|  |  |
| --- | --- |
| **Document Title** | Specification of Port Driver |
| **Document Owner** | AUTOSAR |
| **Document Responsibility** | AUTOSAR |
| **Document Identification No** | 040 |
| **Document Classification** | Standard |
|  | |
| **Document Version** | 3.2.0 |
| **Document Status** | Final |
| **Part of Release** | 4.0 |
| **Revision** | 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Document Change History** | | | |
| **Date** | **Version** | **Changed by** | **Change Description** |
| 03.11.2010 | 3.2.0 | AUTOSAR  Administration | * Removed Port132 and updated Figure 1 |
| * Rephrased Port114 and Port075 |
| * Removed Port210 * Added Chapter 12 |
| 30.11.2009 | 3.1.0 | AUTOSAR  Administration | * A lot of requirements are changed to make them atomic or /and more clear. No technical content changes. Look Chapter 14 for the complete list. |
| * Debugging Concept Inserted. |
| * Insertion of the new configuration parameter to enable or disable the mode   of the port pins at run time.   * Checking of Port\_GetVersionInfo. * Legal disclaimer revised |
| 23.06.2008 | 3.0.2 | AUTOSAR  Administration | Legal disclaimer revised |
| 23.01.2008 | 3.0.1 | AUTOSAR  Administration | Table formatting corrected |

|  |  |  |  |
| --- | --- | --- | --- |
| **Document Change History** | | | |
| 27.11.2007 | 3.0.0 | AUTOSAR  Administration | * Update to Chapter 10 configuration. * Inclusion of Port Container |
|  |  |  | * Inclusion of new SRS general |
|  |  |  | requirements |
|  |  |  | * Removal of redundant function: |
|  |  |  | Dem\_ReportErrorEvent() |
|  |  |  | * Development errors and error codes |
|  |  |  | added |
|  |  |  | * Rewording of requirements (as part of |
|  |  |  | the SWS Improvements) |
|  |  |  | * Renaming of configuration parameter |
|  |  |  | (PORT\_PIN\_DIRECTION\_CHANGES\_ |
|  |  |  | ALLOWED -> |
|  |  |  | PORT\_SEP\_PIN\_DIRECTION\_API) |
|  |  |  | * Technical Office Improvements: wording |
|  |  |  | improvements, alignment of API |
|  |  |  | description. |
|  |  |  | * Document meta information extended |
|  |  |  | * Small layout adaptations made |
| 31.01.2007 | 2.1.0 | AUTOSAR  Administration | * New API introduced: Port\_SetPinMode() |
|  |  |  | * New Module type definition: |
|  |  |  | Port\_PinModeType |
|  |  |  | * Updated to section [5.1.2](#_bookmark13): Inclusion of |
|  |  |  | new file structure information |
|  |  |  | * Inclusion of new pre-processor switch |
|  |  |  | PortSetPinModeApi |
|  |  |  | * New configurable parameter introduced: |
|  |  |  | PortPinInitialMode |
|  |  |  | * Rewording of requirement PORT105. |
|  |  |  | * Removal of redundant requirements |
|  |  |  | PORT119 and PORT026 in |
|  |  |  | requirements matrix. |
|  |  |  | * Legal disclaimer revised |
|  |  |  | * Release Notes added |
|  |  |  | * “Advice for users” revised |
|  |  |  | * “Revision Information” added |
| 28.04.2006 | 2.0.0 | AUTOSAR | Document structure adapted to common |
|  |  | Administration | Release 2.0 SWS Template. |
|  |  |  | * Major changes in chapter [10](#_bookmark109) |
|  |  |  | * Structure of document changed partly |
|  |  |  | * Other changes see chapter [11](#_bookmark126) |
| 30.06.2005 | 1.0.0 | AUTOSAR | Initial Release |
|  |  | Administration |  |

### Disclaimer

This specification 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 specification.

The material contained in this specification is protected by copyright and other types of Intellectual Property Rights. The commercial exploitation of the material contained in this specification requires a license to such Intellectual Property Rights.

This specification 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 specification may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The AUTOSAR specifications have been developed for automotive applications only. They have neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

### Advice for users

AUTOSAR specifications 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 specifications for illustration purposes only, and they themselves are not part of the AUTOSAR Standard. Neither their presence in such specifications, 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. [Introduction and functional overview 6](#_bookmark0)
2. [Acronyms and abbreviations 8](#_bookmark1)
3. [Related documentation 9](#_bookmark2)
   1. [Input documents 9](#_bookmark3)
   2. [Related standards and norms 9](#_bookmark6)
4. [Constraints and assumptions 10](#_bookmark7)
   1. [Limitations 10](#_bookmark8)
   2. [Applicability to car domains 10](#_bookmark9)
5. [Dependencies to other modules 11](#_bookmark10)
   1. [File structure 11](#_bookmark11)
      1. [Code file structure 11](#_bookmark12)
      2. [Header file structure 11](#_bookmark13)
6. [Requirements traceability 14](#_bookmark15)
7. [Functional specification 23](#_bookmark22)
   1. [General Behaviour 23](#_bookmark23)
      1. [Background & Rationale 23](#_bookmark24)
      2. [Requirements 23](#_bookmark25)
         1. [Configuration of Port Pin Properties 23](#_bookmark26)
         2. [Switch port pin direction 24](#_bookmark27)
         3. [Refresh port direction 24](#_bookmark28)
         4. [Configuration of unused Ports and Port Pins 24](#_bookmark29)
         5. [Configuration of symbolic names 24](#_bookmark30)
         6. [Atomicity of port access 24](#_bookmark31)
      3. [Version Check 25](#_bookmark32)
         1. [Background and Rationale 25](#_bookmark33)
         2. [Requirements 25](#_bookmark34)
   2. [Error classification 25](#_bookmark35)
   3. [Error detection 26](#_bookmark36)
   4. [Error notification 26](#_bookmark37)
   5. [API Parameter checking 26](#_bookmark38)
   6. [Debugging Support 27](#_bookmark42)
8. [API specification 29](#_bookmark47)
   1. [Imported types. 29](#_bookmark48)
   2. [Type definitions 29](#_bookmark50)
      1. [Port\_ConfigType 29](#_bookmark52)
      2. [Port\_PinType 30](#_bookmark56)
      3. [Port\_PinDirectionType 30](#_bookmark58)
      4. [Port\_PinModeType 30](#_bookmark61)
   3. [Function definitions 31](#_bookmark64)
      1. [Port\_Init 31](#_bookmark65)
      2. [Port\_SetPinDirection 33](#_bookmark70)
      3. [Port\_RefreshPortDirection 34](#_bookmark74)
      4. [Port\_GetVersionInfo 34](#_bookmark75)
      5. [Port\_SetPinMode 35](#_bookmark77)
   4. [Call-back notifications 36](#_bookmark83)
   5. [Scheduled functions 36](#_bookmark86)
   6. [Expected Interfaces 36](#_bookmark89)
      1. [Mandatory Interfaces 36](#_bookmark91)
      2. [Optional Interfaces 36](#_bookmark93)
      3. [Configurable Interfaces 36](#_bookmark97)
9. [Sequence diagrams 37](#_bookmark98)
   1. [Overall Configuration of Ports 37](#_bookmark99)
   2. [Set the direction of a Port Pin 37](#_bookmark100)
   3. [Refresh the direction of all Port Pins 38](#_bookmark103)
   4. [Change the mode of a Port Pin 38](#_bookmark104)
10. [Configuration specification 39](#_bookmark109)
    1. [How to read this chapter 39](#_bookmark110)
       1. [Configuration and configuration parameters 39](#_bookmark111)
       2. [Containers 39](#_bookmark112)
    2. [Containers and configuration parameters 40](#_bookmark113)
       1. [Variants 40](#_bookmark114)
       2. [Port 40](#_bookmark115)
       3. [PortContainer 40](#_bookmark116)
       4. [PortGeneral 41](#_bookmark117)
       5. [PortPin 42](#_bookmark120)
       6. [PortConfigSet 44](#_bookmark123)
    3. [Published Information 46](#_bookmark124)
11. [Changes to Release 3.0 47](#_bookmark126)
    1. [Deleted SWS Items 47](#_bookmark127)
    2. [Replaced SWS Items 47](#_bookmark128)
    3. [Changed SWS Items 47](#_bookmark129)
    4. [Added SWS Items 47](#_bookmark130)
12. [Changes to Release 4.0 Rev 1 49](#_bookmark132)
    1. [Deleted SWS Items 49](#_bookmark133)
    2. [Replaced SWS Items 49](#_bookmark134)
    3. [Changed SWS Items 49](#_bookmark135)
    4. [Added SWS Items 49](#_bookmark136)

# 1 Introduction and functional overview

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

This driver specification is applicable for on-chip ports and port pins.

