

Document Title	Specification of DIO Driver
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	20

Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R23-11

Document Change History			
Date	Release	Changed by	Description
2023-11-23	R23-11	AUTOSAR Release Management	<ul style="list-style-type: none"> • Editorial changes
2022-11-24	R22-11	AUTOSAR Release Management	<ul style="list-style-type: none"> • Minor corrections / clarifications / editorial changes
2021-11-25	R21-11	AUTOSAR Release Management	<ul style="list-style-type: none"> • Cleaned return codes
2020-11-30	R20-11	AUTOSAR Release Management	<ul style="list-style-type: none"> • Cleaned error classification
2019-11-28	R19-11	AUTOSAR Release Management	<ul style="list-style-type: none"> • MCALMulticoreDistribution (CONC_639) • Changed Document Status from Final to published
2018-10-31	4.4.0	AUTOSAR Release Management	<ul style="list-style-type: none"> • Introduced MaskedWritePort API
2017-12-08	4.3.1	AUTOSAR Release Management	<ul style="list-style-type: none"> • Removed unused artifacts • Editorial changes





2016-11-30	4.3.0	AUTOSAR Release Management	<ul style="list-style-type: none"> • Removed [SWS_Dio_00065] • Replaced content of "7.6.2 Runtime Errors" by "There are no runtime errors." • Replaced content of "7.6.3 Transient Faults" by "There are no transient faults" • Removed the definition of the "configuration variants" from 10.1.1 • Changed Figure 2: Include File Structure
2015-07-31	4.2.2	AUTOSAR Release Management	<ul style="list-style-type: none"> • DET Renaming and Extension Incorporation • Changed DioChannelId, DioPortId precompile configuration
2014-10-31	4.2.1	AUTOSAR Release Management	<ul style="list-style-type: none"> • DIO: ReadChannelGroup / WriteChannelGroup pointer parameters. Provided support for Link time only. • The generation of link-time parameters aggregated by a postBuildChangeable container may not be possible. Reference to SWS_BSW_00380 is removed.
2013-10-31	4.1.2	AUTOSAR Release Management	<ul style="list-style-type: none"> • Formalization of Service Interfaces • Revised return values of Service Interfaces • Editorial changes • Removed chapter(s) on change documentation
2013-03-15	4.1.1	AUTOSAR Administration	<ul style="list-style-type: none"> • Updated chapter 10 for 'Scope' values • Changed 'MemMap.h' to 'Dio_MemMap.h' • Added Subchapter 3.x • Requirement IDs changed to new format
2011-12-22	4.0.3	AUTOSAR Administration	<ul style="list-style-type: none"> • Removed Dem.h from [SWS_Dio_00171] and added new requirement [SWS_Dio_00194]





2010-09-30	3.1.5	AUTOSAR Administration	<ul style="list-style-type: none"> • Added a new API "Dio_LevelType Dio_FlipChannel(Dio_ChannelType ChannelId)" to flip (change from 1 to 0 or from 0 to 1) the level of a channel and return the level of the channel after flip. • Removed requirement DIO174 and rephrased [SWS_Dio_00106]. • Added requirements [SWS_DIO_00188] and [SWS_Dio_00189], to report DET error DIO_E_PARAM_POINTER from Dio_GetVersionInfo.
2010-02-02	3.1.4	AUTOSAR Administration	<ul style="list-style-type: none"> • Clarification of [SWS_Dio_00014] • DioVersionInfoApi added to DIO071 • Clean up of configuration parameters and header file inclusion structure • Legal disclaimer revised
2008-08-13	3.1.1	AUTOSAR Administration	<ul style="list-style-type: none"> • Legal disclaimer revised
2007-12-21	3.0.1	AUTOSAR Administration	<ul style="list-style-type: none"> • Harmonized initialization with MCAL modules • Optional inclusion of the DEM header file • Added explanation on dependency between DIO_PORT_MASK and DIO_PORT_OFFSET • Document meta information extended • Small layout adaptations made



△

2007-01-24	2.1.15	AUTOSAR Administration	<ul style="list-style-type: none"> • File structure update • Removed BSW00324 • In the configuration where pre-compile and link time is possible the variant for pre-compile is now always "PC" and not "All variants". • Added Chapter 8.6 • Changes in referencing symbolic naming • Updated traceability matrix regarding SRS_BSW_00435 and SRS_BSW_00436 • Legal disclaimer revised • "Advice for users" revised • "Revision Information" added
2006-05-16	2.0	AUTOSAR Administration	<ul style="list-style-type: none"> • Major changes in chapter 10, Configuration specification • Structure of document changed partly • Readback support moved to PORT Driver
2005-05-31	1.0	AUTOSAR Administration	<ul style="list-style-type: none"> • 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.

Contents

1	Introduction and functional overview	8
2	Acronyms and Abbreviations	10
3	Related documentation	11
3.1	Input documents & related standards and norms	11
3.2	Related specification	11
4	Constraints and assumptions	12
4.1	Limitations	12
4.2	Applicability to car domains	12
5	Dependencies to other modules	13
5.1	Port Driver Module	13
5.2	Default Error Tracer	13
6	Requirements Tracing	14
7	Functional specification	16
7.1	General Behaviour	16
7.1.1	Background & Rationale	16
7.1.2	Requirements	16
7.2	Initialization	18
7.2.1	Background & Rationale	18
7.2.2	Requirements	18
7.3	Runtime Reconfiguration	18
7.3.1	Background & Rationale	18
7.3.2	Requirements	18
7.4	DIO Write Service	18
7.4.1	Background & Rationale	18
7.4.2	Requirements	18
7.4.2.1	DIO Channel Write Service	19
7.4.2.2	DIO Port Write Service	19
7.4.2.3	DIO Channel Group Write Service	19
7.4.2.4	DIO Masked Port Write Service	19
7.5	DIO Read Service	20
7.5.1	Background & Rationale	20
7.5.2	Requirements	20
7.5.2.1	DIO Channel Read Service	20
7.5.2.2	DIO Port Read Service	20
7.5.2.3	DIO Channel Group Read Service	20
7.5.2.4	DIO Readback of Output Pins	20
7.6	Error Classification	21
7.6.1	Development Errors	21
7.6.2	Runtime Errors	21

7.6.3	Transient Faults	21
7.6.4	Production Errors	21
7.6.5	Extended Production Errors	21
7.7	Security Events	22
8	API specification	23
8.1	Imported types	23
8.2	Type definitions	23
8.2.1	Dio_ChannelType	23
8.2.2	Dio_PortType	24
8.2.3	Dio_ChannelGroupType	24
8.2.4	Dio_LevelType	25
8.2.5	Dio_PortLevelType	25
8.3	Function definitions	25
8.3.1	Dio_ReadChannel	26
8.3.2	Dio_WriteChannel	26
8.3.3	Dio_ReadPort	27
8.3.4	Dio_WritePort	28
8.3.5	Dio_ReadChannelGroup	29
8.3.6	Dio_WriteChannelGroup	29
8.3.7	Dio_GetVersionInfo	30
8.3.8	Dio_FlipChannel	31
8.3.9	Dio_MaskedWritePort	32
8.4	Callback notifications	32
8.5	Scheduled functions	32
8.6	Expected interfaces	33
8.6.1	Mandatory Interfaces	33
8.6.2	Optional Interfaces	33
9	Sequence diagrams	34
9.1	Read a value from a digital I/O	34
9.2	Write a value to a digital I/O	34
10	Configuration specification	35
10.1	Containers and configuration parameters	35
10.1.1	Variants	35
10.1.2	Dio	36
10.1.3	DioGeneral	36
10.1.4	DioPort	38
10.1.5	DioChannel	39
10.1.6	DioChannelGroup	41
10.1.7	DioConfig	43
10.2	Published Information	43
A	Not applicable requirements	44

1 Introduction and functional overview

This specification specifies the functionality, API and the configuration of the AUTOSAR Basic Software module DIO Driver.

This specification is applicable to drivers only for on chip DIO pins and ports.

The DIO Driver provides services for reading and writing to / from

- DIO Channels (Pins)
- DIO Ports
- DIO Channel Groups

The behaviour of those services is synchronous.

This module works on pins and ports which are configured by the PORT driver for this purpose. For this reason, there is no configuration and initialization of this port structure in the DIO Driver.

The diagram below identifies the DIO Driver functions, and the structure of the PORT Driver and DIO Driver within the MCAL software layer.

