Document Title	Requirements on RAM Test
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	114
Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R23-11

Document Change History

Date	Release	Changed by	Change Description
2023-11-23	R23-11	AUTOSAR Release Management	• No content changes
2022-11-24	R22-11	AUTOSAR Release Management	• No content changes
2021-11-25	R21-11	AUTOSAR Release Management	• No content changes
2020-11-30	R20-11	AUTOSAR Release Management	• Editorial changes
2019-11-28	R19-11	AUTOSAR Release Management	• No content changes • Changed Document Status from Final to published
2018-10-31	4.4.0	AUTOSAR Release Management	• Editorial changes
2017-12-08	4.3.1	AUTOSAR Release Management	• Removed obsolete references
2016-11-30	4.3.0	AUTOSAR Release Management	• Add section 5 Requirements Tracing
2015-07-31	4.2.2	AUTOSAR Release Management	• Update of the document for Diverse corrections • Editorial changes
2014-10-31	4.2.1	AUTOSAR Release Management	• Editorial changes

Document Change History

Date	Release	Changed by	Change Description
2013-10-31	4.1.2	AUTOSAR Release Management	• Editorial changes
2013-03-15	4.1.1	AUTOSAR Administration	• Formal update of the document template and requirements tracing • Adjustment to ISO 26262
2010-09-30	3.1.5	AUTOSAR Administration	• Modified SRS_RamTst_13801 • Legal disclaimer revised
2008-08-13	3.1.1	AUTOSAR Administration	• Legal disclaimer revised
2007-12-21	3.0.1	AUTOSAR Administration	• Document meta information extended • Small layout adaptations made
2007-01-24	2.1.15	AUTOSAR Administration	• "Advice for users" revised • "Revision Information" added
2006-11-28	2.1	AUTOSAR Administration	• [BSW13814] rejected "modified Hamming code" test removed • Legal disclaimer revised
2006-05-16	2.0	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.

Table of Contents

1 Scope of this document

This document specifies requirements on the module RAM Test.

This document covers requirements only for software algorithms to check the RAM. A hardware RAM check (like ECC check) is not in the scope of this document.

2 How to read this document

Each requirement has its unique identifier starting with the prefix “BSW” (for “Basic Software”). For any review annotations, remarks or questions please refer to this unique ID rather than chapter or page numbers!

2.1 Conventions used

- The representation of requirements in AUTOSAR documents follows the table specified in [5].

- In requirements, the following specific semantics are used

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted . Note that the requirement level of the document in which they are used modifies the force of these words.

- MUST: This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.
- MUST NOT: This phrase, or the phrase „SHALL NOT“, means that the definition is an absolute prohibition of the specification.
- SHOULD: This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
- SHOULD NOT: This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behaviour is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behaviour described with this label.
- MAY: This word, or the adjective „OPTIONAL“, means that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation, which does not include a particular option, MUST be prepared to interoperate with another implementation, which does include the option, though perhaps with reduced functionality. In the same vein an implementation, which does include a particular option, MUST be prepared to interoperate with another implementation, which does not include the option (except, of course, for the feature the option provides.)

2.2 Requirement structure

Each module specific chapter contains a short functional description of the Basic Software Module. Requirements of the same kind within each chapter are grouped under the following headlines (where applicable):

Functional Requirements:

- Configuration (which elements of the module need to be configurable)
- Initialization
- Normal Operation
- Shutdown Operation
- Fault Operation
- ...

Non-Functional Requirements:

- Timing Requirements
- Resource Usage
- Usability
- Output for other WPs (e.g. Description Templates, Tooling,...)
- ...

3 Acronyms and abbreviations

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

Acronym:	***Description:***
ECU	Electric Control Unit
EOL	End Of Line Often used in the term 'EOL Programming' or 'EOL Configuration'
MAL	Old name of Microcontroller Abstraction Layer (replaced by MCAL because 'MAL' is a french term meaning 'bad')
MCAL	Microcontroller Abstraction Layer
MCU	Microcontroller Unit
NMI	Non maskable interrupt
OS	Operating System
SPAL	The name of this working group
SFR	Special Function Register
RTE	Runtime environment
WP	Work Package

Abbreviation:	***Description:***
STD	Standard
REQ	Requirement
UNINIT	Uninitialized (= not initialized)

As this is a document from professionals for professionals, all other terms are expected to be known.