This module shall provide the service for initializing the whole PORT structure of the microcontroller. Many ports and port pins can be assigned to various functionalities, e.g.

* General purpose I/O
* ADC
* SPI
* SCI
* PWM
* CAN
* LIN
* etc

For this reason, there shall be an overall configuration and initialization of this port structure. The configuration and mode of these port pins is microcontroller and ECU dependent.

Port initialisation data shall be written to each port as efficiently as possible.

This PORT driver module shall complete the overall configuration and initialisation of the port structure which is used in the DIO driver module. Therefore, the DIO driver works on pins and ports which are configured by the PORT driver.

The PORT driver shall be initialised prior to use of the DIO functions. Otherwise DIO functions will exhibit undefined behaviour.

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

|  |  |  |  |
| --- | --- | --- | --- |
| ***Driver*** | ***Name for a Port Pin*** | ***Name for Subset of Adjacent pins on one port*** | ***Name for a whole port*** |
| **DIO Driver** | Channel | Channel Group | Port |
| **PORT Driver** | Port pin | -- | Port |

**IO HW**

**Abstraction Software**

Inside this dotted line is PORT Driver specific

**MCAL**

I/O HW Abstraction Driver Functions

**Software**

PORT\_Init

DIO\_W riteChannel

DIO\_ReadPort

**PORT Driver DIO Driver**

PORT\_Config

DIO\_W riteGroup

Other PORT Driver Functions

Other DIO Driver Functions

Port Function

Port Data Read

**On-Chip Registers**

Processor Specific

Processor Specific

Port Direction

Port Data W rite

**On-Chip Hardware**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PORT | | | | | |
| PIN0 | PIN1 | PIN2 | PIN3 | PIN... | PINn |

# 2 Acronyms and abbreviations

The following table summarizes the expressions used within the PORT driver.

|  |  |
| --- | --- |
| ***Abbreviation / Acronym:*** | ***Description:*** |
| DEM | Diagnostic Event Manager |
| DET | Development Error Tracer |
| MCU | MicroController Unit |
| Port Pin | Represents a single configurable input or output pin on an MCU device. |
| Port | Represents a whole configurable port on an MCU device. |
| Physical Level (Input) | Two states are possible: LOW/HIGH |
| Physical Level (Output) | Two states are possible: LOW/HIGH |

# Related documentation

## Input documents

[1] List of Basic Software Modules,

AUTOSAR\_TR\_BSWModuleList.pdf

[2] Layered Software Architecture,

AUTOSAR\_EXP\_LayeredSoftwareArchitecture.pdf

[3] General Requirements on Basic Software Modules,

AUTOSAR\_SRS\_BSWGeneral.pdf

[4] Specification of Development Error Tracer,

AUTOSAR\_SWS\_DevelopmentErrorTracer.pdf

[5] Specification of ECU Configuration,

AUTOSAR\_TPS\_ECUConfiguration.pdf

[6] Specification of Diagnostic Event Manager,

AUTOSAR\_SWS\_DiagnosticEventManager.pdf

[7] Specification of ECU State Manager,

AUTOSAR\_SWS\_ECUStateManager.pdf

[8] General Requirements on SPAL,

AUTOSAR\_SRS\_SPALGeneral.pdf

[9] Requirements on PORT driver,

AUTOSAR\_SRS\_PORTDriver.pdf

[10] Specification of Standard Types,

AUTOSAR\_SWS\_StandardTypes.pdf

[11] Basic Software Module Description Template, AUTOSAR\_TPS\_BSWModuleDescriptionTemplate.pdf

## Related standards and norms

[12] EC 7498-1 The Basic Model, IEC Norm, 1994

# Constraints and assumptions

## Limitations

Limitations for the PORT driver are specified as followed:

* + - It is the user’s responsibility to ensure that the same Port/Port pin is not being accessed in parallel by different entities in the same system, e.g. by two tasks configuring the same port or two tasks configuring the same pin, or two tasks configuring different pins on the same port.

## Applicability to car domains

No restrictions

# Dependencies to other modules

Other driver modules may be dependent on the PORT driver depending on the available functionality of individual port pins on an MCU. For example, an MCU pin may be configurable as a DIO or SPI pin. Therefore, the DIO and/or the SPI driver modules may be dependent on the PORT module to configure the pin for the desired functionality.

## File structure

### Code file structure

The code file structure shall not be defined within this specification completely.

**PORT108:** The code file structure shall include the following files named:

* + - * Port\_Lcfg.c – for link time configurable parameters and
      * Port\_PBcfg.c – for post build time configurable parameters.

These files shall contain all link time and post-build time configurable parameters.

### Header file structure

**PORT080:** The include file structure of the Port Driver shall be as follows:



«header» Port\_Cfg.h

«header» Port.h

«header» Std\_Types.h

«includes»

«includes»

«header» MemMap.h

«source» Port\_PBcfg.c

«source» Port.c

«optional,

des»

«header» Port\_Cbk.h

«header» Det.h

«optional,includes»

«includes» «includes»

«includes» «includes» «includes»

«includes»

«includes»

«includes»

includes»

«includes»

«includes»

«optional,inclu

«optional,includes»

«optional,includes»

«header» EcuM.h

«header» Compiler.h

«header» Platform\_Types.h

«source» Port\_Lcfg.c

«header» Dem.h

«header» SchM\_Port.h

**Figure 1 Header File Structure**

**PORT130:** Port.h shall include Port\_Cfg.h for the API pre-compiler switches,

Std\_Types.h and EcuM.h

**PORT131:** Port.c shall include Port.h, MemMap.h, Det.h, SchM\_Port.h

(see PORT114 for inclusion checking).

**PORT204:** Port.c shall optionally include Port\_Cbk.h and Dem.h Port.c has implicit access to the Port\_Cfg.h file through the Port.h file. **PORT205:** Port\_Lcfg.c shall include MemMap.h and Port.h **PORT133:** Port\_PBcfg.c shall include MemMap.h and Port.h

The implicit include of Port\_Cfg.h via Port.h in the files Port\_Lcfg.c and

Port\_PBcfg.c is necessary to solve the following construct:

Port.h

----------

#ifdef xxx\_VERSION\_INFO\_API xxx\_GetVersionInfo(...)

#endif

Port\_Cfg.h

---------------

#define xxx\_VERSION\_INFO\_API

Note: A separate file type is not required for the PORT driver as the Port Types depend on the platform and are not configurable.

# Requirements traceability

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

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

Document: AUTOSAR requirements on Basic Software, general [[3](#_bookmark14)]

|  |  |
| --- | --- |
| ***Requirement*** | ***Satisfied by*** |
| [BSW003] Version identification | [PORT222](#_bookmark20) |
| [BSW004] Version check | [PORT114](#_bookmark51), [Chapter 7.1.3](#_bookmark49) |
| [BSW005] No hard coded horizontal interfaces within MCAL | Not applicable  (Because architectural AUTOSAR concept is the basis for this concept) |
| [BSW006] Platform independency | Not applicable  (because the module is not above the MCAL) |
| [BSW007] HIS MISRA C | Not applicable  (Because it is requirement on implementation) |
| [BSW009] Module User Documentation | [Section 3.1](#_bookmark14) |
| [BSW010] Memory resource documentation | Not applicable  (Because this is a requirement for the implementer) |
| [BSW101] Initialization interface | [PORT001](#_bookmark39), [PORT002](#_bookmark94), [PORT041](#_bookmark81), [PORT042](#_bookmark82) |
| [BSW158] Separation of configuration from implementation | [Figure 5.1: Header File](#_bookmark17) |
| [BSW159] Tool-based configuration | [PORT004](#_bookmark40) |
| [BSW160] Human-readable configuration data | Not applicable  (Because it only applies to the configuration. Requirement on implementation) |
| [BSW161] Microcontroller abstraction | Not applicable  (Because architectural AUTOSAR concept is the basis for this concept) |
| [BSW162] ECU layout abstraction | Not applicable  (Because architectural AUTOSAR concept is the basis for this concept) |
| [BSW164] Implementation of interrupt service routines Not | applicable  (Because the PORT does not provide interrupt functionality |
| [BSW167] Static configuration checking Not Applicable | (Because this is a requirement for the configuration tool). |
| [BSW168] Diagnostic Interface of SW components | Not applicable  (Because the PORT does not provide diagnostic capabilities |
| [BSW170] Data for reconfiguration of AUTOSAR SW- Components | Not applicable  (Because it only affects the configuration) |

