



LPTIM internal peripheral



Contents

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



A quality version of this page, approved on 5 January 2021, was based off this revision.

Warning

This section has not been yet updated/tested for/with the STM32 MPU ecosystem-v3 flow. Information provided in this section may not be not fully applicable for v3 and you may need to proceed to some adaptations. We apologize for this inconvenient.

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 **LPTIM** 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 LPTIM peripheral



2 Peripheral overview

The **LPTIM** peripheral is a single channel low-power timer unit, that can continue to run even during low power modes when it selects a clock source that remains active in **RCC**.

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.

The LPTIM peripheral is available in different configurations, depending on the selected instance :

- LPTIM1 and LPTIM2 can act as PWM, quadrature encoder^[1], external event counter, trigger source for other internal peripherals like: ADC^[2], DAC^[3], DFSDM^[4].
- LPTIM3 can act as PWM, external event counter, trigger source for other internal peripherals like ADC^[2], DFSDM^[4].
- LPTIM4 and LPTIM5 can act as PWM.

2.2 Security support

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



3 Peripheral usage and associated software

3.1 Boot time

The LPTIM is not used at boot time.

3.2 Runtime

3.2.1 Overview

LPTIM instances can be allocated to:

- the Arm® Cortex®-A7 non-secure to be used under Linux® with PWM and/or IIO frameworks.
- or
- the Arm® Cortex®-M4 to be used with STM32Cube MPU Package with LPTIM HAL driver

3.2.2 Software frameworks

Domain	Peripheral	Software components		Comment
OP-TEE	Linux	STM32Cube		
Core/Timers	LPTIM		Linux PWM framework Linux IIO framework	STM32Cube LPTIM driver

3.2.3 Peripheral configuration

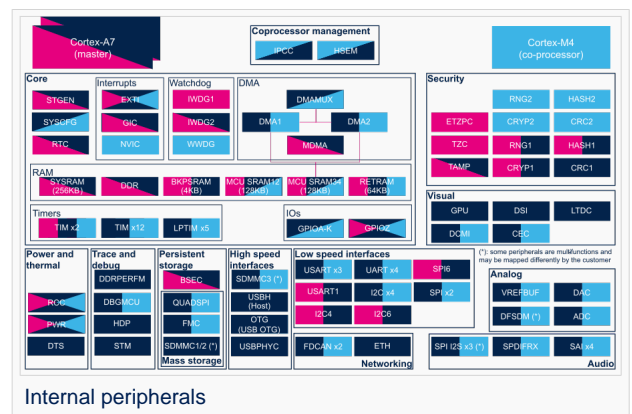
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via STM32CubeMX tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information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

For Linux kernel configuration, please refer to LPTIM device tree configuration and STM32 LPTIM 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	Periphera	Runtime allocation			Comment
Instance	I Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)		
Core/Timers	LPTIM	LPTIM1			Assignment (single choice)
		LPTIM2			Assignment (single choice)
		LPTIM3			Assignment (single choice)
		LPTIM4			Assignment (single choice)
		LPTIM5			Assignment (single choice)



4 References

- Quadrature encoder
- 2.02.1 ADC internal peripheral
- DAC internal peripheral
- 4.04.1 DFSDM internal peripheral

Stable: 16.11.2021 - 14:47 / Revision: 16.11.2021 - 14:47

Warning

This section has not been yet updated/tested for/with the STM32 MPU ecosystem-v3 flow. Information provided in this section may not be not fully applicable for v3 and you may need to proceed to some adaptations. We apologize for this inconvenient.

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 **LPTIM** 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 LPTIM peripheral



2 Peripheral overview

The **LPTIM** peripheral is a single channel low-power timer unit, that can continue to run even during low power modes when it selects a clock source that remains active in **RCC**.

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.

The LPTIM peripheral is available in different configurations, depending on the selected instance :

- LPTIM1 and LPTIM2 can act as PWM, quadrature encoder^[1], external event counter, trigger source for other internal peripherals like: ADC^[2], DAC^[3], DFSDM^[4].
- LPTIM3 can act as PWM, external event counter, trigger source for other internal peripherals like ADC^[2], DFSDM^[4].
- LPTIM4 and LPTIM5 can act as PWM.

2.2 Security support

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



3 Peripheral usage and associated software

3.1 Boot time

The LPTIM is not used at boot time.

3.2 Runtime

3.2.1 Overview

LPTIM instances can be allocated to:

- the Arm® Cortex®-A7 non-secure to be used under Linux® with PWM and/or IIO frameworks.
- or
- the Arm® Cortex®-M4 to be used with STM32Cube MPU Package with LPTIM HAL driver

3.2.2 Software frameworks

Domain	Peripheral	Software components		Comment
OP-TEE	Linux	STM32Cube		
Core/Timers	LPTIM		Linux PWM framework Linux IIO framework	STM32Cube LPTIM driver

3.2.3 Peripheral configuration

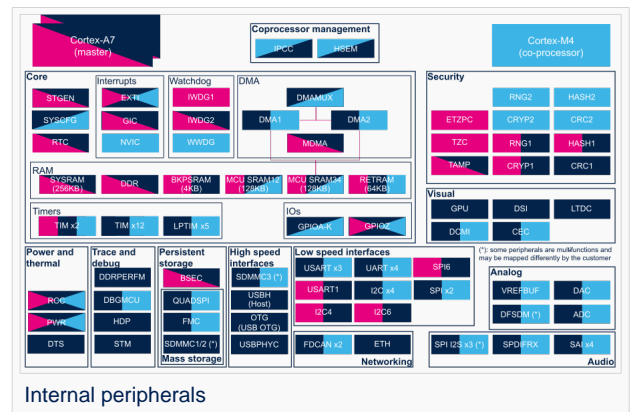
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via STM32CubeMX tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information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

For Linux kernel configuration, please refer to LPTIM device tree configuration and STM32 LPTIM 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	Periphera	Runtime allocation			Comment
Instance	I Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)		
Core/Timers	LPTIM	LPTIM1			Assignment (single choice)
		LPTIM2			Assignment (single choice)
		LPTIM3			Assignment (single choice)
		LPTIM4			Assignment (single choice)
		LPTIM5			Assignment (single choice)



4 References

- Quadrature encoder
- 2.02.1 ADC internal peripheral
- DAC internal peripheral
- 4.04.1 DFSDM internal peripheral

Stable: 12.02.2020 - 16:42 / Revision: 12.02.2020 - 16:41

Warning

This section has not been yet updated/tested for/with the STM32 MPU ecosystem-v3 flow. Information provided in this section may not be not fully applicable for v3 and you may need to proceed to some adaptations. We apologize for this inconvenient.