4 Functional Overview

This module has the task to test the RAM memory area by software.

5 Requirements Tracing

Requirement	Description	Satisfied by
RS_BRF_00057	AUTOSAR shall define a memory mapping mechanism	SRS_RamTst_13802
RS_BRF_00129	AUTOSAR shall support data corruption detection and protection	SRS_RamTst_13803, SRS_RamTst_13804, SRS_RamTst_13809, SRS_RamTst_13811, SRS_RamTst_13822, SRS_RamTst_13823, SRS_RamTst_13824
RS_BRF_01048	AUTOSAR module design shall support modules to cooperate in a multitasking environment	SRS_RamTst_13809
RS_BRF_01056	AUTOSAR BSW modules shall provide standardized interfaces	SRS_RamTst_13810
RS_BRF_01064	AUTOSAR BSW shall provide callback functions in order to access upper layer modules	SRS_RamTst_13820
RS_BRF_01472	AUTOSAR shall support modes	SRS_RamTst_13822, SRS_RamTst_13823, SRS_RamTst_13824
RS_BRF_01504	AUTOSAR shall handle memory corruption resulting from ECU sleep	SRS_RamTst_13804
RS_BRF_02040	AUTOSAR BSW and RTE shall ensure data consistency	SRS_RamTst_13816
RS_BRF_02048	AUTOSAR shall support usage of hardware memory protection features to enhance safety	SRS_RamTst_13825
RS_BRF_02064	AUTOSAR shall use hardware communication data integrity mechanisms	SRS_RamTst_13825
RS_BRF_02224	AUTOSAR shall support run-time hardware tests	SRS_RamTst_13800, SRS_RamTst_13801, SRS_RamTst_13804, SRS_RamTst_13809, SRS_RamTst_13811, SRS_RamTst_13812, SRS_RamTst_13822, SRS_RamTst_13823, SRS_RamTst_13824

6 Requirements Specification

6.1 Functional Requirements

6.1.1 Configuration

6.1.1.1 [SRS_RamTst_13800] The number of tested cells shall be changeable at runtime

⌈

Type:	Valid
Description:	To react on different requirements (sleep, driving cycle) the user shall have the possibility to change the number of tested cells per cycle “online”.
Rationale:	Influences the interrupt disable times.
Use Case:	When car is driven the system interrupt locking time must be much shorter than in case of sleep mode.
Dependencies:	--
Supporting Material:	--

⌋(RS_BRF_02224)

6.1.1.2 [SRS_RamTst_13801] Test cell size shall be a published parameter

⌈

Type:	Valid
Description:	The implementer shall chose the test cell size (bit, byte, word, long word) for a specific test implementation according to the controller properties. This parameter shall be published to the integrator along with a specific implementation..
Rationale:	--
Use Case:	Runtime optimization by implementer due to controller properties.
Dependencies:	--
Supporting Material:	--

⌋(RS_BRF_02224)

6.1.1.3 [SRS_RamTst_13802] Multiple RAM areas shall be configurable at post build/ link time

⌈

Type:	valid
Description:	It shall be possible to configure multiple RAM areas (by configuring their start and end address). If two RAM areas overlap and an error is detected in the overlapping region while testing one of the blocks, the driver does not guarantee to update the status of the other block.
Rationale:	--
Use Case:	User shall have the possibility to configure the memory mapping.
Dependencies:	--
Supporting Material:	--

⌋(RS_BRF_00057)

6.1.1.4 [SRS_RamTst_13803] A subset of available RAM Test algorithms shall be selectable at pre-compile time

⌈

Type:	valid
Description:	The user shall select at pre-compile time the available algorithms which matches to the project safety requirements.
Rationale:	Avoid unused code.
Use Case:	Depending on ECU safety analysis different RAM test algorithms should be selectable. To save ROM space the algorithms should be selectable at compile time
Dependencies:	--
Supporting Material:	--

