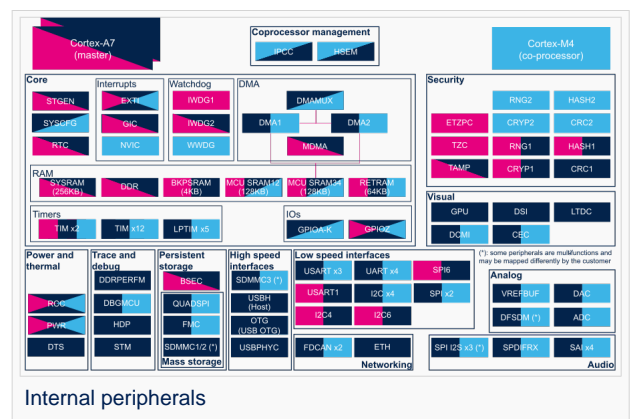
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via the STM32CubeMX tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information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

For the Linux kernel configuration, please refer to DAC device tree configuration and DAC Linux driver articles.

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.

Domain	Peripheral	Runtime allocation			Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)		
Analog	DAC	DAC			Assignment (single choice)



4 References

- TIM internal peripheral
- LPTIM internal peripheral
- EXTI internal peripheral
- RCC internal peripheral
- DMA internal peripheral
- VREFBUF internal peripheral
- Regulator overview

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

low-power timer (STM32 specific)

External Interrupt

Low Speed Internal oscillator (STM32 clock source)

Reset and Clock Control

Direct Memory Access

voltage reference buffer (STM32 specific)

Arm[®] is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



Cortex[®]

Linux[®] is a registered trademark of Linus Torvalds.

Open Portable Trusted Execution Environment

Stable: 08.03.2021 - 16:13 / Revision: 16.02.2021 - 17:11

Contents

1 Article purpose	29
2 Peripheral overview	30
2.1 Features	30
2.2 Security support	30
3 Peripheral usage and associated software	31
3.1 Boot time	31
3.2 Runtime	31
3.2.1 Overview	31
3.2.2 Software frameworks	31
3.2.3 Peripheral configuration	31
3.2.4 Peripheral assignment	31
4 References	33



1 Article purpose

The purpose of this article is to

- briefly introduce the DAC 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 DAC peripheral.



2 Peripheral overview

The **DAC** peripheral is a voltage output digital-to-analog converter.

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.

- It may be configured in **8- or 12-bit mode** (data could be left- or right-aligned)
- It has **two output channels**, each with its **own converter**
- The dual DAC channel mode could be done **independently or simultaneously**
- It has built-in noise and triangle **waveform generator** and supports **triggers** for conversions, using: TIM^[1], LPTIM^[2] or EXTI^[3]
- The DAC output buffer allows a high drive output current
- It can operate under **Normal mode** or low-power **Sample and Hold mode** (uses LSI clock, from RCC^[4]).
- It may be used in conjunction with the **DMA^[5]** controller (with underrun error detection)
- The **common reference voltage**, can be provided by either VREFBUF^[6] or any other external regulator^[7] wired to VREF+ pin.

2.2 Security support

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



3 Peripheral usage and associated software

3.1 Boot time

The DAC is not used at boot time.

3.2 Runtime

3.2.1 Overview

The DAC instances can be allocated to:

- the Arm®Cortex®-A7 non-secure core to be used under Linux® with the IIO framework
- or
- the Arm®Cortex®-M4 core to be used with STM32Cube Package with STM32Cube DAC driver.

The Peripheral assignment chapter describes which peripheral instance can be assigned to which context.

3.2.2 Software frameworks

Domain	Peripheral	Software components		Comment
OP-TEE	Linux	STM32Cube		
Analog	DAC	Linux IIO framework	STM32Cube DAC driver	

3.2.3 Peripheral configuration

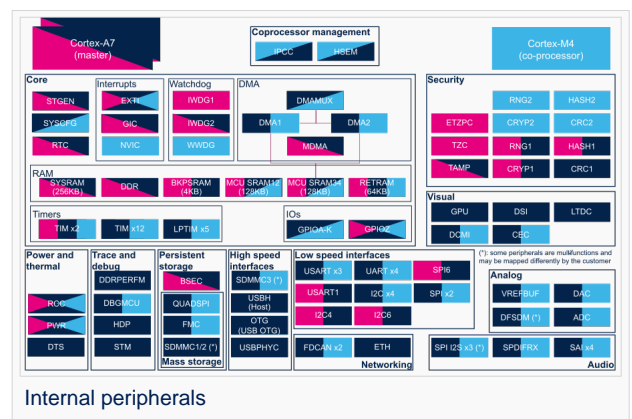
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via the STM32CubeMX tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information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

For the Linux kernel configuration, please refer to DAC device tree configuration and DAC Linux driver articles.

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.

Domain	Peripheral	Runtime allocation			Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)		
Analog	DAC	DAC			Assignment (single choice)



4 References

- TIM internal peripheral
- LPTIM internal peripheral
- EXTI internal peripheral
- RCC internal peripheral
- DMA internal peripheral
- VREFBUF internal peripheral
- Regulator overview

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

low-power timer (STM32 specific)

External Interrupt

Low Speed Internal oscillator (STM32 clock source)

Reset and Clock Control

Direct Memory Access

voltage reference buffer (STM32 specific)

Arm® is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



Cortex®

Linux® is a registered trademark of Linus Torvalds.

