Document Title	Specification of Flash Test
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	261

Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R24-11

Document Change History

Date	Release	Changed by	Description
2024-11-27	R24-11	AUTOSAR Release Management	• No content changes
2023-11-23	R23-11	AUTOSAR Release Management	• Editorial changes
2022-11-24	R22-11	AUTOSAR Release Management	• Changed [SWS_FlsTst_00166] to [SWS_FlsTst_NA_00166]
2021-11-25	R21-11	AUTOSAR Release Management	• [SWS_FlsTst_00019] removed
2020-11-30	R20-11	AUTOSAR Release Management	• Included Development Errors as artifact
2019-11-28	R19-11	AUTOSAR Release Management	• “DRAFT” status of [ECUC_FlsTst_00175] removed • Changed Document Status from Final to published
2018-10-31	4.4.0	AUTOSAR Release Management	• FlsTstBlockBgndConfigSet and FlsTstBlockFgndConfigSet removed • FlsTstEcucPartitionRef configuration parameter added
2017-12-08	4.3.1	AUTOSAR Release Management	• Numeric value definition • minor corrections / clarifications / editorial changes; for details please refer to the ChangeDocumentation

▽

△

2016-11-30	4.3.0	AUTOSAR Release Management	• [ECUC_FlsTst_00172]: FlsTstMainFunctionPeriod added • [SWS_FlsTst_00081] removed • Unresolved references BSW00431, BSW00434, [SRS_BSW_00326], [SRS_BSW_00435], [SRS_BSW_00436] deleted
2015-07-31	4.2.2	AUTOSAR Release Management	• Debugging support marked as obsolete • [ECUC_FlsTst_00119] set to obsolete • [ECUC_FlsTst_00161] created • [ECUC_FlsTst_00151] modified (precompile only) • [SWS_FlsTst_00023], [SWS_FlsTst_00026], [SWS_FlsTst_00133] removed • [SWS_FlsTst_00007]: Error code 0x05 modified • [SWS_FlsTst_00168] created: Extended production error table with pass/fail criteria • [SWS_FlsTst_00161] modified • [SWS_FlsTst_00167] created • Renaming from Development Error Tracer to Default Error Tracer; changes in abbreviations, chapter 3.1, [SWS_FlsTst_00011] • Template changes; chapters runtime errors and transient faults added

▽

△

2014-10-31	4.2.1	AUTOSAR Release Management	• Formal text modifications in: [SWS_FlsTst_00138], [SWS_FlsTst_00140], [SWS_FlsTst_00142], [SWS_FlsTst_00143], [SWS_FlsTst_00071], [SWS_FlsTst_00115], [SWS_FlsTst_00116], [SWS_FlsTst_00117], [ECUC_FlsTst_00160], Figure 7/8/9/10 • [ECUC_FlsTst_00086]: configuration FlsTstConfigurationOfOptApiServices added
2013-10-31	4.1.2	AUTOSAR Release Management	• [SWS_FlsTst_00066]: VARIABLE_CYCLIC_OR_ON_PRECONDITION in table removed • Editorial changes • Removed chapter(s) on change documentation
2013-03-15	4.1.1	AUTOSAR Administration	• Rework according to the new SWS_BSWGeneral document • Added Subchapter 3.x due to SWS General Rollout • Chapter 10: scope of configuration parameters are changed to “local” • [SWS_FlsTst_00003]: Rename MemMap.h to FlsTst_MemMap.h • [SWS_FlsTst_00007]: production errors removed • New chapters Production Errors and Extended Production Errors created
2011-12-22	4.0.3	AUTOSAR Administration	• [SWS_FlsTst_00026]: minor text change • Figure1: IRQ files removed • [SWS_FlsTst_00052]: parameter range modified • [SWS_FlsTst_00053]: minor text correction

▽

△

2010-09-30	3.1.5	AUTOSAR Administration	• FlsTst_BlockIdFgndType: type change to uint8-32 • Limit range of the following parameters to max. value “0xFFFFFFFF” • FlsTstBlockNumberBgnd: • FlsTstBlockNumberFgnd: • FlsTstBlockIndex: • FlsTstBlockSize: • FlsTstNumberOfTestedCells: • FlsTstNumberOfTestedCellsAtomic: • FlsTstTestIntervalIdEndValue: • FlsTst015 removed • [ECUC_FlsTst_00119]: configuration for each block • [ECUC_FlsTst_00158]: multiplicity changed to “1”. • FlsTstDemEventParameterRefs table included
2010-02-02	3.1.4	AUTOSAR Administration	• Initial release

Disclaimer

This work (specification and/or software implementation) and the material contained in it, as released by AUTOSAR, is for the purpose of information only. AUTOSAR and the companies that have contributed to it shall not be liable for any use of the work.

The material contained in this work is protected by copyright and other types of intellectual property rights. The commercial exploitation of the material contained in this work requires a license to such intellectual property rights.

This work may be utilized or reproduced without any modification, in any form or by any means, for informational purposes only. For any other purpose, no part of the work may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The work has been developed for automotive applications only. It has neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Contents

1 Introduction and functional overview

This specification specifies the functionality, API and the configuration of the AUTOSAR Basic Software module Flash Test driver.

This Flash test module provides algorithm to test invariable memory. Invariable memory can be data/program flash, program SRAM, locked cache and is either embedded in the microcontroller or memory mapped connected to the microcontroller. For simplification the SW module is called Flash Test driver.

The test service can be executed at any time after MCU initialization and it is up to the user of the Flash Test Driver to choose the suitable test algorithm and the right execution place to fulfill the safety requirements of the system. The test service itself is dependant on the storage concept of the system. Therefore, the availability of different test algorithms is configurable.

The Flash Test driver is intended to be integrated in the overall safety concept and will not provide the required diagnostic coverage on its own.

2 Acronyms and Abbreviations

Acronyms and abbreviations that have a local scope are not contained in the AUTOSAR glossary. These appear in a local glossary below.

Acronym:	Description:
BSW	BasicSoftWare
PC	PreCompile
PB	PostBuild

Table 2.1: acronyms used in the scope of this Document

Abbreviation:	Description:
DEM	Diagnostic Event Manager.
DET	Default Error Tracer.
MCU	Micro Controller Unit.
PLL	Phase Locked Loop.
ISR	Interrupt Service Routine.

Table 2.2: abbreviations used in the scope of this Document

The following table lists important Term and Definition, which are used within this document.

Term:	Description:
Background test	Background test is called periodically by a scheduler, and is interruptible. The test is split up over many scheduled tasks.
Foreground test	Foreground test is called via users call.
Flash cell	Smallest entity to be addressed, in this case bytes shall be used
Invariable memory	Invariable memory can be program flash, program SRAM, locked cache and ROM
Test block	Defined memory area to be tested in foreground and background mode.
Test interval	Interval of a complete Flash test in background mode
Test time	Time for partial test defined within one scheduled task.
Signature	Unique calculation result of the content of a specific memory block
Memory block	Defined memory area
Partial test	Test to be executed in one scheduler interval
Test Interval Id	Identifier of a test interval, which shall be incremented by each start of a new test interval

Table 2.3: Terms and definitions used in the scope of this Document

3 Related documentation

3.1 Input documents & related standards and norms

[1] General Specification of Basic Software Modules
AUTOSAR_CP_SWS_BSWGeneral

[2] General Requirements on Basic Software Modules
AUTOSAR_CP_RS_BSWGeneral

[3] General Requirements on SPAL
AUTOSAR_CP_RS_SPALGeneral

[4] Requirements on Flash Test
AUTOSAR_CP_RS_FlashTest

3.2 Related specification

AUTOSAR provides a General Specification on Basic Software modules [1, SWS BSW General], which is also valid for Flash Test.

Thus, the specification SWS BSW General shall be considered as additional and required specification for Flash Test.

4 Constraints and assumptions

4.1 Limitations

During Flash Test operation, the Flash area under test shall not be modified.

4.2 Applicability to car domains

No restrictions.

5 Dependencies to other modules

The Flash Test module depends on the following modules:

- BSW scheduler is required to trigger main function in background mode

5.1 File structure

5.1.1 Code file structure

Note: Refer to SWS_BSWGeneral document [1].

6 Requirements Tracing

The following tables reference the requirements specified in [2], [3], and [4] and links to the fulfillment of these. Please note that if column “Satisfied by” is empty for a specific requirement this means that this requirement is not fulfilled by this document.

Requirement	Description	Satisfied by
[SRS_BSW_00101]	The Basic Software Module shall be able to initialize variables and hardware in a separate initialization function	[SWS_FlsTst_00017]
[SRS_BSW_00304]	All AUTOSAR Basic Software Modules shall use only AUTOSAR data types instead of native C data types	[SWS_FlsTst_00016]
[SRS_BSW_00323]	All AUTOSAR Basic Software Modules shall check passed API parameters for validity	[SWS_FlsTst_00033]
[SRS_BSW_00336]	Basic SW module shall be able to shutdown	[SWS_FlsTst_00027]
[SRS_BSW_00337]	Classification of development errors	[SWS_FlsTst_00007]
[SRS_BSW_00339]	Reporting of production relevant error status	[SWS_FlsTst_00042] [SWS_FlsTst_00060] [SWS_FlsTst_00112]
[SRS_BSW_00357]	For success/failure of an API call a standard return type shall be defined	[SWS_FlsTst_00063]
[SRS_BSW_00377]	A Basic Software Module can return a module specific types	[SWS_FlsTst_00048]
[SRS_BSW_00385]	List possible error notifications	[SWS_FlsTst_00007]
[SRS_BSW_00386]	The BSW shall specify the configuration and conditions for detecting an error	[SWS_FlsTst_00025] [SWS_FlsTst_00033] [SWS_FlsTst_00056] [SWS_FlsTst_00059] [SWS_FlsTst_00062] [SWS_FlsTst_00065] [SWS_FlsTst_00089] [SWS_FlsTst_00091] [SWS_FlsTst_00093] [SWS_FlsTst_00114]
[SRS_BSW_00405]	BSW Modules shall support multiple configuration sets	[SWS_FlsTst_00018]
[SRS_BSW_00406]	API handling in uninitialized state	[SWS_FlsTst_00011]
[SRS_BSW_00407]	Each BSW module shall provide a function to read out the version information of a dedicated module implementation	[SWS_FlsTst_00044]
[SRS_BSW_00409]	All production code error ID symbols are defined by the Dem module and shall be retrieved by the other BSW modules from Dem configuration	[SWS_FlsTst_00007]
[SRS_BSW_00411]	All AUTOSAR Basic Software Modules shall apply a naming rule for enabling/disabling the existence of the API	[SWS_FlsTst_00044]
[SRS_BSW_00438]	Configuration data shall be defined in a structure	[SWS_FlsTst_00018]
[SRS_FlsTst_14208]	Background Flash test shall be interruptible	[SWS_FlsTst_00066] [SWS_FlsTst_00071]
[SRS_FlsTst_14209]	The memory to be tested shall be split into individual smaller pieces	[SWS_FlsTst_00066] [SWS_FlsTst_00071] [SWS_FlsTst_00139]