⌋(RS_BRF_00129)

6.1.1.5 [SRS_RamTst_13804] A subset of the pre-compile time selected RAM Check test algorithms shall be selectable at runtime

⌈

Type:	valid
Description:	The user shall select the test algorithms from those available at runtime to conform to the project safety requirements.
Rationale:	Different levels of testing are available.
Use Case:	During normal operation a simple test is executed and before going to sleep mode a more complex RAM test algorithm will be executed. The complex RAM test can not be executed during normal operation because of stronger interrupt latency requirements.
Dependencies:	[SRS_RamTst_13803]
Supporting Material:	--

⌋(RS_BRF_00129,RS_BRF_02224,RS_BRF_01504)

6.1.2 Normal Operation

6.1.2.1 [SRS_RamTst_13822] A safety mechanism with low coverage shall be available

⌈

Type:	New
Description:	A safety measure which fulfils a diagnostic coverage of 60% shall be available. It may be provided by hardware means, by an appropriate test algorithm, or both.
Rationale:	Detect permanent faults in RAM.
Use Case:	Support of EOL, quick start-up tests and where low diagnostic coverage tests are required, e.g. if system has safety goals with low ISO 26262 ASIL rating only.
Dependencies:	--
Supporting Material:	ISO 26262-5:2018, Tables 4, 5 and D.1 ISO 26262-11:2018 Table 33, clause 5.1.13.5 (RAM Pattern test), 5.1.13.6 (Parity bit), 5.1.13.7 (RAM March test)

⌋(RS_BRF_00129, RS_BRF_02224,RS_BRF_01472)

6.1.2.2 [SRS_RamTst_13823] A Test algorithm with medium coverage shall be available

⌈

Type:	New
Description:	A safety measure which fulfils a diagnostic coverage of 90 % shall be available. It may be provided by hardware means, by an appropriate test algorithm, or both.
Rationale:	Detect permanent faults in RAM.
Use Case:	Support of EOL, start-up tests and where medium diagnostic coverage tests are required, e.g. if the latent fault metric of ISO 26262 for the ASIL level of the safety goals of a system can be achieved with medium coverage.
Dependencies:	--
Supporting Material:	ISO 26262-5:2018, Tables 4, 5 and D.1 ISO 26262-11:2018 Table 33, clause 5.1.13.5 (RAM Pattern test), 5.1.13.6 (Parity bit), 5.1.13.7 (RAM March test)

⌋(RS_BRF_00129, RS_BRF_02224,RS_BRF_01472)

6.1.2.3 [SRS_RamTst_13824] A Test algorithm with high coverage shall be available

⌈

Type:	New
Description:	A safety measure which fulfils a diagnostic coverage of 99 % shall be available. It may be provided by hardware means, by an appropriate test algorithm, or both.
Rationale:	Detect permanent faults in RAM.
Use Case:	Support of EOL, diligent start-up, shut-down or runtime tests and where high diagnostic coverage tests are required, e.g. if system has a safety goal with high ISO 26262 ASIL rating.
Dependencies:	--
Supporting Material:	ISO 26262-5:2018, Tables 4, 5 and D.1 ISO 26262-11:2018 Table 33, clause 5.1.13.5 (RAM Pattern test), 5.1.13.6 (Parity bit), 5.1.13.7 (RAM March test)

⌋(RS_BRF_00129, RS_BRF_02224,RS_BRF_01472)

6.1.2.4 [SRS_RamTst_13809] It shall be possible to divide the RAM test execution into smaller pieces

⌈

Type:	Valid
Description:	It shall be possible to divide the RAM test execution into smaller pieces. With one call of the RAM test it shall be possible to execute only a part of the whole RAM test.
Rationale:	Avoid long interrupt disable times
Use Case:	Drivers who need short interrupt latency times
Dependencies:	--
Supporting Material:	--

⌋(RS_BRF_00129, RS_BRF_02224,RS_BRF_01048)

6.1.2.5 [SRS_RamTst_13810] Current status of RAM test execution per block shall be available through a get status interface