Open Portable Trusted Execution Environment

Stable: 17.02.2021 - 16:24 / Revision: 17.02.2021 - 16:22

Contents

1 Article purpose	34
2 Peripheral overview	35
2.1 Features	35
2.2 Security support	35
3 Peripheral usage and associated software	36
3.1 Boot time	36
3.2 Runtime	36
3.2.1 Overview	36
3.2.2 Software frameworks	36
3.2.3 Peripheral configuration	36
3.2.4 Peripheral assignment	36
4 References	38



1 Article purpose

The purpose of this article is to

- briefly introduce the DAC 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 DAC peripheral.



2 Peripheral overview

The **DAC** peripheral is a voltage output digital-to-analog converter.

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.

- It may be configured in **8- or 12-bit mode** (data could be left- or right-aligned)
- It has **two output channels**, each with its **own converter**
- The dual DAC channel mode could be done **independently or simultaneously**
- It has built-in noise and triangle **waveform generator** and supports **triggers** for conversions, using: TIM^[1], LPTIM^[2] or EXTI^[3]
- The DAC output buffer allows a high drive output current
- It can operate under **Normal mode** or low-power **Sample and Hold mode** (uses LSI clock, from RCC^[4]).
- It may be used in conjunction with the **DMA^[5]** controller (with underrun error detection)
- The **common reference voltage**, can be provided by either VREFBUF^[6] or any other external regulator^[7] wired to VREF+ pin.

2.2 Security support

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



3 Peripheral usage and associated software

3.1 Boot time

The DAC is not used at boot time.

3.2 Runtime

3.2.1 Overview

The DAC instances can be allocated to:

- the Arm®Cortex®-A7 non-secure core to be used under Linux® with the IIO framework
- or
- the Arm®Cortex®-M4 core to be used with STM32Cube Package with STM32Cube DAC driver.

The Peripheral assignment chapter describes which peripheral instance can be assigned to which context.

3.2.2 Software frameworks

Domain	Peripheral	Software components		Comment
OP-TEE	Linux	STM32Cube		
Analog	DAC	Linux IIO framework	STM32Cube DAC driver	

3.2.3 Peripheral configuration

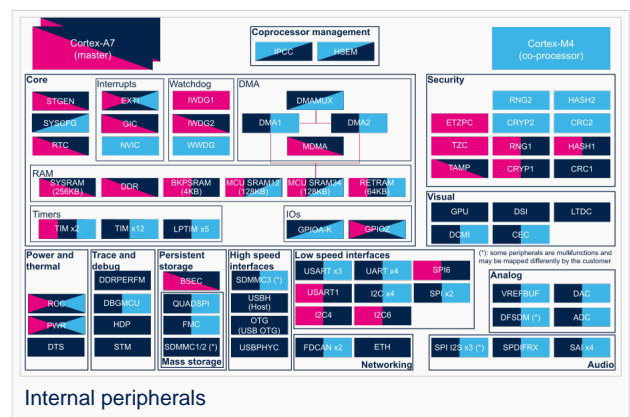
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via the STM32CubeMX tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information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

For the Linux kernel configuration, please refer to DAC device tree configuration and DAC Linux driver articles.

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.

Domain	Peripheral	Runtime allocation			Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)		
Analog	DAC	DAC			Assignment (single choice)



4 References

- TIM internal peripheral
- LPTIM internal peripheral
- EXTI internal peripheral
- RCC internal peripheral
- DMA internal peripheral
- VREFBUF internal peripheral
- Regulator overview

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

low-power timer (STM32 specific)

External Interrupt

Low Speed Internal oscillator (STM32 clock source)

Reset and Clock Control

Direct Memory Access

voltage reference buffer (STM32 specific)

Arm® is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



Cortex®

Linux® is a registered trademark of Linus Torvalds.

Open Portable Trusted Execution Environment

Stable: 30.03.2021 - 13:25 / Revision: 29.03.2021 - 09:46

Contents

1 Article purpose	39
2 Peripheral overview	40
2.1 Features	40
2.2 Security support	40
3 Peripheral usage and associated software	41
3.1 Boot time	41
3.2 Runtime	41
3.2.1 Overview	41
3.2.2 Software frameworks	41
3.2.3 Peripheral configuration	41
3.2.4 Peripheral assignment	41
4 References	43



1 Article purpose

The purpose of this article is to

- briefly introduce the DAC 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 DAC peripheral.



2 Peripheral overview

The **DAC** peripheral is a voltage output digital-to-analog converter.

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.

- It may be configured in **8- or 12-bit mode** (data could be left- or right-aligned)
- It has **two output channels**, each with its **own converter**
- The dual DAC channel mode could be done **independently or simultaneously**
- It has built-in noise and triangle **waveform generator** and supports **triggers** for conversions, using: TIM^[1], LPTIM^[2] or EXTI^[3]
- The DAC output buffer allows a high drive output current
- It can operate under **Normal mode** or low-power **Sample and Hold mode** (uses LSI clock, from RCC^[4]).
- It may be used in conjunction with the **DMA^[5]** controller (with underrun error detection)
- The **common reference voltage**, can be provided by either VREFBUF^[6] or any other external regulator^[7] wired to VREF+ pin.

2.2 Security support

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



3 Peripheral usage and associated software

3.1 Boot time

The DAC is not used at boot time.