|  |  |
| --- | --- |
| [BSW171] Configurability of optional functionality | [PORT117\_Conf](#_bookmark122), [PORT118\_Conf](#_bookmark125) |
| [BSW172] Compatibility and documentation of scheduling strategy | Not applicable  (Because PORT does not have any special scheduling requirements |
| [BSW00300] Module naming convention | [Figure 5.1](#_bookmark16) |
| [BSW00301] Limit imported information | [Figure 5.1](#_bookmark16) |
| [BSW00302] Limit exported information | [Figure 5.1](#_bookmark16) |
| [BSW00304] AUTOSAR integer data types | [Section 8.2](#_bookmark71) |
| [BSW00305] Self-defined data types naming convention | [Section 8.2](#_bookmark71) |
| [BSW00306] Avoid direct use of compiler and platform specific keywords | [Figure 5.1](#_bookmark16) |
| [BSW00307] Global variables naming convention | Not applicable  (Because it is a requirement on implementation) |
| [BSW00308] Definition of global data | Not applicable  (Because this is a requirement for the implementer) |
| [BSW00309] Global data with read-only constraint | Not applicable  (Req for the implementer) |
| [BSW00310] API naming convention | [Section 8.3](#_bookmark78) |
| [BSW00312] Shared code shall be reentrant | [Section 8.3](#_bookmark78) |
| [BSW00314] Separation of interrupt frames and service routines | [Figure 5.1](#_bookmark16) |
| [BSW00318] Format of module version numbers | [PORT222](#_bookmark20) |
| [BSW00321] Enumeration of module version numbers | Not applicable  (Because this is a requirement for the Implementer) |
| [BSW00323] API parameter checking | [PORT087](#_bookmark63) |
| [BSW00325] Runtime of interrupt service routines | Not applicable  (Because the PORT driver does not provide interrupt capabilities. This is a requirement for implementation) |
| [BSW00326] Transition from ISRs to OS tasks | Not applicable  (Because PORT does not provide interrupt capabilities. This is a requirement on implementation) |
| [BSW00327] Error values naming convention | [PORT051](#_bookmark55) |
| [BSW00328] Avoid duplication of code | Not applicable  (Because this is a requirement for the implementer) |
| [BSW00329] Avoidance of generic interfaces | Not applicable  (Because there are no generic interfaces specified within this SWS) |
| [BSW00330] Usage of macros / inline functions instead of functions | Not applicable  (Because this is a requirement for the implementer) |
| [BSW00331] Separation of error and status values | Not applicable  (Because there are no status values specified within this SWS) |

|  |  |
| --- | --- |
| [BSW00333] Documentation of callback function context | Not applicable  (Because it is a non functional requirement. There is no callback functionality in the PORT module) |
| [BSW00334] Provision of XML file | Not applicable  (Because this is specified by WP ECU Configuration) |
| [BSW00335] Status values naming convention | Not applicable  (Because there are no status values specified within the SWS) |
| [BSW00336] Shutdown interface | Not applicable  (Because for the PORT driver there is no need for this requirement. |
| [BSW00337] Classification of errors | [PORT051](#_bookmark55) |
| [BSW00338] Detection and Reporting of development errors | [PORT100](#_bookmark57) |
| [BSW00339] Reporting of production relevant error status | [PORT037](#_bookmark59) |
| [BSW00341] Microcontroller compatibility documentation | Not applicable  (Because this is a requirement for the Implementer) |
| [BSW00342] Usage of source code and object code | Not applicable  (Because it is requirement on implementation) |
| [BSW00343] Specification and configuration of time | Not applicable  (Because it is a non functional requirement. Time configuration is not a requirement of the PORT driver. |
| [BSW00344] Reference to Link-time configuration | Not applicable  (PORT driver does not support Link-time configuration) |
| [BSW00345] Pre-compile-time configuration | [Figure 5.1: Header File](#_bookmark17), [PORT117\_Conf](#_bookmark122) |
| [BSW00346] Basic set of module files | [Figure 5.1: Header File](#_bookmark17) |
| [BSW00347] Naming separation of different instances of BSW drivers | Not applicable  (Because it is a requirement on implementation) |
| [BSW00348] Standard type header | [Figure 5.1](#_bookmark16) |
| [BSW00350] Development error detection keyword | [PORT123\_Conf](#_bookmark121) |
| [BSW00353] Platform specific type header | [Figure 5.1](#_bookmark16) |
| [BSW00355] Do not redefine AUTOSAR integer data types | Not applicable  (Because no integer data types are redefined in this specification) |
| [BSW00357] Standard API return type | Not applicable  (Because this type is not used within the SWS) |
| [BSW00358] Return type of init() functions | [Section 8.3.1](#_bookmark79), [PORT140](#_bookmark80) |
| [BSW00359] Return type of callback functions | Not applicable  (Because the PORT module does not provide a callback mechanism) |
| [BSW00360] Parameters of callback functions | Not applicable  (Because the PORT module does not provide a callback mechanism) |
| [BSW00361] Compiler specific language extension header | [Figure 5.1](#_bookmark16) |

|  |  |
| --- | --- |
| [BSW00369] Do not return development error codes via API | [PORT107](#_bookmark60) |

|  |  |
| --- | --- |
| [BSW00370] Separation of callback interface from API | Not applicable  (Because the PORT driver does not provide a callback mechanism) |
| [BSW00371] Do not pass function pointers via API | Not applicable  (Because no function pointers are passed via API in this SWS) |
| [BSW00373] Main processing function naming convention | Not applicable  (Because it is a non functional requirement. There is no main processing function specified in the PORT driver) |
| [BSW00374] Module vendor identification | [PORT222](#_bookmark20) |
| [BSW00375] Notification of wake-up reason | Not applicable  (Because the PORT driver does not provide a wake-up mechanism) |
| [BSW00376] Return type and parameters of main processing functions | Not applicable  (Because there is no main processing function specified) |
| [BSW00377] Module specific API return types | Not applicable  (Because this type is not used within the SWS) |
| [BSW00378] AUTOSAR boolean type | Not applicable  (data type not defined inside this SWS) |
| [BSW00379] Module identification | [PORT222](#_bookmark20) |
| [BSW00380] Separate C-File for configuration parameters | [Figure 5.1: Header File](#_bookmark16) |
| [BSW00381] Separate configuration header file for pre- compile time parameters | [Figure 5.1: Header File](#_bookmark16) |
| [BSW00383] List dependencies of configuration files | [PORT080](#_bookmark18), Section 5.1.2 |
| [BSW00384] List dependencies to other modules | [PORT080](#_bookmark18), Section 5.1.2 |
| [BSW00385] List possible error notificatons | [PORT051](#_bookmark55) |
| [BSW00386] Configuration for detecting an error | [Chapter 7.2.1](#_bookmark53) |
| [BSW00387] Specify the configuration class of callback function | Not applicable  (Because the PORT driver does not have any callback capability). |
| [BSW00388] Introduce containers | [Chapter 10.2](#_bookmark119) |
| [BSW00389] Containers shall have names | [Chapter 10.2](#_bookmark119) |
| [BSW00390] Parameter content shall be unique within the module | [Chapter 8.3](#_bookmark76) |
| [BSW00391] Parameter shall have unique names | [Chapter 8.3](#_bookmark76) |
| [BSW00392] Parameters shall have a type | [Chapter 8.3](#_bookmark76) |
| [BSW00393] Parameters shall have a range | [Chapter 8.3](#_bookmark76) |
| [BSW00394] Specify the scope of the parameters | [Chapter 8.3](#_bookmark76) |
| [BSW00395] List the required parameters (per parameter)N | ot applicable  (Because none of the parameters of the PORT driver are dependent on other modules) |
| [BSW00396] Configuration classes | [Chapter 10.2](#_bookmark119) |
| [BSW00397] Pre-compile-time parameters | [PORT117\_Conf](#_bookmark122) |
| [BSW00398] Link-time parameters | Not applicable  (No Link-time parameter specified) |
| [BSW00399] Loadable Post-build time parameters | [Chapter 10.2](#_bookmark119) |
| [BSW00400] Selectable Post-build time parameters | [Chapter 10.2](#_bookmark119) |
| [BSW00401] Documentation of multiple instances of configuration parameters | [Chapter 10.2](#_bookmark119) |
| [BSW00402] Published information | [Chapter 10.3](#_bookmark137) |
| [BSW00404] Reference to post build time configuration | [PORT041](#_bookmark81) |
| [BSW00405] Reference to multiple configuration sets | [PORT121\_Conf](#_bookmark131) |