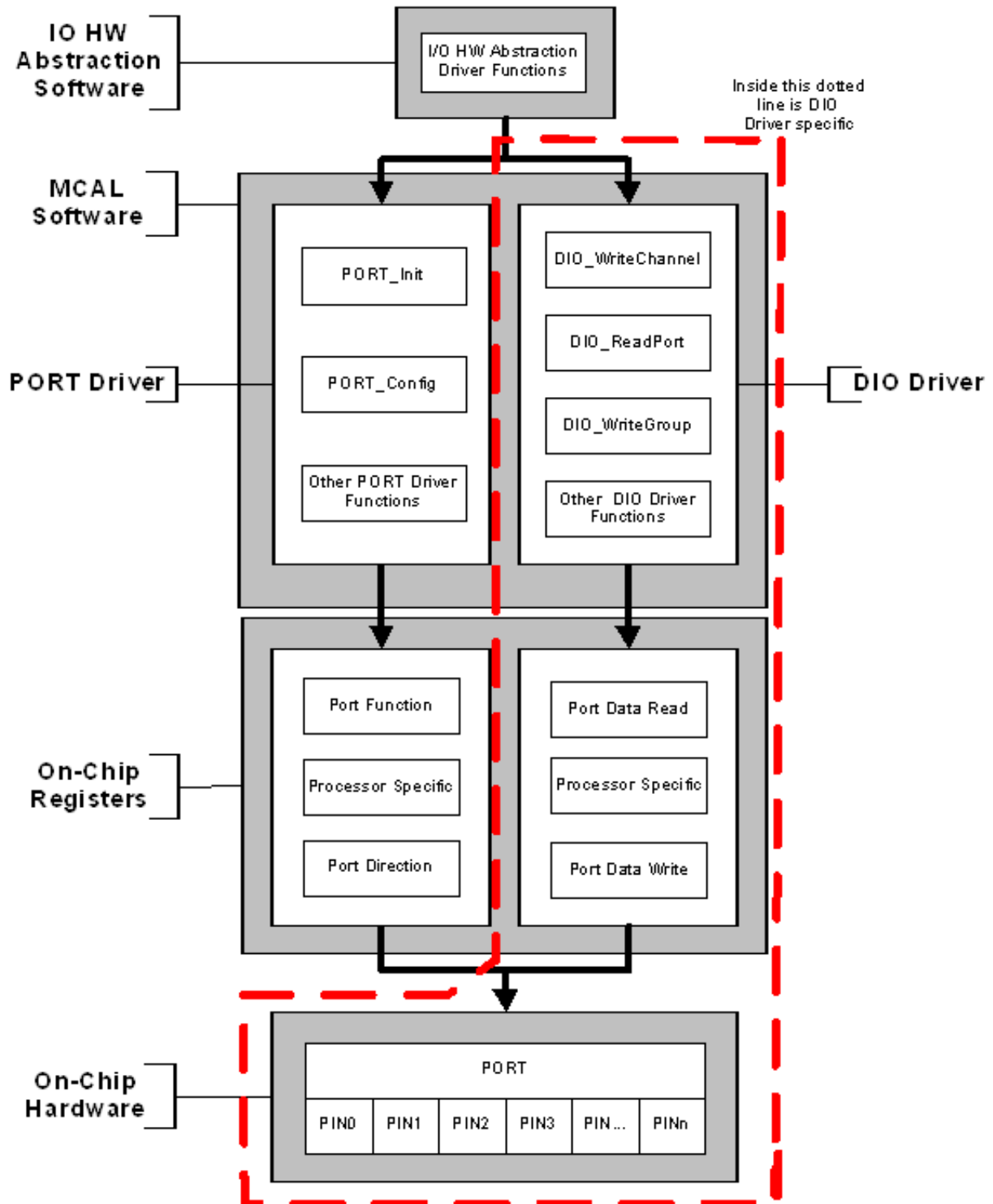


Figure 1.1: DIO Driver Structure and Integration

2 Acronyms and Abbreviations

The glossary below includes acronyms and abbreviations relevant to the DIO Driver module that are not included in the [1, AUTOSAR glossary].

Abbreviation / Acronym	Description
DIO channel	Represents a single general-purpose digital input / output pin
DIO port	Represents several DIO channels that are grouped by hardware (typically controlled by one hardware register). Example: Port A (8 bit) of Freescale HC08
DIO channel group	Represents several adjoining DIO channels represented by a logical group. A DIO channel group shall belong to one DIO port. Example: Port pins 2..6 of an 8 bit port addressing a multiplexer
Physical Level (Input)	Two states possible: LOW / HIGH. A bit value '0' represents a LOW, a bit value '1' represents a HIGH.
Physical Level (Output)	Two states possible: LOW / HIGH. A bit value '0' represents a LOW, a bit value '1' represents a HIGH.
LSB	Least Significant Bit
MSB	Most Significant Bit
DIO	Digital Input Output
ID	Identifier
ADC	Analog to Digital Converter
SPI	Serial Peripheral Interface
PWM	Pulse Width Modulation
ICU	Input Capture Unit
DET	Default Error Tracer
DEM	Diagnostic Event Manager

Table 2.1: Acronyms and abbreviations used in the scope of this Document

3 Related documentation

3.1 Input documents & related standards and norms

- [1] Glossary
AUTOSAR_FO_TR_Glossary
- [2] General Specification of Basic Software Modules
AUTOSAR_CP_SWS_BSWGeneral

3.2 Related specification

AUTOSAR provides a General Specification on Basic Software modules [2, SWS BSW General], which is also valid for DIO Driver.

Thus, the specification SWS BSW General shall be considered as additional and required specification for DIO Driver.

4 Constraints and assumptions

4.1 Limitations

No limitations.

4.2 Applicability to car domains

No restrictions.

5 Dependencies to other modules

5.1 Port Driver Module

Many ports and port pins are assigned by the PORT Driver Module to various functionalities as for example:

- General purpose I/O
- ADC
- SPI
- PWM

[SWS_Dio_00061] [The Dio module shall not provide APIs for overall configuration and initialization of the port structure which is used in the Dio module. These actions are done by the PORT Driver Module.]()

[SWS_Dio_00063] [The Dio module shall adapt its configuration and usage to the microcontroller and ECU.]()

[SWS_Dio_00102] [The Dio module's user shall only use the Dio functions after the Port Driver has been initialized. Otherwise the Dio module will exhibit undefined behavior.]()

5.2 Default Error Tracer

[SWS_Dio_00194] [Dio.c shall include Det.h if detection of development error (DET) is enabled.]([SRS_BSW_00350](#))

6 Requirements Tracing

This chapter refers to input requirements specified in the SRS documents (Software Requirements Specifications) that are applicable for this software module.

The table below lists links to specification items of the DIO driver SWS document, that satisfy the input requirements. Only functional requirements are referenced.

Requirement	Description	Satisfied by
[SRS_BSW_00323]	All AUTOSAR Basic Software Modules shall check passed API parameters for validity	[SWS_Dio_00074] [SWS_Dio_00075] [SWS_Dio_00114] [SWS_Dio_00189]
[SRS_BSW_00344]	BSW Modules shall support link-time configuration	[SWS_Dio_00001] [SWS_Dio_00002]
[SRS_BSW_00350]	All AUTOSAR Basic Software Modules shall allow the enabling/disabling of detection and reporting of development errors.	[SWS_Dio_00194]
[SRS_BSW_00396]	The Basic Software Module specifications shall specify the supported configuration classes for changing values and multiplicities for each parameter/container	[SWS_Dio_00129]
[SRS_BSW_00411]	All AUTOSAR Basic Software Modules shall apply a naming rule for enabling/disabling the existence of the API	[SWS_Dio_00139]
[SRS_Dio_12003]	The DIO Driver shall provide a service that writes a data word to the assigned DIO port	[SWS_Dio_00004] [SWS_Dio_00007] [SWS_Dio_00034] [SWS_Dio_00035] [SWS_Dio_00051] [SWS_Dio_00053] [SWS_Dio_00089] [SWS_Dio_00105] [SWS_Dio_00108] [SWS_Dio_00200] [SWS_Dio_00201] [SWS_Dio_00202] [SWS_Dio_00203] [SWS_Dio_00204]
[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	[SWS_Dio_00008] [SWS_Dio_00021] [SWS_Dio_00039] [SWS_Dio_00040] [SWS_Dio_00051] [SWS_Dio_00056] [SWS_Dio_00089] [SWS_Dio_00090] [SWS_Dio_00091]
[SRS_Dio_12005]	The DIO Driver shall provide a service for write access to single DIO channels	[SWS_Dio_00006] [SWS_Dio_00028] [SWS_Dio_00029] [SWS_Dio_00051] [SWS_Dio_00079] [SWS_Dio_00089] [SWS_Dio_00127] [SWS_Dio_00128]
[SRS_Dio_12006]	The DIO Driver shall provide a service for reading a data word from the assigned DIO port	[SWS_Dio_00013] [SWS_Dio_00031] [SWS_Dio_00051] [SWS_Dio_00053] [SWS_Dio_00089] [SWS_Dio_00104]
[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	[SWS_Dio_00014] [SWS_Dio_00021] [SWS_Dio_00037] [SWS_Dio_00051] [SWS_Dio_00056] [SWS_Dio_00089] [SWS_Dio_00092] [SWS_Dio_00093]
[SRS_Dio_12008]	The DIO Driver shall provide a service for reading one bit of an assigned DIO channel	[SWS_Dio_00011] [SWS_Dio_00027] [SWS_Dio_00051] [SWS_Dio_00089] [SWS_Dio_00127] [SWS_Dio_00128]
[SRS_Dio_12352]	The DIO driver shall allow reading from and writing to DIO ports, channel groups and channels	[SWS_Dio_00012] [SWS_Dio_00064] [SWS_Dio_00070] [SWS_Dio_00083] [SWS_Dio_00084] [SWS_Dio_00109]
[SRS_Dio_12355]	Symbolic names shall be configured	[SWS_Dio_00017] [SWS_Dio_00020] [SWS_Dio_00022] [SWS_Dio_00026]