▽

△

Requirement	Description	Satisfied by
[SRS_FlsTst_14211]	Flash test execution status shall be available	[SWS_FlsTst_00040] [SWS_FlsTst_00041] [SWS_FlsTst_00091]
[SRS_FlsTst_14212]	Flash test execution completion shall be provided by a notification mechanism	[SWS_FlsTst_00077] [SWS_FlsTst_00078]
[SRS_FlsTst_14213]	Calculation signature/checksum of a finalized test shall be provided	[SWS_FlsTst_00054] [SWS_FlsTst_00055] [SWS_FlsTst_00056] [SWS_FlsTst_00057] [SWS_FlsTst_00058] [SWS_FlsTst_00059] [SWS_FlsTst_00115] [SWS_FlsTst_00116]
[SRS_FlsTst_14214]	Service for Flash test execution result shall be provided.	[SWS_FlsTst_00042] [SWS_FlsTst_00043] [SWS_FlsTst_00093] [SWS_FlsTst_00112] [SWS_FlsTst_00113] [SWS_FlsTst_00114]
[SRS_FlsTst_14215]	Suspend Flash test execution shall be possible	[SWS_FlsTst_00034] [SWS_FlsTst_00036] [SWS_FlsTst_00037] [SWS_FlsTst_00088]
[SRS_FlsTst_14216]	Flash test execution shall be resumed when suspended	[SWS_FlsTst_00035] [SWS_FlsTst_00038] [SWS_FlsTst_00039] [SWS_FlsTst_00089]
[SRS_FlsTst_14217]	Flash test execution shall be stopped when wanted	[SWS_FlsTst_00030] [SWS_FlsTst_00031] [SWS_FlsTst_00032]
[SRS_FlsTst_14219]	Foreground Flash test shall be available	[SWS_FlsTst_00033] [SWS_FlsTst_00050] [SWS_FlsTst_00051] [SWS_FlsTst_00137] [SWS_FlsTst_00143] [SWS_FlsTst_00149]
[SRS_FlsTst_14223]	Flash Test Error details shall be reported	[SWS_FlsTst_00060] [SWS_FlsTst_00061] [SWS_FlsTst_00062]
[SRS_FlsTst_14224]	ECC Circuitry shall be tested	[SWS_FlsTst_00063] [SWS_FlsTst_00064] [SWS_FlsTst_00065]
[SRS_FlsTst_14225]	Each Flash test Interval shall have an Identifier	[SWS_FlsTst_00153] [SWS_FlsTst_00154] [SWS_FlsTst_00155]
[SRS_SPAL_00157]	All drivers and handlers of the AUTOSAR Basic Software shall implement notification mechanisms of drivers and handlers	[SWS_FlsTst_00040] [SWS_FlsTst_00042] [SWS_FlsTst_00054] [SWS_FlsTst_00057] [SWS_FlsTst_00060] [SWS_FlsTst_00072] [SWS_FlsTst_00073] [SWS_FlsTst_00077] [SWS_FlsTst_00112]
[SRS_SPAL_12057]	All driver modules shall implement an interface for initialization	[SWS_FlsTst_00017] [SWS_FlsTst_00020]
[SRS_SPAL_12125]	All driver modules shall only initialize the configured resources	[SWS_FlsTst_00022]
[SRS_SPAL_12163]	All driver modules shall implement an interface for de-initialization	[SWS_FlsTst_00027] [SWS_FlsTst_00028]
[SRS_SPAL_12448]	All driver modules shall have a specific behavior after a development error detection	[SWS_FlsTst_00025] [SWS_FlsTst_00033] [SWS_FlsTst_00039]

Table 6.1: Requirements Tracing

7 Functional specification

7.1 General behavior

[SWS_FlsTst_00137]

Upstream requirements: SRS_FlsTst_14219

⌈The Flash test module provides test execution services in background and foreground mode.⌋

[SWS_FlsTst_00138] ⌈The memory blocks to be tested shall be configurable for background and foreground mode separately.⌋

[SWS_FlsTst_00139]

Upstream requirements: SRS_FlsTst_14209

⌈In background mode the test blocks shall be tested in the same order they are configured in configuration structure. When all blocks are tested, one test interval is completed. In background testing the partial tests shall be triggered via FlsTst_MainFunction (see [SWS_FlsTst_00066]).⌋

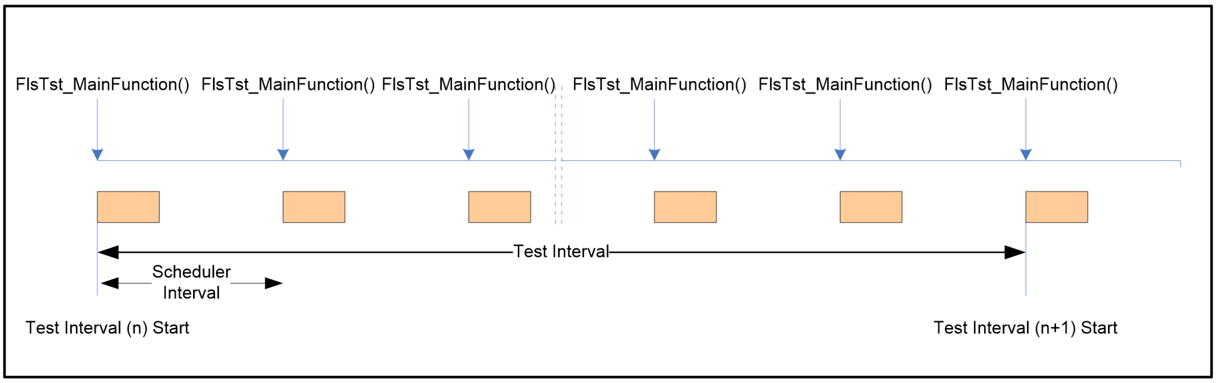

Figure 7.1: Background Test: Test Interval

[SWS_FlsTst_00140] ⌈The length of a partial test is defined by the number of tested cells, which shall be tested in one scheduled task. (see [ECUC_FlsTst_00161]). The required time for a partial test without interruption is defined as "Test time".⌋

Note: The partial test can be interrupted by a higher priority task at any time, because the Flash test does not require atomic sequences. It is the responsibility of the user to ensure that the interruptible partial test is finished before the scheduler interval is started(See Figure 7.2).

[SWS_FlsTst_00142] ⌈A background test shall be aborted or suspended via the API services FlsTst_Abort() or FlsTst_Suspended(). The maximum latency time until the API call request is processed, shall be configurable (see [ECUC_FlsTst_00120]).⌋

[SWS_FlsTst_00156] ⌈Each Flash test Interval shall have an Identifier, which shall be incremented by each start of a new test interval in background mode.⌋

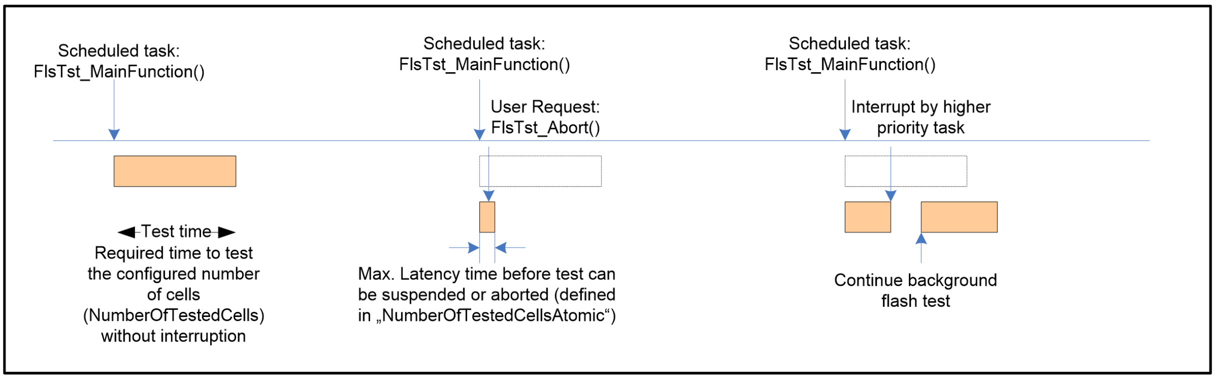

Figure 7.2: Background Test: Test Process

7.1.1 State Diagram

The Flash test driver states in background mode are described in Figure 7.3. The described states are driver states in background operation mode.

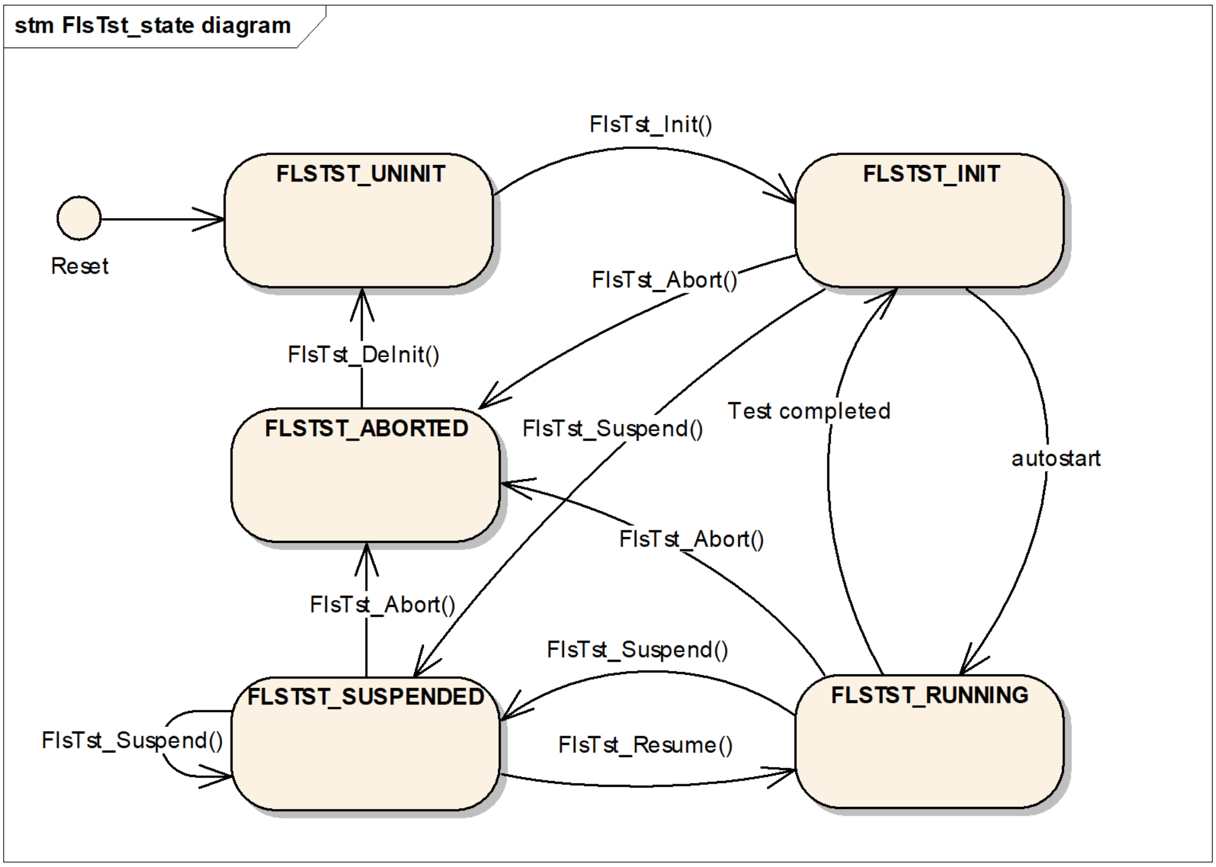

Figure 7.3: State Diagram - Background mode

[SWS_FlsTst_00143]

Upstream requirements: SRS_FlsTst_14219

⌈Foreground tests are defined as synchronous tests which shall not be interrupted. The execution of Foreground tests is configurable (see [ECUC_FlsTst_00086]) and can be called after module initialization at any time.⌋

7.2 Error Classification

Section "Error Handling" of the document [1] "General Specification of Basic Software Modules" describes the error handling of the Basic Software in detail. Above all, it constitutes a classification scheme consisting of five error types which may occur in BSW modules.

