Document Title	Requirements on RAM Test
Document Owner	AUTOSAR GbR
Document Responsibility	AUTOSAR GbR
Document Identification No	114
Document Classification	Auxiliary

Document Version	1.1.3
Document Status	Final
Part of Release	3.1
Revision	0001

Document Change History

Date	Version	Changed by	Change Description
23.06.2008	1.1.3	AUTOSAR Administration	Legal disclaimer revised
31.10.2007	1.1.2	AUTOSAR Administration	• Document meta information extended • Small layout adaptations made
24.01.2007	1.1.1	AUTOSAR Administration	• “Advice for users” revised • “Revision Information” added
01.12.2006	1.1.0	AUTOSAR Administration	• [BSW13814] rejected “modified Hamming code” test removed • Legal disclaimer revised
14.03.2006	1.0.0	AUTOSAR Administration	Initial release

Page left intentionally blank

Disclaimer

Advice to users of AUTOSAR Specification Documents:

AUTOSAR Specification Documents may contain exemplary items (exemplary reference models, "use cases", and/or references to exemplary technical solutions, devices, processes or software).
Any such exemplary items are contained in the Specification Documents for illustration purposes only, and they themselves are not part of the AUTOSAR Standard. Neither their presence in such Specification Documents, nor any later documentation of AUTOSAR conformance of products actually implementing such exemplary items, imply that intellectual property rights covering such exemplary items are licensed under the same rules as applicable to the AUTOSAR Standard.

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

In requirements, the following specific semantics are used (taken from Request for Comment RFC 2119 from the Internet Engineering Task Force IETF)

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119. 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'
HIS	Herstellerinitiative Software
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 Requirement Specification

4.1 RAM Test

4.1.1 Functional Overview

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

4.1.2 Functional Requirements

4.1.2.1 Configuration

4.1.2.1.1 [BSW13800] Number of tested cells shall be changeable at runtime

Initiator:	Delphi
Date:	22.07.2005
Short Description:	The number of tested cells shall be changeable at runtime.
Type:	New
Importance:	Medium
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:	--
Conflicts:	--
Supporting Material:	--

4.1.2.1.2 [BSW13801] Test cell size shall be configurable at pre-compile time

Initiator:	Delphi
Date:	23.11.2005
Short Description:	Test cell size shall be configurable at pre-compile time.
Type:	Changed
Importance:	High
Description:	The user shall have the possibility to change test cell size (bit, byte, word, long word) dependent on the controller properties.
Rationale:	--
Use Case:	Runtime optimization due to controller properties.
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.2.1.3 [BSW13802] Multiple RAM areas shall be configurable at post build/ link time

Initiator:	Delphi
Date:	23.11.2005
Short Description:	Multiple RAM areas shall be configurable at post build/ link time.
Type:	Changed
Importance:	High
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:	--
Conflicts:	--
Supporting Material:	--

4.1.2.1.4 [BSW13803] A subset of available RAM Test algorithms shall be selectable at pre-compile time

Initiator:	Delphi
Date:	23.11.2005
Short Description:	A subset of available RAM Test algorithms shall be selectable at pre-compile time.
Type:	Changed
Importance:	High
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:	--
Conflicts:	--
Supporting Material:	--

4.1.2.1.5 [BSW13804] A subset of the pre-compile time selected RAM Check test algorithms shall be selectable at runtime

Initiator:	WP4.2.2.1.12
Date:	23.11.2005
Short Description:	A subset of the pre-compile time selected RAM Check test algorithms shall be selectable at runtime.
Type:	Changed
Importance:	High
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:	[BSW13803]
Conflicts:	--
Supporting Material:	--

4.1.2.2 Normal Operation

4.1.2.2.1 [BSW13805] Checkerboard test algorithm shall be available

Initiator:	Delphi
Date:	01.06.2005
Short Description:	A checkerboard test algorithm shall be available
Type:	New
Importance:	High
Description:	A test algorithm, which fulfils an error detection level of 60 % shall be available. The Checkerboard test algorithm shall be selectable.
Rationale:	--
Use Case:	Support of EOL, quick start-up tests and where low diagnostic coverage tests are required.
Dependencies:	--
Conflicts:	--
Supporting Material:	ISO 61508-2 (table A.6), ISO 61508-7 (table A.5.1)

4.1.2.2.2 [BSW13807] March test algorithm shall be available

Initiator:	Delphi
Date:	01.06.2005
Short Description:	The March test algorithm shall be available
Type:	New
Importance:	High
Description:	A test algorithm, which fulfils an error detection level of 60 % shall be available. The March test algorithm shall be selectable.
Rationale:	--
Use Case:	Support of EOL, quick start-up tests and where low diagnostic coverage tests are required.
Dependencies:	--
Conflicts:	--
Supporting Material:	ISO 61508-2 (table A.6) , ISO 61508-7 (table A.5.1)

4.1.2.2.3 [BSW13806] Walk path test algorithm shall be available

Initiator:	Delphi
Date:	01.06.2005
Short Description:	A medium diagnostic coverage test algorithm shall be available.
Type:	New
Importance:	High
Description:	A medium diagnostic coverage test (90% failure coverage) algorithm shall be available. The Walk path test algorithm shall be selectable.

