LPTIM internal peripheral

Contents

Stable: [illegible] / Revision: [illegible]

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

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

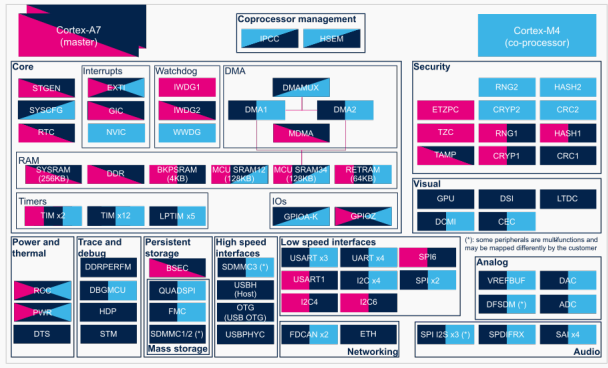
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration by itself can be performed via STM32CubeMX tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information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



Internal peripherals

Refer to How to assign an internal peripheral to a runtime context for more information on how to assign peripherals manually or via STM32CubeMX.
The present chapter describes STMicroelectronics recommendations or choice of implementation. Additional possiblities might be described in STM32MP15 reference manuals.

Domain	Periphera l	Runtime allocation				Comment
Instance	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