Based on this foundation, the following section specifies particular errors arranged in the respective subsections below.

7.2.1 Development Errors

[SWS_FlsTst_00007] Definiton of development errors in module FlsTst

Upstream requirements: SRS_BSW_00337, SRS_BSW_00409, SRS_BSW_00385

⌈

Type of error	*Related error code*	*Error value*
Failure within Flash Test execution state	FLSTST_E_STATE_FAILURE	0x01
API parameter out of specified range	FLSTST_E_PARAM_INVALID	0x02
API service used without module initialization	FLSTST_E_UNINIT	0x03
Flash Test module is already initialized	FLSTST_E_ALREADY_INITIALIZED	0x04
For Variant PB: Configuration pointer is a NULL pointer. For Variant PC: Configuration pointer is NOT a NULL pointer.	FLSTST_E_INIT_FAILED	0x05
Pointer is a NULL pointer	FLSTST_E_PARAM_POINTER	0x06

⌋

7.2.2 Runtime Errors

There are no runtime errors.

7.2.3 Production Errors

There are no production errors.

7.2.4 Extended Production Errors

[SWS_FlsTst_00168] ⌈

Error Name:	FLSTST_E_FLSTST_FAILURE	
Short Description:	Failure detection in background mode	
Long Description:	This Extended Production Error shall be issued in case a failure is detected in background mode within a test interval.	
Detection Criteria:	Fail	At least one block within a test interval in background mode is NOT OK (see SWS_FlsTst_00167)
	Pass	All blocks within a test interval in background mode are tested with the result OK.(see SWS_FlsTst_00161)
Secondary Parameters:	N/A	

▽

△

Time Required:	N/A
Monitor Frequency	continuous

⌋

7.3 Initialization Sequence

[SWS_FlsTst_00011]

Upstream requirements: SRS_BSW_00406

⌈The function FlsTst_Init shall be called first before calling any other Flash Test functions except the function FlsTst_GetCurrentState. If this sequence is not respected, the error code FLSTST_E_UNINIT shall be reported to the Default Error Tracer (if development error detection is enabled).⌋

7.4 Version Check

Note: Refer to SWS_BSWGeneral document [1].

7.5 Debugging Support

No requirement defined.

8 API specification

8.1 Imported types

This chapter lists data type definitions for the included variables and constants.

[SWS_FlsTst_00016] Definition of imported datatypes of module FlsTst

Upstream requirements: SRS_BSW_00304

⌈

Module	*Header File*	*Imported Type*
Dem	Rte_Dem_Type.h	Dem_EventIdType
	Rte_Dem_Type.h	Dem_EventStatusType
Std	Std_Types.h	Std_ReturnType
	Std_Types.h	Std_VersionInfoType

⌋

8.2 Type definitions

8.2.1 FlsTst_ConfigType

[SWS_FlsTst_00018] Definition of datatype FlsTst_ConfigType

Upstream requirements: SRS_BSW_00405, SRS_BSW_00438

⌈

Name	FlsTst_ConfigType	
Kind	Structure	
Elements	implementation specific	
	Type	–
	Comment	implementation specific
Description	This type of external data structure shall contain the initialization data for the Flash Test.	
Available via	FlsTst.h	

⌋

8.2.2 FlsTst_StateType

[SWS_FlsTst_00048] Definition of datatype FlsTst_StateType

Upstream requirements: SRS_BSW_00377

⌈

Name	FlsTst_StateType		
Kind	Enumeration		
Range	FLSTST_UNINIT	0x00	The Flash Test is not initialized or not usable.
	FLSTST_INIT	0x01	The Flash Test is initialized and ready to be started.
	FLSTST_RUNNING	0x02	The Flash Test is currently running.
	FLSTST_ABORTED	0x03	The Flash Test is aborted.
	FLSTST_SUSPENDED	0x04	The Flash Test is waiting to be resumed or is waiting to start foreground mode test
Description	This is a state value returned by the API service FlsTst_GetCurrentState().		
Available via	FlsTst.h		

⌋

[SWS_FlsTst_00049] ⌈For the type FlsTst_StateType, the enumeration value FLSTST_UNINIT shall be the default value after a reset.⌋

8.2.3 FlsTst_TestResultFgndType

[SWS_FlsTst_00052] Definition of datatype FlsTst_TestResultFgndType ⌈

Name	FlsTst_TestResultFgndType		
Kind	Enumeration		
Range	FLSTST_NOT_TESTED	0x00	There is no result available.
	FLSTST_OK	0x01	The last Flash Test has been tested with OK result.
	FLSTST_NOT_OK	0x02	The last Flash Test has been tested with NOT_OK result.
Description	Return type of API service FlsTst_GetResultFgnd().		
Available via	FlsTst.h		

⌋

[SWS_FlsTst_00053] ⌈For the type FlsTst_TestResultFgndType, the enumeration value FLSTST_NOT_TESTED shall be the default value after a reset.⌋

8.2.4 FlsTst_TestResultBgndType

[SWS_FlsTst_00153] Definition of datatype FlsTst_TestResultBgndType

Upstream requirements: SRS_FlsTst_14225

⌈

Name	FlsTst_TestResultBgndType	
Kind	Structure	
Elements	0..<FlsTstTestIntervalIdEndValue>	
	Type	uint8, uint16, uint32
	Comment	current value of FlsTstTestIntervalId, which is incremented by each new start of an test interval.
	result	
	Type	FlsTst_TestResultType
	Comment	–
Description	Return type of API service FlsTst_GetTestResultBgnd().	
Available via	FlsTst.h	

⌋

[SWS_FlsTst_00154]

Upstream requirements: SRS_FlsTst_14225

⌈For the type FlsTst_TestResultBgndType, the enumeration value FLSTST_RESULT_NOT_TESTED shall be the default value after a reset.⌋

8.2.5 FlsTst_BlockIdFgndType

[SWS_FlsTst_00100] Definition of datatype FlsTst_BlockIdFgndType ⌈

Name	FlsTst_BlockIdFgndType		
Kind	Type		
Derived from	***Basetype***	***Variation***	
	uint16	–	
	uint32	–	
	uint8	–	
Range	0..<FlsTstBlock Number Fgnd > -1	–	The range is dependent on the number of Foreground Flash blocks defined in the configuration structure. The type shall be chosen depending on the MCU platform for best performance.
Description	This type specifies the identification (ID) for a Flash block to be tested in foreground mode, which is configured in the configuration structure.		
Available via	FlsTst.h		

⌋

8.2.6 FlsTst_ErrorDetailsType

[SWS_FlsTst_00108] Definition of datatype FlsTst_ErrorDetailsType ⌈

Name	FlsTst_ErrorDetailsType	
Kind	Structure	
Elements	implementation specific	
	Type	–
	Comment	implementation specific
Description	This type shall specify implementation specific error information monitored in the Flash test module.	
Available via	FlsTst.h	

⌋

8.2.7 FlsTst_TestSignatureFgndType

[SWS_FlsTst_00109] Definition of datatype FlsTst_TestSignatureFgndType ⌈

Name	FlsTst_TestSignatureFgndType	
Kind	Structure	
Elements	implementation specific	
	Type	–
	Comment	Implementation specific type
Description	Type for test signature in foreground mode	
Available via	FlsTst.h	

⌋

8.2.8 FlsTst_TestSignatureBgndType

[SWS_FlsTst_00155] Definition of datatype FlsTst_TestSignatureBgndType

Upstream requirements: SRS_FlsTst_14225

⌈

Name	FlsTst_TestSignatureBgndType	
Kind	Structure	
Elements	0..<FlsTstTestIntervalIdEndValue>	
	Type	uint8, uint16, uint32

▽

△

	Comment	current value of FlsTstTestIntervalId, which is incremented by each new start of an test interval.
	Implementation specific	
	Type	uint8, uint16, uint32
	Comment	It represents the signature value of the last completed test interval. Value might be generated from several block signatures.
Description	Type for test signature in background mode.	
Available via	FlsTst.h	

⌋

8.2.9 FlsTst_TestResultType

[SWS_FlsTst_00164] Definition of datatype FlsTst_TestResultType ⌈

Name	FlsTst_TestResultType		
Kind	Enumeration		
Range	FLSTST_RESULT_NOT_TESTED	0x00	There is no test result available.
	FLSTST_RESULT_OK	0x01	The last Flash Test interval has been tested with OK result
	FLSTST_RESULT_NOT_OK	0x02	The last Flash Test interval has been tested with NOT-OK result.
Description	–		
Available via	FlsTst.h		

⌋

8.3 Function definitions

This is a list of functions provided for upper layer modules.

8.3.1 FlsTst_Init

[SWS_FlsTst_00017] Definition of API function FlsTst_Init

Upstream requirements: SRS_BSW_00101, SRS_SPAL_12057

⌈

Service Name	FlsTst_Init	

Syntax	`void FlsTst_Init (`
	`  const FlsTst_ConfigType* ConfigPtr`
	`)`	

Service ID [hex]	0x00	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	ConfigPtr	Pointer to configuration set
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Service for Flash Test initialization.	
Available via	FlsTst.h	

⌋

[SWS_FlsTst_00020]

Upstream requirements: SRS_SPAL_12057

⌈The function FlsTst_Init shall initialize all Flash Test relevant registers and global variables and change the execution state to FLSTST_INIT.⌋

[SWS_FlsTst_00022]

Upstream requirements: SRS_SPAL_12125

⌈The function FlsTst_Init shall only initialize the configured resources and shall not touch resources that are not configured in the configuration file.⌋

[SWS_FlsTst_00025]

Upstream requirements: SRS_BSW_00386, SRS_SPAL_12448

⌈If development error detection is enabled, calling the routine FlsTst_Init while the Flash Test driver is already initialized shall cause development error FLSTST_E_ALREADY_INITIALIZED. The function shall be left without any action.⌋

Note: The FlsTst_Init function shall be called only once after a reset, unless an FlsTst_DeInit call is made before calling FlsTst_Init again.