3.2 Runtime

3.2.1 Overview

The DAC instances can be allocated to:

- the Arm®Cortex®-A7 non-secure core to be used under Linux® with the IIO framework
- or
- the Arm®Cortex®-M4 core to be used with STM32Cube Package with STM32Cube DAC driver.

The Peripheral assignment chapter describes which peripheral instance can be assigned to which context.

3.2.2 Software frameworks

Domain	Peripheral	Software components		Comment
OP-TEE	Linux	STM32Cube		
Analog	DAC	Linux IIO framework	STM32Cube DAC driver	

3.2.3 Peripheral configuration

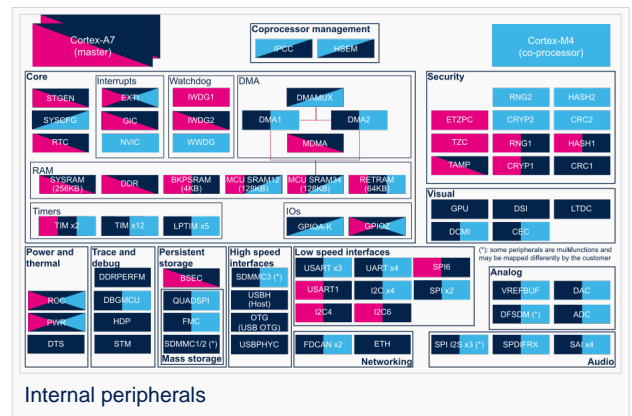
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via the STM32CubeMX tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information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

For the Linux kernel configuration, please refer to DAC device tree configuration and DAC Linux driver articles.

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.

Domain	Peripheral	Runtime allocation			Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)		
Analog	DAC	DAC			Assignment (single choice)



4 References

- TIM internal peripheral
- LPTIM internal peripheral
- EXTI internal peripheral
- RCC internal peripheral
- DMA internal peripheral
- VREFBUF internal peripheral
- Regulator overview

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

low-power timer (STM32 specific)

External Interrupt

Low Speed Internal oscillator (STM32 clock source)

Reset and Clock Control

Direct Memory Access

voltage reference buffer (STM32 specific)

Arm[®] is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



Cortex[®]

Linux[®] is a registered trademark of Linus Torvalds.

Open Portable Trusted Execution Environment

Stable: 25.09.2020 - 09:10 / Revision: 25.09.2020 - 09:09

Contents

1 Article purpose	44
2 Peripheral overview	45
2.1 Features	45
2.2 Security support	45
3 Peripheral usage and associated software	46
3.1 Boot time	46
3.2 Runtime	46
3.2.1 Overview	46
3.2.2 Software frameworks	46
3.2.3 Peripheral configuration	46
3.2.4 Peripheral assignment	46
4 References	48



1 Article purpose

The purpose of this article is to

- briefly introduce the DAC 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 DAC peripheral.



2 Peripheral overview

The **DAC** peripheral is a voltage output digital-to-analog converter.

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.

- It may be configured in **8- or 12-bit mode** (data could be left- or right-aligned)
- It has **two output channels**, each with its **own converter**
- The dual DAC channel mode could be done **independently or simultaneously**
- It has built-in noise and triangle **waveform generator** and supports **triggers** for conversions, using: TIM^[1], LPTIM^[2] or EXTI^[3]
- The DAC output buffer allows a high drive output current
- It can operate under **Normal mode** or low-power **Sample and Hold mode** (uses LSI clock, from RCC^[4]).
- It may be used in conjunction with the **DMA^[5]** controller (with underrun error detection)
- The **common reference voltage**, can be provided by either VREFBUF^[6] or any other external regulator^[7] wired to VREF+ pin.

2.2 Security support

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



3 Peripheral usage and associated software

3.1 Boot time

The DAC is not used at boot time.

3.2 Runtime

3.2.1 Overview

The DAC instances can be allocated to:

- the Arm®Cortex®-A7 non-secure core to be used under Linux® with the IIO framework
- or
- the Arm®Cortex®-M4 core to be used with STM32Cube Package with STM32Cube DAC driver.

The Peripheral assignment chapter describes which peripheral instance can be assigned to which context.

3.2.2 Software frameworks

Domain	Peripheral	Software components		Comment
OP-TEE	Linux	STM32Cube		
Analog	DAC	Linux IIO framework	STM32Cube DAC driver	

3.2.3 Peripheral configuration

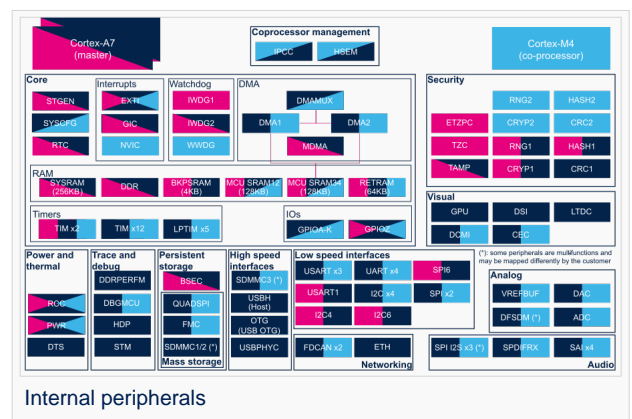
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via the STM32CubeMX tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information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

For the Linux kernel configuration, please refer to DAC device tree configuration and DAC Linux driver articles.

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.