⌈

Type:	Valid
Description:	RAM test execution status per block (RESULT_NOT_TESTED, RESULT_OK, RESULT_NOT_OK, and RESULT_UNDEFINED) shall be provided to the user. User shall have the possibility to get the status of the RAM test at any time. This shall be implemented as a get status interface and shall be configurable during compile time. This function shall be optional.
Rationale:	--
Use Case:	Diagnostics may need to know if there has been errors occurred or not.
Dependencies:	--
Supporting Material:	--

⌋(RS_BRF_01056)

6.1.2.6 [SRS_RamTst_13820] RAM test execution status shall be provided by a notification mechanism

⌈

Type:	Valid
Description:	Information when error has been detected or test has been finished shall be provided to the user by a notification mechanism. This function shall be optional.
Rationale:	--
Use Case:	Diagnostics may need to know immediately if an error has been detected or not.
Dependencies:	--
Supporting Material:	--

⌋(RS_BRF_01064)

6.1.2.7 [SRS_RamTst_13811] The RAM test module shall be able to perform its tests in a non-destructive manner

⌈

Type:	Valid
Description:	The RAM test module shall be able to perform its tests in a non-destructive manner.
Rationale:	Original data shall be preserved
Use Case:	Destroying of all RAM data may lead to longer reaction times (e.g. wake up), higher resource consumption (e.g.: EEPROM)
Dependencies:	--
Supporting Material:	--

⌋(RS_BRF_02224,RS_BRF_00129)

6.1.2.8 [SRS_RamTst_13812] The RAM test module shall be able to perform its tests in a destructive manner

⌈

Type:	Valid
Description:	The RAM test module shall be able to perform its tests in a destructive manner. The state of the RAM after testing shall be defined.
Rationale:	Original data does not need to be preserved
Use Case:	--

Dependencies:	--
Supporting Material:	--

⌋(RS_BRF_02224)

6.2 Non-Functional Requirements

6.2.1 [SRS_RamTst_13816] Effects of Instruction / Data queue shall be taken into account

⌈

Type:	valid
Description:	When writing to a cell and after reading back this may lead to the problem that the read-back value comes from the data queue and not from the RAM cell to be tested. In that case instruction(s) have to be injected to eliminate such an effect.
Rationale:	Read back the value from a tested cell
Use Case:	Controller with instruction or data queue may have such effects
Dependencies:	--
Supporting Material:	--

⌋(RS_BRF_02040)

6.2.2 [SRS_RamTst_13825] The RAM Test Module shall be usable to comply with requirements of the different ASIL levels of ISO 26262.

⌈

Type:	New
Description:	The RAM Test Module shall provide and document (fault models and fault coverage) diagnostic capability for permanent faults in RAMs to enable fulfillment of the latent fault metric targets of ISO 26262 for the different ASIL levels.
Rationale:	Usability of AUTOSAR for sytems which need to comply with ISO 26262.
Use Case:	--
Dependencies:	--
Supporting Material:	ISO 26262-5:2018, Tables 4, 5 and D.1 ISO 26262-11:2018 Table 33, clause 5.1.13.5 (RAM Pattern test), 5.1.13.6 (Parity bit), 5.1.13.7 (RAM March test)

⌋(RS_BRF_02048,RS_BRF_02064)

7 References

7.1 Deliverables of AUTOSAR

[1] Glossary
AUTOSAR_FO_TR_Glossary.pdf

[2] Layered Software Architecture
AUTOSAR_CP_EXP_LayeredSoftwareArchitecture.pdf

[3] General Requirements on Basic Software Modules
AUTOSAR_CP_SRS_BSWGeneral.pdf

[4] General Requirements on SPAL
AUTOSAR_CP_SRS_SPALGeneral.pdf

[5] Software Standardization Template
AUTOSAR_FO_TPS_StandardizationTemplate.pdf

7.2 Related standards and norms

[6] CEI/IEC 61508-2:2000: Requirements for electrical/electronic/programmable electronic safety-related systems

[7] ISO 26262 (all parts) - Road vehicles - Functional Safety http://www.iso.org