8.3.2 FlsTst_DeInit

[SWS_FlsTst_00027] Definition of API function FlsTst_DeInit

Upstream requirements: SRS_BSW_00336, SRS_SPAL_12163

⌈

Service Name	FlsTst_DeInit

Syntax	`void FlsTst_DeInit (`
	`  void`
	`)`

Service ID [hex]	0x01
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return value	None
Description	Service for Flash Test De-Initialization.
Available via	FlsTst.h

⌋

[SWS_FlsTst_00028]

Upstream requirements: SRS_SPAL_12163

⌈The function FlsTst_DeInit shall de-initialize all Flash Test relevant registers and global variables that were initialized by FlsTst_Init.⌋

[SWS_FlsTst_00029] ⌈The function FlsTst_DeInit shall set the Flash Test module state to FLSTST_UNINIT.⌋

8.3.3 FlsTst_StartFgnd

[SWS_FlsTst_00149] Definition of API function FlsTst_StartFgnd

Upstream requirements: SRS_FlsTst_14219

⌈

Service Name	FlsTst_StartFgnd

Syntax	`Std_ReturnType FlsTst_StartFgnd (`
	`  FlsTst_BlockIdFgndType FgndBlockId`
	`)`

Service ID [hex]	0x02

▽

△

Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	FgndBlockId	Number of the foreground test to be executed. This is dependent on configuration.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: Foreground test processed `E_NOT_OK`: Foreground test not accepted
Description	Service for executing foreground Flash Test.	
Available via	FlsTst.h	

⌋

[SWS_FlsTst_00050]

Upstream requirements: SRS_FlsTst_14219

⌈The function FlsTst_StartFgnd is only applicable for Foreground mode Flash Test operation.⌋

[SWS_FlsTst_00051]

Upstream requirements: SRS_FlsTst_14219

⌈The function FlsTst_StartFgnd shall be pre compile time configurable On/Off by the configuration parameter: FlsTst_StartFgndApi.⌋

[SWS_FlsTst_00033]

Upstream requirements: SRS_BSW_00323, SRS_BSW_00386, SRS_SPAL_12448, SRS_FlsTst_14219

⌈If development error detection is enabled and the parameter FgndBlockId is out of range, the DET error value FLSTST_E_PARAM_INVALID shall be raised and the function shall return without any action with return value E_NOT_OK.⌋

8.3.4 FlsTst_Abort

[SWS_FlsTst_00030] Definition of API function FlsTst_Abort

Upstream requirements: SRS_FlsTst_14217

⌈

Service Name	FlsTst_Abort

Syntax	`void FlsTst_Abort (`
	`  void`
	`)`

Service ID [hex]	0x03
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return value	None
Description	Service for aborting the Flash Test.
Available via	FlsTst.h

⌋

[SWS_FlsTst_00031]

Upstream requirements: SRS_FlsTst_14217

⌈This function shall abort Flash test background operation and set the state to FLSTST_ABORTED. When the FlsTst_Abort function is called, FlsTst_MainFunction shall finish the current atomic sequence it is running.⌋

[SWS_FlsTst_00032]

Upstream requirements: SRS_FlsTst_14217

⌈After an FlsTst_Abort call, FlsTst_MainFunction shall not begin testing again when called by the scheduler until after a complete re-initialization of the Flash test module.⌋

8.3.5 FlsTst_Suspend

[SWS_FlsTst_00034] Definition of API function FlsTst_Suspend

Upstream requirements: SRS_FlsTst_14215

⌈

Service Name	FlsTst_Suspend

Syntax	`void FlsTst_Suspend (`
	`  void`
	`)`

Service ID [hex]	0x04
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return value	None
Description	Service for suspending current operation of the Flash Test, until FlsTst_Resume is called.
Available via	FlsTst.h

⌋

[SWS_FlsTst_00036]

Upstream requirements: SRS_FlsTst_14215

⌈The function FlsTst_Suspend is only applicable for Background mode Flash Test operation.⌋

[SWS_FlsTst_00037]

Upstream requirements: SRS_FlsTst_14215

⌈The function FlsTst_Suspend shall set the Flash Test execution state to FLSTST_SUSPENDED in case the execution state was FLSTST_RUNNING or FLSTST_INIT.⌋

[SWS_FlsTst_00088]

Upstream requirements: SRS_FlsTst_14215

⌈The function FlsTst_Suspend shall be pre compile time configurable On/Off by the configuration parameter: FlsTst_SuspendResumeApi.⌋

8.3.6 FlsTst_Resume

[SWS_FlsTst_00035] Definition of API function FlsTst_Resume

Upstream requirements: SRS_FlsTst_14216

⌈

Service Name	FlsTst_Resume

Syntax	`void FlsTst_Resume (`
	`  void`
	`)`

Service ID [hex]	0x05
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return value	None
Description	Service for continuing the Flash Test at the point it was suspended.
Available via	FlsTst.h

⌋

[SWS_FlsTst_00038]

Upstream requirements: SRS_FlsTst_14216

⌈The function FlsTst_Resume shall change the execution state to FLSTST_RUNNING when commanded to continue and the current execution state is FLSTST_SUSPENDED.⌋

[SWS_FlsTst_00039]

Upstream requirements: SRS_SPAL_12448, SRS_FlsTst_14216

⌈If development error detection is enabled and the execution state of the Flash Test module is not FLSTST_SUSPENDED, the Flash Test module shall report the error value FLSTST_E_STATE_FAILURE to the DET, and then immediately return from the function.⌋

[SWS_FlsTst_00162] ⌈The function FlsTst_Resume is only applicable for Background mode Flash Test operation.⌋

[SWS_FlsTst_00089]

Upstream requirements: SRS_BSW_00386, SRS_FlsTst_14216

⌈The function FlsTst_Resume shall be pre compile time configurable On/Off by the configuration parameter: FlsTst_SuspendResumeApi.⌋

8.3.7 FlsTst_GetCurrentState

[SWS_FlsTst_00040] Definition of API function FlsTst_GetCurrentState

Upstream requirements: SRS_SPAL_00157, SRS_FlsTst_14211

⌈

Service Name	FlsTst_GetCurrentState	

Syntax	`FlsTst_StateType FlsTst_GetCurrentState (`
	`  void`
	`)`	

Service ID [hex]	0x06	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	None	
Return value	FlsTst_StateType	`FLSTST_UNINIT` The Flash Test is not initialized or not usable. `FLSTST_INIT` The Flash Test is initialized and ready to be started. `FLSTST_RUNNING` The Flash Test is currently running. `FLSTST_ABORTED` The Flash Test is aborted. `FLSTST_SUSPENDED` The Flash Test is waiting to be resumed or is waiting to start forground mode test
Description	Service returns the current Flash Test exection state.	
Available via	FlsTst.h	

⌋

[SWS_FlsTst_00041]

Upstream requirements: SRS_FlsTst_14211

⌈The function FlsTst_GetCurrentState shall return the current Flash Test execution state.⌋

[SWS_FlsTst_00091]

Upstream requirements: SRS_BSW_00386, SRS_FlsTst_14211

⌈The function FlsTst_GetCurrentState shall be pre compile time configurable On/Off by the configuration parameter: FlsTst_GetCurrentStateApi.⌋

8.3.8 FlsTst_GetTestResultBgnd

[SWS_FlsTst_00042] Definition of API function FlsTst_GetTestResultBgnd

Upstream requirements: SRS_BSW_00339, SRS_SPAL_00157, SRS_FlsTst_14214

⌈

Service Name	FlsTst_GetTestResultBgnd	

Syntax	`FlsTst_TestResultBgndType FlsTst_GetTestResultBgnd (`
	`  void`
	`)`	

Service ID [hex]	0x07	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	None	
Return value	FlsTst_TestResultBgndType	See type definition
Description	Service returns the Background Flash Test result.	
Available via	FlsTst.h	

⌋

[SWS_FlsTst_00043]

Upstream requirements: SRS_FlsTst_14214

⌈The function FlsTst_GetTestResultBgnd shall return the Flash test result and Test Interval Id of the last background test.⌋

[SWS_FlsTst_00093]

Upstream requirements: SRS_BSW_00386, SRS_FlsTst_14214

⌈The function FlsTst_GetTestResultBgnd shall be pre compile time configurable On/ Off by the configuration parameter: FlsTst_GetTestResultBgndApi.⌋

8.3.9 FlsTst_GetTestResultFgnd

[SWS_FlsTst_00112] Definition of API function FlsTst_GetTestResultFgnd

Upstream requirements: SRS_BSW_00339, SRS_SPAL_00157, SRS_FlsTst_14214

⌈

Service Name	FlsTst_GetTestResultFgnd	

Syntax	`FlsTst_TestResultFgndType FlsTst_GetTestResultFgnd (`
	`  void`
	`)`	

Service ID [hex]	0x0f	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	None	
Return value	FlsTst_TestResultFgnd Type	See type definition
Description	Service returns the Foreground Flash Test result.	
Available via	FlsTst.h	

⌋

[SWS_FlsTst_00113]

Upstream requirements: SRS_FlsTst_14214

⌈The function FlsTst_GetTestResultFgnd shall return the Flash test result of the last foreground test.⌋

[SWS_FlsTst_00114]

Upstream requirements: SRS_BSW_00386, SRS_FlsTst_14214

⌈The function FlsTst_GetTestResultFgnd shall be pre compile time configurable On/Off by the configuration parameter: FlsTst_GetTestResultFgndApi.⌋

8.3.10 FlsTst_GetVersionInfo

[SWS_FlsTst_00044] Definition of API function FlsTst_GetVersionInfo

Upstream requirements: SRS_BSW_00407, SRS_BSW_00411

⌈

Service Name	FlsTst_GetVersionInfo	

Syntax	`void FlsTst_GetVersionInfo (`
	`  Std_VersionInfoType* versioninfo`
	`)`	

Service ID [hex]	0x08	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	versioninfo	Pointer to where to store the version information of this module.
Return value	None	
Description	Service returns the version information of this module.	
Available via	FlsTst.h	

⌋

8.3.11 FlsTst_GetTestSignatureBgnd

[SWS_FlsTst_00054] Definition of API function FlsTst_GetTestSignatureBgnd

Upstream requirements: SRS_FlsTst_14213, SRS_SPAL_00157

⌈

Service Name	FlsTst_GetTestSignatureBgnd	

Syntax	`FlsTst_TestSignatureBgndType FlsTst_GetTestSignatureBgnd (`
	`  void`
	`)`	

Service ID [hex]	0x09	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	None	
Return value	FlsTst_TestSignatureBgndType	See type definition
Description	Service returns the Flash Test result in background mode.	
Available via	FlsTst.h	

⌋

[SWS_FlsTst_00055]

Upstream requirements: SRS_FlsTst_14213

⌈The function FlsTst_GetTestSignatureBgnd shall return the signature and Test Interval Id of the last background test.⌋

[SWS_FlsTst_00056]

Upstream requirements: SRS_BSW_00386, SRS_FlsTst_14213

⌈The function FlsTst_GetTestSignatureBgnd shall be pre compile time configurable On/Off by the configuration parameter: FlsTst_GetTestSignatureBgndApi.⌋

[SWS_FlsTst_00115]

Upstream requirements: SRS_FlsTst_14213

⌈If no signature is available, the function FlsTst_GetTestSignatureBgnd shall return the default signature value "0x0".⌋