Requirement	Description	Satisfied by
[SRS_Dio_12424]	Provide atomicity of DIO access	[SWS_Dio_00005] [SWS_Dio_00060]
[SRS_Dio_12900]	The DIO Driver shall provide a service to flip	[SWS_Dio_00191] [SWS_Dio_00192] [SWS_Dio_00193]
[SRS_SPAL_12064]	All driver modules shall raise an error if the change of the operation mode leads to degradation of running operations	[SWS_Dio_00001] [SWS_Dio_00002]
[SRS_SPAL_12263]	The implementation of all driver modules shall allow the configuration of specific module parameter types at link time	[SWS_Dio_00017] [SWS_Dio_00020] [SWS_Dio_00022]
[SRS_SPAL_12448]	All driver modules shall have a specific behavior after a development error detection	[SWS_Dio_00074] [SWS_Dio_00075] [SWS_Dio_00114] [SWS_Dio_00119]
[SRS_SPAL_12461]	Specific rules regarding initialization of controller registers shall apply to all driver implementations	[SWS_Dio_00001] [SWS_Dio_00002]
[SRS_SPAL_12462]	The register initialization settings shall be published	[SWS_Dio_00001] [SWS_Dio_00002]
[SRS_SPAL_12463]	The register initialization settings shall be combined and forwarded	[SWS_Dio_00001] [SWS_Dio_00002]

Table 6.1: RequirementsTracing

7 Functional specification

7.1 General Behaviour

7.1.1 Background & Rationale

The DIO Driver abstracts the access to the microcontroller's hardware pins. Furthermore, it allows the grouping of those pins.

7.1.2 Requirements

The Dio SWS shall define functions allowing

- Port-
- Channel-
- Channel-group -

-based read and write access to the internal general purpose I/O ports.

[SWS_Dio_00051] [The Dio module shall not buffer data when providing read and write services.]

The Dio SWS shall define synchronous read / write services.] ([SRS_Dio_12003](#), [SRS_Dio_12004](#), [SRS_Dio_12005](#), [SRS_Dio_12006](#), [SRS_Dio_12007](#), [SRS_Dio_12008](#))

[SWS_Dio_00005] [The Dio module's read and write services shall ensure for all services, that the data is consistent (Interruptible read-modify-write sequences are not allowed).] ([SRS_Dio_12424](#))

[SWS_Dio_00089] [Values used by the DIO Driver for the software level of Channels are either `STD_HIGH` or `STD_LOW`.] ([SRS_Dio_12003](#), [SRS_Dio_12004](#), [SRS_Dio_12005](#), [SRS_Dio_12006](#), [SRS_Dio_12007](#), [SRS_Dio_12008](#))

[SWS_Dio_00128] [A general-purpose digital IO pin represents a DIO channel.] ([SRS_Dio_12005](#), [SRS_Dio_12008](#))

[SWS_Dio_00127] [The Port module shall configure a DIO channel as input or output ([[SWS_Dio_00001](#)] and [[SWS_Dio_00002](#)]).] ([SRS_Dio_12005](#), [SRS_Dio_12008](#))

[SWS_Dio_00053] [In the DIO Driver, it shall be possible to group several DIO channels by hardware (typically controlled by one hardware register) to represent a DIO port.] ([SRS_Dio_12003](#), [SRS_Dio_12006](#))

Note: The single DIO channel levels inside a DIO port represent a bit in the DIO port value, depending on their position inside the port.

[SWS_Dio_00056] [A channel group is a formal logical combination of several adjoining DIO channels within a DIO port.] ([SRS_Dio_12004](#), [SRS_Dio_12007](#))

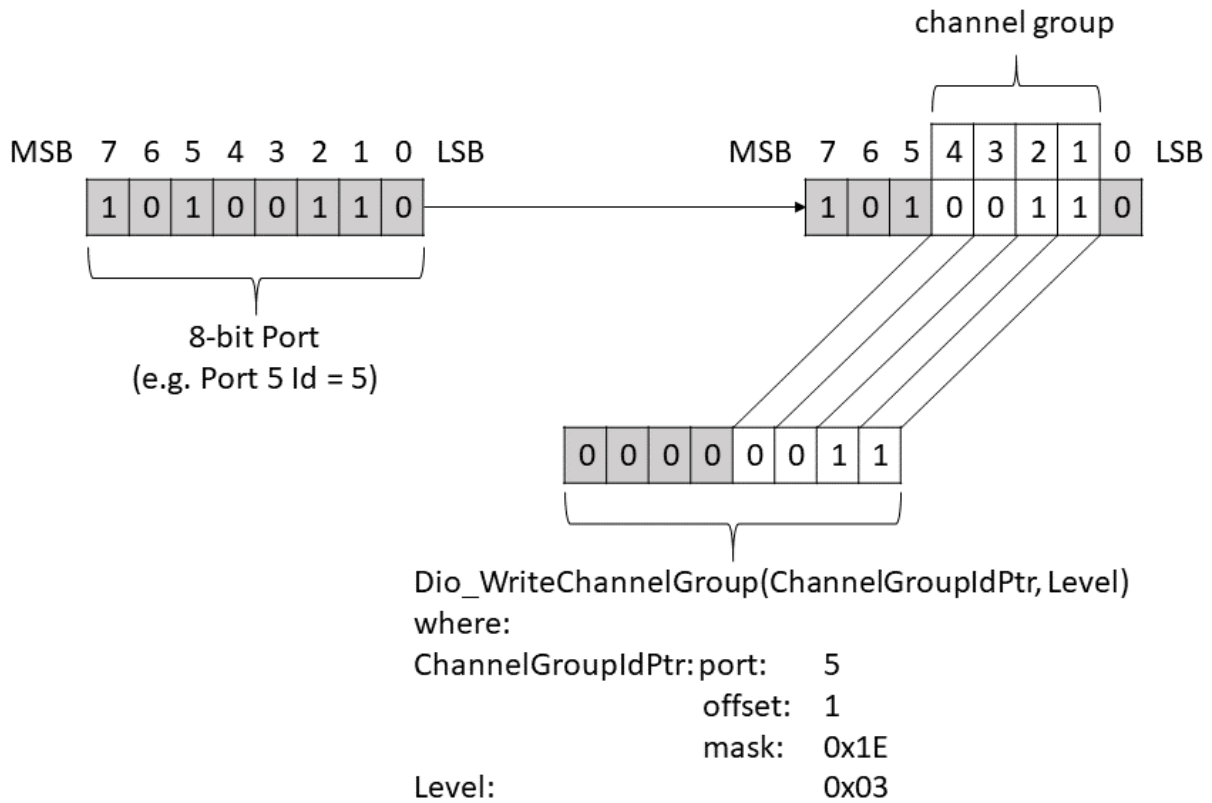


Figure 7.1: Schematic description of a ChannelGroup

The DIO Driver provides the following services:

- The Dio SWS shall define functions to modify the levels of output channels individually, for a port or for a channel group.
- The Dio SWS shall define functions to read the level of input and output (see [SWS_Dio_00083]) channels individually, for a port or for a channel group.

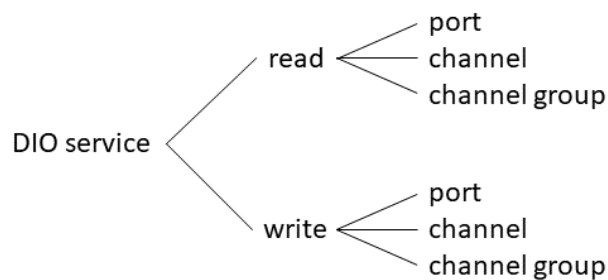


Figure 7.2: DIO Services

[SWS_Dio_00060] [All read and write functions of the Dio module shall be re-entrant.

Reason: The DIO Driver may be accessed by different upper layer handlers or drivers. These upper layer modules may access the driver concurrently.](SRS_Dio_12424)

[SWS_Dio_00026] [The configuration process for Dio module shall provide symbolic names for each configured DIO channel, port and group.](SRS_Dio_12355)

7.2 Initialization

7.2.1 Background & Rationale

Initialization of the hardware is done by the PORT Driver.