Domain	Peripheral	Runtime allocation			Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)		
Analog	DAC	DAC			Assignment (single choice)



4 References

- TIM internal peripheral
- LPTIM internal peripheral
- EXTI internal peripheral
- RCC internal peripheral
- DMA internal peripheral
- VREFBUF internal peripheral
- Regulator overview

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

low-power timer (STM32 specific)

External Interrupt

Low Speed Internal oscillator (STM32 clock source)

Reset and Clock Control

Direct Memory Access

voltage reference buffer (STM32 specific)

Arm[®] is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



Cortex[®]

Linux[®] is a registered trademark of Linus Torvalds.

Open Portable Trusted Execution Environment

Stable: 11.06.2020 - 12:50 / Revision: 11.06.2020 - 12:12

Contents

1 Article purpose	49
2 Peripheral overview	50
2.1 Features	50
2.2 Security support	50
3 Peripheral usage and associated software	51
3.1 Boot time	51
3.2 Runtime	51
3.2.1 Overview	51
3.2.2 Software frameworks	51
3.2.3 Peripheral configuration	51
3.2.4 Peripheral assignment	51
4 References	53



1 Article purpose

The purpose of this article is to

- briefly introduce the DAC 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 DAC peripheral.



2 Peripheral overview

The **DAC** peripheral is a voltage output digital-to-analog converter.

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.

- It may be configured in **8- or 12-bit mode** (data could be left- or right-aligned)
- It has **two output channels**, each with its **own converter**
- The dual DAC channel mode could be done **independently or simultaneously**
- It has built-in noise and triangle **waveform generator** and supports **triggers** for conversions, using: TIM^[1], LPTIM^[2] or EXTI^[3]
- The DAC output buffer allows a high drive output current
- It can operate under **Normal mode** or low-power **Sample and Hold mode** (uses LSI clock, from RCC^[4]).
- It may be used in conjunction with the **DMA^[5]** controller (with underrun error detection)
- The **common reference voltage**, can be provided by either VREFBUF^[6] or any other external regulator^[7] wired to VREF+ pin.

2.2 Security support

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



3 Peripheral usage and associated software

3.1 Boot time

The DAC is not used at boot time.

3.2 Runtime

3.2.1 Overview

The DAC instances can be allocated to:

- the Arm®Cortex®-A7 non-secure core to be used under Linux® with the IIO framework
- or
- the Arm®Cortex®-M4 core to be used with STM32Cube Package with STM32Cube DAC driver.

The Peripheral assignment chapter describes which peripheral instance can be assigned to which context.

3.2.2 Software frameworks

Domain	Peripheral	Software components		Comment
OP-TEE	Linux	STM32Cube		
Analog	DAC	Linux IIO framework	STM32Cube DAC driver	

3.2.3 Peripheral configuration

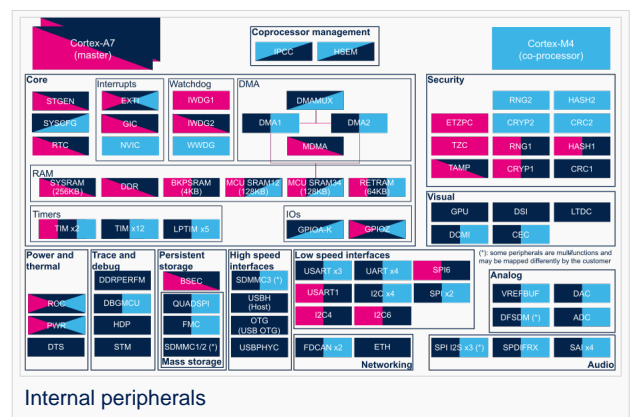
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via the STM32CubeMX tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information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

For the Linux kernel configuration, please refer to DAC device tree configuration and DAC Linux driver articles.

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.

Domain	Peripheral	Runtime allocation			Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)		
Analog	DAC	DAC			Assignment (single choice)



4 References

- TIM internal peripheral
- LPTIM internal peripheral
- EXTI internal peripheral
- RCC internal peripheral
- DMA internal peripheral
- VREFBUF internal peripheral
- Regulator overview

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

low-power timer (STM32 specific)

External Interrupt

Low Speed Internal oscillator (STM32 clock source)

Reset and Clock Control

Direct Memory Access

voltage reference buffer (STM32 specific)

Arm[®] is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



Cortex[®]

Linux[®] is a registered trademark of Linus Torvalds.

Open Portable Trusted Execution Environment

Stable: 31.03.2021 - 11:58 / Revision: 23.03.2021 - 14:07

Contents

1 Article purpose	54
2 Peripheral overview	55
2.1 Features	55
2.2 Security support	55
3 Peripheral usage and associated software	56
3.1 Boot time	56
3.2 Runtime	56
3.2.1 Overview	56
3.2.2 Software frameworks	56
3.2.3 Peripheral configuration	56
3.2.4 Peripheral assignment	56
4 References	58



1 Article purpose

The purpose of this article is to

- briefly introduce the DAC 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 DAC peripheral.



2 Peripheral overview

The **DAC** peripheral is a voltage output digital-to-analog converter.

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.

- It may be configured in **8- or 12-bit mode** (data could be left- or right-aligned)
- It has **two output channels**, each with its **own converter**
- The dual DAC channel mode could be done **independently or simultaneously**
- It has built-in noise and triangle **waveform generator** and supports **triggers** for conversions, using: TIM^[1], LPTIM^[2] or EXTI^[3]
- The DAC output buffer allows a high drive output current
- It can operate under **Normal mode** or low-power **Sample and Hold mode** (uses LSI clock, from RCC^[4]).
- It may be used in conjunction with the **DMA^[5]** controller (with underrun error detection)
- The **common reference voltage**, can be provided by either VREFBUF^[6] or any other external regulator^[7] wired to VREF+ pin.