8.3.12 FlsTst_GetTestSignatureFgnd

[SWS_FlsTst_00057] Definition of API function FlsTst_GetTestSignatureFgnd

Upstream requirements: SRS_FlsTst_14213, SRS_SPAL_00157

⌈

Service Name	FlsTst_GetTestSignatureFgnd	

Syntax	`FlsTst_TestSignatureFgndType FlsTst_GetTestSignatureFgnd (`
	`  void`
	`)`	

Service ID [hex]	0x0a	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	None	
Return value	FlsTst_TestSignatureFgndType	See type definition
Description	Service returns the Flash Test result in foreground mode.	
Available via	FlsTst.h	

⌋

[SWS_FlsTst_00058]

Upstream requirements: SRS_FlsTst_14213

⌈The function FlsTst_GetTestSignatureFgnd shall return the signature of the last foreground test.⌋

[SWS_FlsTst_00059]

Upstream requirements: SRS_BSW_00386, SRS_FlsTst_14213

⌈The function FlsTst_GetTestSignatureFgnd shall be pre compile time configurable On/Off by the configuration parameter: FlsTst_GetTestSignatureFgndApi.⌋

[SWS_FlsTst_00116]

Upstream requirements: SRS_FlsTst_14213

⌈If no signature is available, the function FlsTst_GetTestSignatureFgnd shall return the default signature value "0x0".⌋

8.3.13 FlsTst_GetErrorDetails

[SWS_FlsTst_00060] Definition of API function FlsTst_GetErrorDetails

Upstream requirements: SRS_BSW_00339, SRS_SPAL_00157, SRS_FlsTst_14223

⌈

Service Name	FlsTst_GetErrorDetails	

Syntax	`FlsTst_ErrorDetailsType FlsTst_GetErrorDetails (`
	`  void`
	`)`	

Service ID [hex]	0x0b	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	None	
Return value	FlsTst_ErrorDetailsType	See type definition
Description	Service returns error detais monitored from the Flash module.	
Available via	FlsTst.h	

⌋

[SWS_FlsTst_00061]

Upstream requirements: SRS_FlsTst_14223

⌈The function FlsTst_GetErrorDetails shall return the error details monitored from the Flash Test driver.⌋

[SWS_FlsTst_00062]

Upstream requirements: SRS_BSW_00386, SRS_FlsTst_14223

⌈The function FlsTst_GetErrorDetails shall be pre compile time configurable On/Off by the configuration parameter: FlsTst_GetErrorDetailsApi.⌋

8.3.14 FlsTst_TestEcc

[SWS_FlsTst_00063] Definition of API function FlsTst_TestEcc

Upstream requirements: SRS_BSW_00357, SRS_FlsTst_14224

⌈

Service Name	FlsTst_TestEcc	

Syntax	`Std_ReturnType FlsTst_TestEcc (`
	`  void`
	`)`	

Service ID [hex]	0x0c	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	see type definition
Description	Service executes a test of ECC hardware. This is only applicable in case the hardware provices such functionality.	
Available via	FlsTst.h	

⌋

[SWS_FlsTst_00064]

Upstream requirements: SRS_FlsTst_14224

⌈The function FlsTst_TestEcc shall execute a test of the ECC circuitry.⌋

[SWS_FlsTst_00065]

Upstream requirements: SRS_BSW_00386, SRS_FlsTst_14224

⌈The function FlsTst_TestEcc shall be pre compile time configurable On/Off by the configuration parameter: FlsTst_TestEccApi.⌋

8.4 Callback notifications

Since the Flash Test is a driver module, it does not provide any callback functions for lower layer modules.

8.5 Scheduled functions

8.5.1 FlsTst_MainFunction

[SWS_FlsTst_00066] Definition of scheduled function FlsTst_MainFunction

Upstream requirements: SRS_FlsTst_14208, SRS_FlsTst_14209

⌈

Service Name	FlsTst_MainFunction

Syntax	`void FlsTst_MainFunction (`
	`  void`
	`)`

Service ID [hex]	0x0d
Description	Service for executing the Flash Test in background mode.
Available via	SchM_FlsTst.h

⌋

[SWS_FlsTst_00067] ⌈The function FlsTst_MainFunction shall test the defined flash blocks in background mode, starting with the first flash block in the FlsTstConfigParams.⌋

[SWS_FlsTst_00068] ⌈The function FlsTst_MainFunction shall set the Flash Test execution state from FLSTST_INIT to FLSTST_RUNNING when calling the function the first time after initialization or after a complete test interval.⌋

[SWS_FlsTst_00069] ⌈When FlsTstTestResultSignature is true, the function FlsTst_MainFunction shall provide the test signatures of all blocks within a test interval.⌋

[SWS_FlsTst_00161] ⌈When FlsTstTestResultSignature is disabled, the function FlsTst_MainFunction shall set the overall result status for a test interval to FLSTST_RESULT_OK if all blocks are tested with result status OK. Additionally the DEM FLSTST_E_FLSTST_FAILURE shall be triggered with the detection criteria "Pass".⌋

[SWS_FlsTst_00167] ⌈When FlsTstTestResultSignature is disabled, the function FlsTst_MainFunction shall set the overall result status for a test interval to FLSTST_RESULT_NOT_OK if at least one block test result is not ok regardless whether all blocks are already tested or not. Additionally the DEM FLSTST_E_FLSTST_FAILURE shall be triggered with the detection criteria "Fail".⌋

[SWS_FlsTst_00070] ⌈After the function FlsTst_MainFunction has completed testing all flash blocks, the next call of the function FlsTst_MainFunction shall restart the test from the beginning.⌋

[SWS_FlsTst_00071]

Upstream requirements: SRS_FlsTst_14208, SRS_FlsTst_14209

⌈The function FlsTst_MainFunction shall test a defined number of flash cells within one call. The defined number is specified by configuration (see [ECUC_FlsTst_00161]).⌋

[SWS_FlsTst_00117] ⌈The function FlsTst_MainFunction shall test a defined number of flash cells without checking user request for Abort or Suspend. The defined number is specified by configuration (see [ECUC_FlsTst_00120]).⌋

[SWS_FlsTst_00121] ⌈The function FlsTst_MainFunction shall increment the Test Interval Id by 1 before start of a new test interval. The first test interval shall have the Test Interval Id = "0". If the end value = FlsTstIntervalIdEndValue is reached, Test Interval Id shall start with value "0" again. The value shall be provided as part of the return values of FlsTst_GetTestResultBgnd and FlsTst_GetTestSignatureBgnd.⌋

8.6 Expected interfaces

In this chapter, all interfaces required from other modules are listed.

8.6.1 Mandatory Interfaces

This chapter defines all interfaces that are required to fulfill the core functionality of the module.

[SWS_FlsTst_00072] Definition of mandatory interfaces required by module FlsTst

Upstream requirements: SRS_SPAL_00157

⌈

API Function	***Header File***	***Description***
Dem_SetEventStatus	Dem.h	Called by SW-Cs or BSW modules to report monitor status information to the Dem. BSW modules calling Dem_SetEventStatus can safely ignore the return value. This API will be available only if ({Dem/DemConfigSet/DemEventParameter/DemEventReportingType} == STANDARD_REPORTING)

⌋

8.6.2 Optional Interfaces

This chapter defines all interfaces that are required to fulfill an optional functionality of the module.

[SWS_FlsTst_00073] Definition of optional interfaces requested by module Fls Tst

Upstream requirements: SRS_SPAL_00157

⌈

API Function	***Header File***	***Description***
Det_ReportError	Det.h	Service to report development errors.

⌋

8.6.3 Configurable interfaces

In this chapter, all interfaces are listed where the target function could be configured. The target function is usually a callback function. The names of these kinds of interfaces are not fixed because they are configurable.

[SWS_FlsTst_00074] ⌈The callback notifications shall be configurable as function pointers within the initialization data structure (FlsTst_ConfigType).⌋

[SWS_FlsTst_00075] ⌈The callback notifications shall have no parameters and no return value.⌋

[SWS_FlsTst_00076] ⌈If a callback notification is configured as null pointer, the Flash Test module shall not execute the callback.⌋

8.6.3.1 FlsTst_TestCompleted Notification

[SWS_FlsTst_00077] Definition of API function FlsTst_TestCompletedNotification

Upstream requirements: SRS_SPAL_00157, SRS_FlsTst_14212

⌈

Service Name	FlsTst_TestCompletedNotification

Syntax	`void FlsTst_TestCompletedNotification (`
	`  void`
	`)`

Service ID [hex]	0x0e
Sync/Async	Synchronous
Reentrancy	Don't care
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return value	None
Description	The function FlsTst_TestCompleted shall be called every time when a complete test cycle had been tested.
Available via	FlsTst.h

⌋

[SWS_FlsTst_00078]

Upstream requirements: SRS_FlsTst_14212

⌈The Flash Test module shall call the callback notification FlsTst_TestCompleted every time when it has tested a complete test cycle of a flash test in background mode.⌋

[SWS_FlsTst_00159] ⌈The call of function FlsTst_TestCompleted shall be pre compile time configurable On/Off by the configuration parameter FlsTstTestCompletedNotificationSupported.⌋

9 Sequence diagrams

9.1 Initialization

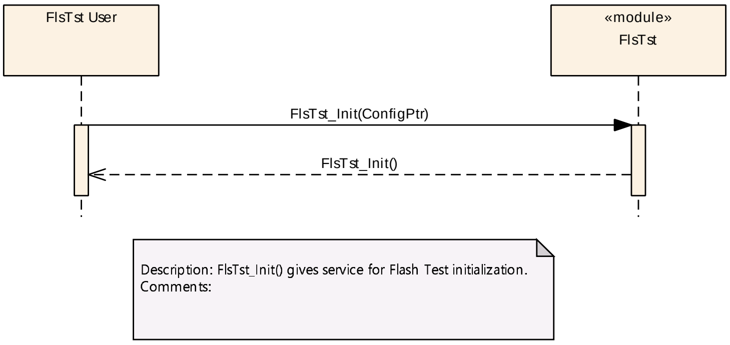
FlsTst User
«module»
FlsTst
FlsTst_Init(ConfigPtr)
FlsTst_Init()
Description: FlsTst_Init() gives service for Flash Test initialization.
Comments:

Figure 9.1: Flash test driver initialization

9.2 De-initialization

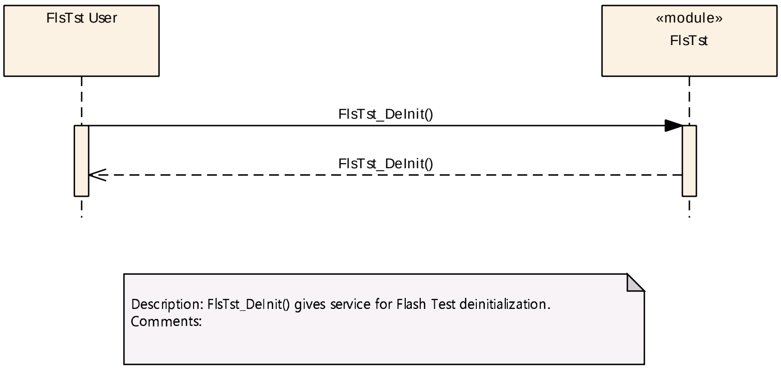
FlsTst User
«module»
FlsTst
FlsTst_DeInit()
FlsTst_DeInit()
Description: FlsTst_DeInit() gives service for Flash Test deinitialization.
Comments:

