Document Title	Requirements on Port Driver
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	196

Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R22-11

Document Change History			
Date	**Release**	**Changed by**	**Description**
2022-11-24	R22-11	AUTOSAR Release Management	• Editorial Changes
2021-11-25	R21-11	AUTOSAR Release Management	• Editorial Changes
2020-11-30	R20-11	AUTOSAR Release Management	• No content changes
2019-11-28	R19-11	AUTOSAR Release Management	• No content changes • Changed Document Status from Final to published
2018-10-31	4.4.0	AUTOSAR Release Management	• Editorial changes
2017-12-08	4.3.1	AUTOSAR Release Management	• minor corrections clarifications editorial changes; For details please refer to the ChangeDocumentation
2016-11-30	4.3.0	AUTOSAR Release Management	• Editorial changes
2014-10-31	4.2.1	AUTOSAR Release Management	• Editorial changes
2013-03-15	4.1.11	AUTOSAR Administration	• [TPS_STDT_00078] formatting • Traceability of BSWAndRTE_Features

2010-09-30	3.1.5	AUTOSAR Administration	• Legal disclaimer revised
2008-08-13	3.1.1	AUTOSAR Administration	• Legal disclaimer revised
2007-12-21	3.0.1	AUTOSAR Administration	• Document meta information extended • Small layout adaptations made
2007-01-24	2.1.15	AUTOSAR Administration	• "Advice for users" revised • "Revision Information" added
2006-11-28	2.1	AUTOSAR Administration	• Legal disclaimer revised
2006-05-16	2.0	AUTOSAR Administration	• Release as a separate document. The SRS SPAL V1.0.0 has been split into 12 independent documents for Release 2.0 • minor changes regarding formal issues
2005-05-31	1.0	AUTOSAR Administration	• Initial release as a part of the SRS SPAL V1.0.0

Disclaimer

Contents

1 Scope of document

This document specifies requirements on the module PORT Driver.

Constraints

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

2 How to Read this Document

Each requirement has its unique identifier starting with the prefix "BSW" (for "Basic Software"). For any review annotations, remarks or questions please refer to this unique ID rather than chapter or page numbers!

2.1 Conventions used

- The representation of requirements in AUTOSAR documents follows the table specified in [TPS_STDT_00078], see [1].
- In requirements, the following specific semantics are used

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as follows.

Note that the requirement level of the document in which they are used modifies the force of these words.

- MUST: This word, or the adjective "LEGALLY REQUIRED", means that the definition is an absolute requirement of the specification due to legal issues.
- MUST NOT: This phrase, or the phrase "MUST NOT", means that the definition is an absolute prohibition of the specification due to legal issues.
- SHALL: This phrase, or the adjective "REQUIRED", means that the definition is an absolute requirement of the specification.
- SHALL NOT: This phrase means that the definition is an absolute prohibition of the specification.
- SHOULD: This word, or the adjective "RECOMMENDED", means that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
- SHOULD NOT: This phrase, or the phrase "NOT RECOMMENDED", means that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
- MAY: This word, or the adjective "OPTIONAL", means that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item.

An implementation, which does not include a particular option, SHALL be prepared to interoperate with another implementation, which does include the option, though perhaps with reduced functionality. In the same vein an implementation, which does include a particular option, SHALL be prepared to interoperate with another implementation, which does not include the option (except, of course, for the feature the option provides.)

2.2 Requirements structure

Each module specific chapter contains a short functional description of the Basic Software Module. Requirements of the same kind within each chapter are grouped under the following headlines (where applicable):

Functional Requirements

- Configuration (which elements of the module need to be configurable)
- Initialisation
- Normal Operation
- Shutdown Operation
- Fault Operation
- ...

Non-Functional Requirements

- Timing Requirements
- Resource Usage
- Usability
- Output for other WPs (e.g. Description Templates, Tooling,...)
- ...

3 Acronyms and abbreviations

Acronyms and abbreviations that have a local scope are not contained in the AUTOSAR glossary [2]. These must appear in a local glossary.

Acronym:	Description:
ADC	Analogue to Digital Converter
DIO	Digital Input Output
ICU	Input Capture Unit
MCAL	Microconroller Abstraction Layer
MCU	Microcontroller Unit
OS	Operating System
PWM	Pulse Width Modulation
SCI	Serial Communication Interface
SPAL	The name of this working group (Standard Peripheral Abstraction Layer)
SPI	Serial Peripheral Interface
WP	Work Package

Table 3.1: Acronyms used in the scope of this Document

Abbreviation:	Description:
STD	Standard
REQ	Requirement
UNINIT	Uninitialized (= not initialized)

Table 3.2: Abbreviations used in the scope of this Document

As this is a document from professionals for professionals, all other terms are expected to be known.