7.2.2 Requirements

[SWS_Dio_00001] [The Dio module shall not provide an interface for initialization of the hardware. The Port Driver performs this.] ([SRS_BSW_00344](#), [SRS_SPAL_12064](#), [SRS_SPAL_12461](#), [SRS_SPAL_12462](#), [SRS_SPAL_12463](#))

7.3 Runtime Reconfiguration

7.3.1 Background & Rationale

Runtime reconfiguration is provided by the PORT Driver.

7.3.2 Requirements

[SWS_Dio_00002] [The PORT driver shall provide the reconfiguration of the port pin direction during runtime.] ([SRS_BSW_00344](#), [SRS_SPAL_12064](#), [SRS_SPAL_12461](#), [SRS_SPAL_12462](#), [SRS_SPAL_12463](#))

7.4 DIO Write Service

7.4.1 Background & Rationale

The DIO Driver provides services to transfer data to the microcontroller's pins.

7.4.2 Requirements

[SWS_Dio_00064] [The Dio module's write functions shall work on input and output channels.] ([SRS_Dio_12352](#))

[SWS_Dio_00070] [If a Dio write function is used on an input channel, it shall have no effect on the physical output level.] ([SRS_Dio_12352](#))

[SWS_Dio_00109] [If supported by hardware, the Dio module shall set / clear the output data latch of an input channel so that the required level is output from the pin when the port driver configures the pin as a DIO output pin.] ([SRS_Dio_12352](#))

[SWS_Dio_00119] [If development errors are enabled and an error occurred, the Dio module's write functions shall NOT process the write command.] ([SRS_SPAL_12448](#))

7.4.2.1 DIO Channel Write Service

[SWS_Dio_00006] [The `Dio_WriteChannel` function shall set the level of a single DIO channel to `STD_HIGH` or `STD_LOW`.] ([SRS_Dio_12005](#))

7.4.2.2 DIO Port Write Service

[SWS_Dio_00007] [The `Dio_WritePort` function shall simultaneously set the levels of all output channels. A bit value '0' sets the corresponding channel to physical `STD_LOW`, a bit value '1' sets the corresponding channel to physical `STD_HIGH`.] ([SRS_Dio_12003](#))

[SWS_Dio_00004] [The `Dio_WritePort` function shall ensure that the functionality of the input channels of that port is not affected.] ([SRS_Dio_12003](#))

7.4.2.3 DIO Channel Group Write Service

[SWS_Dio_00008] [The `Dio_WriteChannelGroup` function shall simultaneously set an adjoining subset of DIO channels (channel group). A bit value '0' sets the corresponding channel to physical `STD_LOW`, a bit value '1' sets the corresponding channel to physical `STD_HIGH`.] ([SRS_Dio_12004](#))

7.4.2.4 DIO Masked Port Write Service

[SWS_Dio_00200] [The `Dio_MaskedWritePort` function shall simultaneously set the levels of the selected output channels.

Mask argument specifies which bits are selected: a bit value '0' means the corresponding channel is not selected, a bit value '1' means the corresponding channel is selected.

Level argument specifies the physical levels: a bit value '0' sets the corresponding channel to physical `STD_LOW`, a bit value '1' sets the corresponding channel to physical `STD_HIGH`.] ([SRS_Dio_12003](#))

[SWS_Dio_00201] [The `Dio_MaskedWritePort` function shall ensure that the functionality of the input channels of that port is not affected.] ([SRS_Dio_12003](#))

7.5 DIO Read Service

7.5.1 Background & Rationale

The DIO Driver provides services to transfer data from the microcontroller's pins.

7.5.2 Requirements

[SWS_Dio_00012] [The Dio module's read functions shall work on input and output channels.] ([SRS_Dio_12352](#))

7.5.2.1 DIO Channel Read Service

[SWS_Dio_00011] [The [Dio_ReadChannel](#) function shall read the level of a single DIO channel.] ([SRS_Dio_12008](#))

7.5.2.2 DIO Port Read Service

[SWS_Dio_00013] [The [Dio_ReadPort](#) function shall read the levels of all channels of one port. A bit value '0' indicates that the corresponding channel is physical `STD_LOW`, a bit value '1' indicates that the corresponding channel is physical `STD_HIGH`.] ([SRS_Dio_12006](#))

7.5.2.3 DIO Channel Group Read Service

[SWS_Dio_00014] [The [Dio_ReadChannelGroup](#) function shall read the levels of a DIO channel group. A bit value '0' indicates that the corresponding channel is physical `STD_LOW`, a bit value '1' indicates that the corresponding channel is physical `STD_HIGH`.] ([SRS_Dio_12007](#))

7.5.2.4 DIO Readback of Output Pins

[SWS_Dio_00083] [If the microcontroller supports the direct read-back of a pin value, the Dio module's read functions shall provide the real pin level, when they are used on a channel which is configured as an output channel.] ([SRS_Dio_12352](#))

[SWS_Dio_00084] [If the microcontroller does not support the direct read-back of a pin value, the Dio module's read functions shall provide the value of the output register, when they are used on a channel which is configured as an output channel.] ([SRS_Dio_12352](#))

7.6 Error Classification

Section "Error Handling" of the document [2] "General Specification of Basic Software Modules" describes the error handling of the Basic Software in detail. Above all, it constitutes a classification scheme consisting of five error types which may occur in BSW modules.

Based on this foundation, the following section specifies particular errors arranged in the respective subsections below.

7.6.1 Development Errors

[SWS_Dio_91001] Definiton of development errors in module Dio [

<i>Type of error</i>	<i>Related error code</i>	<i>Error value</i>
Invalid channel requested	DIO_E_PARAM_INVALID_CHANNEL_ID	0x0A
Invalid port requested	DIO_E_PARAM_INVALID_PORT_ID	0x14
Invalid channel group requested	DIO_E_PARAM_INVALID_GROUP	0x1F
API service called with a NULL pointer	DIO_E_PARAM_POINTER	0x20

]()

7.6.2 Runtime Errors

There are no runtime errors.

7.6.3 Transient Faults

There are no transient faults.

7.6.4 Production Errors

There are no production errors.

7.6.5 Extended Production Errors

There are no extended production errors.

7.7 Security Events

The module does not report security events.

8 API specification

8.1 Imported types

In this chapter all types included from the following modules are listed:

[SWS_Dio_00131] Definition of imported datatypes of module Dio [

Module	Header File	Imported Type
Std	Std_Types.h	Std_ReturnType
	Std_Types.h	Std_VersionInfoType

]()

8.2 Type definitions

[SWS_Dio_00103] [The port width within the types defined for the DIO Driver shall be the size of the largest port on the MCU which may be accessed by the DIO Driver.]()

8.2.1 Dio_ChannelType

[SWS_Dio_00182] Definition of datatype Dio_ChannelType [

Name	Dio_ChannelType		
Kind	Type		
Derived from	uint		
Range	This is implementation specific but not all values may be valid within the type.	–	Shall cover all available DIO channels
Description	Numeric ID of a DIO channel.		
Available via	Dio.h		

]()

[SWS_Dio_00015] [Parameters of type [Dio_ChannelType](#) contain the numeric ID of a DIO channel.]()

[SWS_Dio_00180] [The mapping of the ID is implementation specific but not configurable.]()

[SWS_Dio_00017] [For parameter values of type [Dio_ChannelType](#), the Dio's user shall use the symbolic names provided by the configuration description.

Furthermore, [SWS_Dio_00103] applies to the type [Dio_ChannelType](#).]([SRS_SPAL_12263](#), [SRS_Dio_12355](#))

8.2.2 Dio_PortType

[SWS_Dio_00183] Definition of datatype Dio_PortType [

Name	Dio_PortType		
Kind	Type		
Derived from	uint		
Range	0..<number of ports>	–	Shall cover all available DIO Ports.
Description	Numeric ID of a DIO port.		
Available via	Dio.h		

]()

[SWS_Dio_00018] [Parameters of type [Dio_PortType](#) contain the numeric ID of a DIO port.]()

[SWS_Dio_00181] [The mapping of ID is implementation specific but not configurable.]()

[SWS_Dio_00020] [For parameter values of type [Dio_PortType](#), the user shall use the symbolic names provided by the configuration description.

Furthermore, [SWS_Dio_00103] applies to the type [Dio_PortType](#).] ([SRS_SPAL_-12263](#), [SRS_Dio_12355](#))

8.2.3 Dio_ChannelGroupType

[SWS_Dio_00184] Definition of datatype Dio_ChannelGroupType [

Name	Dio_ChannelGroupType		
Kind	Structure		
Elements	mask		
	Type	uint8/16/32	
	Comment	This element mask which defines the positions of the channel group.	
	offset		
	Type	uint8	
	Comment	This element shall be the position of the Channel Group on the port, counted from the LSB.	
	port		
	Type	Dio_PortType	
Comment	This shall be the port on which the Channel group is defined.		
Description	Type for the definition of a channel group, which consists of several adjoining channels within a port.		
Available via	Dio.h		

]()

[SWS_Dio_00021] [[Dio_ChannelGroupType](#) is the type for the definition of a channel group, which consists of several adjoining channels within a port.] ([SRS_Dio_-12004](#), [SRS_Dio_12007](#))

[SWS_Dio_00022] [For parameter values of type [Dio_ChannelGroupType](#), the user shall use the symbolic names provided by the configuration description.