Figure 9.2: Flash test driver de-initialization

9.3 Background Test

9.3.1 Test Result Calculation within Flash test driver

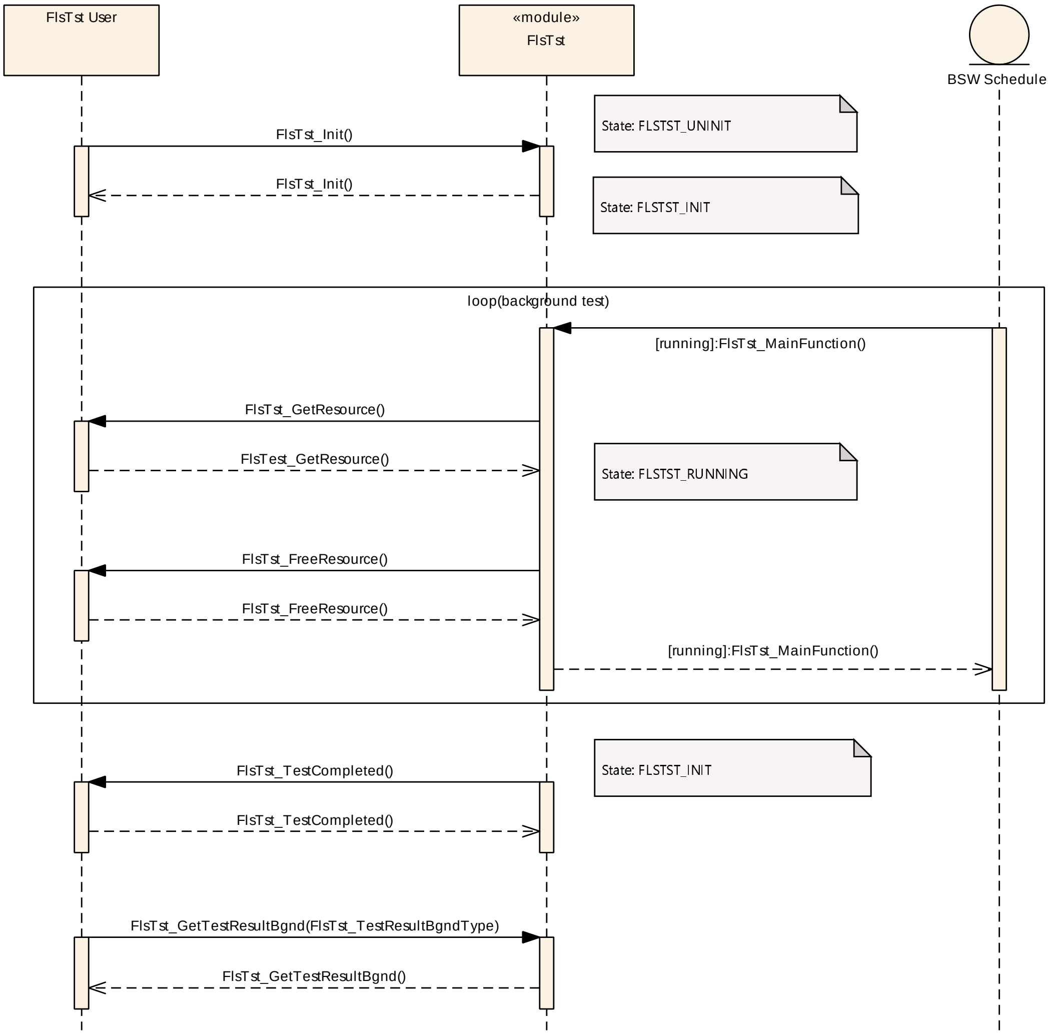

Figure 9.3: Background Test – Test result calculation in Flash test driver

9.3.2 Test signature provided to caller

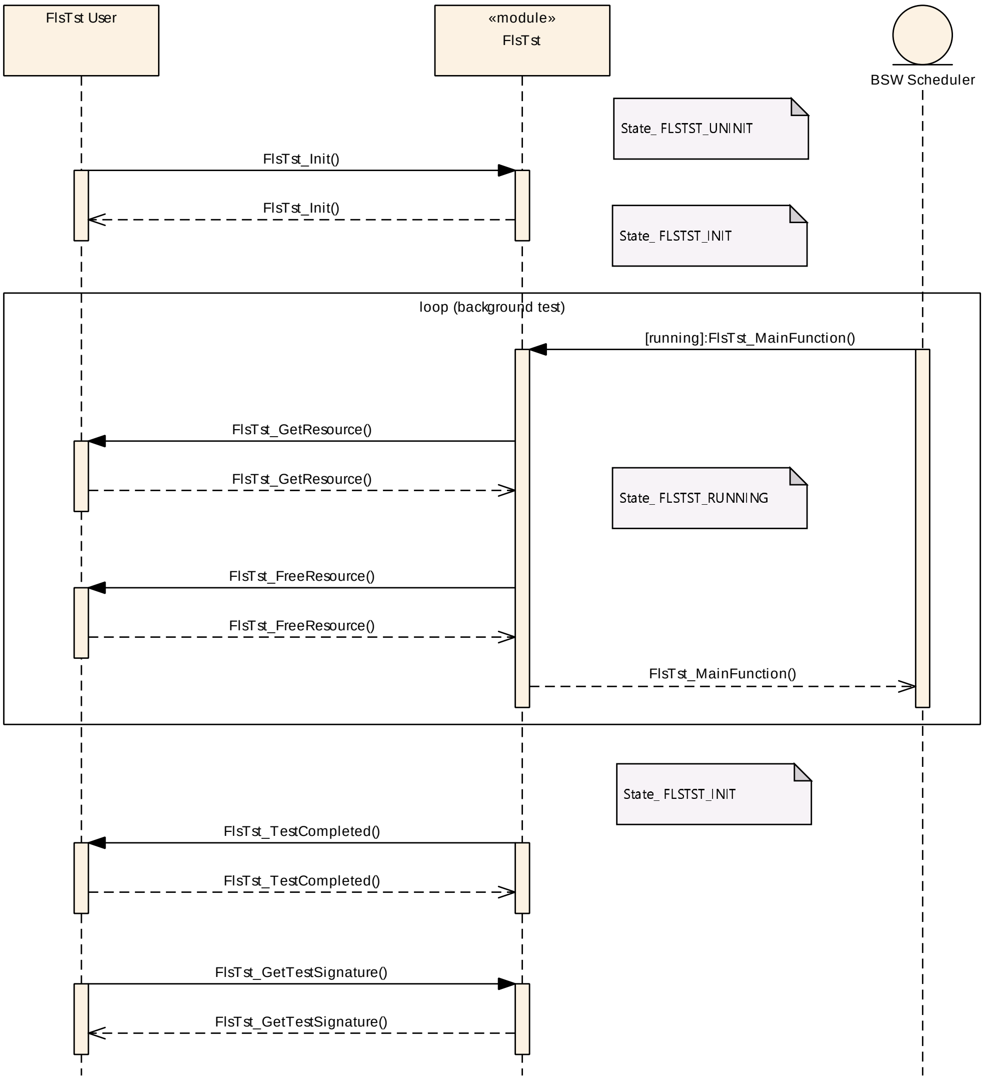

Figure 9.4: Background Test – Test signature provided to caller

9.4 Suspend and Resume Background Testing

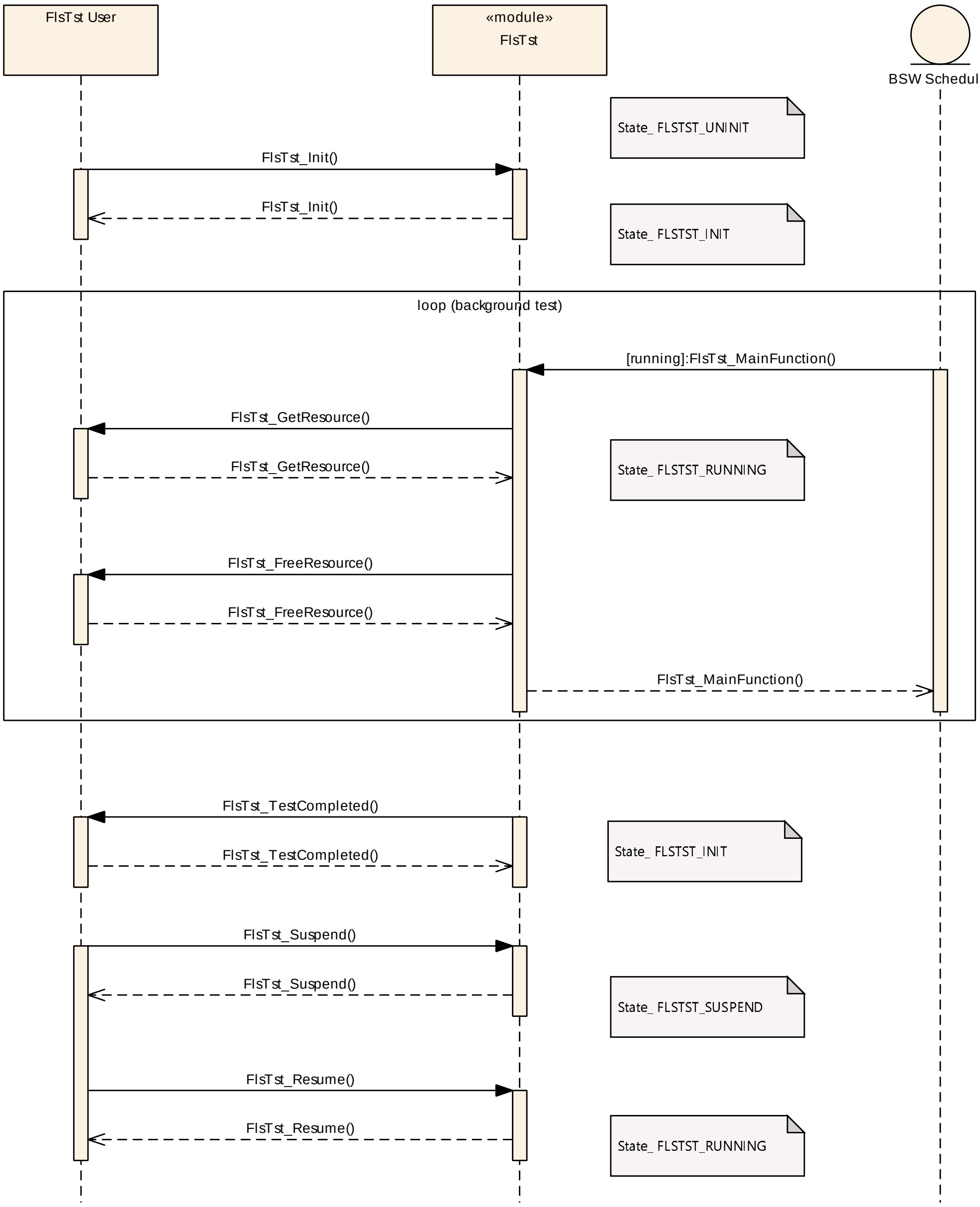

Figure 9.5: Suspend and Resume Background Testing

9.5 Foreground Task interrupts Background Task

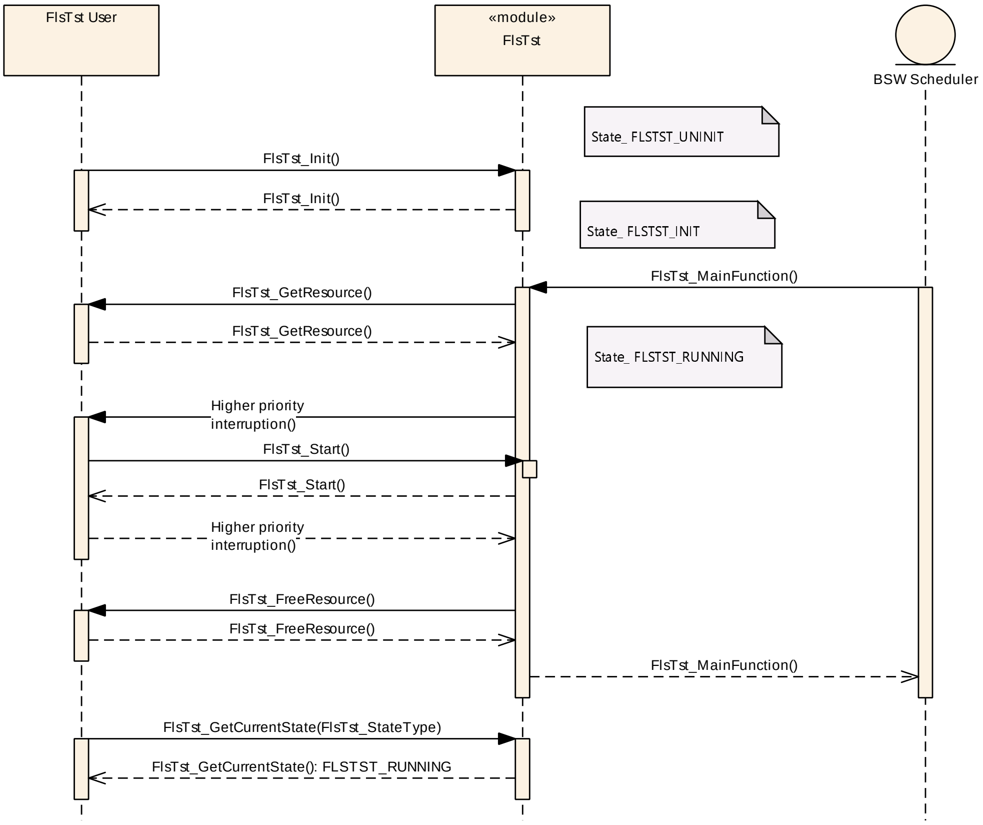

Figure 9.6: Foreground task interrupts Background Task

10 Configuration specification

10.1 Specification template for configuration parameters

Label	Description
x	The configuration parameter shall be of configuration class Pre-compile time.
–	The configuration parameter shall never be of configuration class Pre-compile time.

Link time - specifies whether the configuration parameter shall be of configuration class Link time or not

Label	Description
x	The configuration parameter shall be of configuration class Link time.
–	The configuration parameter shall never be of configuration class Link time.

Post Build - specifies whether the configuration parameter shall be of configuration class Post Build or not

Label	Description
x	The configuration parameter shall be of configuration class Post Build and no specific implementation is required.
L	Loadable - the configuration parameter shall be of configuration class Post Build and only one configuration parameter set resides in the ECU.
M	Multiple - the configuration parameter shall be of configuration class Post Build and is selected out of a set of multiple parameters by passing a dedicated pointer to the init function of the module.
–	The configuration parameter shall never be of configuration class Post Build.

10.2 Containers and configuration parameters

The following chapters summarize all configuration parameters. The detailed meanings of the parameters describe Chapters Functional specification and Chapter API specification.

10.2.1 Variants

See [1] for definition of variants.

10.2.2 FlsTst

[ECUC_FlsTst_00135] Definition of EcucModuleDef FlsTst ⌈

Module Name	FlsTst
Description	Configuration of the FlsTst module.
Post-Build Variant Support	true
Supported Config Variants	VARIANT-POST-BUILD, VARIANT-PRE-COMPILE

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
FlsTstConfigSet	1	This container contains the configuration parameters and sub containers of the AUTOSAR FlsTst module.
FlsTstConfigurationOfOptApiServices	1	–
FlsTstDemEventParameterRefs	0..1	Container for the references to DemEventParameter elements which shall be invoked using the API Dem_SetEventStatus in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId symbolic value. The standardized errors are provided in this container and can be extended by vendor-specific error references.
FlsTstGeneral	1	–

⌋

10.2.3 FlsTstConfigSet

[ECUC_FlsTst_00152] Definition of EcucParamConfContainerDef FlsTstConfigSet ⌈

Container Name	FlsTstConfigSet
Parent Container	FlsTst
Description	This container contains the configuration parameters and sub containers of the AUTOSAR FlsTst module.
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
FlsTstBlockNumberBgnd	1	[ECUC_FlsTst_00122]
FlsTstBlockNumberFgnd	1	[ECUC_FlsTst_00124]
FlsTstTestCompletedNotification	1	[ECUC_FlsTst_00102]

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
FlsTstBlockBgnd	0..*	This container specifies configuration parameters for an individual background test block.
FlsTstBlockFgnd	0..*	This container specifies configuration parameters for an individual foreground test block.

⌋

[ECUC_FlsTst_00122] Definition of EcucIntegerParamDef FlsTstBlockNumberBgnd ⌈

Parameter Name	FlsTstBlockNumberBgnd		
Parent Container	FlsTstConfigSet		
Description	This parameter shall represent the number of test blocks available for the background test. calculationFormula = Number of configured FlsTstBlocks in the FlsTstBlockBgndConfigSet (or 0 if no FlsTstBlocks are configured).		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default value	–		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

⌋

[ECUC_FlsTst_00124] Definition of EcucIntegerParamDef FlsTstBlockNumberFgnd ⌈

Parameter Name	FlsTstBlockNumberFgnd		
Parent Container	FlsTstConfigSet		
Description	This parameter shall represent the number of test blocks available for the foreground test. calculationFormula = Number of configured FlsTstBlocks in the FlsTstBlockFgndConfigSet (or 0 if no FlsTstBlocks are configured).		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default value	–		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

⌋

[ECUC_FlsTst_00102] Definition of EcucFunctionNameDef FlsTstTestCompleted Notification ⌈

Parameter Name	FlsTstTestCompletedNotification		
Parent Container	FlsTstConfigSet		
