STM internal peripheral

STM internal peripheral

Stable: 26.09.2019 - 13:44 / Revision: 26.09.2019 - 13:43

Contents

1 Article purpose

The purpose of this article is to:

- briefly introduce the STM 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, when necessary, how to configure the STM peripheral.

2 Peripheral overview

The **STM** peripheral is used to log STM trace into the embedded trace FIFO (ETF). This trace can include hardware events (the list is given in the STM32MP15 reference manuals) or direct 'printf like' log from the Cortex®-A7. Once in the ETF buffer, the trace can directly be dumped from the Cortex®-A7 or to the trace port interface unit (TPIU), connected to an external probe able to decode it.

2.1 Features

Refer to the STM32MP15 reference manuals for the complete list of features, and to the software components, introduced below, to see which features are really implemented.

2.2 Security support

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

3 Peripheral usage and associated software

3.1 Boot time

The STM is not used at boot time.

3.2 Runtime

3.2.1 Overview

The STM can be assigned to the Cortex®-A7 non-secure for using in Linux with coresight framework.
This driver allows to select the hardware events (listed in the STM32MP15 reference manuals) to log via the STM peripheral into the ETF and dump it in the Linux console for analysis.

3.2.2 Software frameworks

Domain	Peripheral	Software frameworks			Comment
Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Trace & Debug	STM		Linux Coresight framework		

3.2.3 Peripheral configuration

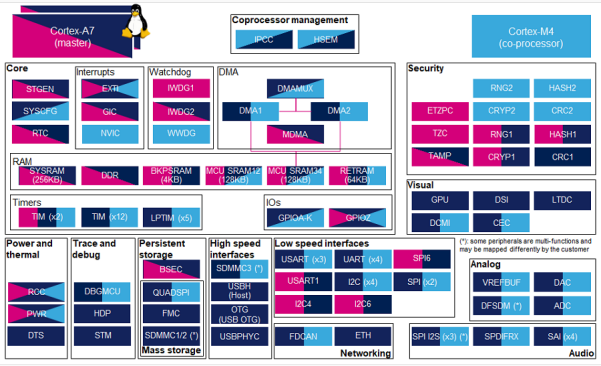
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (particularly for external peripherals), according to the information given in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

3.2.4 Peripheral assignment

Check boxes illustrate the possible peripheral allocations supported by STM32 MPU Embedded Software:

- â means that the peripheral can be assigned (â) to the given runtime context.
- â 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)			
Trace & Debug	STM	STM		â		

4 References