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 **LPTIM** 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 LPTIM peripheral



2 Peripheral overview

The **LPTIM** peripheral is a single channel low-power timer unit, that can continue to run even during low power modes when it selects a clock source that remains active in **RCC**.

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.

The LPTIM peripheral is available in different configurations, depending on the selected instance :

- LPTIM1 and LPTIM2 can act as PWM, quadrature encoder^[1], external event counter, trigger source for other internal peripherals like: ADC^[2], DAC^[3], DFSDM^[4].
- LPTIM3 can act as PWM, external event counter, trigger source for other internal peripherals like ADC^[2], DFSDM^[4].
- LPTIM4 and LPTIM5 can act as PWM.

2.2 Security support

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



3 Peripheral usage and associated software

3.1 Boot time

The LPTIM is not used at boot time.

3.2 Runtime

3.2.1 Overview

LPTIM instances can be allocated to:

- the Arm® Cortex®-A7 non-secure to be used under Linux® with PWM and/or IIO frameworks.
- or
- the Arm® Cortex®-M4 to be used with STM32Cube MPU Package with LPTIM HAL driver

3.2.2 Software frameworks

Domain	Peripheral	Software components		Comment
OP-TEE	Linux	STM32Cube		
Core/Timers	LPTIM		Linux PWM framework Linux IIO framework	STM32Cube LPTIM driver

3.2.3 Peripheral configuration

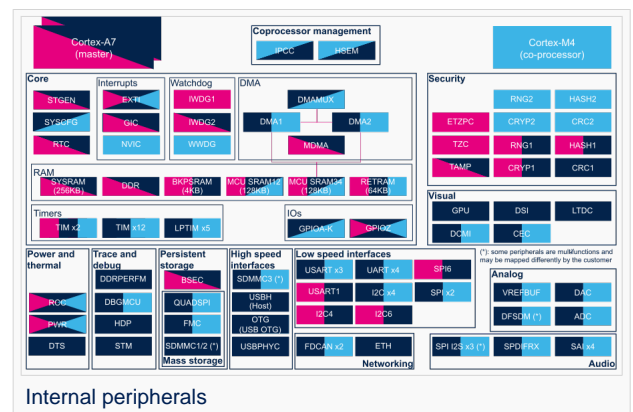
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via STM32CubeMX tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information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

For Linux kernel configuration, please refer to LPTIM device tree configuration and STM32 LPTIM 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	Periphera	Runtime allocation			Comment
Instance	I Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)		
Core/Timers	LPTIM	LPTIM1			Assignment (single choice)
		LPTIM2			Assignment (single choice)
		LPTIM3			Assignment (single choice)
		LPTIM4			Assignment (single choice)
		LPTIM5			Assignment (single choice)



4 References

- Quadrature encoder
- 2.02.1 ADC internal peripheral
- DAC internal peripheral
- 4.04.1 DFSDM internal peripheral

Stable: 12.02.2020 - 16:42 / Revision: 12.02.2020 - 16:41

Warning

This section has not been yet updated/tested for/with the STM32 MPU ecosystem-v3 flow. Information provided in this section may not be not fully applicable for v3 and you may need to proceed to some adaptations. We apologize for this inconvenient.

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 **LPTIM** 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 LPTIM peripheral



2 Peripheral overview

The **LPTIM** peripheral is a single channel low-power timer unit, that can continue to run even during low power modes when it selects a clock source that remains active in **RCC**.

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.

The LPTIM peripheral is available in different configurations, depending on the selected instance :

- LPTIM1 and LPTIM2 can act as PWM, quadrature encoder^[1], external event counter, trigger source for other internal peripherals like: ADC^[2], DAC^[3], DFSDM^[4].
- LPTIM3 can act as PWM, external event counter, trigger source for other internal peripherals like ADC^[2], DFSDM^[4].
- LPTIM4 and LPTIM5 can act as PWM.

2.2 Security support

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



3 Peripheral usage and associated software

3.1 Boot time

The LPTIM is not used at boot time.

3.2 Runtime

3.2.1 Overview

LPTIM instances can be allocated to:

- the Arm® Cortex®-A7 non-secure to be used under Linux® with PWM and/or IIO frameworks.
- or
- the Arm® Cortex®-M4 to be used with STM32Cube MPU Package with LPTIM HAL driver

3.2.2 Software frameworks

Domain	Peripheral	Software components		Comment
OP-TEE	Linux	STM32Cube		
Core/Timers	LPTIM		Linux PWM framework Linux IIO framework	STM32Cube LPTIM driver

3.2.3 Peripheral configuration

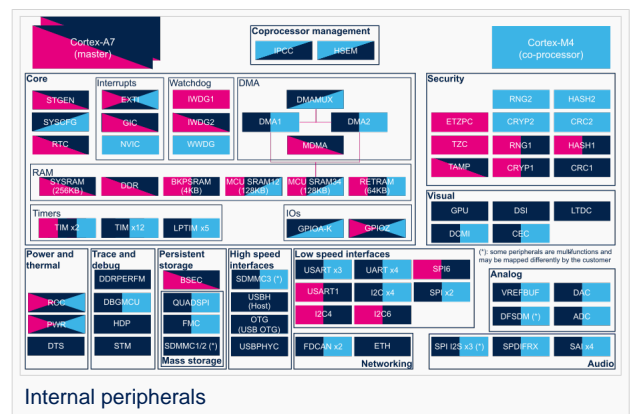
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via STM32CubeMX tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information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