Description	Pointer to function, which shall be called after finishing the background Flash test interval.		
Multiplicity	1		
Type	EcucFunctionNameDef		
Default value	–		
Regular Expression	–		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

⌋

10.2.4 FlsTstGeneral

[ECUC_FlsTst_00082] Definition of EcucParamConfContainerDef FlsTstGeneral ⌈

Container Name	FlsTstGeneral
Parent Container	FlsTst
Description	–
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
FlsTstDevErrorDetect	1	[ECUC_FlsTst_00083]
FlsTstMainFunctionPeriod	1	[ECUC_FlsTst_00172]
FlsTstNumberOfTestedCells	0..1	[ECUC_FlsTst_00161]
FlsTstNumberOfTestedCellsAtomic	1	[ECUC_FlsTst_00120]
FlsTstTestCompletedNotificationSupported	1	[ECUC_FlsTst_00084]
FlsTstTestIntervalIdEndValue	1	[ECUC_FlsTst_00158]
FlsTstTestResultSignature	1	[ECUC_FlsTst_00160]
FlsTstEcucPartitionRef	0..1	[ECUC_FlsTst_00175]

No Included Containers

⌋

[ECUC_FlsTst_00083] Definition of EcucBooleanParamDef FlsTstDevErrorDetect ⌈

Parameter Name	FlsTstDevErrorDetect		
Parent Container	FlsTstGeneral		
Description	Switches the development error detection and notification on or off. • true: detection and notification is enabled. • false: detection and notification is disabled.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_FlsTst_00172] Definition of EcucFloatParamDef FlsTstMainFunctionPeriod ⌈

Parameter Name	FlsTstMainFunctionPeriod		
Parent Container	FlsTstGeneral		
Description	Determines the frequency at which the FlsTstMainFunction is called in [s].		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	]0 .. INF[		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

⌋

[ECUC_FlsTst_00161] Definition of EcucIntegerParamDef FlsTstNumberOfTestedCells ⌈

Parameter Name	FlsTstNumberOfTestedCells	
Parent Container	FlsTstGeneral	
Description	Configures the Number of cells to be tested in background mode during one scheduled task (FlsTst_MainFunction() call).	
Multiplicity	0..1	
Type	EcucIntegerParamDef	
Range	0 .. 4294967295	

▽

△

Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_FlsTst_00120] Definition of EcucIntegerParamDef FlsTstNumberOfTestedCellsAtomic ⌈

Parameter Name	FlsTstNumberOfTestedCellsAtomic		
Parent Container	FlsTstGeneral		
Description	Configures the Number of cells to be tested in background mode without checking user requests (Abort, Suspend).		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_FlsTst_00084] Definition of EcucBooleanParamDef FlsTstTestCompletedNotificationSupported ⌈

Parameter Name	FlsTstTestCompletedNotificationSupported		
Parent Container	FlsTstGeneral		
Description	Switch to indicate that the notification is supported.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	true		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_FlsTst_00158] Definition of EcucIntegerParamDef FlsTstTestIntervalIdEndValue ⌈

Parameter Name	FlsTstTestIntervalIdEndValue		
Parent Container	FlsTstGeneral		
Description	Defines the end value of the Test Interval Id.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_FlsTst_00160] Definition of EcucBooleanParamDef FlsTstTestResultSignature ⌈

Parameter Name	FlsTstTestResultSignature		
Parent Container	FlsTstGeneral		
Description	Configures the result of the test in background mode: True: Test Result is a signature (see SWS_FlsTst_00155, SWS_FlsTst_00054) False: Test Result is ok/not ok (see SWS_FlsTst_00153, SWS_FlsTst_00042)		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_FlsTst_00175] Definition of EcucReferenceDef FlsTstEcucPartitionRef ⌈

Parameter Name	FlsTstEcucPartitionRef		
Parent Container	FlsTstGeneral		
Description	Maps the Flash test driver to zero or one ECUC partition to make the driver API available in this partition.		
Multiplicity	0..1		
Type	Reference to EcucPartition		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants

▽

△

	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

⌋

10.2.5 FlsTstConfigurationOfOptApiServices

[ECUC_FlsTst_00085] Definition of EcucParamConfContainerDef FlsTstConfigurationOfOptApiServices ⌈

Container Name	FlsTstConfigurationOfOptApiServices
Parent Container	FlsTst
Description	–
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
FlsTstGetCurrentStateApi	1	[ECUC_FlsTst_00092]
FlsTstGetErrorDetailsApi	1	[ECUC_FlsTst_00098]
FlsTstGetTestResultBgndApi	1	[ECUC_FlsTst_00094]
FlsTstGetTestResultFgndApi	1	[ECUC_FlsTst_00150]
FlsTstGetTestSignatureBgndApi	1	[ECUC_FlsTst_00096]
FlsTstGetTestSignatureFgndApi	1	[ECUC_FlsTst_00097]
FlsTstStartFgndApi	1	[ECUC_FlsTst_00086]
FlsTstSuspendResumeApi	1	[ECUC_FlsTst_00087]
FlsTstTestEccApi	1	[ECUC_FlsTst_00099]
FlsTstVersionInfoApi	1	[ECUC_FlsTst_00095]

No Included Containers

⌋

[ECUC_FlsTst_00092] Definition of EcucBooleanParamDef FlsTstGetCurrentStateApi ⌈

Parameter Name	FlsTstGetCurrentStateApi
Parent Container	FlsTstConfigurationOfOptApiServices
Description	Adds / removes the service FlsTst_GetCurrentState() from the code.

