

Document Title	Requirements on EEPROM Driver
Document Owner	AUTOSAR GbR
Document Responsibility	AUTOSAR GbR
Document Identification No	192
Document Classification	Auxiliary

Document Version	2.0.4
Document Status	Final
Part of Release	3.1
Revision	0001

Document Change History			
Date	Version	Changed by	Change Description
23.06.2008	2.0.4	AUTOSAR Administration	Legal disclaimer revised
30.10.2007	2.0.3	AUTOSAR Administration	<ul style="list-style-type: none"> Document meta information extended Small layout adaptations made
24.01.2007	2.0.2	AUTOSAR Administration	<ul style="list-style-type: none"> “Advice for users” revised “Revision Information” added
28.11.2006	2.0.1	AUTOSAR Administration	Legal disclaimer revised
24.01.2006	2.0.0	AUTOSAR Administration	Release as a separate document. The SRS SPAL V1.0.0 has been split into 12 independent documents for Release 2.0 <ul style="list-style-type: none"> minor changes regarding formal issues
11.07.2005	1.0.0	AUTOSAR Administration	Initial release as a part of the SRS SPAL V1.0.0

Page left intentionally blank

Disclaimer

This document of a specification as released by the AUTOSAR Development Partnership is intended **for the purpose of information only**. The commercial exploitation of material contained in this specification requires membership of the AUTOSAR Development Partnership or an agreement with the AUTOSAR Development Partnership. The AUTOSAR Development Partnership will not be liable for any use of this specification. Following the completion of the development of the AUTOSAR specifications commercial exploitation licenses will be made available to end users by way of written License Agreement only.

No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher." The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Copyright © 2004-2008 AUTOSAR Development Partnership. All rights reserved.

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	5
2	How to read this document.....	6
2.1	Conventions used.....	6
2.2	Requirements structure	7
3	Acronyms and abbreviations	8
4	Requirement Specification.....	9
4.1	Internal EEPROM Driver	9
4.1.1	Functional Overview.....	9
4.1.2	Functional Requirements	9
4.1.2.1	Configuration.....	9
4.1.2.1.1	[BSW096] EEPROM driver static configuration	9
4.1.2.1.2	[BSW12071] Publication of EEPROM properties.....	10
4.1.2.2	Normal Operation.....	10
4.1.2.2.1	[BSW087] EEPROM read function	10
4.1.2.2.2	[BSW088] EEPROM write function	11
4.1.2.2.3	[BSW089] EEPROM erase function.....	11
4.1.2.2.4	[BSW12091] EEPROM compare function.....	12
4.1.2.2.5	[BSW090] EEPROM cancel function	12
4.1.2.2.6	[BSW091] EEPROM status function	13
4.1.2.2.7	[BSW12156] EEPROM mode selection function.....	13
4.1.2.2.8	[BSW092] EEPROM write cycle reduction.....	14
4.1.2.2.9	[BSW094] EEPROM segmentation handling	14
4.1.2.2.10	[BSW095] EEPROM job management	15
4.1.2.2.11	[BSW12047] EEPROM job processing function.....	15
4.1.2.2.12	[BSW12157] Job processing – normal mode.....	16
4.1.2.2.13	[BSW12072] Job processing – fast mode	16
4.1.3	Non-Functional Requirements (Qualities)	17
4.1.3.1	[BSW12050] EEPROM job processing execution time	17
4.2	External EEPROM Driver	18
4.2.1	Functional Overview.....	18
4.2.2	Functional Requirements	18
4.2.2.1	General	18
4.2.2.1.1	[BSW12051] Functional scope.....	18
4.2.2.2	Configuration.....	18
4.2.2.2.1	[BSW12164] SPI channel configuration	18
4.2.2.3	Normal Operation.....	19
4.2.2.3.1	[BSW12124] SPI access modes	19
4.2.3	Non-Functional Requirements (Qualities)	19
4.2.3.1	[BSW12052] Common EEPROM API	19
4.2.3.2	[BSW12053] Microcontroller independency	20
5	References	21
5.1	Deliverables of AUTOSAR	21
5.2	Related standards and norms	21

1 Scope of this document

This document specifies requirements on the module EEPROM Driver.