Furthermore, [SWS_Dio_00056] applies to the type [Dio_ChannelGroupType](#).] ([SRS_SPAL_12263](#), [SRS_Dio_12355](#))

8.2.4 Dio_LevelType

[SWS_Dio_00185] Definition of datatype [Dio_LevelType](#) [

Name	Dio_LevelType		
Kind	Type		
Derived from	uint8		
Range	STD_LOW	0x00	Physical state 0V
	STD_HIGH	0x01	Physical state 5V or 3.3V
Description	These are the possible levels a DIO channel can have (input or output)		
Available via	Dio.h		

]()

[SWS_Dio_00023] [[Dio_LevelType](#) is the type for the possible levels that a DIO channel can have (input or output).]()

8.2.5 Dio_PortLevelType

[SWS_Dio_00186] Definition of datatype [Dio_PortLevelType](#) [

Name	Dio_PortLevelType		
Kind	Type		
Derived from	uint		
Range	0...xxx	–	If the μ C owns ports of different port widths (e.g. 4, 8,16...Bit) Dio_PortLevelType inherits the size of the largest port
	If the μ C owns ports of different port widths (e.g. 4, 8,16...Bit) Dio_PortLevelType inherits the size of the largest port.		
Description	If the μ C owns ports of different port widths (e.g. 4, 8,16...Bit) Dio_PortLevelType inherits the size of the largest port.		
Available via	Dio.h		

]()

[SWS_Dio_00024] [[Dio_PortLevelType](#) is the type for the value of a DIO port.

Furthermore, [SWS_Dio_00103] applies to the type [Dio_PortLevelType](#).]()

8.3 Function definitions

This is a list of functions provided for upper layer modules.

8.3.1 Dio_ReadChannel

[SWS_Dio_00133] Definition of API function Dio_ReadChannel [

Service Name	Dio_ReadChannel	
Syntax	<pre>Dio_LevelType Dio_ReadChannel (Dio_ChannelType ChannelId)</pre>	
Service ID [hex]	0x00	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	ChannelId	ID of DIO channel
Parameters (inout)	None	
Parameters (out)	None	
Return value	Dio_LevelType	STD_HIGH The physical level of the corresponding Pin is STD_HIGH STD_LOW The physical level of the corresponding Pin is STD_LOW
Description	Returns the value of the specified DIO channel.	
Available via	Dio.h	

]()

[SWS_Dio_00027] [The [Dio_ReadChannel](#) function shall return the value of the specified DIO channel.

Regarding the return value of the [Dio_ReadChannel](#) function, the requirements [SWS_Dio_00083] and [SWS_Dio_00084] are applicable.

Furthermore, the requirements [SWS_Dio_00005] and [SWS_Dio_00026] are applicable to the [Dio_ReadChannel](#) function.](SRS_Dio_12008)

[SWS_Dio_00074] [If development error detection is enabled, the services [Dio_ReadChannel](#), [Dio_WriteChannel](#) and [Dio_FlipChannel](#) shall check the "ChannelId" parameter to be valid within the current configuration. If the "ChannelId" parameter is invalid, the functions shall report the error code [DIO_E_PARAM_INVALID_CHANNEL_ID](#) to the DET.](SRS_BSW_00323, SRS_SPAL_12448)

8.3.2 Dio_WriteChannel

[SWS_Dio_00134] Definition of API function Dio_WriteChannel [

Service Name	Dio_WriteChannel	
Syntax	<pre>void Dio_WriteChannel (Dio_ChannelType ChannelId, Dio_LevelType Level)</pre>	
Service ID [hex]	0x01	
Sync/Async	Synchronous	
Reentrancy	Reentrant	





Parameters (in)	ChannelId	ID of DIO channel
	Level	Value to be written
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Service to set a level of a channel.	
Available via	Dio.h	

]()

[SWS_Dio_00028] [If the specified channel is configured as an output channel, the `Dio_WriteChannel` function shall set the specified Level for the specified channel.] (*SRS_Dio_12005*)

[SWS_Dio_00029] [If the specified channel is configured as an input channel, the `Dio_WriteChannel` function shall have no influence on the physical output.] (*SRS_Dio_12005*)

[SWS_Dio_00079] [If the specified channel is configured as an input channel, the `Dio_WriteChannel` function shall have no influence on the result of the next Read-Service.

Furthermore, the requirements [\[SWS_Dio_00005\]](#), [\[SWS_Dio_00119\]](#) and [\[SWS_Dio_00026\]](#) are applicable to the `Dio_WriteChannel` function.] (*SRS_Dio_12005*)

8.3.3 Dio_ReadPort

[SWS_Dio_00135] Definition of API function Dio_ReadPort [

Service Name	Dio_ReadPort	
Syntax	<pre>Dio_PortLevelType Dio_ReadPort (Dio_PortType PortId)</pre>	
Service ID [hex]	0x02	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	PortId	ID of DIO Port
Parameters (inout)	None	
Parameters (out)	None	
Return value	Dio_PortLevelType	Level of all channels of that port
Description	Returns the level of all channels of that port.	
Available via	Dio.h	

]()

[SWS_Dio_00031] [The `Dio_ReadPort` function shall return the level of all channels of that port.] (*SRS_Dio_12006*)

[SWS_Dio_00104] [When reading a port which is smaller than the `Dio_PortLevelType` using the `Dio_ReadPort` function (see [\[SWS_Dio_00103\]](#)), the function shall set the bits corresponding to undefined port pins to 0.

Furthermore, the requirements [\[SWS_Dio_00005\]](#) and [\[SWS_Dio_00026\]](#) are applicable to the `Dio_ReadPort` function.]([SRS_Dio_12006](#))

[SWS_Dio_00075] [If development error detection is enabled, the functions `Dio_ReadPort`, `Dio_WritePort` and `Dio_MaskedWritePort` shall check the "PortId" parameter to be valid within the current configuration. If the "PortId" parameter is invalid, the functions shall report the error code `DIO_E_PARAM_INVALID_PORT_ID` to the DET.]([SRS_BSW_00323](#), [SRS_SPAL_12448](#))

8.3.4 Dio_WritePort

[SWS_Dio_00136] Definition of API function Dio_WritePort [

Service Name	Dio_WritePort	
Syntax	<pre>void Dio_WritePort (Dio_PortType PortId, Dio_PortLevelType Level)</pre>	
Service ID [hex]	0x03	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	PortId	ID of DIO Port
	Level	Value to be written
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Service to set a value of the port.	
Available via	Dio.h	

]()

[SWS_Dio_00034] [The `Dio_WritePort` function shall set the specified value for the specified port.]([SRS_Dio_12003](#))

[SWS_Dio_00035] [When the `Dio_WritePort` function is called, DIO Channels that are configured as input shall remain unchanged.]([SRS_Dio_12003](#))

[SWS_Dio_00105] [When writing a port which is smaller than the `Dio_PortLevelType` using the `Dio_WritePort` function (see [\[SWS_Dio_00103\]](#)), the function shall ignore the MSB.]([SRS_Dio_12003](#))

[SWS_Dio_00108] [The `Dio_WritePort` function shall have no effect on channels within this port which are configured as input channels.

Furthermore, the requirements [\[SWS_Dio_00005\]](#), [\[SWS_Dio_00119\]](#) and [\[SWS_Dio_00026\]](#) are applicable to the `Dio_WritePort` function.]([SRS_Dio_12003](#))

8.3.5 Dio_ReadChannelGroup

[SWS_Dio_00137] Definition of API function Dio_ReadChannelGroup [

Service Name	Dio_ReadChannelGroup	
Syntax	<pre>Dio_PortLevelType Dio_ReadChannelGroup (const Dio_ChannelGroupType* ChannelGroupIdPtr)</pre>	
Service ID [hex]	0x04	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	ChannelGroupIdPtr	Pointer to ChannelGroup
Parameters (inout)	None	
Parameters (out)	None	
Return value	Dio_PortLevelType	Level of a subset of the adjoining bits of a port
Description	This Service reads a subset of the adjoining bits of a port.	
Available via	Dio.h	

]()

[SWS_Dio_00037] [The [Dio_ReadChannelGroup](#) function shall read a subset of the adjoining bits of a port (channel group).] ([SRS_Dio_12007](#))

[SWS_Dio_00092] [The [Dio_ReadChannelGroup](#) function shall do the masking of the channel group.] ([SRS_Dio_12007](#))

[SWS_Dio_00093] [The [Dio_ReadChannelGroup](#) function shall do the shifting so that the values read by the function are aligned to the LSB.