▽

△

Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_FlsTst_00098] Definition of EcucBooleanParamDef FlsTstGetErrorDetailsApi ⌈

Parameter Name	FlsTstGetErrorDetailsApi		
Parent Container	FlsTstConfigurationOfOptApiServices		
Description	Adds / removes the service FlsTst_GetErrorDetails() from the code.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_FlsTst_00094] Definition of EcucBooleanParamDef FlsTstGetTestResultBgndApi ⌈

Parameter Name	FlsTstGetTestResultBgndApi		
Parent Container	FlsTstConfigurationOfOptApiServices		
Description	Adds / removes the service FlsTst_GetTestResultBgnd() from the code.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_FlsTst_00150] Definition of EcucBooleanParamDef FlsTstGetTestResultFgndApi ⌈

Parameter Name	FlsTstGetTestResultFgndApi		
Parent Container	FlsTstConfigurationOfOptApiServices		
Description	Adds / removes the service FlsTst_GetTestResultFgnd() from the code.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_FlsTst_00096] Definition of EcucBooleanParamDef FlsTstGetTestSignatureBgndApi ⌈

Parameter Name	FlsTstGetTestSignatureBgndApi		
Parent Container	FlsTstConfigurationOfOptApiServices		
Description	Adds / removes the service FlsTst_GetTestSignatureBgnd() from the code.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_FlsTst_00097] Definition of EcucBooleanParamDef FlsTstGetTestSignatureFgndApi ⌈

Parameter Name	FlsTstGetTestSignatureFgndApi		
Parent Container	FlsTstConfigurationOfOptApiServices		
Description	Adds / removes the service FlsTst_GetTestSignatureFgnd() from the code.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	

▽

△

Scope / Dependency	scope: local

⌋

[ECUC_FlsTst_00086] Definition of EcucBooleanParamDef FlsTstStartFgndApi ⌈

Parameter Name	FlsTstStartFgndApi		
Parent Container	FlsTstConfigurationOfOptApiServices		
Description	Adds / removes the service FlsTst_StartFgnd() from the code.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_FlsTst_00087] Definition of EcucBooleanParamDef FlsTstSuspendResumeApi ⌈

Parameter Name	FlsTstSuspendResumeApi		
Parent Container	FlsTstConfigurationOfOptApiServices		
Description	Adds / removes the services FlsTst_Suspend() and FlsTst_Resume() from the code.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_FlsTst_00099] Definition of EcucBooleanParamDef FlsTstTestEccApi ⌈

Parameter Name	FlsTstTestEccApi
Parent Container	FlsTstConfigurationOfOptApiServices
Description	Adds / removes the service FlsTst_TestEcc() from the code.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	false

▽

△

Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_FlsTst_00095] Definition of EcucBooleanParamDef FlsTstVersionInfoApi ⌈

Parameter Name	FlsTstVersionInfoApi		
Parent Container	FlsTstConfigurationOfOptApiServices		
Description	Adds / removes the service FlsTst_GetVersionInfo() from the code.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

10.2.6 FlsTstDemEventParameterRefs

[ECUC_FlsTst_00170] Definition of EcucParamConfContainerDef FlsTstDemEventParameterRefs ⌈

Container Name	FlsTstDemEventParameterRefs
Parent Container	FlsTst
Description	Container for the references to DemEventParameter elements which shall be invoked using the API Dem_SetEventStatus in case the corresponding error occurs. The Event Id is taken from the referenced DemEventParameter's DemEventId symbolic value. The standardized errors are provided in this container and can be extended by vendor-specific error references.
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
FLSTST_E_FLSTST_FAILURE	0..1	[ECUC_FlsTst_00171]

No Included Containers

⌋

[ECUC_FlsTst_00171] Definition of EcucReferenceDef FLSTST_E_FLSTST_FAILURE ⌈

Parameter Name	FLSTST_E_FLSTST_FAILURE		
Parent Container	FlsTstDemEventParameterRefs		
Description	Reference to the DemEventParameter which shall be issued when the error "Flash Failure" has occurred.		
Multiplicity	0..1		
Type	Symbolic name reference to DemEventParameter		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

10.2.7 FlsTstBlockBgnd

[ECUC_FlsTst_00173] Definition of EcucParamConfContainerDef FlsTstBlockBgnd ⌈

Container Name	FlsTstBlockBgnd
Parent Container	FlsTstConfigSet
Description	This container specifies configuration parameters for an individual background test block.
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
FlsTstBgndBlockIndex	1	[ECUC_FlsTst_00174]
FlsTstBlockBaseAddress	1	[ECUC_FlsTst_00106]
FlsTstBlockSize	1	[ECUC_FlsTst_00107]
FlsTstSignatureAddress	1	[ECUC_FlsTst_00123]
FlsTstTestAlgorithm	1	[ECUC_FlsTst_00101]

No Included Containers

⌋

[ECUC_FlsTst_00174] Definition of EcucIntegerParamDef FlsTstBgndBlockIndex ⌈

Parameter Name	FlsTstBgndBlockIndex		
Parent Container	FlsTstBlockBgnd		
Description	The scheduling for background test shall follow an order defined by this index. ’0’ means highest priority.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 4294967295		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_FlsTst_00106] Definition of EcucIntegerParamDef FlsTstBlockBaseAddress ⌈

Parameter Name	FlsTstBlockBaseAddress		
Parent Container	FlsTstBlockBgnd, FlsTstBlockFgnd		
Description	Start Address of the Flash block.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 18446744073709551615		
Default value	–		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

⌋

[ECUC_FlsTst_00107] Definition of EcucIntegerParamDef FlsTstBlockSize ⌈

Parameter Name	FlsTstBlockSize	
Parent Container	FlsTstBlockBgnd, FlsTstBlockFgnd	
Description	This parameter shall represent the Flash Test block size.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 4294967295	
Default value	–	
Post-Build Variant Value	true	

▽

△

Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

⌋

[ECUC_FlsTst_00123] Definition of EcucIntegerParamDef FlsTstSignatureAddress ⌈

Parameter Name	FlsTstSignatureAddress		
Parent Container	FlsTstBlockBgnd, FlsTstBlockFgnd		
Description	Address of the signature reference value of the Flash test block.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 18446744073709551615		
Default value	–		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

⌋

[ECUC_FlsTst_00101] Definition of EcucEnumerationParamDef FlsTstTestAlgorithm ⌈

Parameter Name	FlsTstTestAlgorithm		
Parent Container	FlsTstBlockBgnd, FlsTstBlockFgnd		
Description	This is the configuration of the test algorithm. The availability of algorithm is implementation specific.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	FLSTST_16BIT_CRC	–	
	FLSTST_32BIT_CRC	–	
	FLSTST_8BIT_CRC	–	
	FLSTST_CHECKSUM	–	
	FLSTST_DUPLICATED_ MEMORY	–	
	FLSTST_ECC	–	
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	X	VARIANT-POST-BUILD

▽

△

Scope / Dependency	scope: local

⌋

10.2.8 FlsTstBlockFgnd

[ECUC_FlsTst_00105] Definition of EcucParamConfContainerDef FlsTstBlockFgnd ⌈

Container Name	FlsTstBlockFgnd
Parent Container	FlsTstConfigSet
Description	This container specifies configuration parameters for an individual foreground test block.
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
FlsTstBlockBaseAddress	1	[ECUC_FlsTst_00106]
FlsTstBlockSize	1	[ECUC_FlsTst_00107]
FlsTstFgndBlockIndex	1	[ECUC_FlsTst_00151]
FlsTstSignatureAddress	1	[ECUC_FlsTst_00123]
FlsTstTestAlgorithm	1	[ECUC_FlsTst_00101]

No Included Containers

⌋

For parameter table [ECUC_FlsTst_00106] FlsTstBlockBaseAddress, see definition below container FlsTstBlockBgnd.

For parameter table [ECUC_FlsTst_00107] FlsTstBlockSize, see definition below container FlsTstBlockBgnd.

[ECUC_FlsTst_00151] Definition of EcucIntegerParamDef FlsTstFgndBlockIndex ⌈

Parameter Name	FlsTstFgndBlockIndex	
Parent Container	FlsTstBlockFgnd	
Description	Index identifies block to be tested by FlsTst_StartFgnd();	
Multiplicity	1	
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)	
Range	0 .. 4294967295	

▽

△

Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

For parameter table [ECUC_FlsTst_00123] FlsTstSignatureAddress, see definition below container FlsTstBlockBgnd.

For parameter table [ECUC_FlsTst_00101] FlsTstTestAlgorithm, see definition below container FlsTstBlockBgnd.

10.3 Published Information

For details refer to the chapter 10.3 “Published Information” in [1].

A Not applicable requirements

[SWS_FlsTst_NA_00166]

Upstream requirements: SRS_BSW_00344, SRS_BSW_00159, SRS_BSW_00167, SRS_BSW_00170, SRS_BSW_00419, SRS_BSW_00398, SRS_BSW_00375, SRS_BSW_00416, SRS_BSW_00168, SRS_BSW_00423, SRS_BSW_00424, SRS_BSW_00425, SRS_BSW_00426, SRS_BSW_00427, SRS_BSW_00428, SRS_BSW_00429, SRS_BSW_00432, SRS_BSW_00433, SRS_BSW_00422, SRS_BSW_00417, SRS_BSW_00437, SRS_SPAL_12267, SRS_SPAL_12461, SRS_SPAL_12462, SRS_SPAL_12463, SRS_SPAL_12068, SRS_SPAL_12069, SRS_SPAL_12169, SRS_SPAL_12075, SRS_SPAL_12129, SRS_SPAL_12064, SRS_SPAL_12067, SRS_SPAL_12077, SRS_SPAL_12078, SRS_SPAL_12092, SRS_SPAL_12265, SRS_FlsTst_14221

⌈These requirements are not applicable to this specification.⌋

B Change history of AUTOSAR traceable items

Please note that the lists in this chapter also include traceable items that have been removed from the specification in a later version. These items do not appear as hyperlinks in the document.

B.1 Traceable item history of this document according to AUTOSAR Release R24-11

B.1.1 Added Specification Items in R24-11

none

B.1.2 Changed Specification Items in R24-11

none

B.1.3 Deleted Specification Items in R24-11

none