For Linux kernel configuration, please refer to LPTIM device tree configuration and STM32 LPTIM 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	Periphera	Runtime allocation			Comment
Instance	I Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)		
Core/Timers	LPTIM	LPTIM1			Assignment (single choice)
		LPTIM2			Assignment (single choice)
		LPTIM3			Assignment (single choice)
		LPTIM4			Assignment (single choice)
		LPTIM5			Assignment (single choice)



4 References

- Quadrature encoder
- 2.02.1 ADC internal peripheral
- DAC internal peripheral
- 4.04.1 DFSDM internal peripheral

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

Warning

This section has not been yet updated/tested for/with the STM32 MPU ecosystem-v3 flow. Information provided in this section may not be not fully applicable for v3 and you may need to proceed to some adaptations.
We apologize for this inconvenient.

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 **LPTIM** 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 LPTIM peripheral



2 Peripheral overview

The **LPTIM** peripheral is a single channel low-power timer unit, that can continue to run even during low power modes when it selects a clock source that remains active in **RCC**.

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.

The LPTIM peripheral is available in different configurations, depending on the selected instance :

- LPTIM1 and LPTIM2 can act as PWM, quadrature encoder^[1], external event counter, trigger source for other internal peripherals like: ADC^[2], DAC^[3], DFSDM^[4].
- LPTIM3 can act as PWM, external event counter, trigger source for other internal peripherals like ADC^[2], DFSDM^[4].
- LPTIM4 and LPTIM5 can act as PWM.

2.2 Security support

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



3 Peripheral usage and associated software

3.1 Boot time

The LPTIM is not used at boot time.

3.2 Runtime

3.2.1 Overview

LPTIM instances can be allocated to:

- the Arm® Cortex®-A7 non-secure to be used under Linux® with PWM and/or IIO frameworks.
- or
- the Arm® Cortex®-M4 to be used with STM32Cube MPU Package with LPTIM HAL driver

3.2.2 Software frameworks

Domain	Peripheral	Software components		Comment
OP-TEE	Linux	STM32Cube		
Core/Timers	LPTIM		Linux PWM framework Linux IIO framework	STM32Cube LPTIM driver

3.2.3 Peripheral configuration

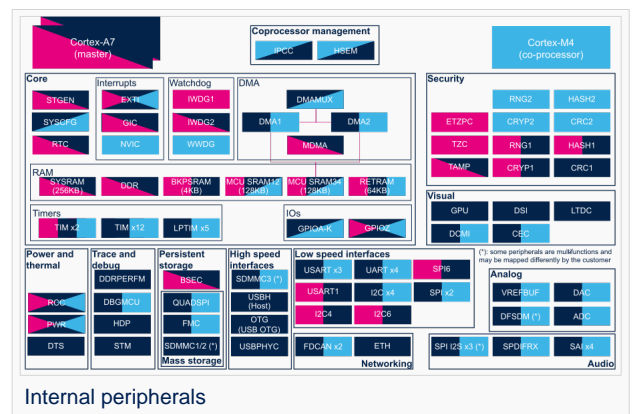
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via *STM32CubeMX* tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information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

For Linux kernel configuration, please refer to *LPTIM device tree configuration* and *STM32 LPTIM 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	Periphera	Runtime allocation			Comment
Instance	I Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)		
Core/Timers	LPTIM	LPTIM1			Assignment (single choice)
		LPTIM2			Assignment (single choice)
		LPTIM3			Assignment (single choice)
		LPTIM4			Assignment (single choice)
		LPTIM5			Assignment (single choice)



4 References

- Quadrature encoder
- 2.02.1 ADC internal peripheral
- DAC internal peripheral
- 4.04.1 DFSDM internal peripheral

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

Warning

This section has not been yet updated/tested for/with the STM32 MPU ecosystem-v3 flow. Information provided in this section may not be not fully applicable for v3 and you may need to proceed to some adaptations. We apologize for this inconvenient.

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 **LPTIM** 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 LPTIM peripheral



2 Peripheral overview

The **LPTIM** peripheral is a single channel low-power timer unit, that can continue to run even during low power modes when it selects a clock source that remains active in **RCC**.

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.

The LPTIM peripheral is available in different configurations, depending on the selected instance :

- LPTIM1 and LPTIM2 can act as PWM, quadrature encoder^[1], external event counter, trigger source for other internal peripherals like: ADC^[2], DAC^[3], DFSDM^[4].
- LPTIM3 can act as PWM, external event counter, trigger source for other internal peripherals like ADC^[2], DFSDM^[4].
- LPTIM4 and LPTIM5 can act as PWM.

2.2 Security support

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



3 Peripheral usage and associated software

3.1 Boot time

The LPTIM is not used at boot time.

