



## SPDIFRX internal peripheral



A quality version of this page, approved on 11 February 2019, was based off this revision.

Template:ArticleMainWriter Template:ArticleApprovedVersion

## Contents

1 Article purpose .....	3
2 Peripheral overview .....	4
2.1 Features .....	4
2.2 Security support .....	4
3 Peripheral usage and associated software .....	5
3.1 Boot time .....	5
3.2 Runtime .....	5
3.2.1 Overview .....	5
3.2.2 Software frameworks .....	5
3.2.3 Peripheral configuration .....	5
3.2.3.1 Configuration in Cortex-A7 non-secure software .....	5
3.2.3.2 Arm® Cortex®-M4 software configuration .....	5
3.2.4 Peripheral assignment .....	5
4 How to go further .....	7
5 References .....	8



---

## 1 Article purpose

---

The purpose of this article is to:

- briefly introduce the SPDIFRX 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 SPDIFRX peripheral.



---

## 2 Peripheral overview

---

The **SPDIFRX** peripheral, is designed to receive an S/PDIF flow compliant with IEC-60958 and IEC-61937. The SPDIFRX receiver provides two separated paths to retrieve the audio data and the user and channel information.

### 2.1 Features

Refer to the [STM32MP15 reference manuals](#) for the complete feature list, and to the software components, introduced below, to see which features are implemented.

### 2.2 Security support

The SPDIFRX is a **non secure** peripheral.



## 3 Peripheral usage and associated software

### 3.1 Boot time

The SPDIFRX is not used at boot time.

### 3.2 Runtime

#### 3.2.1 Overview

The SPDIFRX instance can be allocated to:

- the Arm® Cortex®-A7 non-secure for use in Linux with ALSA framework
- the Cortex-M4 for use in STM32Cube with STM32Cube SPDIFRX driver

Chapter #Peripheral assignment exposes which instance can be assigned to which context.

#### 3.2.2 Software frameworks

Domain	Peripheral	Software components			Comment
OP-TEE	Linux	STM32Cube			
Audio	SPDIFRX		ALSA framework	STM32Cube SPDIFRX 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 can be done alone via the *STM32CubeMX* tool for all internal peripherals, and then manually completed (particularly for external peripherals), according to the information given in the corresponding software framework article.

##### 3.2.3.1 Configuration in Cortex-A7 non-secure software

When the Arm® Cortex®-A7 core operates in non-secure access mode, the SPDIFRX is controlled by the Linux kernel framework. Refer to the *SPDIFRX Linux driver* to drive the SPDIFRX through Linux kernel *ALSA framework*. Refer to *Soundcard configuration* and *SPDIFRX device tree configuration* to configure the SPDIFRX through Linux kernel device tree<sup>[1]</sup>.

##### 3.2.3.2 Arm® Cortex®-M4 software configuration

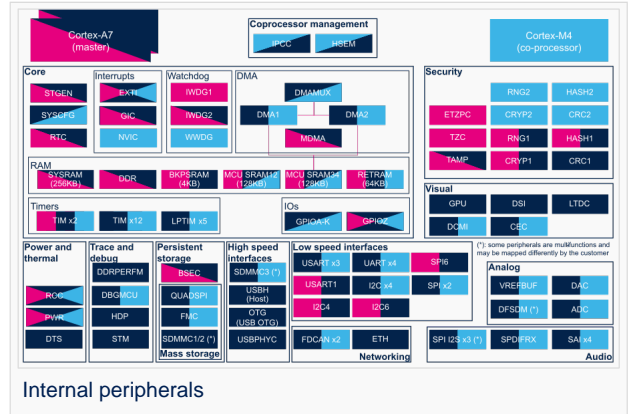
#### 3.2.4 Peripheral assignment

**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	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)	
Audio	SPDIFRX	SPDIFRX		Assignment (single choice)



---

## 4 How to go further

---

The STM32H7 SPDIFRX training <sup>[2]</sup>, introduces the STM32 S/PDIF Receiver interface on the STM32H7. This training also applies to the STM32 MPU SPDIFRX internal peripheral.



---

## 5 References

---

- Device tree
- STM32H7 SPDIFRX training