Furthermore, the requirements [[SWS_Dio_00005](#)], [[SWS_Dio_00056](#)], [[SWS_Dio_00083](#)], [[SWS_Dio_00084](#)] and [[SWS_Dio_00026](#)] are applicable to the [Dio_ReadChannelGroup](#) function.] ([SRS_Dio_12007](#))

[SWS_Dio_00114] [If development error detection is enabled, the functions [Dio_ReadChannelGroup](#) and [Dio_WriteChannelGroup](#) shall check the "Channel GroupIdPtr" parameter to be valid within the current configuration. If the "Channel GroupIdPtr" parameter is invalid, the functions shall report the error code [DIO_E_PARAM_INVALID_GROUP](#) to the DET.] ([SRS_BSW_00323](#), [SRS_SPAL_12448](#))

8.3.6 Dio_WriteChannelGroup

[SWS_Dio_00138] Definition of API function Dio_WriteChannelGroup [

Service Name	Dio_WriteChannelGroup	
Syntax	<pre>void Dio_WriteChannelGroup (const Dio_ChannelGroupType* ChannelGroupIdPtr, Dio_PortLevelType Level)</pre>	





Service ID [hex]	0x05	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	ChannelGroupIdPtr	Pointer to ChannelGroup
	Level	Value to be written
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Service to set a subset of the adjoining bits of a port to a specified level.	
Available via	Dio.h	

]()

[SWS_Dio_00039] [The [Dio_WriteChannelGroup](#) function shall set a subset of the adjoining bits of a port (channel group) to a specified level.] ([SRS_Dio_12004](#))

[SWS_Dio_00040] [The [Dio_WriteChannelGroup](#) shall not change the remaining channels of the port and channels which are configured as input.] ([SRS_Dio_12004](#))

[SWS_Dio_00090] [The [Dio_WriteChannelGroup](#) function shall do the masking of the channel group.] ([SRS_Dio_12004](#))

[SWS_Dio_00091] [The function [Dio_WriteChannelGroup](#) shall do the shifting so that the values written by the function are aligned to the LSB.

Furthermore, the requirements [\[SWS_Dio_00005\]](#), [\[SWS_Dio_00056\]](#), [\[SWS_Dio_00119\]](#) and [\[SWS_Dio_00026\]](#) are applicable for the [Dio_WriteChannelGroup](#) function.] ([SRS_Dio_12004](#))

8.3.7 Dio_GetVersionInfo

[SWS_Dio_00139] Definition of API function Dio_GetVersionInfo [

Service Name	Dio_GetVersionInfo	
Syntax	<pre>void Dio_GetVersionInfo (Std_VersionInfoType* VersionInfo)</pre>	
Service ID [hex]	0x12	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	VersionInfo	Pointer to where to store the version information of this module.
Return value	None	
Description	Service to get the version information of this module.	
Available via	Dio.h	

] ([SRS_BSW_00411](#))

[SWS_Dio_00189] [If DET is enabled for the DIO Driver module, the function `Dio_GetVersionInfo` shall raise `DIO_E_PARAM_POINTER`, if the argument is NULL pointer and return without any action.](SRS_BSW_00323)

8.3.8 Dio_FlipChannel

[SWS_Dio_00190] Definition of API function `Dio_FlipChannel` [

Service Name	Dio_FlipChannel	
Syntax	<code>Dio_LevelType Dio_FlipChannel (</code> <code> Dio_ChannelType ChannelId</code> <code>)</code>	
Service ID [hex]	0x11	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	ChannelId	ID of DIO channel
Parameters (inout)	None	
Parameters (out)	None	
Return value	<code>Dio_LevelType</code>	<code>STD_HIGH</code> : The physical level of the corresponding Pin is <code>STD_HIGH</code> . <code>STD_LOW</code> : The physical level of the corresponding Pin is <code>STD_LOW</code> .
Description	Service to flip (change from 1 to 0 or from 0 to 1) the level of a channel and return the level of the channel after flip.	
Available via	Dio.h	

]()

[SWS_Dio_00191] [If the specified channel is configured as an output channel, the `Dio_FlipChannel` function shall read level of the channel (requirements [SWS_Dio_00083] and [SWS_Dio_00084] are applicable) and invert it, then write the inverted level to the channel. The return value shall be the inverted level of the specified channel.](SRS_Dio_12900)

[SWS_Dio_00192] [If the specified channel is configured as an input channel, the `Dio_FlipChannel` function shall have no influence on the physical output.

The return value shall be the level of the specified channel.](SRS_Dio_12900)

[SWS_Dio_00193] [If the specified channel is configured as an input channel, the `Dio_FlipChannel` function shall have no influence on the result of the next Read-Service.

Furthermore, the requirements [SWS_Dio_00005], [SWS_Dio_00119] and [SWS_Dio_00026] are applicable to the `Dio_FlipChannel` function.](SRS_Dio_12900)

8.3.9 Dio_MaskedWritePort

[SWS_Dio_00300] Definition of API function Dio_MaskedWritePort [

Service Name	Dio_MaskedWritePort	
Syntax	<pre>void Dio_MaskedWritePort (Dio_PortType PortId, Dio_PortLevelType Level, Dio_PortLevelType Mask)</pre>	
Service ID [hex]	0x13	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	PortId	ID of DIO Port
	Level	Value to be written
	Mask	Channels to be masked in the port
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Service to set the value of a given port with required mask.	
Available via	Dio.h	

]()

[SWS_Dio_00202] [The `Dio_MaskedWritePort` function shall set the specified value for the channels in the specified port if the corresponding bit in Mask is '1'.] (*SRS_Dio_12003*)

[SWS_Dio_00203] [When the `Dio_MaskedWritePort` function is called, DIO Channels that are configured as input shall remain unchanged.] (*SRS_Dio_12003*)

[SWS_Dio_00204] [When writing a port which is smaller than the `Dio_PortLevelType` using the `Dio_MaskedWritePort` function (see [SWS_Dio_00103]), the function shall ignore the MSB.] (*SRS_Dio_12003*)

8.4 Callback notifications

This chapter lists all functions provided by the Dio module to lower layers.

The Dio module does not provide any callback notifications. Callbacks related to the functionality of the Dio module are implemented in another module (ICU Driver and / or complex drivers).

8.5 Scheduled functions

This chapter lists all functions called directly by the Basic Software Module Scheduler.

The Dio module has no scheduled functions.

8.6 Expected interfaces

This chapter lists all functions the Dio module requires from other modules.

8.6.1 Mandatory Interfaces

None

8.6.2 Optional Interfaces

This chapter defines all interfaces which are required to fulfill an optional functionality of the module.

[SWS_Dio_00140] Definition of optional interfaces in module Dio [

<i>API Function</i>	<i>Header File</i>	<i>Description</i>
Det_ReportError	Det.h	Service to report development errors.

]()

9 Sequence diagrams

The diagrams below show the sequences when calling the `Dio_ReadChannel` and `Dio_WriteChannel` service, they show normal operation mode. Since all other services which are defined in Chapter 8.3 have exactly the same synchronous behavior concerning, there are intentionally no further sequence diagrams in this document.

9.1 Read a value from a digital I/O

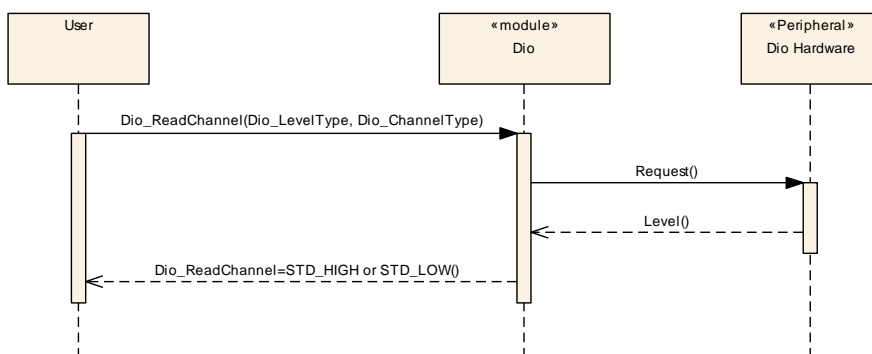


Figure 9.1: Read Service Sequence Chart (normal operation mode)

9.2 Write a value to a digital I/O

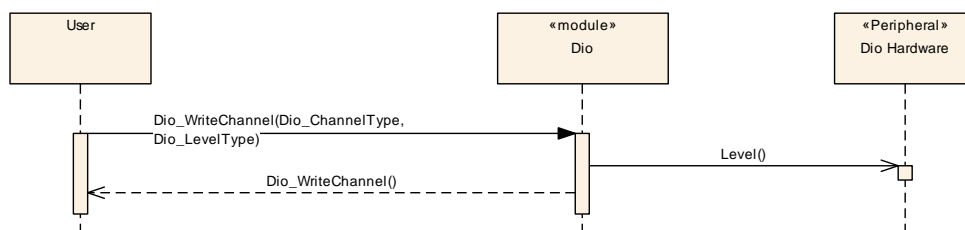


Figure 9.2: Write Service Sequence Chart (normal operation mode)

10 Configuration specification

This chapter defines configuration parameters and their clustering into containers.