3.2 Runtime

3.2.1 Overview

LPTIM instances can be allocated to:

- the Arm® Cortex®-A7 non-secure to be used under Linux® with PWM and/or IIO frameworks.
- or
- the Arm® Cortex®-M4 to be used with STM32Cube MPU Package with LPTIM HAL driver

3.2.2 Software frameworks

Domain	Peripheral	Software components		Comment
OP-TEE	Linux	STM32Cube		
Core/Timers	LPTIM		Linux PWM framework Linux IIO framework	STM32Cube LPTIM driver

3.2.3 Peripheral configuration

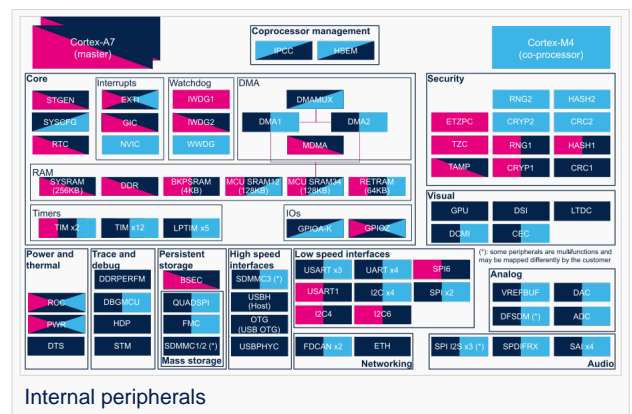
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via *STM32CubeMX* tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information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

For Linux kernel configuration, please refer to *LPTIM device tree configuration* and *STM32 LPTIM 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	Periphera	Runtime allocation			Comment
Instance	I Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)		
Core/Timers	LPTIM	LPTIM1			Assignment (single choice)
		LPTIM2			Assignment (single choice)
		LPTIM3			Assignment (single choice)
		LPTIM4			Assignment (single choice)
		LPTIM5			Assignment (single choice)



4 References

- Quadrature encoder
- 2.02.1 ADC internal peripheral
- DAC internal peripheral
- 4.04.1 DFSDM internal peripheral

Stable: 30.03.2021 - 13:09 / Revision: 29.03.2021 - 09:34

Warning

This section has not been yet updated/tested for/with the STM32 MPU ecosystem-v3 flow. Information provided in this section may not be not fully applicable for v3 and you may need to proceed to some adaptations. We apologize for this inconvenient.

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 **LPTIM** 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 LPTIM peripheral



2 Peripheral overview

The **LPTIM** peripheral is a single channel low-power timer unit, that can continue to run even during low power modes when it selects a clock source that remains active in **RCC**.

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.

The LPTIM peripheral is available in different configurations, depending on the selected instance :

- LPTIM1 and LPTIM2 can act as PWM, quadrature encoder^[1], external event counter, trigger source for other internal peripherals like: ADC^[2], DAC^[3], DFSDM^[4].
- LPTIM3 can act as PWM, external event counter, trigger source for other internal peripherals like ADC^[2], DFSDM^[4].
- LPTIM4 and LPTIM5 can act as PWM.

2.2 Security support

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



3 Peripheral usage and associated software

3.1 Boot time

The LPTIM is not used at boot time.

3.2 Runtime

3.2.1 Overview

LPTIM instances can be allocated to:

- the Arm® Cortex®-A7 non-secure to be used under Linux® with PWM and/or IIO frameworks.
- or
- the Arm® Cortex®-M4 to be used with STM32Cube MPU Package with LPTIM HAL driver

3.2.2 Software frameworks

Domain	Peripheral	Software components		Comment
OP-TEE	Linux	STM32Cube		
Core/Timers	LPTIM		Linux PWM framework Linux IIO framework	STM32Cube LPTIM driver

3.2.3 Peripheral configuration

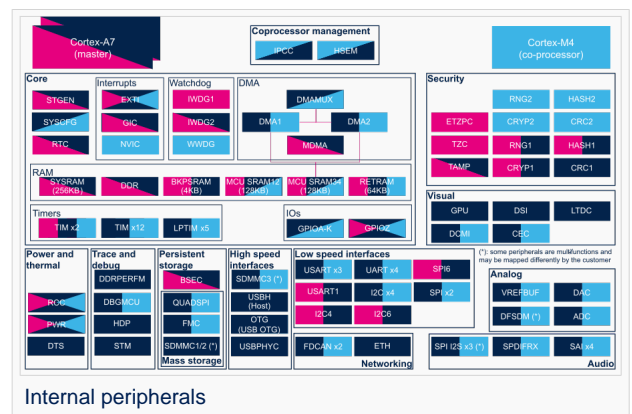
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via STM32CubeMX tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information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

For Linux kernel configuration, please refer to LPTIM device tree configuration and STM32 LPTIM 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	Periphera	Runtime allocation				Comment
Instance	I Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Core/Timers	LPTIM	LPTIM1				Assignment (single choice)
		LPTIM2				Assignment (single choice)
		LPTIM3				Assignment (single choice)
		LPTIM4				Assignment (single choice)
		LPTIM5				Assignment (single choice)



4 References

- Quadrature encoder
- 2.02.1 ADC internal peripheral
- DAC internal peripheral
- 4.04.1 DFSDM internal peripheral

Stable: 30.03.2021 - 13:05 / Revision: 29.03.2021 - 09:33

Warning

This section has not been yet updated/tested for/with the STM32 MPU ecosystem-v3 flow. Information provided in this section may not be not fully applicable for v3 and you may need to proceed to some adaptations. We apologize for this inconvenient.

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 **LPTIM** 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 LPTIM peripheral



2 Peripheral overview

The **LPTIM** peripheral is a single channel low-power timer unit, that can continue to run even during low power modes when it selects a clock source that remains active in **RCC**.

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.

The LPTIM peripheral is available in different configurations, depending on the selected instance :

- LPTIM1 and LPTIM2 can act as PWM, quadrature encoder^[1], external event counter, trigger source for other internal peripherals like: ADC^[2], DAC^[3], DFSDM^[4].
- LPTIM3 can act as PWM, external event counter, trigger source for other internal peripherals like ADC^[2], DFSDM^[4].
- LPTIM4 and LPTIM5 can act as PWM.

2.2 Security support

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



3 Peripheral usage and associated software

3.1 Boot time

The LPTIM is not used at boot time.