2.2 Security support

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



3 Peripheral usage and associated software

3.1 Boot time

The DAC is not used at boot time.

3.2 Runtime

3.2.1 Overview

The DAC instances can be allocated to:

- the Arm®Cortex®-A7 non-secure core to be used under Linux® with the IIO framework
- or
- the Arm®Cortex®-M4 core to be used with STM32Cube Package with STM32Cube DAC driver.

The Peripheral assignment chapter describes which peripheral instance can be assigned to which context.

3.2.2 Software frameworks

Domain	Peripheral	Software components		Comment
OP-TEE	Linux	STM32Cube		
Analog	DAC		Linux IIO framework	STM32Cube DAC driver

3.2.3 Peripheral configuration

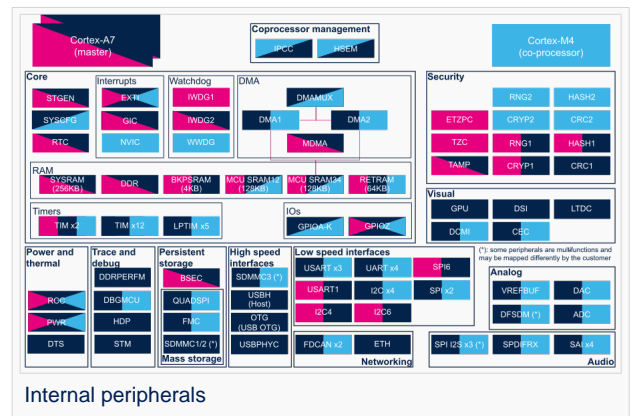
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via the STM32CubeMX tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information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

For the Linux kernel configuration, please refer to DAC device tree configuration and DAC Linux driver articles.

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.

Domain	Peripheral	Runtime allocation			Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)		
Analog	DAC	DAC			Assignment (single choice)



4 References

- TIM internal peripheral
- LPTIM internal peripheral
- EXTI internal peripheral
- RCC internal peripheral
- DMA internal peripheral
- VREFBUF internal peripheral
- Regulator overview

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

low-power timer (STM32 specific)

External Interrupt

Low Speed Internal oscillator (STM32 clock source)

Reset and Clock Control

Direct Memory Access

voltage reference buffer (STM32 specific)

Arm® is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



Cortex®

Linux® is a registered trademark of Linus Torvalds.

Open Portable Trusted Execution Environment

Stable: 23.09.2020 - 13:22 / Revision: 12.06.2020 - 13:25

Contents

1 Article purpose	59
2 Peripheral overview	60
2.1 Features	60
2.2 Security support	60
3 Peripheral usage and associated software	61
3.1 Boot time	61
3.2 Runtime	61
3.2.1 Overview	61
3.2.2 Software frameworks	61
3.2.3 Peripheral configuration	61
3.2.4 Peripheral assignment	61
4 References	63



1 Article purpose

The purpose of this article is to

- briefly introduce the DAC 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 DAC peripheral.



2 Peripheral overview

The **DAC** peripheral is a voltage output digital-to-analog converter.

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.

- It may be configured in **8- or 12-bit mode** (data could be left- or right-aligned)
- It has **two output channels**, each with its **own converter**
- The dual DAC channel mode could be done **independently or simultaneously**
- It has built-in noise and triangle **waveform generator** and supports **triggers** for conversions, using: TIM^[1], LPTIM^[2] or EXTI^[3]
- The DAC output buffer allows a high drive output current
- It can operate under **Normal mode** or low-power **Sample and Hold mode** (uses LSI clock, from RCC^[4]).
- It may be used in conjunction with the **DMA^[5]** controller (with underrun error detection)
- The **common reference voltage**, can be provided by either VREFBUF^[6] or any other external regulator^[7] wired to VREF+ pin.

2.2 Security support

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



3 Peripheral usage and associated software

3.1 Boot time

The DAC is not used at boot time.

3.2 Runtime

3.2.1 Overview

The DAC instances can be allocated to:

- the Arm®Cortex®-A7 non-secure core to be used under Linux® with the IIO framework
- or
- the Arm®Cortex®-M4 core to be used with STM32Cube Package with STM32Cube DAC driver.

The Peripheral assignment chapter describes which peripheral instance can be assigned to which context.

3.2.2 Software frameworks

Domain	Peripheral	Software components		Comment
OP-TEE	Linux	STM32Cube		
Analog	DAC		Linux IIO framework	STM32Cube DAC driver

3.2.3 Peripheral configuration

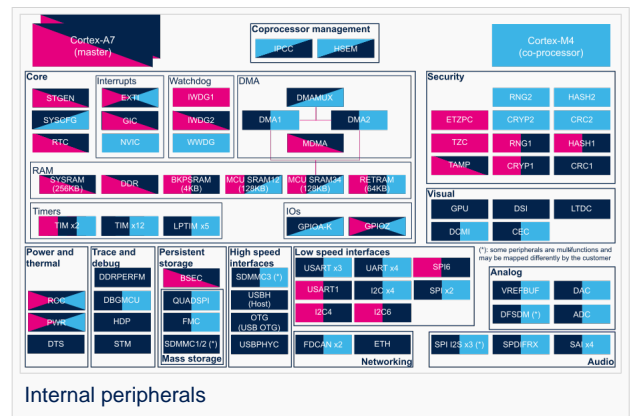
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via the STM32CubeMX tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information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

For the Linux kernel configuration, please refer to DAC device tree configuration and DAC Linux driver articles.