10.1 Containers and configuration parameters

The following chapters summarize all configuration parameters.

[SWS_Dio_00205] [The DIO module shall reject configurations with partition mappings which are not supported by the implementation.] ()

10.1.1 Variants

[SWS_Dio_00129] [At least one of the following variants has to be supported by implementation: VARIANT-PRE-COMPILE, VARIANT-LINK-TIME.] ([SRS_BSW_00396](#))

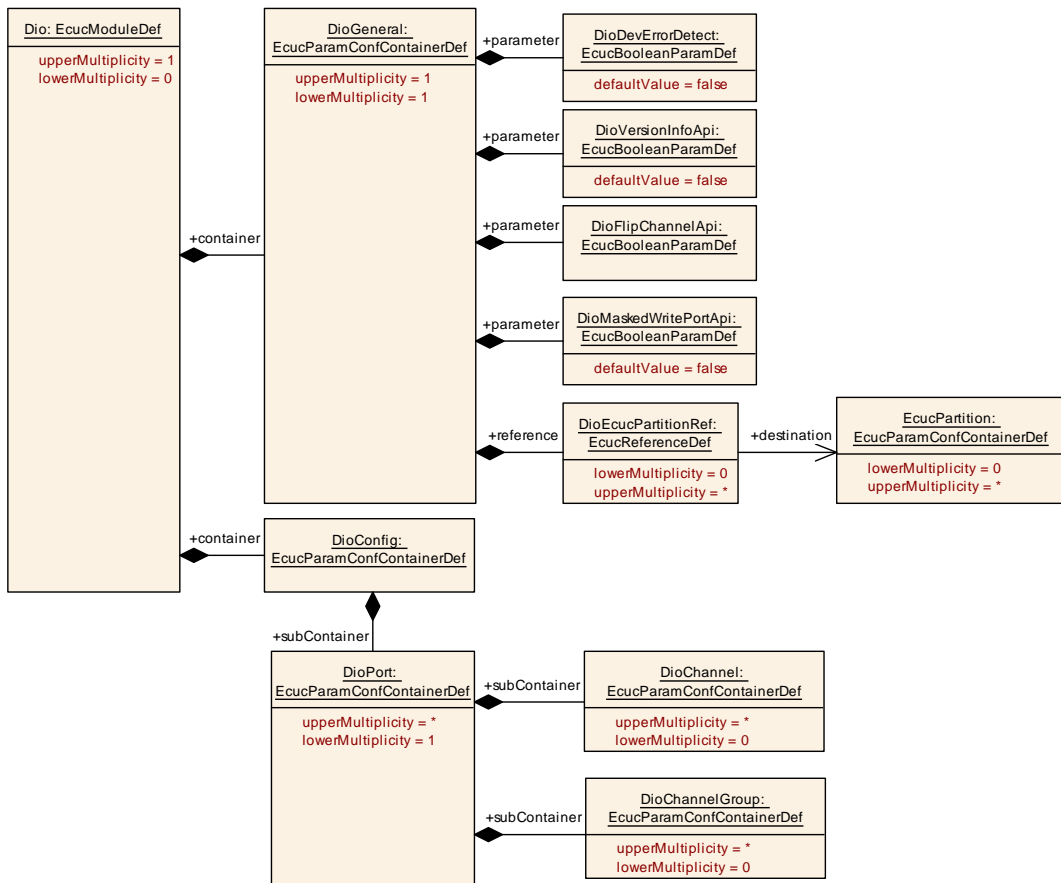


Figure 10.1: DIO Module Configuration

10.1.2 Dio

SWS Item	[ECUC_Dio_00154]
Module Name	Dio
Description	Configuration of the Dio (Digital IO) module.
Post-Build Variant Support	false
Supported Config Variants	VARIANT-LINK-TIME, VARIANT-PRE-COMPILE

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DioConfig	1	This container contains the configuration parameters and sub containers of the AUTOSAR DIO module.
DioGeneral	1	General DIO module configuration parameters.

10.1.3 DioGeneral

SWS Item	[ECUC_Dio_00141]
Container Name	DioGeneral
Parent Container	Dio
Description	General DIO module configuration parameters.
Configuration Parameters	

SWS Item	[ECUC_Dio_00142]		
Parameter Name	DioDevErrorDetect		
Parent Container	DioGeneral		
Description	Switches the development error detection and notification on or off. <ul style="list-style-type: none"> • true: detection and notification is enabled. • false: detection and notification is disabled. 		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	[ECUC_Dio_00153]
Parameter Name	DioFlipChannelApi
Parent Container	DioGeneral
Description	Adds / removes the service Dio_FlipChannel() from the code.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Post-Build Variant Value	false





Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	[ECUC_Dio_00155]		
Parameter Name	DioMaskedWritePortApi		
Parent Container	DioGeneral		
Description	Adds / removes the service Dio_MaskedWritePort() from the code.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	[ECUC_Dio_00143]		
Parameter Name	DioVersionInfoApi		
Parent Container	DioGeneral		
Description	Adds / removes the service Dio_GetVersionInfo() from the code.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	[ECUC_Dio_00156]		
Parameter Name	DioEcucPartitionRef		
Parent Container	DioGeneral		
Description	Maps the DIO driver to zero or multiple ECUC partitions to make the modules API available in this partition.		
Multiplicity	0..*		
Type	Reference to EcucPartition		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

No Included Containers

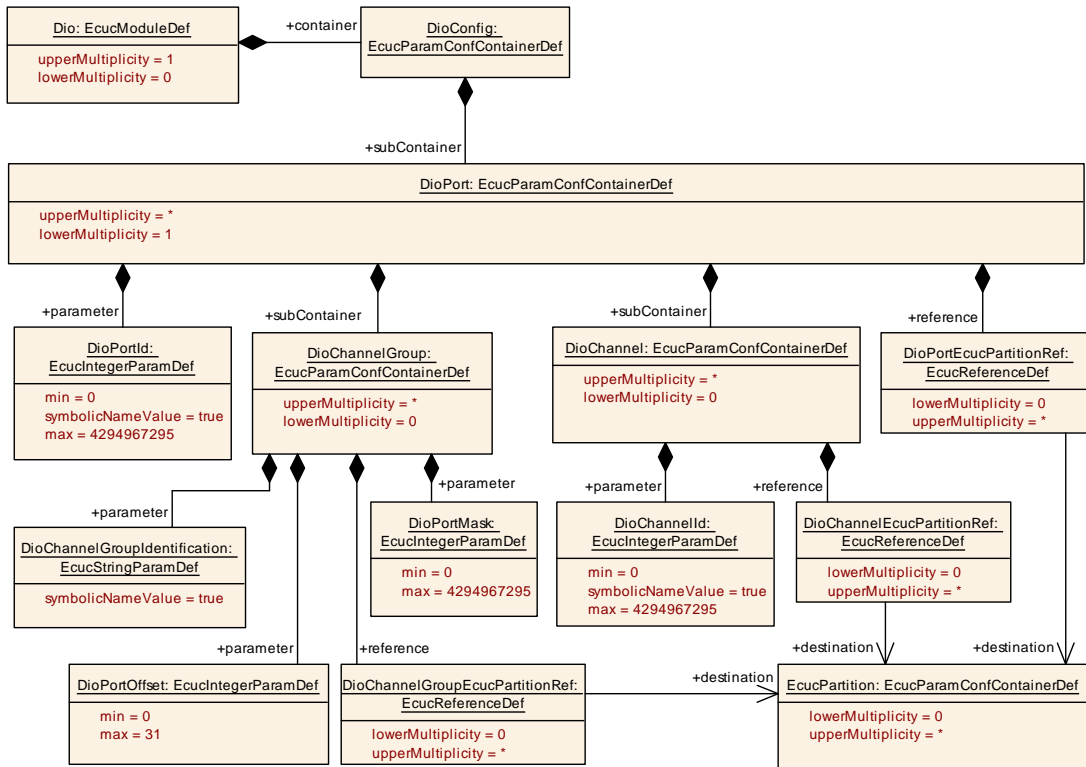


Figure 10.2: DioPort Configuration Container

10.1.4 DioPort

SWS Item	[ECUC_Dio_00144]
Container Name	DioPort
Parent Container	DioConfig
Description	Configuration of individual DIO ports, consisting of channels and possible channel groups. Note that this container definition does not explicitly define a symbolic name parameter. Instead, the container's short name will be used in the Ecu Configuration Description to specify the symbolic name of the port.
Configuration Parameters	

SWS Item	[ECUC_Dio_00145]
Parameter Name	DioPortId
Parent Container	DioPort
Description	Numeric identifier of the DIO port. Not all MCU ports may be used for DIO, thus there may be "gaps" in the list of all IDs. This value will be assigned to the DIO port symbolic name (i.e. the SHORT-NAME of the DioPort container).
Multiplicity	1
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)
Range	0 .. 4294967295
Default value	–
Post-Build Variant Value	false
Value Configuration Class	Pre-compile time
	X
	All Variants





	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

SWS Item	[ECUC_Dio_00157]		
Parameter Name	DioPortEcucPartitionRef		
Parent Container	DioPort		
Description	Maps the DIO ports to zero or multiple ECUC partitions. The ECUC partitions referenced are a subset of the ECUC partitions where the DIO driver is mapped to.		
Multiplicity	0..*		
Type	Reference to EcucPartition		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DioChannel	0..*	Configuration of an individual DIO channel.
DioChannelGroup	0..*	Definition and configuration of DIO channel groups. A channel group represents several adjoining DIO channels represented by a logical group. Note that this container definition does not explicitly define a symbolic name parameter. Instead, the container's short name will be used in the Ecu Configuration Description to specify the symbolic name of the channel group.