3.2 Runtime

3.2.1 Overview

LPTIM instances can be allocated to:

- the Arm[®] Cortex[®]-A7 non-secure to be used under Linux[®] with PWM and/or IIO frameworks.
- or
- the Arm[®] Cortex[®]-M4 to be used with STM32Cube MPU Package with LPTIM HAL driver

3.2.2 Software frameworks

Domain	Peripheral	Software components		Comment
OP-TEE	Linux	STM32Cube		
Core/Timers	LPTIM		Linux PWM framework Linux IIO framework	STM32Cube LPTIM driver

3.2.3 Peripheral configuration

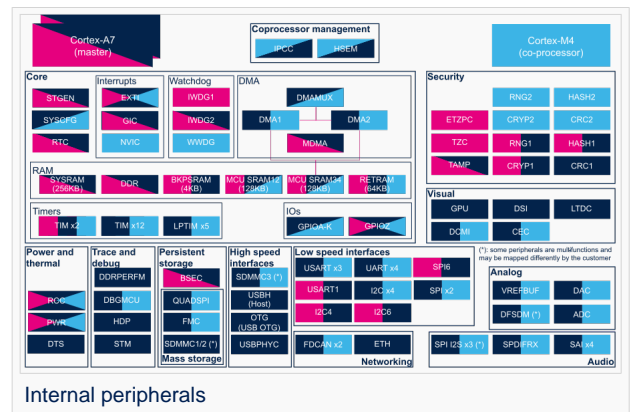
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via *STM32CubeMX* tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information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

For Linux kernel configuration, please refer to *LPTIM device tree configuration* and *STM32 LPTIM 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	Periphera	Runtime allocation			Comment
Instance	I Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)		
Core/Timers	LPTIM	LPTIM1			Assignment (single choice)
		LPTIM2			Assignment (single choice)
		LPTIM3			Assignment (single choice)
		LPTIM4			Assignment (single choice)
		LPTIM5			Assignment (single choice)



4 References

- Quadrature encoder
- 2.02.1 ADC internal peripheral
- DAC internal peripheral
- 4.04.1 DFSDM internal peripheral

Stable: 18.02.2021 - 13:27 / Revision: 18.02.2021 - 10:37

Warning

This section has not been yet updated/tested for/with the STM32 MPU ecosystem-v3 flow. Information provided in this section may not be not fully applicable for v3 and you may need to proceed to some adaptations.
We apologize for this inconvenient.

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 **LPTIM** 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 LPTIM peripheral



2 Peripheral overview

The **LPTIM** peripheral is a single channel low-power timer unit, that can continue to run even during low power modes when it selects a clock source that remains active in **RCC**.

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.

The LPTIM peripheral is available in different configurations, depending on the selected instance :

- LPTIM1 and LPTIM2 can act as PWM, quadrature encoder^[1], external event counter, trigger source for other internal peripherals like: ADC^[2], DAC^[3], DFSDM^[4].
- LPTIM3 can act as PWM, external event counter, trigger source for other internal peripherals like ADC^[2], DFSDM^[4].
- LPTIM4 and LPTIM5 can act as PWM.

2.2 Security support

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



3 Peripheral usage and associated software

3.1 Boot time

The LPTIM is not used at boot time.

3.2 Runtime

3.2.1 Overview

LPTIM instances can be allocated to:

- the Arm® Cortex®-A7 non-secure to be used under Linux® with PWM and/or IIO frameworks.
- or
- the Arm® Cortex®-M4 to be used with STM32Cube MPU Package with LPTIM HAL driver

3.2.2 Software frameworks

Domain	Peripheral	Software components		Comment
OP-TEE	Linux	STM32Cube		
Core/Timers	LPTIM		Linux PWM framework Linux IIO framework	STM32Cube LPTIM driver

3.2.3 Peripheral configuration

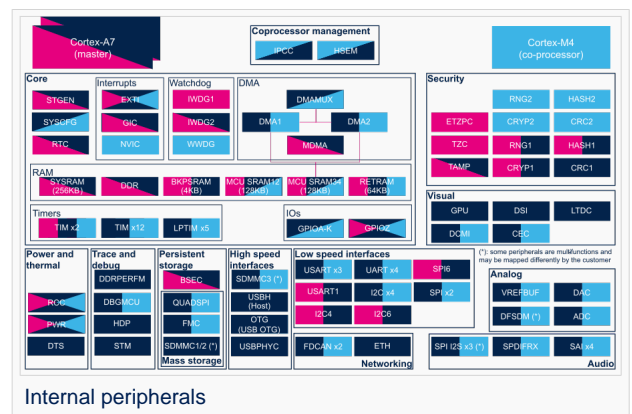
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via STM32CubeMX tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information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

For Linux kernel configuration, please refer to LPTIM device tree configuration and STM32 LPTIM 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	Periphera	Runtime allocation			Comment
Instance	I Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)		
Core/Timers	LPTIM	LPTIM1			Assignment (single choice)
		LPTIM2			Assignment (single choice)
		LPTIM3			Assignment (single choice)
		LPTIM4			Assignment (single choice)
		LPTIM5			Assignment (single choice)



4 References

- Quadrature encoder
- 2.02.1 ADC internal peripheral
- DAC internal peripheral
- 4.04.1 DFSDM internal peripheral

Stable: 01.12.2020 - 10:35 / Revision: 01.12.2020 - 09:20

Warning

This section has not been yet updated/tested for/with the STM32 MPU ecosystem-v3 flow. Information provided in this section may not be not fully applicable for v3 and you may need to proceed to some adaptations.
We apologize for this inconvenient.

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 **LPTIM** 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 LPTIM peripheral



2 Peripheral overview

The **LPTIM** peripheral is a single channel low-power timer unit, that can continue to run even during low power modes when it selects a clock source that remains active in **RCC**.

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.

The LPTIM peripheral is available in different configurations, depending on the selected instance :

- LPTIM1 and LPTIM2 can act as PWM, quadrature encoder^[1], external event counter, trigger source for other internal peripherals like: ADC^[2], DAC^[3], DFSDM^[4].
- LPTIM3 can act as PWM, external event counter, trigger source for other internal peripherals like ADC^[2], DFSDM^[4].
- LPTIM4 and LPTIM5 can act as PWM.