Constraints

First scope for specification of requirements on basic software modules is systems which are not safety relevant. For this reason safety requirements are assigned to medium priority.

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 behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior 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 Requirements 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)
- Initialisation
- 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:
CS	Chip select
DIO	Digital Input Output
ECU	Electric Control Unit
EOL	End Of Line Often used in the term 'EOL Programming' or 'EOL Configuration'
HIS	Herstellerinitiative Software
ICU	Interrupt Capture Unit
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
MMU	Memory Management Unit
Master	A device controlling other devices (slaves, see below)
Slave	A device beeing completely controlled by a master device
NMI	Non maskable interrupt
OS	Operating System
PLL	Phase Locked Loop
PWM	Pulse Width Modulation
RX	Reception (in the context of bus communication)
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 Internal EEPROM Driver

4.1.1 Functional Overview

The internal EEPROM driver provides services for initialization and reading, writing, erasing to/from internal EEPROM.

4.1.2 Functional Requirements

4.1.2.1 Configuration

4.1.2.1.1 [BSW096] EEPROM driver static configuration

Initiator:	BMW
Date:	10.01.2004
Short Description:	EEPROM driver static configuration
Type:	Changed (EEPROM size detailed, Interrupt control removed)
Importance:	High
Description:	<p>The following constants of the EEPROM driver shall be statically configurable:</p> <ol style="list-style-type: none"> 1. EEPROM base address 2. EEPROM size (can be equal or smaller than physical EEPROM size) 3. maximum block sizes (write, erase) processed within the job processing function 4. maximum read block sizes for normal and fast EEPROM mode processed within the job processing function 5. call cycle of cyclic job processing function for read, write, erase
Rationale:	Basic configuration
Use Case:	→ 5: needed if the EEPROM hardware does not provide this timing and/or deadline checks are necessary
Dependencies:	→ 4: see [BSW12072] Job processing – fast mode and [BSW12157] Job processing – normal mode
Conflicts:	--
Supporting Material:	BMW Specification MCAL V1.0a

4.1.2.1.2 [BSW12071] Publication of EEPROM properties

Initiator:	BMW
Date:	07.12.2005
Short Description:	Publication of EEPROM properties
Type:	Changed (use case removed)
Importance:	High
Description:	<p>The EEPROM driver description shall publish the following EEPROM properties:</p> <ul style="list-style-type: none"> • total physical EEPROM size • value of erased EEPROM cell • size of one EEPROM cell (e.g. 8bit, 16bit, ...) • Physical memory segmentation (minimum writable/readable/erasable units)
Rationale:	For configuration of higher layers
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.2.2 Normal Operation

4.1.2.2.1 [BSW087] EEPROM read function

Initiator:	BMW
Date:	10.01.2004
Short Description:	EEPROM read function
Type:	New
Importance:	High
Description:	The EEPROM driver shall provide an asynchronous read function that reads a data block starting from the requested EEPROM address with the passed length from the internal EEPROM.
Rationale:	Basic functionality
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	BMW Specification MCAL V1.0a

4.1.2.2.2 [BSW088] EEPROM write function

Initiator:	BMW
Date:	10.01.2004
Short Description:	EEPROM write function
Type:	New
Importance:	High
Description:	<p>The EEPROM driver shall provide an asynchronous write function that writes a data block starting from the requested EEPROM address with the passed length to the internal EEPROM.</p> <p>If an addressed EEPROM cell is not empty, an erase operation shall be done automatically before the write command is executed.</p> <p>If the erasable block doesn't align to the write request, the driver shall buffer and rewrite the additional affected data in the block.</p>
Rationale:	Basic functionality
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	BMW Specification MCAL V1.0a

4.1.2.2.3 [BSW089] EEPROM erase function

Initiator:	BMW
Date:	25.04.2004
Short Description:	EEPROM erase function
Type:	Changed
Importance:	High
Description:	<p>The EEPROM driver shall provide an asynchronous erase function that erases a data block starting from the requested EEPROM address with the passed length from the internal EEPROM.</p> <p>If the erasable block doesn't align to the erase request, the driver shall buffer and rewrite the additional affected data in the block.</p> <p>The EEPROM driver shall choose the optimal erase strategy internally. E.g. use block erase commands if supported by EEPROM hardware and the passed borders of the data block to be erased fit to a physically erasable block.</p>
Rationale:	For some EEPROM types an erased EEPROM can be programmed much faster.
Use Case:	<ul style="list-style-type: none"> • ECU Production and fast EOL programming • Fast saving of crash data • Get sure that data is really erased
Dependencies:	--
Conflicts:	--
Supporting Material:	BMW Specification MCAL V1.0a