[SWS_Dio_00206] [The ECUC partitions referenced by [DioPortEcucPartitionRef](#) shall be a subset of the ECUC partitions referenced by [DioEcucPartitionRef](#).]
 ()

[SWS_Dio_00209] [If [DioEcucPartitionRef](#) references one or more ECUC partitions, [DioPortEcucPartitionRef](#) shall have a multiplicity of greater than zero and reference one or several of these ECUC partitions as well.]
 ()

10.1.5 DioChannel

SWS Item	[ECUC_Dio_00146]
Container Name	DioChannel
Parent Container	DioPort
Description	Configuration of an individual DIO channel.
Configuration Parameters	

SWS Item	[ECUC_Dio_00147]		
Parameter Name	DioChannelId		
Parent Container	DioChannel		
Description	Channel Id of the DIO channel. This value will be assigned to the symbolic names.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 4294967295		
Default value	-		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	-	
	Post-build time	-	
Scope / Dependency	scope: ECU		

SWS Item	[ECUC_Dio_00158]		
Parameter Name	DioChannelEcucPartitionRef		
Parent Container	DioChannel		
Description	Maps a DIO channel to zero or multiple ECUC partitions. The ECUC partitions referenced are a subset of the ECUC partitions where the related DIO port is mapped to.		
Multiplicity	0..*		
Type	Reference to EcucPartition		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	-	
	Post-build time	-	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	-	
	Post-build time	-	
Scope / Dependency	scope: ECU		

No Included Containers

[SWS_Dio_00207] [The ECUC partitions referenced by [DioChannelEcucPartitionRef](#) shall be a subset of the ECUC partitions referenced by [DioPortEcucPartitionRef](#).]()

[SWS_Dio_00210] [If [DioEcucPartitionRef](#) references one or more ECUC partitions, [DioChannelEcucPartitionRef](#) shall have a multiplicity of greater than zero and reference one or several of these ECUC partitions as well.]()

10.1.6 DioChannelGroup

SWS Item	[ECUC_Dio_00148]
Container Name	DioChannelGroup
Parent Container	DioPort
Description	Definition and configuration of DIO channel groups. A channel group represents several adjoining DIO channels represented by a logical group. Note that this container definition does not explicitly define a symbolic name parameter. Instead, the container's short name will be used in the Ecu Configuration Description to specify the symbolic name of the channel group.
Configuration Parameters	

SWS Item	[ECUC_Dio_00149]		
Parameter Name	DioChannelGroupIdentification		
Parent Container	DioChannelGroup		
Description	<p>The DIO channel group is identified in DIO API by a pointer to a data structure (of type Dio_ChannelGroupType). That data structure contains the channel group information.</p> <p>This parameter contains the code fragment that has to be inserted in the API call of the calling module to get the address of the variable in memory which holds the channel group information. Example values are "&MyDioGroup1" or "&MyDioGroupArray[0]"</p>		
Multiplicity	1		
Type	EcucStringParamDef (Symbolic Name generated for this parameter)		
Default value	-		
Regular Expression	-		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	-	
	Post-build time	-	
Scope / Dependency	scope: ECU		

SWS Item	[ECUC_Dio_00150]		
Parameter Name	DioPortMask		
Parent Container	DioChannelGroup		
Description	This shall be the mask which defines the positions of the channel group. The channels shall consist of adjoining bits in the same port. The data type depends on the port width.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default value	-		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	-	
Scope / Dependency	scope: local		

SWS Item	[ECUC_Dio_00151]		
Parameter Name	DioPortOffset		
Parent Container	DioChannelGroup		





Description	The position of the Channel Group on the port, counted from the LSB. This value can be derived from DioPortMask. calculationFormula = Position of the first bit of DioPortMask which is set to '1' counted from LSB		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 31		
Default value	-		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	-	
Scope / Dependency	scope: local		

SWS Item	[ECUC_Dio_00159]		
Parameter Name	DioChannelGroupEcucPartitionRef		
Parent Container	DioChannelGroup		
Description	Maps a DIO channel group to zero or multiple ECUC partitions. The ECUC partitions referenced are a subset of the ECUC partitions where the related DIO port is mapped to.		
Multiplicity	0..*		
Type	Reference to EcucPartition		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	-	
	Post-build time	-	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	-	
	Post-build time	-	
Scope / Dependency	scope: ECU		

No Included Containers

[SWS_Dio_00208] [The ECUC partitions referenced by [DioChannelGroupEcucPartitionRef](#) shall be a subset of the ECUC partitions referenced by [DioPortEcucPartitionRef](#).]()

[SWS_Dio_00211] [If [DioPortEcucPartitionRef](#) references one or more ECUC partitions, [DioChannelGroupEcucPartitionRef](#) shall have a multiplicity of greater than zero and reference one or several of these ECUC partitions as well.]()

10.1.7 DioConfig

SWS Item	[ECUC_Dio_00152]
Container Name	DioConfig
Parent Container	Dio
Description	This container contains the configuration parameters and sub containers of the AUTOSAR DIO module.
Configuration Parameters	

Included Containers		
Container Name	Multiplicity	Scope / Dependency
DioPort	1..*	Configuration of individual DIO ports, consisting of channels and possible channel groups. Note that this container definition does not explicitly define a symbolic name parameter. Instead, the container's short name will be used in the Ecu Configuration Description to specify the symbolic name of the port.

10.2 Published Information

For details refer to the chapter 10.3 "Published Information" in SWS_BSWGeneral.

A Not applicable requirements

[SWS_Dio_NA_00195] [These requirements are not applicable to this specification.] (SRS_BSW_00101, SRS_BSW_00005, SRS_BSW_00006, SRS_BSW_00007, SRS_BSW_00009, SRS_BSW_00010, SRS_BSW_00160, SRS_BSW_00161, SRS_BSW_00162, SRS_BSW_00164, SRS_BSW_00167, SRS_BSW_00168, SRS_BSW_00170, SRS_BSW_00172, SRS_BSW_00304, SRS_BSW_00306, SRS_BSW_00307, SRS_BSW_00308, SRS_BSW_00309, SRS_BSW_00314, SRS_BSW_00321, SRS_BSW_00325, SRS_BSW_00328, SRS_BSW_00330, SRS_BSW_00331, SRS_BSW_00333, SRS_BSW_00334, SRS_BSW_00335, SRS_BSW_00336, SRS_BSW_00339, SRS_BSW_00341, SRS_BSW_00342, SRS_BSW_00343, SRS_BSW_00347, SRS_BSW_00357, SRS_BSW_00359, SRS_BSW_00360, SRS_BSW_00369, SRS_BSW_00373, SRS_BSW_00375, SRS_BSW_00377, SRS_BSW_00378, SRS_BSW_00384, SRS_BSW_00399, SRS_BSW_00400, SRS_BSW_00404, SRS_BSW_00405, SRS_BSW_00406, SRS_BSW_00413, SRS_BSW_00416, SRS_BSW_00417, SRS_BSW_00422, SRS_BSW_00423, SRS_BSW_00424, SRS_BSW_00425, SRS_BSW_00426, SRS_BSW_00427, SRS_BSW_00428, SRS_BSW_00429, SRS_BSW_00432, SRS_BSW_00433, SRS_SPAL_00157, SRS_SPAL_12057, SRS_SPAL_12063, SRS_SPAL_12067, SRS_SPAL_12068, SRS_SPAL_12069, SRS_SPAL_12075, SRS_SPAL_12077, SRS_SPAL_12078, SRS_SPAL_12092, SRS_SPAL_12125, SRS_SPAL_12129, SRS_SPAL_12163, SRS_SPAL_12169, SRS_SPAL_12265, SRS_SPAL_12267)