2.2 Security support

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



3 Peripheral usage and associated software

3.1 Boot time

The LPTIM is not used at boot time.

3.2 Runtime

3.2.1 Overview

LPTIM instances can be allocated to:

- the Arm® Cortex®-A7 non-secure to be used under Linux® with PWM and/or IIO frameworks.
- or
- the Arm® Cortex®-M4 to be used with STM32Cube MPU Package with LPTIM HAL driver

3.2.2 Software frameworks

Domain	Peripheral	Software components		Comment
OP-TEE	Linux	STM32Cube		
Core/Timers	LPTIM		Linux PWM framework Linux IIO framework	STM32Cube LPTIM driver

3.2.3 Peripheral configuration

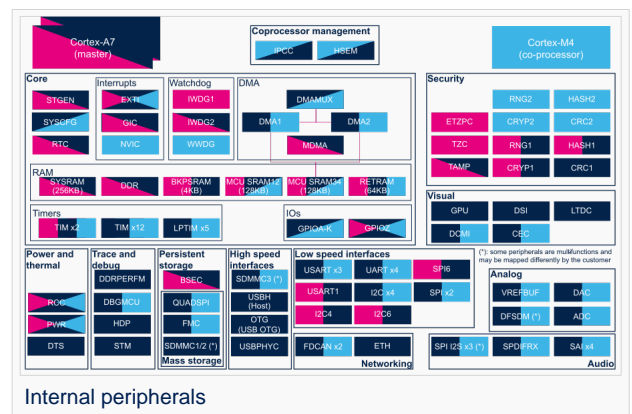
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via *STM32CubeMX* tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information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

For Linux kernel configuration, please refer to *LPTIM device tree configuration* and *STM32 LPTIM 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	Periphera	Runtime allocation			Comment
Instance	I Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)		
Core/Timers	LPTIM	LPTIM1			Assignment (single choice)
		LPTIM2			Assignment (single choice)
		LPTIM3			Assignment (single choice)
		LPTIM4			Assignment (single choice)
		LPTIM5			Assignment (single choice)



4 References

- Quadrature encoder
- 2.02.1 ADC internal peripheral
- DAC internal peripheral
- 4.04.1 DFSDM internal peripheral

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

Warning

This section has not been yet updated/tested for/with the STM32 MPU ecosystem-v3 flow. Information provided in this section may not be not fully applicable for v3 and you may need to proceed to some adaptations.
We apologize for this inconvenient.

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 **LPTIM** 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 LPTIM peripheral



2 Peripheral overview

The **LPTIM** peripheral is a single channel low-power timer unit, that can continue to run even during low power modes when it selects a clock source that remains active in **RCC**.

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.

The LPTIM peripheral is available in different configurations, depending on the selected instance :

- LPTIM1 and LPTIM2 can act as PWM, quadrature encoder^[1], external event counter, trigger source for other internal peripherals like: ADC^[2], DAC^[3], DFSDM^[4].
- LPTIM3 can act as PWM, external event counter, trigger source for other internal peripherals like ADC^[2], DFSDM^[4].
- LPTIM4 and LPTIM5 can act as PWM.

2.2 Security support

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



3 Peripheral usage and associated software

3.1 Boot time

The LPTIM is not used at boot time.

3.2 Runtime

3.2.1 Overview

LPTIM instances can be allocated to:

- the Arm® Cortex®-A7 non-secure to be used under Linux® with PWM and/or IIO frameworks.
- or
- the Arm® Cortex®-M4 to be used with STM32Cube MPU Package with LPTIM HAL driver

3.2.2 Software frameworks

Domain	Peripheral	Software components		Comment
OP-TEE	Linux	STM32Cube		
Core/Timers	LPTIM		Linux PWM framework Linux IIO framework	STM32Cube LPTIM driver

3.2.3 Peripheral configuration

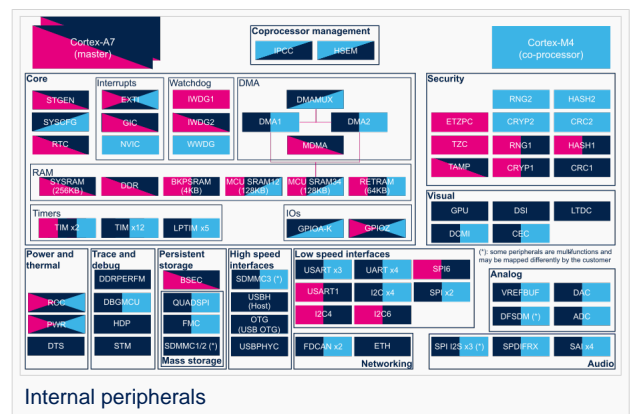
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via STM32CubeMX tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information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

For Linux kernel configuration, please refer to LPTIM device tree configuration and STM32 LPTIM 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	Periphera	Runtime allocation				Comment
Instance	I Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Core/Timers	LPTIM	LPTIM1				Assignment (single choice)
		LPTIM2				Assignment (single choice)
		LPTIM3				Assignment (single choice)
		LPTIM4				Assignment (single choice)
		LPTIM5				Assignment (single choice)



4 References

- Quadrature encoder
- 2.02.1 ADC internal peripheral
- DAC internal peripheral
- 4.04.1 DFSDM internal peripheral

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

Warning

This section has not been yet updated/tested for/with the STM32 MPU ecosystem-v3 flow. Information provided in this section may not be not fully applicable for v3 and you may need to proceed to some adaptations. We apologize for this inconvenient.

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 **LPTIM** 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 LPTIM peripheral



2 Peripheral overview

The **LPTIM** peripheral is a single channel low-power timer unit, that can continue to run even during low power modes when it selects a clock source that remains active in **RCC**.

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.