|  |  |
| --- | --- |
| [BSW00406] Check module initialization | [PORT051](#_bookmark55), [PORT087](#_bookmark63) (PORT\_E\_UNINIT) |
| [BSW00407] Function to read out published parameters | [PORT102](#_bookmark108) |
| [BSW00408] Configuration parameter naming convention | [Chapter 10.2](#_bookmark119) |
| [BSW00409] Header files for production code error IDs | [PORT115](#_bookmark54) |
| [BSW00410] Compiler switches shall have defined values | [Chapter 10.2](#_bookmark119) |
| [BSW00411] Get version info keyword | [PORT103](#_bookmark109) |
| [BSW00412] Separate H-File for configuration parameters | [Figure 5.1: Header file](#_bookmark17) |
| [BSW00413] Accessing instances of BSW modules | Not applicable  (Because this is a requirement on implementation) |
| [BSW00414] Parameter of init function | [PORT121](#_bookmark98) |
| [BSW00415] User dependent include files | [Figure 5.1](#_bookmark17) |
| [BSW00416] Sequence of Initialization | Not applicable  (Because this requirement describes the initialization of the whole SPAL layer). |
| [BSW00417] Reporting of Error Events by Non-Basic Software | Not applicable  (Because this driver is part of the basic software layer. This requirement applies only for non-BSW modules). |
| [BSW00419] Separate C-Files for pre-compile time configuration parameters | Not applicable  C file for pre-compile parameter is not defined in this SWS. |
| [BSW00420] Production relevant error event rate detection | Not applicable  (Because it is a non functional requirement and applies only for DEM) |
| [BSW00421] Reporting of production relevant error events | Not applicable  (Req rejected from the SRS General) |
| [BSW00422] Debouncing of production relevant error status | [PORT037](#_bookmark59) |
| [BSW00423] Usage of SW-C template to describe BSW modules with AUTOSAR Interfaces | Not applicable  (Because it is a non-functional requirement. The PORT driver has no AUTOSAR interface) |
| [BSW00424] BSW main processing function task allocation | Not applicable  (Because the PORT driver does not contain any main processing functions) |
| [BSW00425] Trigger conditions for schedulable objects | Not applicable  (Because the PORT driver does not contain any schedulable objects/services. This is a requirement for the Implementer). |
| [BSW00426] Exclusive areas in BSW modules | Not applicable  (Because applies only for the module descriptions template) |
| [BSW00427] ISR description for BSW modules | Not applicable  (Because this is a requirement for the implementer) |
| [BSW00428] Execution order dependencies of main processing functions | Not applicable  (Because there is no main processing function specified). |
| [BSW00429] Restricted BSW OS functionality access | Not applicable  (Requirement on the implementation ) |
| [BSW00431] The BSW Scheduler module implements task bodies | Not applicable  (Because this requirement is for an upper layer. There is no scheduling functionality in the PORT module). |

|  |  |
| --- | --- |
| [BSW00432] Modules should have separate main processing functions for read/receive and write/transmit data path | Not applicable  (Because the PORT driver does not contain any main processing functions). |
| [BSW00433] Calling of main processing functions | Not applicable  (Because for the PORT driver there is no main processing function specified). |
| [BSW00434] The Schedule Module shall provide an API for exclusive areas | Not applicable  (Because it is a non functional requirement. There is no scheduling functionality in the PORT driver) |
| [BSW00435] Header files Structure for the Basic Software Scheduler. | See Section [5.1.2](#_bookmark17) ([PORT131](#_bookmark19) ) |
| [BSW00436] Module Header File Structure for the Basic Software Memory Mapping. | See Section [5.1.2](#_bookmark17) ([PORT131](#_bookmark19)) |
| [BSW00437] NoInit—Area in RAM | Not applicable  (Because this is a requirement for the implementer) |
| [BSW00438] Post build Configuration data Structure See S | ection [10](#_bookmark118). |
| [BSW00439] Declaration of Interrupt Handler and ISR | Not applicable  (Port driver does not have Interrupt or Isr to manage) |
| [BSW00440] Function prototype for callback functions Not a | pplicable  (There aren’t callbacks in the Port Driver |
|  | ) |
| [BSW00442] Debugging Support | [PORT200](#_bookmark66),[PORT201](#_bookmark67) [PORT202](#_bookmark68), [PORT203](#_bookmark69) |

Document: AUTOSAR requirements on SPAL document [8]

b

|  |  |
| --- | --- |
| ***Requirement*** | ***Satisfied by*** |
| [BSW12263] Object code compatible configuration concept | [PORT041](#_bookmark81) and [Port\_ConfigType](#_bookmark72) |
| [BSW12056] Configuration of notification mechanism Not A | pplicable  (Because the PORT driver does not include notification functionality). |
| [BSW12267] Configuration of wakeup sources Not Applica | le  (Because there is no wake-up functionality associated to the PORT driver). |
| [BSW12057] Driver module initialization | [PORT041](#_bookmark81), [PORT042](#_bookmark82), [PORT043](#_bookmark92) |
| [BSW12125] Initialization of hardware resources | [PORT041](#_bookmark81), [PORT042](#_bookmark82) |
| [BSW12163] Driver module deinitialization | [PORT003](#_bookmark95) |
| [BSW12068] MCAL initialization sequence | Not applicable  (Because this requirement describes the initialisation of the whole SPAL layer). |
| [BSW12069] Wake-up notification of ECU State Manager N | ot applicable  (Because the PORT driver has no wake- up functionality). |
| [BSW157] Notification mechanisms of drivers and handlers | Not applicable  (Because there is no notification functionality associated to the PORT driver. |
| [BSW12169] Control of operation mode | Not applicable  (Because there is no set mode functionality in the PORT driver). |
| [BSW12063] Raw value mode | Not applicable  (Because there is no functionality in the PORT driver for the raw value mode). |

|  |  |
| --- | --- |
| [BSW12075] Use of application buffers | Not applicable  (Because there is no random streaming capability) |
| [BSW12129] Resetting of interrupt flags | Not applicable  (Because the interrupt functionality is not part of the PORT driver). |
| [BSW12064] Change of operation mode during running operation | Not applicable  (Because this is a non-functional requirement concerning system design). |
| [BSW12067] Setting of wake-up conditions | Not applicable  (Because the PORT driver has no wake- up conditions). |
| [BSW12448] Behaviour after development error detection | [PORT077](#_bookmark62) |
| [BSW12077] Non-blocking implementation | Not applicable  (Because this is a requirement for the implementer) |
| [BSW12078] Runtime and memory efficiency | Not applicable  (Because this is a requirement for the implementer) |
| [BSW12092] Access to drivers | Not applicable  (Because this is a non-functional requirement concerning the system design). |
| [BSW12265] Configuration data shall be kept constant | Not applicable  (Because this is a requirement for the implementer) |
| [BSW12264] Specification of configuration items Chapter 1 | 0 - Configuration specification. |
| [BSW12461] Responsibility for register initialisation | [PORT113](#_bookmark84),[PORT214](#_bookmark85), [PORT215](#_bookmark87), [PORT217](#_bookmark88), [PORT218](#_bookmark90) |
| [BSW12462] Provide settings for register initialisation. Not | applicable  (Requirement for the implementers) |

|  |  |
| --- | --- |
| [BSW12463] Combine and forward settings for register initialisation. | Not applicable  (Because this is a requirement for a configuration tool). |

Document: Requirements on PORT Driver SRS [9]

|  |  |
| --- | --- |
| ***Requirement*** | ***Satisfied by*** |
| [BSW12001] Configuration of Port Pin Properties | [PORT004](#_bookmark40), [PORT079](#_bookmark41), [PORT072](#_bookmark73) |
| [BSW12302] Configuration of symbolic names | [PORT006](#_bookmark46) |
| [BSW12405] Set port pin direction | [PORT063](#_bookmark101), [PORT086](#_bookmark102), [PORT138](#_bookmark45) |
| [BSW12406] Refresh port direction | [PORT060](#_bookmark105), [PORT061](#_bookmark106) |
| [BSW12300] Configuration of unused port pins | [PORT005](#_bookmark44) |
| [BSW12423] Provide atomicity of port access | [PORT075](#_bookmark43) |

# Functional specification

## General Behaviour

### Background & Rationale

**PORT001:** The PORT Driver module shall initialize the whole port structure of the microcontroller.

Note: Defining the order in which the ports and port pins are configured is the task of the configuration tool.

### Requirements

### Configuration of Port Pin Properties

**PORT004:** The PORT Driver module shall allow the configuration of different functionality for each port and port pin, e.g. ADC, SPI, DIO etc.

The configuration of the port (i.e. whole port or single port pin) is microcontroller dependent.

**PORT079:** The PORT Driver module shall provide additional configurations for the MCU port/port pins:

* + - * + Pin direction (input/output)
        + Pin level initial value
        + Pin direction changeable during runtime (yes/no).
        + Port mode changeable during runtime.

**PORT081:** The PORT Driver module shall provide a number of optional configurations for the MCU ports and port pins (if supported by hardware):

* + - * + Slew rate control
        + Activation of internal pull-ups
        + Input Thresholds
        + Pin driven mode (push-pull / open drain).
        + Type of Readback support (pin level, output register value).

**PORT082:** The PORT Driver module shall not provide the facility to configure pin level inversion. The default value shall be set (i.e. not inverted).

Note: The IO Hardware Abstraction layer shall carry out level inversion.