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.

Domain	Peripheral	Runtime allocation			Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)		
Analog	DAC	DAC			Assignment (single choice)



4 References

- TIM internal peripheral
- LPTIM internal peripheral
- EXTI internal peripheral
- RCC internal peripheral
- DMA internal peripheral
- VREFBUF internal peripheral
- Regulator overview

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

low-power timer (STM32 specific)

External Interrupt

Low Speed Internal oscillator (STM32 clock source)

Reset and Clock Control

Direct Memory Access

voltage reference buffer (STM32 specific)

Arm® is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



Cortex®

Linux® is a registered trademark of Linus Torvalds.

Open Portable Trusted Execution Environment

Stable: 20.07.2021 - 09:02 / Revision: 11.03.2021 - 08:07

Contents

1 Article purpose	64
2 Peripheral overview	65
2.1 Features	65
2.2 Security support	65
3 Peripheral usage and associated software	66
3.1 Boot time	66
3.2 Runtime	66
3.2.1 Overview	66
3.2.2 Software frameworks	66
3.2.3 Peripheral configuration	66
3.2.4 Peripheral assignment	66
4 References	68



1 Article purpose

The purpose of this article is to

- briefly introduce the DAC 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 DAC peripheral.



2 Peripheral overview

The **DAC** peripheral is a voltage output digital-to-analog converter.

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.

- It may be configured in **8- or 12-bit mode** (data could be left- or right-aligned)
- It has **two output channels**, each with its **own converter**
- The dual DAC channel mode could be done **independently or simultaneously**
- It has built-in noise and triangle **waveform generator** and supports **triggers** for conversions, using: TIM^[1], LPTIM^[2] or EXTI^[3]
- The DAC output buffer allows a high drive output current
- It can operate under **Normal mode** or low-power **Sample and Hold mode** (uses LSI clock, from RCC^[4]).
- It may be used in conjunction with the **DMA^[5]** controller (with underrun error detection)
- The **common reference voltage**, can be provided by either VREFBUF^[6] or any other external regulator^[7] wired to VREF+ pin.

2.2 Security support

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



3 Peripheral usage and associated software

3.1 Boot time

The DAC is not used at boot time.

3.2 Runtime

3.2.1 Overview

The DAC instances can be allocated to:

- the Arm®Cortex®-A7 non-secure core to be used under Linux® with the IIO framework
- or
- the Arm®Cortex®-M4 core to be used with STM32Cube Package with STM32Cube DAC driver.

The Peripheral assignment chapter describes which peripheral instance can be assigned to which context.

3.2.2 Software frameworks

Domain	Peripheral	Software components		Comment
OP-TEE	Linux	STM32Cube		
Analog	DAC	Linux IIO framework	STM32Cube DAC driver	

3.2.3 Peripheral configuration

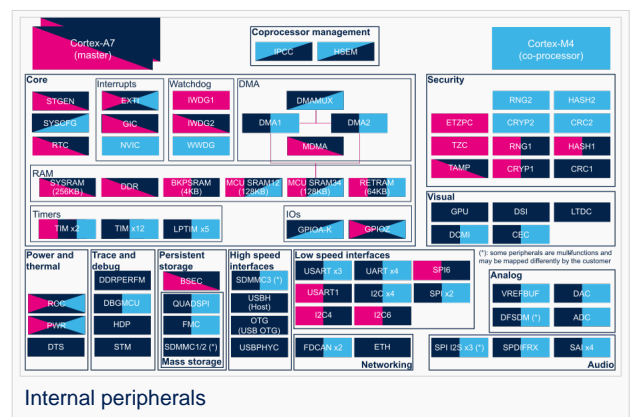
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via the STM32CubeMX tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information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

For the Linux kernel configuration, please refer to DAC device tree configuration and DAC Linux driver articles.

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.

Domain	Peripheral	Runtime allocation			Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)		
Analog	DAC	DAC			Assignment (single choice)



4 References

- TIM internal peripheral
- LPTIM internal peripheral
- EXTI internal peripheral
- RCC internal peripheral
- DMA internal peripheral
- VREFBUF internal peripheral
- Regulator overview

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

low-power timer (STM32 specific)

External Interrupt

Low Speed Internal oscillator (STM32 clock source)

Reset and Clock Control

Direct Memory Access

voltage reference buffer (STM32 specific)

Arm[®] is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



Cortex[®]

Linux[®] is a registered trademark of Linus Torvalds.

Open Portable Trusted Execution Environment

Stable: 26.03.2021 - 11:32 / Revision: 12.03.2021 - 11:07

Contents

1 Article purpose	69
2 Peripheral overview	70
2.1 Features	70
2.2 Security support	70
3 Peripheral usage and associated software	71
3.1 Boot time	71
3.2 Runtime	71
3.2.1 Overview	71
3.2.2 Software frameworks	71
3.2.3 Peripheral configuration	71
3.2.4 Peripheral assignment	71
4 References	73



1 Article purpose

The purpose of this article is to

- briefly introduce the DAC 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 DAC peripheral.