The LPTIM peripheral is available in different configurations, depending on the selected instance :

- LPTIM1 and LPTIM2 can act as PWM, quadrature encoder^[1], external event counter, trigger source for other internal peripherals like: ADC^[2], DAC^[3], DFSDM^[4].
- LPTIM3 can act as PWM, external event counter, trigger source for other internal peripherals like ADC^[2], DFSDM^[4].
- LPTIM4 and LPTIM5 can act as PWM.

2.2 Security support

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



3 Peripheral usage and associated software

3.1 Boot time

The LPTIM is not used at boot time.

3.2 Runtime

3.2.1 Overview

LPTIM instances can be allocated to:

- the Arm® Cortex®-A7 non-secure to be used under Linux® with PWM and/or IIO frameworks.
- or
- the Arm® Cortex®-M4 to be used with STM32Cube MPU Package with LPTIM HAL driver

3.2.2 Software frameworks

Domain	Peripheral	Software components		Comment
OP-TEE	Linux	STM32Cube		
Core/Timers	LPTIM		Linux PWM framework Linux IIO framework	STM32Cube LPTIM driver

3.2.3 Peripheral configuration

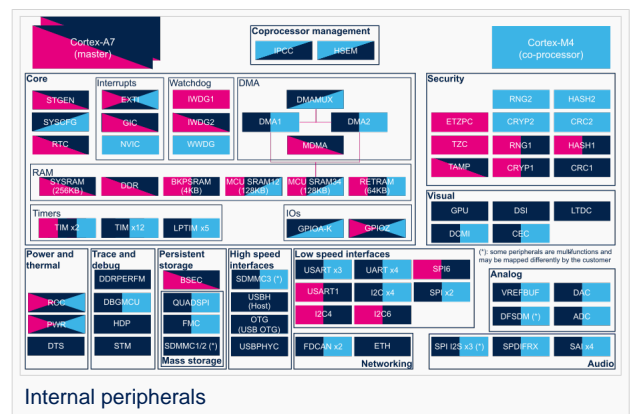
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via STM32CubeMX tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information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

For Linux kernel configuration, please refer to LPTIM device tree configuration and STM32 LPTIM 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	Periphera	Runtime allocation			Comment
Instance	I Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)		
Core/Timers	LPTIM	LPTIM1			Assignment (single choice)
		LPTIM2			Assignment (single choice)
		LPTIM3			Assignment (single choice)
		LPTIM4			Assignment (single choice)
		LPTIM5			Assignment (single choice)



4 References

- Quadrature encoder
- 2.02.1 ADC internal peripheral
- DAC internal peripheral
- 4.04.1 DFSDM internal peripheral

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

Warning

This section has not been yet updated/tested for/with the STM32 MPU ecosystem-v3 flow. Information provided in this section may not be not fully applicable for v3 and you may need to proceed to some adaptations.
We apologize for this inconvenient.

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 **LPTIM** 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 LPTIM peripheral



2 Peripheral overview

The **LPTIM** peripheral is a single channel low-power timer unit, that can continue to run even during low power modes when it selects a clock source that remains active in **RCC**.

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.

The LPTIM peripheral is available in different configurations, depending on the selected instance :

- LPTIM1 and LPTIM2 can act as PWM, quadrature encoder^[1], external event counter, trigger source for other internal peripherals like: ADC^[2], DAC^[3], DFSDM^[4].
- LPTIM3 can act as PWM, external event counter, trigger source for other internal peripherals like ADC^[2], DFSDM^[4].
- LPTIM4 and LPTIM5 can act as PWM.

2.2 Security support

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



3 Peripheral usage and associated software

3.1 Boot time

The LPTIM is not used at boot time.

3.2 Runtime

3.2.1 Overview

LPTIM instances can be allocated to:

- the Arm® Cortex®-A7 non-secure to be used under Linux® with PWM and/or IIO frameworks.
- or
- the Arm® Cortex®-M4 to be used with STM32Cube MPU Package with LPTIM HAL driver

3.2.2 Software frameworks

Domain	Peripheral	Software components		Comment
OP-TEE	Linux	STM32Cube		
Core/Timers	LPTIM		Linux PWM framework Linux IIO framework	STM32Cube LPTIM driver

3.2.3 Peripheral configuration

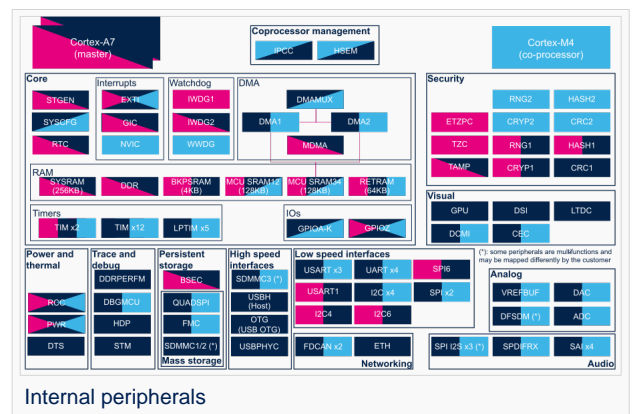
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via STM32CubeMX tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information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

For Linux kernel configuration, please refer to LPTIM device tree configuration and STM32 LPTIM 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	Periphera	Runtime allocation				Comment
Instance	I Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Core/Timers	LPTIM	LPTIM1				Assignment (single choice)
		LPTIM2				Assignment (single choice)
		LPTIM3				Assignment (single choice)
		LPTIM4				Assignment (single choice)
		LPTIM5				Assignment (single choice)



4 References

- Quadrature encoder
- 2.02.1 ADC internal peripheral
- DAC internal peripheral
- 4.04.1 DFSDM internal peripheral

Stable: 17.11.2021 - 16:41 / Revision: 17.11.2021 - 10:47

Warning

This section has not been yet updated/tested for/with the STM32 MPU ecosystem-v3 flow. Information provided in this section may not be not fully applicable for v3 and you may need to proceed to some adaptations.
We apologize for this inconvenient.

