

DTS internal peripheral

Stable: 11.02.2019 - 12:21 / Revision: 07.01.2019 - 17:31

Contents

1 Peripheral overview	1
1.1 Features	1
1.2 Security support	1
2 Peripheral usage and associated software	1
2.1 Boot time	1
2.2 Runtime	2
2.2.1 Overview	2
2.2.2 Software frameworks	2
2.2.3 Peripheral configuration	2
2.2.4 Peripheral assignment	2
3 References	3

1 Peripheral overview

The **DTS** peripheral is used to monitor the device temperature and take some preventive action (like frequency scaling or peripheral disabling) in case it is becoming too high and before destroying the component.

1.1 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 implemented.

1.2 Security support

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

2 Peripheral usage and associated software

2.1 Boot time

DTS is not used at boot time.

2.2 Runtime

2.2.1 Overview

The device cannot warm up if the Cortex[®]-M4 is running alone, as a consequence the monitoring is only done from the Cortex-A7 non-secure context with Linux[®] [thermal management framework](#).

2.2.2 Software frameworks

Do main	Peripheral	Software frameworks			Comment
		Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)	
Power & Thermal	DTS		Linux thermal framework		

2.2.3 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.

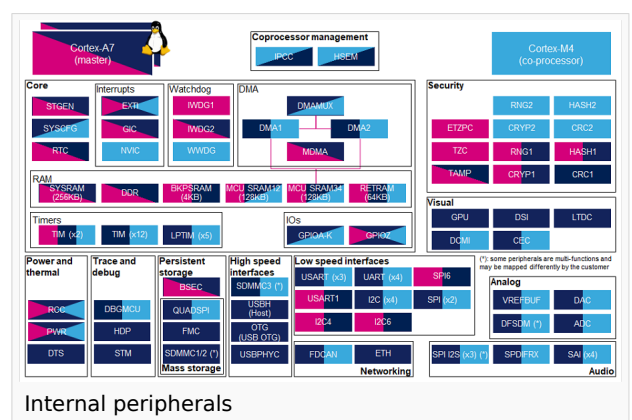
2.2.4 Peripheral assignment

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](#).



Internal peripherals

Do ma in	Pe ri ph e ra l	Runtime allocation			Comme nt
		Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	
Po we r & Th er ma l	DT S	DTS		<input type="checkbox"/>	

3 References

Device Tree Source (in software context) or Digital Temperature Sensor (in peripheral context)

Open Portable Trusted Execution Environment