Rationale:	--
Use Case:	Support of medium diagnostic coverage tests.
Dependencies:	--
Conflicts:	--
Supporting Material:	ISO 61508-2 (table A.6), ISO 61508-7 (table A.5.2)

4.1.2.2.4 [BSW13808] Galpat test algorithm shall be available

Initiator:	Delphi
Date:	01.06.2005
Short Description:	A high diagnostic coverage test algorithm shall be available. The Galpat test algorithm shall be selectable
Type:	New
Importance:	High
Description:	A high diagnostic coverage test algorithm (99% failure coverage) shall be available. The Galpat test algorithm shall be selectable.
Rationale:	--
Use Case:	Support of high diagnostic coverage tests
Dependencies:	--
Conflicts:	--
Supporting Material:	ISO 61508-2 (table A.6), ISO 61508-7 (table A.5.3)

4.1.2.2.5 [BSW13818] Transparent Galpat test algorithm shall be available

Initiator:	Delphi
Date:	07.07.2005
Short Description:	Transparent Galpat test algorithm shall be available
Type:	New
Importance:	High
Description:	A high diagnostic coverage test algorithm (99% failure coverage) shall be available. The Transparent Galpat test algorithm shall be selectable.
Rationale:	--
Use Case:	Support of high diagnostic coverage tests
Dependencies:	--
Conflicts:	--
Supporting Material:	ISO 61508-2 (table A.6), ISO 61508-7 (table A.5.3)

4.1.2.2.6 [BSW13813] Abraham test algorithm shall be available

Initiator:	PSA
Date:	16.06.2005
Short Description:	Abraham test algorithm shall be available
Type:	New
Importance:	High
Description:	A high diagnostic coverage test algorithm (99% failure coverage) shall be available. The Abraham test algorithm shall be selectable.
Rationale:	--
Use Case:	Support of high diagnostic coverage tests
Dependencies:	--
Conflicts:	--

Supporting Material:	ISO 61508-2 (table A.6), ISO 61508-7 (table A.5.4)

4.1.2.2.7 [BSW13809] RAM Test Execution Management

Initiator:	Delphi
Date:	16.08.2005
Short Description:	RAM Test Execution Management
Type:	New
Importance:	High
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:	--
Conflicts:	--
Supporting Material:	--

4.1.2.2.8 [BSW13810] Current status of RAM test execution per block shall be available through a get status interface

Initiator:	WP4.2.2.1.12
Date:	23.11.2005
Short Description:	Current status of RAM test execution per block shall be available through a get status interface.
Type:	Changed
Importance:	Medium
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:	--
Conflicts:	--.
Supporting Material:	--

4.1.2.2.9 [BSW13820] RAM test execution status shall be provided by a notification mechanism

Initiator:	Delphi
Date:	22.07.2005
Short Description:	RAM test execution status shall be provided by a notification mechanism.
Type:	New
Importance:	Medium
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:	--
Conflicts:	--
Supporting Material:	--

4.1.2.2.10 [BSW13811] Non Destructive RAM Test

Initiator:	WP4.2.2.1.12
Date:	01.06.2005
Short Description:	Non destructive RAM test
Type:	New
Importance:	High
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:	--
Conflicts:	--
Supporting Material:	--

4.1.2.2.11 [BSW13812] Destructive RAM Test

Initiator:	WP4.2.2.1.12
Date:	01.06.2005
Short Description:	Destructive RAM Test
Type:	New
Importance:	High
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:	--
Conflicts:	--
Supporting Material:	--

4.1.3 Non-Functional Requirements (Qualities)

4.1.3.1 [BSW13816] Effects of Instruction / Data queue shall be taken into account

Initiator:	Delphi
Date:	01.07.2005
Short Description:	Effects of Instruction / Data queue / cache shall be taken into account
Type:	New
Importance:	High
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:	--
Conflicts:	--
Supporting Material:	--

4.1.3.2 [BSW13821] The RAM Test Module shall be usable within SIL3 Environments

Initiator:	Delphi
Date:	26.09.2005
Short Description:	The RAM Test Module shall be usable within SIL3 Environments.
Type:	New
Importance:	High
Description:	The RAM Test Module shall provide algorithms according to IEC 61508-2. Which shall be used in safety relevant systems.
Rationale:	Support of systems which have to fulfill safety requirements regarding to IEC 61508-2 needs special algorithms.
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	IEC 61508-2, Table 2, 3 and A.6,

5 References

5.1 Deliverables of AUTOSAR

[1] Glossary
https:/svn2.autosar.org/repos2/22_Releases
AUTOSAR_Glossary.pdf

[2] Layered Software Architecture
https:/svn2.autosar.org/repos2/22_Releases
AUTOSAR_LayeredSoftwareArchitecture.pdf

[3] General Requirements on Basic
https:/svn2.autosar.org/repos2/22_Releases
AUTOSAR_SRS_General.pdf

[4] General Requirements on SPAL
https:/svn2.autosar.org/repos2/22_Releases
AUTOSAR_SRS_SPAL_General.pdf

5.2 Related standards and norms

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