### Switch port pin direction

**PORT137:** For the port pins configured as changeable using the configuration tool, the PORT driver shall allow the user to change the direction of port pins during runtime.

**PORT138:** If the MCU port control hardware provides an output latch for setting the output level on a port pin, switching the port pin direction shall not alter the level set in this output latch.

### Refresh port direction

**PORT066:** For refreshing of the port on the microcontroller, the PORT driver shall allow the user to refresh the direction of those port pins whose direction is set by configuration and cannot be changed dynamically.

### Configuration of unused Ports and Port Pins

**PORT005:** The PORT Driver module shall configure all ports and port pins that are not used (neither as GPIO nor special purpose IO) to be set to a defined state by the PORT Driver module configuration.

### Configuration of symbolic names

**PORT006:** The user of the PORT Driver module shall configure the symbolic names of the port pins of the MCU.

**PORT207:** These symbolic names for the individual port pins (e.g. PORT\_A\_PIN\_0) shall be defined in the configuration tool.

**PORT076:** The PORT Driver module’s implementer shall define symbolic names in the file Port\_Cfg.h

**PORT208:** The PORT Driver module’s implementer shall publish the symbolic names through the file Port.h

### Atomicity of port access

**PORT075:** The PORT Driver module shall provide atomic access to all ports and port pins.

Note:

An atomic access is a non interruptible access to Microcontroller registers by the use of either atomic instructions or the usage of an exclusive area

(interrupt disabling for example) provided by the basic software scheduler module.

### Version Check

### Background and Rationale

The integration of incompatible files shall be avoided. Minimum implementation is the version check of the header file inside the .c file (version numbers of .c and .h files shall be identical).

### Requirements

The Port module shall avoid the integration of incompatible files by the following pre-processor checks:

### PORT114:

For included (external) header files,

* + - * + <MODULENAME>\_AR\_RELEASE\_MAJOR\_VERSION
        + <MODULENAME>\_AR\_RELEASE\_MINOR\_VERSION shall be verified.

## Error classification

**PORT115:** Values for production code Event Ids are assigned externally by the configuration of the DEM. They are published in the file Dem\_IntErrId.h and included via Dem.h.

By this inclusion the APIs to report errors as well as the required Event Id symbols are included. This specification defines the name of the Event Id symbols which are provided by XML to the DEM configuration tool. The DEM configuration tool assigns ECU dependent values to the Event Id symbols and publishes the symbols in Dem\_IntErrId.h.

**PORT116:** Development error values are of type uint8.

**PORT051:** The following errors and exceptions shall be detectable by the PORT driver depending on its build version (development/production).

|  |  |  |  |
| --- | --- | --- | --- |
| ***Type or error*** | ***Relevance*** | ***Related error code*** | ***Value*** |
| Invalid Port Pin ID requested | Development | PORT\_E\_PARAM\_PIN | 0x0A |
| Port Pin not configured as changeable | Development | PORT\_E\_DIRECTION\_UNCHANGEABLE | 0x0B |
| API Port\_Init service | Development | PORT\_E\_PARAM\_CONFIG | 0x0C |

|  |  |  |  |
| --- | --- | --- | --- |
| called with wrong parameter. |  |  |  |
| API Port\_SetPinMode service called when mode is unchangeable. | Development | PORT\_E\_PARAM\_INVALID\_MODE | 0x0D |
| PORT\_E\_MODE\_UNCHANGEABLE | 0x0E |
| API service called without module initialization | Development | PORT\_E\_UNINIT | 0x0F |
| APIs called with a Null Pointer | Development | PORT\_E\_PARAM\_POINTER | 0x10 |

## Error detection

**PORT100:** The detection of development errors is configurable (ON/OFF) at pre- compile time.

**PORT211:** The switch PortDevErrorDetect (see Chapter 10) shall activate or deactivate the detection of all development errors.

**PORT101:** If the PortDevErrorDetect switch is enabled, API parameter checking is enabled. The detailed description of the detected errors can be found in Chapter 7.2 and Chapter 8.

**PORT139:** The detection of production code errors cannot be switched off.

## Error notification

**PORT037:** Production errors shall be reported to the Diagnostic Event Manager [[Ref.6](#_bookmark14)].

**PORT038:** If it is possible to detect additional errors, because of specific implementation and/or specific hardware properties, then the PORT module’s implementer shall add them to the PORT implementation specification.

The classification and enumeration shall be compatible to the errors listed above [PORT051].

**PORT107:** Detected development errors shall be reported to the Det\_ReportError service of the Development Error Tracer (DET) if the pre-processor switch PortDevErrorDetect is set (see Chapter 10).

## API Parameter checking

**PORT077**: If development error detection is enabled the Port Driver module shall check the function parameters in the order in which they are passed and skip further parameter checking if one check fails.

Example: For the function Port\_SetPinDirection, the first parameter to be passed is the pin ID. This parameter shall identify the relevant port pin of the MCU’s

port. The second parameter passed corresponds to the direction to change on the port pin.

**PORT087**: If development error detection is enabled and the Port Driver module has detected an error, the desired functionality shall be skipped and the requested service shall return without any action.

See table below for a list of the Det errors returned by each function.

|  |  |  |
| --- | --- | --- |
| ***Function*** | ***Error Condition*** | ***Related error value*** |
| Port\_SetPinDirection | Incorrect Port Pin | PORT\_E\_PARAM\_PIN |
|  | ID passed |  |
|  | Port Pin not | PORT\_E\_DIRECTION\_UNCHANGEABLE |
|  | configured as |  |
|  | changeable |  |
| Port\_Init | Port\_Init service | PORT\_E\_PARAM\_CONFIG |
|  | called with wrong |  |
|  | parameter. |  |
| Port\_SetPinMode | Incorrect Port Pin | PORT\_E\_PARAM\_PIN |
|  | ID passed |  |
|  | Port Pin Mode | PORT\_E\_PARAM\_INVALID\_MODE |
|  | passed not valid |  |
|  | Port\_SetPinMode | PORT\_E\_MODE\_UNCHANGEABLE |
|  | service called |  |
|  | when the mode is |  |
|  | unchangeable |  |
| Port\_SetPinDirection, | API service | PORT\_E\_UNINIT |
| Port\_SetPinMode | called prior to |  |
| Port\_GetVersionInfo | module |  |
| Port\_RefreshPortDirection | initialization |  |
| Port\_GetVersionInfo | Api called with a | PORT\_E\_PARAM\_POINTER |
|  | NULL Pointer |  |
|  | Parameter |  |

## Debugging Support

The following requirements deal with the definition of variables and the description of debug information.

PORT200: Each variable that shall be accessible by AUTOSAR Debugging shall be defined as global variable.

PORT201: All type definitions of variables that shall be debugged, shall be accessible by the header file Port.h.

PORT202: The declaration of variables in the header file shall allow to calculate the size of the variables by C-"sizeof".

PORT203: Variables available for debugging shall be described in the respective Basic Software Module Description.

# API specification

## Imported types

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

### PORT129:

|  |  |
| --- | --- |
| ***Module*** | ***Imported Type*** |
| Dem | Dem\_EventIdType |
| Dem\_EventStatusType |
| Std\_Types | Std\_ReturnType |
| Std\_VersionInfoType |

## Type definitions

### Port\_ConfigType

|  |  |  |
| --- | --- | --- |
| ***Name:*** | Port\_ConfigType | |
| ***Type:*** | Structure | |
| ***Range:*** | Hardware Dependent Structure | The contents of the initialization data structure are specific to the microcontroller. |
| ***Description:*** | Type of the external data structure containing the initialization data for this  module. | |

**PORT073:** The type Port\_ConfigType is a type for the external data structure containing the initialization data for the PORT Driver.

Note: The user shall use the symbolic names defined in the configuration tool.

Note: The configuration of each port pin is MCU specific. Therefore, it is not possible to include a complete list of different configurations in this specification.

**PORT072**: A list of possible port configurations for the structure Port\_ConfigType

is given below:

* Pin mode (e.g. DIO, ADC, SPI …) – this port pin configuration is mandatory unless the port pin is configured for DIO.
* Pin direction (input, output) – this port pin configuration is mandatory when the port pin is to be used for DIO.
* Pin level init value (see [PORT055](#_bookmark96)) – this port pin configuration is mandatory when the port pin is used for DIO.
* Pin direction changeable during runtime (STD\_ON/STD\_OFF) – this port pin configuration is MCU dependent.
* Pin mode changeable during runtime (STD\_ON/STD\_OFF) – configuration is MCU dependent.

Optional parameters (if supported by hardware)

* Slew rate control.
* Activation of internal pull-ups.
* Microcontroller specific port pin properties.