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 **LPTIM** 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 LPTIM peripheral



2 Peripheral overview

The **LPTIM** peripheral is a single channel low-power timer unit, that can continue to run even during low power modes when it selects a clock source that remains active in **RCC**.

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.

The LPTIM peripheral is available in different configurations, depending on the selected instance :

- LPTIM1 and LPTIM2 can act as PWM, quadrature encoder^[1], external event counter, trigger source for other internal peripherals like: ADC^[2], DAC^[3], DFSDM^[4].
- LPTIM3 can act as PWM, external event counter, trigger source for other internal peripherals like ADC^[2], DFSDM^[4].
- LPTIM4 and LPTIM5 can act as PWM.

2.2 Security support

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



3 Peripheral usage and associated software

3.1 Boot time

The LPTIM is not used at boot time.

3.2 Runtime

3.2.1 Overview

LPTIM instances can be allocated to:

- the Arm® Cortex®-A7 non-secure to be used under Linux® with PWM and/or IIO frameworks.
- or
- the Arm® Cortex®-M4 to be used with STM32Cube MPU Package with LPTIM HAL driver

3.2.2 Software frameworks

Domain	Peripheral	Software components		Comment
OP-TEE	Linux	STM32Cube		
Core/Timers	LPTIM		Linux PWM framework Linux IIO framework	STM32Cube LPTIM driver

3.2.3 Peripheral configuration

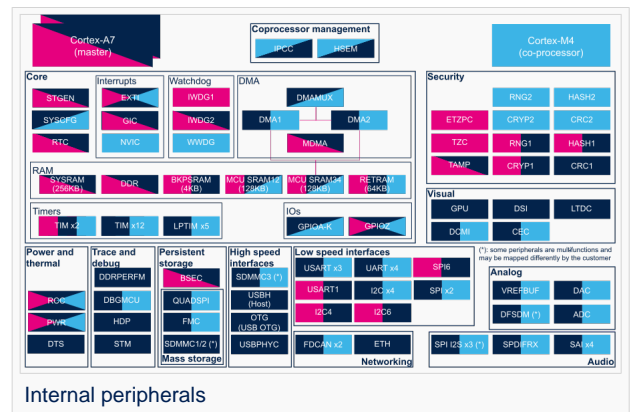
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via STM32CubeMX tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information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

For Linux kernel configuration, please refer to LPTIM device tree configuration and STM32 LPTIM 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	Periphera	Runtime allocation				Comment
Instance	I Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Core/Timers	LPTIM	LPTIM1				Assignment (single choice)
		LPTIM2				Assignment (single choice)
		LPTIM3				Assignment (single choice)
		LPTIM4				Assignment (single choice)
		LPTIM5				Assignment (single choice)



4 References

- Quadrature encoder
- 2.02.1 ADC internal peripheral
- DAC internal peripheral
- 4.04.1 DFSDM internal peripheral

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

Warning

This section has not been yet updated/tested for/with the STM32 MPU ecosystem-v3 flow. Information provided in this section may not be not fully applicable for v3 and you may need to proceed to some adaptations. We apologize for this inconvenient.

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 **LPTIM** 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 LPTIM peripheral



2 Peripheral overview

The **LPTIM** peripheral is a single channel low-power timer unit, that can continue to run even during low power modes when it selects a clock source that remains active in **RCC**.

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.

The LPTIM peripheral is available in different configurations, depending on the selected instance :

- LPTIM1 and LPTIM2 can act as PWM, quadrature encoder^[1], external event counter, trigger source for other internal peripherals like: ADC^[2], DAC^[3], DFSDM^[4].
- LPTIM3 can act as PWM, external event counter, trigger source for other internal peripherals like ADC^[2], DFSDM^[4].
- LPTIM4 and LPTIM5 can act as PWM.

2.2 Security support

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



3 Peripheral usage and associated software

3.1 Boot time

The LPTIM is not used at boot time.

3.2 Runtime

3.2.1 Overview

LPTIM instances can be allocated to:

- the Arm® Cortex®-A7 non-secure to be used under Linux® with PWM and/or IIO frameworks.
- or
- the Arm® Cortex®-M4 to be used with STM32Cube MPU Package with LPTIM HAL driver

3.2.2 Software frameworks

Domain	Peripheral	Software components		Comment
OP-TEE	Linux	STM32Cube		
Core/Timers	LPTIM		Linux PWM framework Linux IIO framework	STM32Cube LPTIM driver

3.2.3 Peripheral configuration

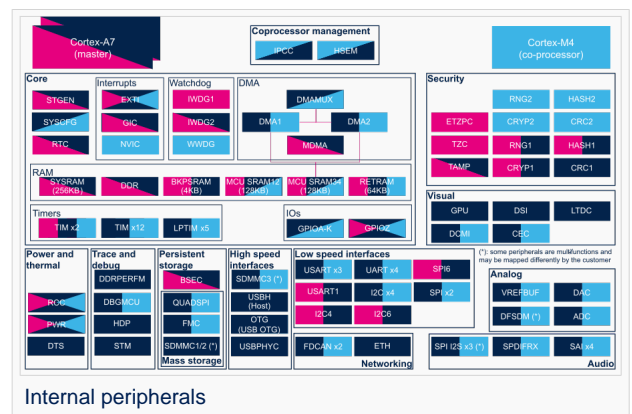
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via STM32CubeMX tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information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

For Linux kernel configuration, please refer to LPTIM device tree configuration and STM32 LPTIM 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	Periphera	Runtime allocation			Comment
Instance	I Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)		
Core/Timers	LPTIM	LPTIM1			Assignment (single choice)
		LPTIM2			Assignment (single choice)
		LPTIM3			Assignment (single choice)
		LPTIM4			Assignment (single choice)
		LPTIM5			Assignment (single choice)



4 References

- Quadrature encoder
- 2.02.1 ADC internal peripheral
- DAC internal peripheral
- 4.04.1 DFSDM internal peripheral