2 Peripheral overview

The **DAC** peripheral is a voltage output digital-to-analog converter.

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.

- It may be configured in **8- or 12-bit mode** (data could be left- or right-aligned)
- It has **two output channels**, each with its **own converter**
- The dual DAC channel mode could be done **independently or simultaneously**
- It has built-in noise and triangle **waveform generator** and supports **triggers** for conversions, using: TIM^[1], LPTIM^[2] or EXTI^[3]
- The DAC output buffer allows a high drive output current
- It can operate under **Normal mode** or low-power **Sample and Hold mode** (uses LSI clock, from RCC^[4]).
- It may be used in conjunction with the **DMA^[5]** controller (with underrun error detection)
- The **common reference voltage**, can be provided by either VREFBUF^[6] or any other external regulator^[7] wired to VREF+ pin.

2.2 Security support

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



3 Peripheral usage and associated software

3.1 Boot time

The DAC is not used at boot time.

3.2 Runtime

3.2.1 Overview

The DAC instances can be allocated to:

- the Arm®Cortex®-A7 non-secure core to be used under Linux® with the IIO framework
- or
- the Arm®Cortex®-M4 core to be used with STM32Cube Package with STM32Cube DAC driver.

The Peripheral assignment chapter describes which peripheral instance can be assigned to which context.

3.2.2 Software frameworks

Domain	Peripheral	Software components		Comment
OP-TEE	Linux	STM32Cube		
Analog	DAC	Linux IIO framework	STM32Cube DAC driver	

3.2.3 Peripheral configuration

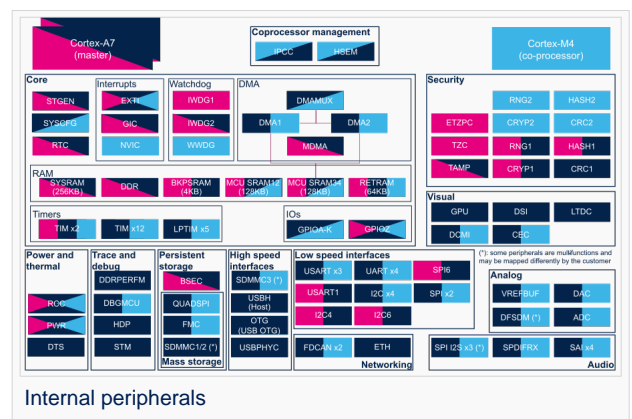
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via the STM32CubeMX tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information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

For the Linux kernel configuration, please refer to DAC device tree configuration and DAC Linux driver articles.

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.

Domain	Peripheral	Runtime allocation			Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)		
Analog	DAC	DAC			Assignment (single choice)



4 References

- TIM internal peripheral
- LPTIM internal peripheral
- EXTI internal peripheral
- RCC internal peripheral
- DMA internal peripheral
- VREFBUF internal peripheral
- Regulator overview

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

low-power timer (STM32 specific)

External Interrupt

Low Speed Internal oscillator (STM32 clock source)

Reset and Clock Control

Direct Memory Access

voltage reference buffer (STM32 specific)

Arm[®] is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



Cortex[®]

Linux[®] is a registered trademark of Linus Torvalds.

Open Portable Trusted Execution Environment

Stable: 30.03.2021 - 13:21 / Revision: 29.03.2021 - 09:46

Contents

1 Article purpose	74
2 Peripheral overview	75
2.1 Features	75
2.2 Security support	75
3 Peripheral usage and associated software	76
3.1 Boot time	76
3.2 Runtime	76
3.2.1 Overview	76
3.2.2 Software frameworks	76
3.2.3 Peripheral configuration	76
3.2.4 Peripheral assignment	76
4 References	78



1 Article purpose

The purpose of this article is to

- briefly introduce the DAC 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 DAC peripheral.



2 Peripheral overview

The **DAC** peripheral is a voltage output digital-to-analog converter.

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.

- It may be configured in **8- or 12-bit mode** (data could be left- or right-aligned)
- It has **two output channels**, each with its **own converter**
- The dual DAC channel mode could be done **independently or simultaneously**
- It has built-in noise and triangle **waveform generator** and supports **triggers** for conversions, using: TIM^[1], LPTIM^[2] or EXTI^[3]
- The DAC output buffer allows a high drive output current
- It can operate under **Normal mode** or low-power **Sample and Hold mode** (uses LSI clock, from RCC^[4]).
- It may be used in conjunction with the **DMA^[5]** controller (with underrun error detection)
- The **common reference voltage**, can be provided by either VREFBUF^[6] or any other external regulator^[7] wired to VREF+ pin.

2.2 Security support

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



3 Peripheral usage and associated software

3.1 Boot time

The DAC is not used at boot time.

3.2 Runtime

3.2.1 Overview

The DAC instances can be allocated to:

- the Arm®Cortex®-A7 non-secure core to be used under Linux® with the IIO framework
- or
- the Arm®Cortex®-M4 core to be used with STM32Cube Package with STM32Cube DAC driver.

The Peripheral assignment chapter describes which peripheral instance can be assigned to which context.