### Port\_PinType

|  |  |  |
| --- | --- | --- |
| ***Name:*** | Port\_PinType | |
| ***Type:*** | uint | |
| ***Range:*** | 0 - <number of port pins:> | Shall cover all available port pins. The type should be chosen for the specific MCU platform (best performance). |
| ***Description:*** | Data type for the symbolic name of a port pin. | |

**PORT013:** The type Port\_PinType shall be used for the symbolic name of a Port Pin.

**PORT219:** The type Port\_PinType shall be uint8, uint16 or uint32 based on the specific MCU platform.

Note: The user shall use the symbolic names provided by the configuration tool.

### Port\_PinDirectionType

|  |  |  |  |
| --- | --- | --- | --- |
| ***Name:*** | Port\_PinDirectionType | | |
| ***Type:*** | Enumeration | | |
| ***Range:*** | PORT\_PIN\_IN | Sets port pin as input. |  |
| PORT\_PIN\_OUT | Sets port pin as output. |  |
| ***Description:*** | Possible directions of a port pin. | | |

**PORT046:** The type Port\_PinDirectionType is a type for defining the direction of a Port Pin.

**PORT220:** The type Port\_PinDirectionType shall be of enumeration type having range as PORT\_PIN\_IN and PORT\_PIN\_OUT.

### Port\_PinModeType

***Name:***

Port\_PinModeType

|  |  |  |
| --- | --- | --- |
| ***Type:*** | uint | |
| ***Range:*** | Implementation specific | As several port pin modes shall be configurable on one pin, the range shall be determined by the implementation. |
| ***Description:*** | Different port pin modes. | |

**PORT124:** A port pin shall be configurable with a number of port pin modes (type

Port\_PinModeType).

**PORT212:** The type Port\_PinModeType shall be used with the function call

Port\_SetPinMode (see Section [8.3.5](#_bookmark107)).

**PORT221**: The type Port\_PinModeType shall be uint8, uint16 or uint32.

## Function definitions

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

### Port\_Init PORT140:

|  |  |  |
| --- | --- | --- |
| ***Service name:*** | Port\_Init | |
| ***Syntax:*** | void Port\_Init(  const Port\_ConfigType\* ConfigPtr  ) | |
| ***Service ID[hex]:*** | 0x00 | |
| ***Sync/Async:*** | Synchronous | |
| ***Reentrancy:*** | Non Reentrant | |
| ***Parameters (in):*** | ConfigPtr | Pointer to configuration set. |
| ***Parameters***  ***(inout):*** | None | |
| ***Parameters (out):*** | None | |
| ***Return value:*** | None | |
| ***Description:*** | Initializes the Port Driver module. | |

**PORT041**: The function Port\_Init shall initialize ALL ports and port pins with the configuration set pointed to by the parameter ConfigPtr.

**PORT078**: The Port Driver module’s environment shall call the function Port\_Init

first in order to initialize the port for use.

**PORT213:** If Port\_Init function is not called first, then no operation can occur on the MCU ports and port pins.

**PORT042**: The function Port\_Init shall initialize all configured resources.

The function Port\_Init shall apply the following rules regarding initialisation of controller registers.

* + - * **PORT113**: If the hardware allows for only one usage of the register, the driver module implementing that functionality is responsible for initializing the register.
      * **PORT214:** If the register can affect several hardware modules and if it is an I/O register it shall be initialised by this PORT driver.
      * **PORT215**: If the register can affect several hardware modules and if it is not an I/O register, it shall be initialised by the MCU driver.
      * **PORT217**: One-time writable registers that require initialisation directly after reset shall be initialised by the startup code.
      * **PORT218:** All the other registers not mentioned before, shall be initialised by the start-up code.

**PORT043**: The function Port\_Init shall avoid glitches and spikes on the affected port pins.

**PORT071**: The Port Driver module’s environment shall call the function Port\_Init

after a reset in order to reconfigure the ports and port pins of the MCU.

**PORT002:** The function Port\_Init shall initialize all variables used by the PORT driver module to an initial state.

**PORT003:** The Port Driver module’s environment may also uses the function Port\_Init to initialize the driver software and reinitialize the ports and port pins to another configured state depending on the configuration set passed to this function.

Note: In some cases, MCU port control hardware provides an output latch for setting the output level on a port pin that may be used as a DIO port pin.

**PORT055:** The function Port\_Init shall set the port pin output latch to a default level (defined during configuration) before setting the port pin direction to output.

Requirement PORT055 ensures that the default level is immediately output on the port pin when it is set to an output port pin.

Example: On some MCU’s, after a power-on-reset, a DIO configurable port pin will be configured as an input pin. If the required configuration of the port pin is an output pin, then the function Port\_Init shall ensure that the default level is set before switching the functionality of the port pin from input to output.

**PORT105:** If development error detection for the Port Driver module is enabled: In case the function Port\_Init is called with a NULL ConfigPtr and if a variant containing postbuild multiple selectable configuration parameters is used (VARIANT-POST- BUILD), the function Port\_Init shall raise the development error PORT\_E\_PARAM\_CONFIG and return without any action.

 Specification of Port Driver

V3.2.0

R4.0 Rev 2

**PORT121:** The function Port\_Init shall always have a pointer as a parameter, even though for the configuration variant VARIANT-PRE-COMPILE, no configuration set shall be given. In this case, the Port Driver module’s environment shall pass a NULL pointer to the function Port\_Init.

The Port Driver module’s environment shall not call the function Port\_Init during a running operation. This shall only apply if there is more than one caller of the PORT module.

Configuration of Port\_Init: All port pins and their functions, and alternate functions shall be configured by the configuration tool.

### Port\_SetPinDirection PORT141:

|  |  |  |
| --- | --- | --- |
| ***Service name:*** | Port\_SetPinDirection | |
| ***Syntax:*** | void Port\_SetPinDirection( Port\_PinType Pin, Port\_PinDirectionType Direction  ) | |
| ***Service ID[hex]:*** | 0x01 | |
| ***Sync/Async:*** | Synchronous | |
| ***Reentrancy:*** | Reentrant | |
| ***Parameters (in):*** | Pin | Port Pin ID number |
| Direction Port Pin Direction |  |
| ***Parameters (inout):*** | None | |
| ***Parameters (out):*** | None | |
| ***Return value:*** | None | |
| ***Description:*** | Sets the port pin direction | |

**PORT063:** The function Port\_SetPinDirection shall set the port pin direction during runtime.

**PORT054:** The function Port\_SetPinDirection shall be re-entrant if accessing different pins independent of a port.

**PORT086:** The function Port\_SetPinDirection shall only be available to the user if the pre-compile parameter PortSetPinDirectionApi is set to TRUE. If set to FALSE, the function Port\_SetPinDirection is not available. (see also section [8.3.2](#_bookmark70).

Configuration of Port\_SetPinDirection: All ports and port pins shall be configured by the configuration tool. See [PORT117](#_bookmark21).



### Port\_RefreshPortDirection PORT142:

Specification of Port Driver

V3.2.0 R4.0 Rev 2

|  |  |
| --- | --- |
| ***Service name:*** | Port\_RefreshPortDirection |
| ***Syntax:*** | void Port\_RefreshPortDirection( void  ) |
| ***Service ID[hex]:*** | 0x02 |
| ***Sync/Async:*** | Synchronous |
| ***Reentrancy:*** | Non Reentrant |
| ***Parameters (in):*** | None |
| ***Parameters (inout):*** | None |
| ***Parameters (out):*** | None |
| ***Return value:*** | None |
| ***Description:*** | Refreshes port direction. |

**PORT060:** The function Port\_RefreshPortDirection shall refresh the direction of all configured ports to the configured direction (PortPinDirection).

**PORT061:** The function Port\_RefreshPortDirection shall exclude those port pins from refreshing that are configured as ‘pin direction changeable during runtime‘.

The configuration tool shall provide names for each configured port pin.

### Port\_GetVersionInfo PORT143:

|  |  |  |
| --- | --- | --- |
| ***Service name:*** | Port\_GetVersionInfo | |
| ***Syntax:*** | void Port\_GetVersionInfo( Std\_VersionInfoType\* versioninfo  ) | |
| ***Service ID[hex]:*** | 0x03 | |
| ***Sync/Async:*** | Synchronous | |
| ***Reentrancy:*** | Non 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:*** | Returns the version information of this module. | |

**PORT102:** The function Port\_GetVersionInfo shall return the version information of this module. The version information includes:

* Module Id
* Vendor Id
* Vendor specific version numbers (BSW00407).

**PORT103:** The function Port\_GetVersionInfo shall be pre compile time configurable On/Off by the configuration parameter PortVersionInfoApi.

**PORT144:** If source code for caller and callee of Port\_GetVersionInfo is available, the PORT Driver module should realize Port\_GetVersionInfo as a macro, defined in the module’s header file.

**PORT225:** if Det is enabled, the parameter versioninfo shall be

checked for being NULL. The error PORT\_E\_PARAM\_POINTER shall be reported in case the value is a NULL pointer.

