BSEC internal peripheral

Stable: 24.09.2019 - 24.09 Revision: 24.09.2019 - 07:57

A quality version of this page, approved on *24 September 2019*, was based off this revision.

Contents

1 Article purpose

The purpose of this article is to

- briefly introduce the BSEC 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 BSEC peripheral.

2 Peripheral overview

The **BSEC** peripheral is used to control an OTP (one time programmable) fuse box, used for on-chip non-volatile storage for device configuration and security parameters.

2.1 Features

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

2.2 Security support

The BSEC is a **secure** peripheral.

3 Peripheral usage and associated software

3.1 Boot time

The BSEC is configured at boot time to set up platform security.

3.2 Runtime

3.2.1 Overview

The BSEC instance is a system peripheral and is controlled by the Arm® Cortex®-A7 secure:

Information

- BSEC lower OTP access can be made available to the Arm® Cortex®-A7 non-secure.
- Upper OTP access can be managed as exceptions (in Trusted Boot Chain only, using TF-A), via "secure monitor calls", managed by TF-A or by OP-TEE. Please refer to BSEC device tree configuration for more details.

3.2.2 Software frameworks

Domain	Peripheral	Software components		Comment
OP-TEE	Linux	STM32Cube		
Security	BSEC	OP-TEE BSEC driver	Linux NVMEM framework	

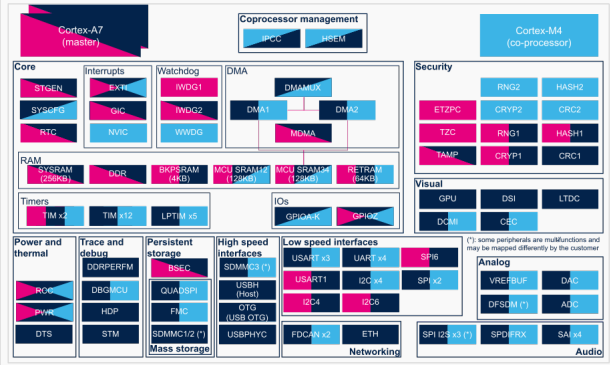
3.2.3 Peripheral configuration

The configuration is based on Device tree, please refer to BSEC device tree configuration article.
It can be applied by the firmware running in a secure context, done in TF-A or in OP-TEE.
It can also be configured by Linux® kernel, please refer to NVMEM overview article.

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)			
Security	BSEC	BSEC				

4 How to go further

5 References