

Document Title	Requirements on DIO Driver
Document Owner	AUTOSAR GbR
Document Responsibility	AUTOSAR GbR
Document Identification No	191
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
31.10.2007	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
14.03.2006	2.0.0	AUTOSAR Administration	Release as a separate document. The SRS SPAL V1.0.0 has been split into 15 independent documents for Release 2.0
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	Requirement structure	7
3	Acronyms and abbreviations	8
4	Requirement Specification	9
4.1	DIO Driver	9
4.1.1	Functional Overview	9
4.1.2	Functional Requirements	10
4.1.2.1	Configuration and Initialization	10
4.1.2.1.1	[BSW12355] Configuration of symbolic names	10
4.1.2.2	Normal Operation	10
4.1.2.2.1	[BSW12003] DIO port write service	10
4.1.2.2.2	[BSW12004] DIO channel group write service	11
4.1.2.2.3	[BSW12005] DIO channel write service	11
4.1.2.2.4	[BSW12006] DIO port read service	11
4.1.2.2.5	[BSW12007] DIO channel group read service	12
4.1.2.2.6	[BSW12008] DIO channel read service	12
4.1.2.2.7	[BSW12352] General read/write behavior	12
4.1.3	Non-Functional Requirements (Qualities)	13
4.1.3.1	[BSW12424] Provide atomicity of DIO access	13
5	References	14
5.1	Deliverables of AUTOSAR	14

1 Scope of this document

This document specifies requirements on the module DIO Driver.

Constraints

First scope for specification of requirements on basic software modules are 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 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)
- 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 being 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 DIO Driver

4.1.1 Functional Overview

The DIO driver provides port and channel based read and write access to the internal general purpose I/O ports. The read and write behavior is unbuffered. The basic behavior of this driver is synchronous.

The following expressions are used within the DIO driver:

<i>Expression</i>	<i>Explanation</i>
DIO channel:	Represents a single general-purpose digital input/output pin
DIO port:	Represents multiple DIO channels that are grouped by hardware and accessible synchronously (typically controlled by one hardware register). Example: Port A (8 bit) of Freescale HC08
DIO channel group:	Represents multiple adjoining DIO channels represented by a logical group. A DIO channel group is a subset of one DIO port, accessible synchronously. Example: Port pins 2..6 of an 8 bit port

4.1.2 Functional Requirements

4.1.2.1 Configuration and Initialization

The configuration and initialization of the port structure is not part of the DIO driver. This is done by the Port driver (see [BSW12001](#) Configuration of port pin properties).

4.1.2.1.1 [BSW12355] Configuration of symbolic names

Initiator:	WP4.2.2.1.12
Date:	13.10.2004
Short Description:	Configuration of symbolic names
Type:	New
Importance:	High
Description:	The DIO driver shall allow the static configuration of the following symbolic names: <ul style="list-style-type: none"> • DIO channel names • DIO channel group names • DIO port names
Rationale:	Provide human readable symbolic names for DIO channels
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.2.2 Normal Operation

4.1.2.2.1 [BSW12003] DIO port write service

Initiator:	BMW
Date:	03.05.2004
Short Description:	DIO port write service
Type:	New
Importance:	High
Description:	The DIO Driver shall provide a service that writes a data word to the assigned DIO port. The operation shall be unbuffered. There shall be no influence to the input functionality of the port .
Rationale:	Basic functionality
Use Case:	Write access to an entire DIO port.
Dependencies:	BSW12352 General read/write behavior
Conflicts:	--
Supporting Material:	BMW Specification MCAL V1.0a, REQ MAL10.3.0

4.1.2.2.2 [BSW12004] DIO channel group write service

Initiator:	BMW
Date:	03.05.2004
Short Description:	DIO channel group write service
Type:	New
Importance:	High
Description:	The DIO Driver shall provide a service that writes a selectable number of adjoining bits to an assigned part of a DIO port. The operation shall be unbuffered.
Rationale:	Allow for simultaneous setting of a group of DIO channels of a DIO port that has multiple external assignments.
Use Case:	Write access to DIO ports with multiple assignments.
Dependencies:	[BSW12352] General read/write behavior
Conflicts:	--
Supporting Material:	BMW Specification MCAL V1.0a, REQ MAL10.3.1

