Document Title	Requirements on DIO Driver
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	191

Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R21-11

Document Change History

Date	Release	Changed by	Change Description
2021-11-25	R21-11	AUTOSAR Release Management	• No content changes
2020-11-30	R20-11	AUTOSAR Release Management	• No content 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 references to BMW and HIS documents • Editorial changes
2016-11-30	4.3.0	AUTOSAR Release Management	• Add section 5 Requirements Tracing
2015-07-31	4.2.2	AUTOSAR Release Management	• Editorial changes
2014-10-31	4.2.1	AUTOSAR Release Management	• Editorial changes
2013-03-15	4.1.1	AUTOSAR Administration	• Requirement IDs changed • Updated for requirements traceability
2011-12-22	4.0.3	AUTOSAR Administration	• Added SRS_Dio_12900
2010-09-30	3.1.5	AUTOSAR Administration	• Legal disclaimer revised

Document Change History

Date	Release	Changed by	Change Description
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	• Legal disclaimer revised
2006-05-16	2.0	AUTOSAR Administration	• Release as a separate document. The SRS SPAL V1.0.0 has been split into 15 independent documents for Release 2.0
2005-05-31	1.0	AUTOSAR Administration	• Initial release as a part of the SRS SPAL V1.0.0

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 document

This document specifies requirements on the module DIO Driver.

1.1 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

- The representation of requirements in AUTOSAR documents follows the table specified in [TPS_STDT_00078].

- 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'
ICU	Interrupt Capture Unit
MAL	Old name of Microconroller Abstraction Layer (replaced by MCAL because 'MAL' is a french term meaning 'bad')
MCAL	Microconroller 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 Functional Overview

4.1 DIO Driver

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:

Expression	*Explanation*
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

5 Requirements Tracing

Requirement	Description	Satisfied by	
RS_BRF_01024	AUTOSAR shall provide naming rules for public symbols	SRS_Dio_12355	
RS_BRF_01864	AUTOSAR microcontroller abstraction shall provide mapping of I/O signals to digital I/O ports	SRS_Dio_12003, SRS_Dio_12005, SRS_Dio_12007, SRS_Dio_12352, SRS_Dio_12900	SRS_Dio_12004, SRS_Dio_12006, SRS_Dio_12008, SRS_Dio_12424,

6 Requirement Specification

6.1 Functional Requirements

6.1.1 DIO Driver

6.1.1.1 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 [SRS_Port_12001] Configuration of port pin properties).

6.1.1.1.1 [SRS_Dio_12355] Symbolic names shall be configured

[

Type:	Valid
Description:	The DIO driver shall allow the static configuration of the following symbolic names: • DIO channel names • DIO channel group names • DIO port names
Rationale:	Provide human readable symbolic names for DIO channels
Use Case:	--
Dependencies:	--
Supporting Material:	--

](RS_BRF_01024)

6.1.1.2 Normal Operation

6.1.1.2.1 [SRS_Dio_12003] The DIO Driver shall provide a service that writes a data word to the assigned DIO port

[

Type:	Valid
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:	[SRS_Dio_12352] General read/write behavior
Supporting Material:	--

](RS_BRF_01864)

6.1.1.2.2 [SRS_Dio_12004] The DIO Driver shall provide a service that writes a selectable number of adjoining bits to an assigned part of a DIO port

[

Type:	Valid
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:	[SRS_Dio_12352] General read/write behavior
Supporting Material:	--

⌋(RS_BRF_01864)

6.1.1.2.3 [SRS_Dio_12005] The DIO Driver shall provide a service for write access to single DIO channels

⌈

Type:	Valid
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:	[SRS_Dio_12352] General read/write behavior
Supporting Material:	--

⌋(RS_BRF_01864)

6.1.1.2.4 [SRS_Dio_12006] The DIO Driver shall provide a service for reading a data word from the assigned DIO port

⌈

Type:	Valid
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:	[SRS_Dio_12352] General read/write behavior
Supporting Material:	--

⌋(RS_BRF_01864)

6.1.1.2.5 [SRS_Dio_12007] The DIO Driver shall provide a service for reading a selectable number of adjoining bits from an assigned part of a DIO port

⌈

Type:	Valid
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:	[SRS_Dio_12352] General read/write behavior
Supporting Material:	--

⌋(RS_BRF_01864)

6.1.1.2.6 [SRS_Dio_12008] The DIO Driver shall provide a service for reading one bit of an assigned DIO channel

[

Type:	Valid
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:	[SRS_Dio_12352] General read/write behavior
Supporting Material:	--

](RS_BRF_01864)

6.1.1.2.7 [SRS_Dio_12900] The DIO Driver shall provide a service to flip

[

Type:	Valid
Description:	The DIO Driver shall provide a service to flip (change from 1 to 0 or from 0 to 1) one bit of an assigned DIO channel (specific port pin) and return the level of the channel after flip. The operation shall be unbuffered.
Rationale:	Efficient handling of single DIO channels.
Use Case:	Read & write access to a particular DIO channel to flip the level.
Dependencies:	[SRS_Dio_12352] General read/write behavior
Supporting Material:	--

](RS_BRF_01864)

6.1.1.2.8 [SRS_Dio_12352] The DIO driver shall allow reading from and writing to DIO ports, channel groups and channels

[

Type:	Valid
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:	--
Supporting Material:	--

](RS_BRF_01864)

6.2 Non-Functional Requirements

6.2.1 DIO Driver

6.2.1.1 [SRS_Dio_12424] Provide atomicity of DIO access

⌈

Type:	Valid
Description:	All re-entrant functions of the DIO Driver shall perform the following access actions in an atomic way: • 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:	--
Supporting Material:	--

⌋(RS_BRF_01864)

7 References

7.1 Deliverables of AUTOSAR

[DOC_LAYERED_ARCH] Layered Software Architecture,
AUTOSAR_EXP_LayeredSoftwareArchitecture.pdf

[AUTOSAR_GLOSSARY] Glossary,
AUTOSAR_TR_Glossary.pdf

[SRS_BSW_GENERAL] General Requirements on Basic Software Modules,
AUTOSAR_SRS_BSWGeneral.pdf

[SRS_BSW_SPAL] General Requirements on SPAL,
AUTOSAR_SRS_SPALGeneral.pdf

[TPS_STDT_0078] Software Standardization Template
AUTOSAR_TPS_StandardizationTemplate.pdf