4.1.2.2.4 [BSW12091] EEPROM compare function

Initiator:	LEAR
Date:	17.06.2004
Short Description:	EEPROM compare function
Type:	New
Importance:	High
Description:	The EEPROM driver shall provide an asynchronous compare function that compares a section in memory with a section in EEPROM with the passed length.
Rationale:	Basic functionality
Use Case:	Compare a complete block after a hot reset could speed up the recovery of the system. This function can also be used for verifying complete data blocks after writing to EEPROM.
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.2.2.5 [BSW090] EEPROM cancel function

Initiator:	BMW
Date:	22.06.2004
Short Description:	EEPROM cancel function
Type:	Changed (this function is explicitly synchronous)
Importance:	High
Description:	The EEPROM driver shall provide a synchronous cancel function that stops the currently processed job. The states and data of the affected EEPROM cells are undefined! The EEPROM driver and controller itself shall be ready for new jobs.
Rationale:	Urgent write commands can be performed without any delay
Use Case:	Writing crash relevant data in case of detected vehicle crash.
Dependencies:	The NVRAM manager shall pay attention to the priority of the jobs. Only it is authorized to use this function!
Conflicts:	--
Supporting Material:	BMW Specification MCAL V1.0a

4.1.2.2.6 [BSW091] EEPROM status function

Initiator:	BMW
Date:	10.01.2004
Short Description:	EEPROM status function
Type:	New
Importance:	High
Description:	The EEPROM driver shall provide a synchronous function which returns the job processing status.
Rationale:	Check if EEPROM driver is busy
Use Case:	<p>Only example (will be specified within API definition):</p> <ul style="list-style-type: none"> • After Reset and before a successful initialization the RW state is UNINIT. • After a successful initialization the RW state is READY. • During job processing the RW state is BUSY. • After canceling a job the RW state is READY. • After an detected error the RW state is the associated ERROR_XXX status.
Dependencies:	--
Conflicts:	--
Supporting Material:	BMW Specification MCAL V1.0a

4.1.2.2.7 [BSW12156] EEPROM mode selection function

Initiator:	WP4.2.2.1.12
Date:	06.07.2004
Short Description:	EEPROM mode selection function
Type:	New
Importance:	High
Description:	<p>The EEPROM driver shall provide a synchronous function which allows to switch the operation mode between normal and fast EEPROM access.</p> <p>Comment: For specification of these two modes see the links below.</p>
Rationale:	Fast read operation during ECU start-up, fast write operation during ECU shut down, "cooperative" operation during normal ECU mode.
Use Case:	--
Dependencies:	[BSW12072] Job processing – fast mode [BSW12157] Job processing – normal mode [BSW12124] Fast SPI access
Conflicts:	--
Supporting Material:	--

4.1.2.2.8 [BSW092] EEPROM write cycle reduction

Initiator:	BMW
Date:	10.01.2004
Short Description:	EEPROM write cycle reduction
Type:	New
Importance:	High
Description:	<p>The EEPROM driver shall only write data if at least one data value of the affected erasable block in the EEPROM is different from the data value to be written.</p> <p>This feature shall be statically configurable (on/off).</p>
Rationale:	An erase and write cycle is only done if necessary. Thereby the lifetime of EEPROM is elongated.
Use Case:	The value '0x45' shall be written to an EEPROM cell. This value is already contained in this EEPROM cell.
Dependencies:	--
Conflicts:	--
Supporting Material:	BMW Specification MCAL V1.0a

4.1.2.2.9 [BSW094] EEPROM segmentation handling

Initiator:	BMW
Date:	10.01.2004
Short Description:	EEPROM segmentation handling
Type:	Changed (erase and compare function added)
Importance:	High
Description:	<p>The EEPROM driver shall handle the EEPROM memory segmentation. The read, write, erase and compare functions of the EEPROM driver shall resolve the physical EEPROM block sizes and segment borders by using read-modify-write operations if necessary.</p>
Rationale:	<p>Identical API behavior for different EEPROM types.</p> <p>The EEPROM driver knows the most efficient way to handle and resolve the segmentation.</p>
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	BMW Specification MCAL V1.0a

4.1.2.2.10 [BSW095] EEPROM job management

Initiator:	BMW
Date:	10.01.2004
Short Description:	EEPROM job management
Type:	New
Importance:	High
Description:	<p>The EEPROM driver shall handle only one job (read or write or erase or compare) at the same time. Requested jobs during a running job shall be rejected and handled as error.</p> <p>This error detection shall be statically configurable (on/off).</p> <p>Further explanation: The calling function is responsible for buffering and queueing of jobs, not the EEPROM driver.</p>
Rationale:	Different operations like read, write, erase, compare can't be handled at the same time and the results are dependent of the execution order.
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	BMW Specification MCAL V1.0a

4.1.2.2.11 [BSW12047] EEPROM job processing function

Initiator:	BMW
Date:	07.12.2005
Short Description:	EEPROM job processing function
Type:	Changed (remove OS package by dedicated module)
Importance:	High
Description:	<p>The EEPROM driver shall provide a function that has to be called for job processing. All job processing shall be done within this function.</p> <p>If supported by hardware, this function can be called from an interrupt. Otherwise, this function should be handled by a dedicated module.</p> <p>Further comments for better understanding: The job processing function usually contains a big state machine which processes the read/write/erase jobs and sets the driver status variable.</p>
Rationale:	Allow flexible possibilities of job processing. Fulfill requirements of OS independency.
Use Case:	Example: The job processing function is called every 10ms.
Dependencies:	--
Conflicts:	--
Supporting Material:	BMW Specification MCAL V1.0a

4.1.2.2.12 [BSW12157] Job processing – normal mode

Initiator:	WP4.2.2.1.12
Date:	06.07.2004
Short Description:	Job processing – normal mode
Type:	New
Importance:	High
Description:	In normal mode, one cycle of the job processing function of the EEPROM driver shall limit the block size that is read from EEPROM to the configured default block size. Simplified comment: Only read a few bytes during one call of the job processing function.
Rationale:	Cooperative, non-blocking scheduling of EEPROM driver in normal operation mode.
Use Case:	Example: In normal EEPROM mode, the maximum block size of read data is 16. In fast EEPROM mode, the maximum block size of read data is 128.
Dependencies:	[BSW096] EEPROM driver static configuration, maximum read block sizes [BSW12156] EEPROM mode selection function
Conflicts:	--
Supporting Material:	--

4.1.2.2.13 [BSW12072] Job processing – fast mode

Initiator:	WP4.2.2.1.12
Date:	26.05.2004
Short Description:	Fast EEPROM access
Type:	Changed during WP4.2.2.1.12 review
Importance:	High
Description:	In fast mode, one cycle of the job processing function of the EEPROM driver shall limit the block size that is read from EEPROM to the configured maximum block size. Simplified comment: Read a big block of data during one call of the job processing function.
Rationale:	Allow fast reading and checking of EEPROM during ECU start-up
Use Case:	Example: In normal EEPROM mode, the maximum block size of read data is 16. In fast EEPROM mode, the maximum block size of read data is 128.
Dependencies:	[BSW096] EEPROM driver static configuration, maximum read block sizes [BSW12156] EEPROM mode selection function
Conflicts:	--
Supporting Material:	--

4.1.3 Non-Functional Requirements (Qualities)

4.1.3.1 [BSW12050] EEPROM job processing execution time

Initiator:	BMW
Date:	10.01.2004
Short Description:	EEPROM job processing execution time
Type:	New
Importance:	High
Description:	The job processing function of the EEPROM driver shall process only as much data as the EEPROM hardware can handle in one step (particularly write operation) or as much as a defined user limit (particularly read operation).
Rationale:	Minimize processor load, reduce blocking times.
Use Case:	The job processing function performs the writing of one byte and the reading of max. 8 bytes during one call.
Dependencies:	[BSW12157] Job processing – normal mode
Conflicts:	--
Supporting Material:	BMW Specification MCAL V1.0a

4.2 External EEPROM Driver

4.2.1 Functional Overview

The external EEPROM driver provides services for initialization and reading, writing, erasing to/from external EEPROM.