4.1.2.2.3 [BSW12005] DIO channel write service

Initiator:	BMW
Date:	03.05.2004
Short Description:	DIO channel write service
Type:	New
Importance:	High
Description:	The DIO Driver shall provide a service for write access to single DIO channels (specific port pins).
Rationale:	Efficient handling of single DIO channels.
Use Case:	Write access to a particular DIO channel (port pin).
Dependencies:	[BSW12352] General read/write behavior
Conflicts:	--
Supporting Material:	BMW Specification MCAL V1.0a, REQ MAL10.5.0

4.1.2.2.4 [BSW12006] DIO port read service

Initiator:	BMW
Date:	03.05.2004
Short Description:	DIO port read service
Type:	New
Importance:	High
Description:	The DIO Driver shall provide a service for reading a data word from the assigned DIO port. The operation shall be unbuffered. There shall be no influence to the output functionality of the port.
Rationale:	Basic functionality
Use Case:	Read access to an entire DIO port.
Dependencies:	[BSW12352] General read/write behavior
Conflicts:	--
Supporting Material:	BMW Specification MCAL V1.0a, REQ MAL10.4.0

4.1.2.2.5 [BSW12007] DIO channel group read service

Initiator:	BMW
Date:	03.05.2004
Short Description:	DIO channel group read service
Type:	New
Importance:	High
Description:	The DIO Driver shall provide a service for reading a selectable number of adjoining bits from an assigned part of a DIO port. The operation shall be unbuffered.
Rationale:	Basic functionality
Use Case:	Read access to DIO ports with multiple assignments.
Dependencies:	[BSW12352] General read/write behavior
Conflicts:	--
Supporting Material:	BMW Specification MCAL V1.0a, REQ MAL10.4.1

4.1.2.2.6 [BSW12008] DIO channel read service

Initiator:	BMW
Date:	03.05.2004
Short Description:	DIO channel read service
Type:	New
Importance:	High
Description:	The DIO Driver shall provide a service for reading one bit of an assigned DIO channel (specific port pin). The operation shall be unbuffered.
Rationale:	Efficient handling of single DIO channels.
Use Case:	Read access to a particular DIO channel.
Dependencies:	[BSW12352] General read/write behavior
Conflicts:	--
Supporting Material:	BMW Specification MCAL V1.0a, REQ MAL10.6.0

4.1.2.2.7 [BSW12352] General read/write behavior

Initiator:	WP4.2.2.1.12
Date:	12.10.2004
Short Description:	General read/write behaviour
Type:	New
Importance:	High
Description:	The DIO driver shall allow reading from and writing to DIO ports, channel groups and channels regardless of the configuration of their direction. If a channel configured as input is written to, the value shall be written to the output register, but does not appear on the physical port pin. If a channel configured as output is read, the value of the real pin level is read, if supported by hardware. Otherwise, the value of the port output register is read.
Rationale:	Simplify implementation of all DIO read and write services. Allow readback of output pins. Allow presetting of output values before switching the port pin to output direction.
Use Case:	See rationale.

Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3 Non-Functional Requirements (Qualities)

4.1.3.1 [BSW12424] Provide atomicity of DIO access

Initiator:	WP4.2.2.1.12
Date:	07.12.2004
Short Description:	Provide atomicity of DIO access
Type:	New
Importance:	High
Description:	All re-entrant functions of the DIO Driver shall perform the following access actions in an atomic way: <ul style="list-style-type: none"> • DIO ports • DIO channels • DIO channel groups
Rationale:	Avoid data integrity problems within concurrent access of DIO Driver API functions.
Use Case:	A specific microcontroller (or a specific compiler) does not provide atomic access to single port pins. For that reason, the implementation has to use read-modify-write operations on the whole port. Concurrent access to pins of the same port will lead to data integrity problems if concurrent access is not blocked.
Dependencies:	--
Conflicts:	--
Supporting Material:	--

5 References

5.1 Deliverables of AUTOSAR

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

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

[SRS_BSW_GENERAL] General Requirements on Basic Software Modules,
https://svn2.autosar.org/repos2/22_Releases
AUTOSAR_SRS_General.pdf

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