### Port\_SetPinMode PORT145:

|  |  |  |
| --- | --- | --- |
| ***Service name:*** | Port\_SetPinMode | |
| ***Syntax:*** | void Port\_SetPinMode( Port\_PinType Pin, Port\_PinModeType Mode  ) | |
| ***Service ID[hex]:*** | 0x04 | |
| ***Sync/Async:*** | Synchronous | |
| ***Reentrancy:*** | Reentrant | |
| ***Parameters (in):*** | Pin | Port Pin ID number |
| Mode New | Port Pin mode to be set on port pin. |
| ***Parameters (inout):*** | None | |
| ***Parameters (out):*** | None | |
| ***Return value:*** | None | |
| ***Description:*** | Sets the port pin mode. | |

**PORT125:** The function Port\_SetPinMode shall set the port pin mode of the referenced pin during runtime.

**PORT128:** The function Port\_SetPinMode shall be re-entrant if accessing different pins, independent of a port.

**PORT223:** If Det is enabled, the function Port\_SetPinMode shall return PORT\_E\_MODE\_UNCHANGEABLE and return without any action, if the parameter PortPinModeChangeable is set to FALSE.

Configuration of Port\_SetPinMode: All ports and port pins shall be configured by the configuration tool. See [PORT117](#_bookmark21).

## Call-back notifications

There are no callback notifications from the PORT driver. The callback notifications are implemented in another module (ICU Driver and/or complex drivers).

## Scheduled functions

There are no scheduled functions within the PORT Driver.

## Expected Interfaces

In this chapter, all interfaces required from other modules are listed.

### Mandatory Interfaces

None

### Optional Interfaces

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

### PORT146:

|  |  |
| --- | --- |
| ***API function*** | ***Description*** |
| Dem\_ReportErrorStatus | Queues the reported events from the BSW modules (API is only used by BSW modules). The interface has an asynchronous behavior, because the processing of the event is done within the Dem main function. |
| Det\_ReportError | Service to report development errors. |

### Configurable Interfaces

None

# Sequence diagrams

## Overall Configuration of Ports



Generic

Elements::User

«module»

Port::Port

Port\_Init(const

Port\_ConfigType\*)

Port\_Init()

All ports are

initialized

## Set the direction of a Port Pin

Generic

Elements::User



«module» Port::Port

Port\_SetPinDirection(Port\_PinType,

Port\_PinDirectionType)

Port\_SetPinDirection()

The port pin

direction is set

## Refresh the direction of all Port Pins



Generic

Elements::User

«module» Port::Port

Port\_RefreshPortDirection()

Port\_RefreshPortDirection()

The Port pin

direction is refreshed.

## Change the mode of a Port Pin



**sd Port\_SetPinMode()**

Generic

Elements::User

Port::Port

Port\_SetPinMode(Pin,Mode)

Port\_SetPinMode

The port pin mode

is set

Description:

Set the mode of a port pin

Comments:

# Configuration specification

In general, this chapter defines configuration parameters and their clustering into containers. In order to support the specification Chapter [10.1](#_bookmark110) describes fundamentals. It also specifies a template (table) you shall use for the parameter specification. We intend to leave Chapter [10.1](#_bookmark110) in the specification to guarantee comprehension.

Chapter [10.2](#_bookmark119) specifies the structure (containers) and the parameters of the module PORT

Chapter [10.3](#_bookmark137) specifies published information of the module PORT.

## How to read this chapter

In addition to this section, it is highly recommended to read the documents:

* + - AUTOSAR Layered Software Architecture [[2](#_bookmark14)].
    - AUTOSAR ECU Configuration Specification [[5](#_bookmark14)]

This document describes the AUTOSAR configuration methodology and the AUTOSAR configuration metamodel in detail.

The following is only a short survey of the topic and it will not replace the ECU Configuration Specification document.

### Configuration and configuration parameters

Configuration parameters define the variability of the generic part(s) of an implementation of a module. This means that only generic or configurable module implementation can be adapted to the environment (software/hardware) in use during system and/or ECU configuration.

The configuration of parameters can be achieved at different times during the software process: before compile time, before link time or after build time. In the following, the term “configuration class” (of a parameter) shall be used in order to refer to a specific configuration point in time.

### Containers

Containers structure the set of configuration parameters. This means:

- *all* configuration parameters are kept in containers.

(sub-) containers can reference (sub-) containers. It is possible to assign a multiplicity to these references. The multiplicity then defines the possible number of instances of the contained parameters.

## Containers and configuration parameters

The following chapters summarize all configuration parameters. The detailed meanings of the parameters describe Chapters [7](#_bookmark22) and Chapter [8](#_bookmark47).

### Variants

**PORT147:** VARIANT-PRE-COMPILE.

Only parameters with "Pre-compile time" configuration are allowed in this variant.

The intention of this variant is to optimize the parameters configuration for a source code delivery.

**PORT148:** VARIANT-POST-BUILD.

Parameters with "Pre-compile time", "Link time" and "Post-build time" are allowed in this variant.

The intention of this variant is to optimize the parameters configuration for a re- loadable binary.

### Port

|  |  |
| --- | --- |
| ***Module Name*** | *Port* |
| ***Module Description*** | Configuration of the Port module. |

|  |  |  |
| --- | --- | --- |
| ***Included Containers*** | | |
| ***Container Name*** | ***Multiplicity*** | ***Scope / Dependency*** |
| PortConfigSet | 1 | This container is the base of a multiple configuration set |
| PortGeneral | 1 | Module wide configuration parameters of the PORT driver. |

### PortContainer

|  |  |
| --- | --- |
| ***SWS Item*** | **PORT122\_Conf :** |
| ***Container Name*** | PortContainer |
| ***Description*** | Container collecting the PortPins. |
| ***Configuration Parameters*** | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **PORT124\_Conf :** | | |
| ***Name*** | PortNumberOfPortPins | | |
| ***Description*** | The number of specified PortPins in this PortContainer. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EcucIntegerParamDef | | |
| ***Range*** | 1 .. 65535 |  | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** |  | | |

|  |  |  |
| --- | --- | --- |
| ***Included Containers*** | | |
| ***Container Name*** | ***Multiplicity*** | ***Scope / Dependency*** |
| PortPin | 1..\* | Configuration of the individual port pins. |

### PortGeneral

|  |  |
| --- | --- |
| ***SWS Item*** | **PORT117\_Conf :** |
| ***Container Name*** | PortGeneral{PORT General configuration} |
| ***Description*** | Module wide configuration parameters of the PORT driver. |
| ***Configuration Parameters*** | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **PORT123\_Conf :** | | |
| ***Name*** | PortDevErrorDetect {PORT\_DEV\_ERROR\_DETECT} | | |
| ***Description*** | Switches the Development Error Detection and Notification on or off. true: Enabled. false: Disabled. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EcucBooleanParamDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: Module | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **PORT131\_Conf :** | | |
| ***Name*** | PortSetPinDirectionApi {PORT\_SET\_PIN\_DIRECTION\_API} | | |
| ***Description*** | Pre-processor switch to enable / disable the use of the function Port\_SetPinDirection(). TRUE: Enabled - Function Port\_SetPinDirection() is available. FALSE: Disabled - Function Port\_SetPinDirection() is not available. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EcucBooleanParamDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: Module | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **PORT132\_Conf :** | | |
| ***Name*** | PortSetPinModeApi {PORT\_SET\_PIN\_MODE\_API} | | |
| ***Description*** | Pre-processor switch to enable / disable the use of the function Port\_SetPinMode(). true: Enabled - Function Port\_SetPinMode() is available. false: Disabled - Function Port\_SetPinMode() is not available. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EcucBooleanParamDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: Module | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **PORT133\_Conf :** | | |
| ***Name*** | PortVersionInfoApi {PORT\_VERSION\_INFO\_API} | | |
| ***Description*** | Pre-processor switch to enable / disable the API to read out the modules version information. true: Version info API enabled. false: Version info API disabled. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EcucBooleanParamDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: Module | | |

***No Included Containers***

The top level Port Driver container holds parameters that apply to the PORT configuration.