3.2.2 Software frameworks

Domain	Peripheral	Software components		Comment
OP-TEE	Linux	STM32Cube		
Analog	DAC	Linux IIO framework	STM32Cube DAC driver	

3.2.3 Peripheral configuration

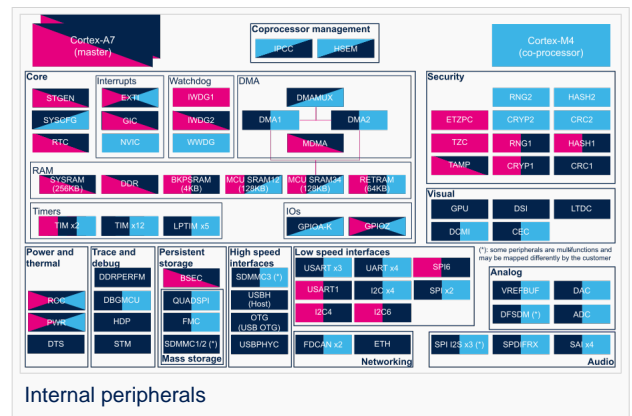
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via the STM32CubeMX tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information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

For the Linux kernel configuration, please refer to DAC device tree configuration and DAC Linux driver articles.

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.

Domain	Peripheral	Runtime allocation			Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)		
Analog	DAC	DAC			Assignment (single choice)



4 References

- TIM internal peripheral
- LPTIM internal peripheral
- EXTI internal peripheral
- RCC internal peripheral
- DMA internal peripheral
- VREFBUF internal peripheral
- Regulator overview

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

low-power timer (STM32 specific)

External Interrupt

Low Speed Internal oscillator (STM32 clock source)

Reset and Clock Control

Direct Memory Access

voltage reference buffer (STM32 specific)

Arm[®] is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



Cortex[®]

Linux[®] is a registered trademark of Linus Torvalds.

Open Portable Trusted Execution Environment

Stable: 25.09.2020 - 09:43 / Revision: 25.09.2020 - 09:37

Contents

1 Article purpose	79
2 Peripheral overview	80
2.1 Features	80
2.2 Security support	80
3 Peripheral usage and associated software	81
3.1 Boot time	81
3.2 Runtime	81
3.2.1 Overview	81
3.2.2 Software frameworks	81
3.2.3 Peripheral configuration	81
3.2.4 Peripheral assignment	81
4 References	83



1 Article purpose

The purpose of this article is to

- briefly introduce the DAC 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 DAC peripheral.



2 Peripheral overview

The **DAC** peripheral is a voltage output digital-to-analog converter.

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.

- It may be configured in **8- or 12-bit mode** (data could be left- or right-aligned)
- It has **two output channels**, each with its **own converter**
- The dual DAC channel mode could be done **independently or simultaneously**
- It has built-in noise and triangle **waveform generator** and supports **triggers** for conversions, using: TIM^[1], LPTIM^[2] or EXTI^[3]
- The DAC output buffer allows a high drive output current
- It can operate under **Normal mode** or low-power **Sample and Hold mode** (uses LSI clock, from RCC^[4]).
- It may be used in conjunction with the **DMA^[5]** controller (with underrun error detection)
- The **common reference voltage**, can be provided by either VREFBUF^[6] or any other external regulator^[7] wired to VREF+ pin.

2.2 Security support

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



3 Peripheral usage and associated software

3.1 Boot time

The DAC is not used at boot time.

3.2 Runtime

3.2.1 Overview

The DAC instances can be allocated to:

- the Arm®Cortex®-A7 non-secure core to be used under Linux® with the IIO framework
- or
- the Arm®Cortex®-M4 core to be used with STM32Cube Package with STM32Cube DAC driver.

The Peripheral assignment chapter describes which peripheral instance can be assigned to which context.

3.2.2 Software frameworks

Domain	Peripheral	Software components		Comment
OP-TEE	Linux	STM32Cube		
Analog	DAC	Linux IIO framework	STM32Cube DAC driver	

3.2.3 Peripheral configuration

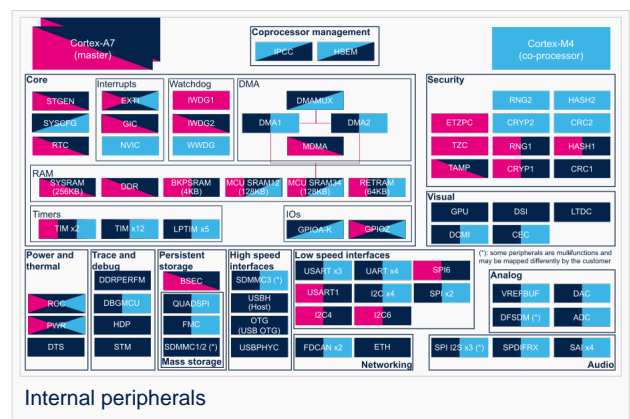
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via the STM32CubeMX tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information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

For the Linux kernel configuration, please refer to DAC device tree configuration and DAC Linux driver articles.

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.

Domain	Peripheral	Runtime allocation			Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)		
Analog	DAC	DAC			Assignment (single choice)



4 References

- TIM internal peripheral
- LPTIM internal peripheral
- EXTI internal peripheral
- RCC internal peripheral
- DMA internal peripheral
- VREFBUF internal peripheral
- Regulator overview

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

low-power timer (STM32 specific)

External Interrupt

Low Speed Internal oscillator (STM32 clock source)

Reset and Clock Control

Direct Memory Access

voltage reference buffer (STM32 specific)

Arm[®] is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere. **arm**

Cortex[®]

Linux[®] is a registered trademark of Linus Torvalds.

Open Portable Trusted Execution Environment