4.2.2 Functional Requirements

4.2.2.1 General

4.2.2.1.1 [BSW12051] Functional scope

Initiator:	BMW
Date:	13.05.2004
Short Description:	Functional scope
Type:	New
Importance:	High
Description:	For an external EEPROM driver the same requirements shall apply like for an internal EEPROM driver.
Rationale:	Make no functional differences between internal and external EEPROM. Keep the functional scope the same.
Use Case:	The STAR12 has internal EEPROM. The NEC V850 needs external EEPROM. On both microcontrollers the same NVRAM Manager shall be used.
Dependencies:	Requirements on internal EEPROM driver
Conflicts:	--
Supporting Material:	--

4.2.2.2 Configuration

4.2.2.2.1 [BSW12164] SPI channel configuration

Initiator:	WP4.2.2.1.12
Date:	07.07.2004
Short Description:	SPI channel configuration
Type:	Changed (generalized)
Importance:	High
Description:	A driver for an external SPI EEPROM shall allow the static configuration of the required SPI parameters. Those parameters are specified by the SPI Handler specification.
Rationale:	Basic configuration of SPI access
Use Case:	Use the SPI EEPROM driver together with other SPI device drivers on the same SPI bus.
Dependencies:	[BSW096] EEPROM driver static configuration.
Conflicts:	--
Supporting Material:	AUTOSAR SWS SPI Handler

4.2.2.3 Normal Operation

4.2.2.3.1 [BSW12124] SPI access modes

Initiator:	SVDO
Date:	05.07.2004
Short Description:	SPI access modes
Type:	New
Importance:	High
Description:	<p>The EEPROM driver for an external SPI EEPROM device shall access the SPI depending on the current EEPROM mode:</p> <ul style="list-style-type: none"> • normal EEPROM mode: SPI channel with single byte/word mode • fast EEPROM mode: SPI channel with burst mode
Rationale:	Fast read operation during ECU start-up, non-blocking SPI usage during normal operation.
Use Case:	<p>External SPI EEPROM, SPI baud rate 500kbaud.</p> <p>During ECU start-up the EEPROM is operated in burst mode in order to reduce start-up time. An EEPROM access (32bytes+header) blocks the SPI bus for approx. 560µs.</p> <p>During normal operation, the EEPROM is operated in normal mode so that the SPI bus access of the EEPROM driver does not block a communication request with higher priority.</p>
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.2.3 Non-Functional Requirements (Qualities)

4.2.3.1 [BSW12052] Common EEPROM API

Initiator:	BMW
Date:	13.05.2004
Short Description:	Common EEPROM API
Type:	New
Importance:	High
Description:	The external EEPROM driver shall have a semantically identical API as an internal EEPROM driver.
Rationale:	Ease Memory Abstraction. Keep handling of internal and external EEPROM similar.
Use Case:	The STAR12 has internal EEPROM. The NEC V850 needs external EEPROM. On both microcontrollers the same NVRAM Manager shall be used.
Dependencies:	Requirements on internal EEPROM driver
Conflicts:	--
Supporting Material:	--

4.2.3.2 [BSW12053] Microcontroller independency

Initiator:	BMW
Date:	13.05.2004
Short Description:	Microcontroller independency
Type:	New
Importance:	High
Description:	The source code of the external EEPROM driver shall be independent from the underlying microcontroller.
Rationale:	Reuse of external EEPROM driver across multiple microcontrollers
Use Case:	The same external EEPROM driver for an SPI EEPROM device can be used on a NEC V850 and on a MPC563 without any modification using the standardized SPI Handler interface.
Dependencies:	--
Conflicts:	--
Supporting Material:	--

5 References

5.1 Deliverables of AUTOSAR

- [1] List of Basic Software Modules
https://svn2.autosar.org/repos2/22_Releases
AUTOSAR_BasicSoftwareModules.pdf
- [2] Layered Software Architecture
https://svn2.autosar.org/repos2/22_Releases
AUTOSAR_LayeredSoftwareArchitecture.pdf
- [3] General Requirements on Basic Software Modules
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] HIS API IO Driver
<http://www.automotive-his.de/results.htm>
API_IODriver_2_1_1.pdf