### PortPin

|  |  |
| --- | --- |
| ***SWS Item*** | **PORT118\_Conf :** |
| ***Container Name*** | PortPin |
| ***Description*** | Configuration of the individual port pins. |
| ***Configuration Parameters*** | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **PORT125\_Conf :** | | |
| ***Name*** | PortPinDirection {PORT\_PIN\_DIRECTION} | | |
| ***Description*** | The initial direction of the pin (IN or OUT). If the direction is not changeable, the value configured here is fixed. The direction must match the pin mode.  E.g. a pin used for an ADC must be configured to be an in port. Implementation Type: Port\_PinDirectionType | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EcucEnumerationParamDef | | |
| ***Range*** | PORT\_PIN\_IN | Port Pin direction set as input | |
| PORT\_PIN\_OUT Port Pin direction set a | s output | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | -- |  |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Module | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **PORT126\_Conf :** | | |
| ***Name*** | PortPinDirectionChangeable {PORT\_PIN\_DIRECTION\_CHANGEABLE} | | |
| ***Description*** | Parameter to indicate if the direction is changeable on a port pin during runtime. true: Port Pin direction changeable enabled. false: Port Pin direction changeable disabled. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EcucBooleanParamDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | -- |  |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Module | | |

|  |  |
| --- | --- |
| ***SWS Item*** | **PORT127\_Conf :** |
| ***Name*** | PortPinId |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Description*** | Pin Id of the port pin. This value will be assigned to the symbolic name derived from the port pin container short name. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EcucIntegerParamDef (Symbolic Name generated for this parameter) | | |
| ***Range*** | 1 .. 65535 |  | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: Module | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **PORT128\_Conf :** | | |
| ***Name*** | PortPinInitialMode {PORT\_PIN\_INITIAL\_MODE} | | |
| ***Description*** | Port pin mode from mode list for use with Port\_Init() function. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EcucEnumerationParamDef | | |
| ***Range*** | PORT\_PIN\_MODE\_ADC | Port Pin used by ADC | |
| PORT\_PIN\_MODE\_CAN | Port Pin used for CAN | |
| PORT\_PIN\_MODE\_DIO | Port Pin configured for DIO. It shall be used under control of the DIO driver. | |
| PORT\_PIN\_MODE\_DIO\_GPT | Port Pin configured for DIO. It shall be used under control of the general purpose timer driver. | |
| PORT\_PIN\_MODE\_DIO\_WDG | Port Pin configured for DIO. It shall be used under control of the watchdog driver. | |
| PORT\_PIN\_MODE\_FLEXRAY | Port Pin used for FlexRay | |
| PORT\_PIN\_MODE\_ICU | Port Pin used by ICU | |
| PORT\_PIN\_MODE\_LIN | Port Pin used for LIN | |
| PORT\_PIN\_MODE\_MEM | Port Pin used for external memory under control of a memory driver. | |
| PORT\_PIN\_MODE\_PWM | Port Pin used by PWM | |
| PORT\_PIN\_MODE\_SPI | Port Pin used by SPI | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | -- |  |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Module | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **PORT129\_Conf :** | | |
| ***Name*** | PortPinLevelValue {PORT\_PIN\_LEVEL\_VALUE} | | |
| ***Description*** | Port Pin Level value from Port pin list. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EcucEnumerationParamDef | | |
| ***Range*** | PORT\_PIN\_LEVEL\_HIGH | Port Pin level is High | |
| PORT\_PIN\_LEVEL\_LOW Port Pin level is L | OW | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | -- |  |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Module | | |

|  |  |
| --- | --- |
| ***SWS Item*** | **PORT130\_Conf :** |
| ***Name*** | PortPinMode {PORT\_PIN\_MODE} |
| ***Description*** | Port pin mode from mode list. Note that more than one mode is allowed by |

|  |  |  |  |
| --- | --- | --- | --- |
|  | default. That way it is e.g. possible to combine DIO with another mode such as ICU. More HW specific literals may be added by module implementor. E.g. SPI may be further broken down into SpiCsHw, SpiCsSw, etc. if necessary. | | |
| ***Multiplicity*** | 1..\* | | |
| ***Type*** | EcucEnumerationParamDef | | |
| ***Range*** | PORT\_PIN\_MODE\_ADC | Port Pin used by ADC | |
| PORT\_PIN\_MODE\_CAN | Port Pin used for CAN | |
| PORT\_PIN\_MODE\_DIO | Port Pin configured for DIO. It shall be used under control of the DIO driver. | |
| PORT\_PIN\_MODE\_DIO\_GPT | Port Pin configured for DIO. It shall be used under control of the general purpose timer driver. | |
| PORT\_PIN\_MODE\_DIO\_WDG | Port Pin configured for DIO. It shall be used under control of the watchdog driver. | |
| PORT\_PIN\_MODE\_FLEXRAY | Port Pin used for FlexRay | |
| PORT\_PIN\_MODE\_ICU | Port Pin used by ICU | |
| PORT\_PIN\_MODE\_LIN | Port Pin used for LIN | |
| PORT\_PIN\_MODE\_MEM | Port Pin used for external memory under control of a memory driver. | |
| PORT\_PIN\_MODE\_PWM | Port Pin used by PWM | |
| PORT\_PIN\_MODE\_SPI | Port Pin used by SPI | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | -- |  |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Module | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **PORT134\_Conf :** | | |
| ***Name*** | PortPinModeChangeable {PORT\_PIN\_MODE\_CHANGEABLE} | | |
| ***Description*** | Parameter to indicate if the mode is changeable on a port pin during runtime. True: Port Pin mode changeable allowed. False: Port Pin mode changeable not permitted. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EcucBooleanParamDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | -- |  |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Module | | |

***No Included Containers***

### PortConfigSet

|  |  |
| --- | --- |
| ***SWS Item*** | **PORT121\_Conf :** |
| ***Container Name*** | PortConfigSet [Multi Config Container] |
| ***Description*** | This container is the base of a multiple configuration set |
| ***Configuration Parameters*** | |

|  |  |  |
| --- | --- | --- |
| ***Included Containers*** | | |
| ***Container Name*** | ***Multiplicity*** | ***Scope / Dependency*** |

|  |  |  |
| --- | --- | --- |
| PortContainer | 1..\* | Container collecting the PortPins. |

## 10.3 Published Information

[PORT001\_PI] The standardized common published parameters as required by BSW00402 in the General Requirements on Basic Software Modules [[3]](#_bookmark5) shall be published within the header file of this module and need to be provided in the BSW Module Description. The according module abbreviation can be found in the List of Basic Software Modules [[1]](#_bookmark4).

Additional module-specific published parameters are listed below if applicable.

# Changes to Release 3.0

## Deleted SWS Items

|  |  |
| --- | --- |
| ***SWS Item*** | ***Rationale*** |
| Port031 | Redundant, it’s already covered by Port051 |

## Replaced SWS Items

|  |  |  |
| --- | --- | --- |
| ***SWS Item*** | ***replaced by SWS Item*** | ***Rationale*** |
| Port006 | Port006,Port207 | Made atomic |
| Port076 | Port076,Port208 | // |
| Port078 | Port078, Port213 | // |
| Port100 | Port100,Port211 | // |
| Port113 | Port113, Port214, Port215, Port217, Port218 | // |
| Port114 | Port114, Port210 | // |
| Port124 | Port124, Port212 | // |

## Changed SWS Items

|  |  |
| --- | --- |
| ***SWS Item*** | ***Rationale*** |
| Port004 | Made SWS item atomic |
| Port082 | Rephrased |
| Port086 | // |
| Port108 | Made SWS Item atomic |
| Port114 | Rephrased |
| Port130 | CTPSEC issues |
| Port131 | // |
| Port132 | // |
| Port133 | Made SWS Item atomic |
| Port147 | Rephrased |
| Port148 | Rephrased |
| Port038 | // |
| Port124\_Conf, Port127\_Conf | Min and Max values in the Range field changed. |

## Added SWS Items

|  |  |
| --- | --- |
| ***SWS Item*** | ***Rationale*** |
| Port200 | Debugging Concept |
| Port201 | // |
| Port202 | // |
| Port203 | // |
| Port204 | Inclusion files modification - CTSPEC issues |
| Port205 | // |
| Port219 | Data Types changes - CTSPEC issues |
| Port220 | // |
| Port221 | // |

|  |  |
| --- | --- |
| Port222 | ID added in the published information – CTSPEC issues |
| Port223, Port134\_Conf | Addition of a new parameter to enable/disable pin mode changing at run time |
| Port225 | Port\_GetVersionInfo parameter checking. |
| PORT001\_PI | Rework of Published Information |

# Changes to Release 4.0 Rev 1

## Deleted SWS Items

|  |  |
| --- | --- |
| ***SWS Item*** | ***Rationale*** |
| Port210 | Due to a change to the General Requirement BSW004. (Autosar does not specify any version check for the internal files but this is completely left to the implementer) |
| Port132 | Removed Reference to Port\_Irq.c (Bugzilla |

## Replaced SWS Items

|  |  |  |
| --- | --- | --- |
| ***SWS Item*** | ***replaced by SWS Item*** | ***Rationale*** |
| Port222 | Port001 PI | Rework of the published information by the AUTOSAR Administration |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Changed SWS Items

|  |  |
| --- | --- |
| ***SWS Item*** | ***Rationale*** |
| Port075 | Made Requirement more clear and ready to be tested. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## Added SWS Items

|  |  |
| --- | --- |
| ***SWS Item*** | ***Rationale*** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |