Category:Timers peripherals

Contents

Stable: 17.06.2020 - 13:27 / Revision: 16.01.2020 - 12:54

A quality version of this page, approved on *17 June 2020*, was based off this revision.

This category groups together all articles related to the **timers** internal peripherals (hardware blocks) embedded in the STM32 MPUs microprocessor devices.

Pages in category "Timers peripherals"

The following 2 pages are in this category, out of 2 total.

- LPTIM internal peripheral
- TIM internal peripheral

Stable: 30.03.2021 - 15:41 / Revision: 29.03.2021 - 09:57

A quality version of this page, approved on *30 March 2021*, was based off this revision.

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, IIO, *Counter* or/and *Clock events* 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 frameworks			Comment
		Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)	
Core/Timers	LPTIM		PWM framework, IIO framework, *Counter* framework, *Clock events* 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.

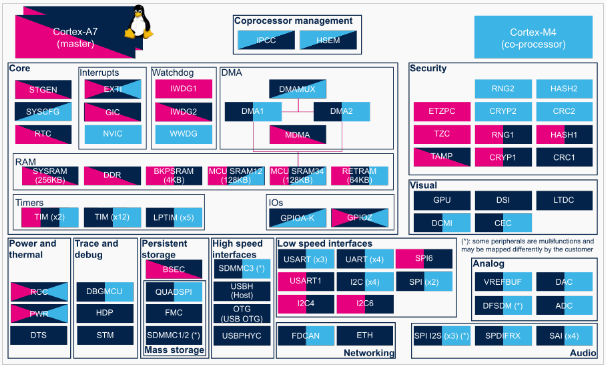
For Linux kernel configuration, please refer to LPTIM device tree configuration and STM32 LPTIM Linux driver articles.

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 possiblities might be described in STM32MP15 reference manuals
.



Internal peripherals

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

low-power timer (STM32 specific)

Pulse Width Modulation

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

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

Digital Filter for Sigma-Delta Modulator

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.

Microprocessor Unit

Open Portable Trusted Execution Environment

Stable: 30.03.2021 - 15:47 / Revision: 29.03.2021 - 09:58

A quality version of this page, approved on *30 March 2021*, was based off this revision.

Contents

1 Article purpose

The purpose of this article is to

- briefly introduce the **TIM** 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 TIM peripheral

2 Peripheral overview

The TIM peripheral is a multi-channel timer unit, available in various configurations, depending on the instance used. There are basically following categories: advanced-control timers, general-purpose timers and basic timers.

The TIM can provide: PWM with complementary output and dead-time insertion, break detection, input capture[1], quadrature encoder[2] interface (typically used for rotary encoders), trigger source for other internal peripherals like: ADC[3], DAC[4], DFSDM[5].

2.1 Features

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

- TIM1 and TIM8 are advanced-control timers, with 6 independent channels.
- TIM2, TIM3, TIM4 and TIM5 are general-purpose timers, with 4 independent channels.
- TIM12, TIM13 and TIM14 are general-purpose timers, with 2 (TIM12) or 1 (TIM13 and TIM14) independent channels.
- TIM15, TIM16 and TIM17 are also general-purpose timers, with 2 (TIM15) or 1 (TIM16 and TIM17) independent channels.

Compare to TIM12, TIM13 and TIM14, this configuration brings some features that are very useful for motor control (like break function, DMA burst mode control, complementary output with dead-time insertion, ...)

- TIM6 and TIM7 are basic timers

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.

2.2 Security support

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

3 Peripheral usage and associated software

3.1 Boot time

The TIM is not used at boot time.

3.2 Runtime

3.2.1 Overview

TIM12 and/or **TIM15** can be allocated to:

- the Arm®Cortex®-A7 secure core to be controlled in the secure monitor (TF-A or OP-TEE), to perform HSI and CSI calibrations[6] in RCC.

All TIM instances can be allocated to:

- the Arm®Cortex®-A7 non-secure to be controlled in Linux® by the PWM, the IIO, and/or the *Counter* frameworks.

or

- the Arm®Cortex®-M4 to be controlled in STM32Cube MPU Package by TIM HAL driver

Note that RCC[7] owns one prescaler per **TIM group** corresponding to **APB1** and **APB2** buses: TIMG1PRE and TIMG2PRE, respectively. The allocation to Cortex-A7 or the Cortex-M4 should ideally be done on a per group basis to get independent clocking setup on each side, this is why the TIM instances groups are shown in the summary table below (#Peripheral assignment).

3.2.2 Software frameworks

Domain	Peripheral	Software frameworks			Comment
Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Core/Timers	TIM	TF-A TIM driver OP-TEE TIM driver	PWM framework IIO framework, *Counter* framework	STM32Cube TIM 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.

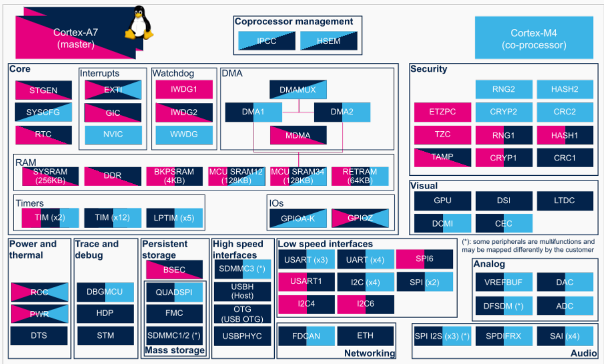
For Linux kernel configuration, please refer to TIM device tree configuration and TIM Linux driver articles.

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 possiblities might be described in STM32MP15 reference manuals
.



Internal peripherals

Domain	Peripheral	Runtime allocation				Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
		TIM1 (APB2 group)				Assignment (single choice)
		TIM2 (APB1 group)				Assignment (single choice)
		TIM3 (APB1 group)				Assignment (single choice)
		TIM4 (APB1 group)				Assignment (single choice)
		TIM5 (APB1 group)				Assignment (single choice)
		TIM6 (APB1 group)				Assignment (single choice)
		TIM7 (APB1 group)				Assignment (single choice)

Domain	Periphera	Runtime allocation				Comment
Core/Timers	TIM	TIM8 (APB2 group)				Assignment (single choice)
		TIM12 (APB1 group)				Assignment (single choice)
		TIM13 (APB1 group)				Assignment (single choice)
		TIM14 (APB1 group)				Assignment (single choice)
		TIM15 (APB2 group)				Assignment (single choice)
		TIM16 (APB2 group)				Assignment (single choice)
		TIM17 (APB2 group)				Assignment (single choice)

4 How to go further

STM32 cross-series timer overview[8] application note.

5 References

- Input capture
- Quadrature encoder
- ADC internal peripheral
- DAC internal peripheral
- DFSDM internal peripheral
- How to activate HSI and CSI oscillators calibration
- RCC internal peripheral
- STM32 cross-series timer overview application note

Pulse Width Modulation

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

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

Digital Filter for Sigma-Delta Modulator

Direct Memory Access

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

Cortex®

High Speed Internal oscillator (STM32 clock source) or High Speed Synchronous Serial Interface (MIPI® Alliance standard)

Multi Speed Internal oscillator (STM32 clock source)

Linux® is a registered trademark of Linus Torvalds.

Microprocessor Unit

Reset and Clock Control

Open Portable Trusted Execution Environment