4 Requirement Specification

This chapter describes all requirements driving the work to define the Port Driver.

4.1 Functional Overview

This module initializes the whole port structure of the microcontroller. Many ports and port pins can be assigned to various functionalities like e.g.

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

For this reason there has to be an overall configuration and initialization of this port structure. The configuration and usage of those port pins is microcontroller and ECU dependent.

The following expressions are used within the Port driver:

Expression:	Explanation:
Physical Level (Input):	Two states possible: LOW/HIGH
Physical Level (Output):	three states possible: LOW/HIGH/High Impedance
Logical Level:	This level is seen within the software: TRUE/FALSE

4.2 Functional Requirements

4.2.1 Configuration

[SRS_Port_12001] The Port driver shall allow the static configuration of the following options for each port ⌈

Description:	The Port driver shall allow the static configuration of the following options for each port. The granularity of configuration (whole port or single port pin) is microcontroller dependent. Mandatory parameters: • pin usage (e.g. DIO, ADC, SPI, ...) • pin direction (input, output) • pin level init value • pin direction changeable during runtime (yes/no) Optional parameters (only if supported by hardware): • activation of internal pull-ups/pull-downs • slew rate control • input thresholds • pin driver mode (push-pull/open drain) • further microcontroller specific properties Level inversion features shall not be configurable, but be set to the default value (not inverted). Level inversion is task of the I/O Hardware Abstraction.
Rationale:	Basic configuration; Pin direction changeable during runtime: this is information necessary for port refreshing and runtime direction change, see [SRS_Port_12405] Set port pin direction and [SRS_Port_12406] Refresh port direction.
Use Case:	–
Dependencies:	–
Supporting Material:	–

⌋*(RS_BRF_01864)*

[SRS_Port_12302] The port driver shall allow the static configuration of the port pin names ⌈

Description:	The port driver shall allow the static configuration of the following symbolic names: • port pin names

▽

△

Rationale:	Provide human readable symbolic names for microcontroller ports and port pins.
Use Case:	Examples: • PORT_A_PIN_0
Dependencies:	–
Supporting Material:	–

⌋(*RS_BRF_01864*)

4.2.2 Normal Operation

[SRS_Port_12405] The Port driver shall provide a service for setting the direction of port pins during runtime ⌈

Description:	The Port driver shall provide a service for setting the direction of port pins during runtime. The Port driver shall only allow to change the direction of those port pins that are configured as changeable.
Rationale:	–
Use Case:	Single wire bidirectional communication with ASICs.
Dependencies:	[SRS_Port_12001] Configuration of port pin properties
Supporting Material:	–

⌋(*RS_BRF_01056*, *RS_BRF_01864*)

[SRS_Port_12406] The Port driver shall provide a service to refresh the direction of all configured ports ⌈

Description:	The Port driver shall provide a service to refresh the direction of all configured ports to the configured direction. The Port driver shall exclude those port pins from refreshing that are configured as "pin direction changeable during runtime".
Rationale:	Make system more robust against EMC and application software bugs (port data direction register corruption).
Use Case:	–
Dependencies:	[SRS_Port_12001] Configuration of port pin properties
Supporting Material:	–

⌋(*RS_BRF_01056*, *RS_BRF_01864*)

4.3 Non-Functional Requirements

[SRS_Port_12423] All re-entrant functions of the Port Driver shall perform port register access actions in an atomic way ⌈

Description:	All re-entrant functions of the Port Driver shall perform port register access actions in an atomic way.
Rationale:	Avoid data integrity problems within concurrent access of Port Driver API functions.
Use Case:	A specific microcontroller (or a specific compiler) does not provide atomic access to single port pins. For that reason, the implementation has to use read-modify-write operations on the whole port. Concurrent access to pins of the same port will lead to data integrity problems if concurrent access it not blocked.
Dependencies:	–
Supporting Material:	–

⌋*(RS_BRF_01864)*

4.3.1 Process Requirements

[SRS_Port_12300] Ports and port pins that are not used shall be set to a defined state ⌈

Description:	Ports and port pins that are not used (neither as general purpose I/O nor as special purpose I/O) shall be set to a defined state by the Port module configuration.
Rationale:	Ensure that all ports and port pins are in a defined state.
Use Case:	–
Dependencies:	–
Supporting Material:	–

⌋*(RS_BRF_01864)*

5 Requirements traceability

The following table references the features specified in [3] and links to the fulfillments of these.

Requirement	Description	Satisfied by
[RS_BRF_01056]	AUTOSAR BSW modules shall provide standardized interfaces	[SRS_Port_12405] [SRS_Port_12406]
[RS_BRF_01864]	AUTOSAR microcontroller abstraction shall provide mapping of I/O signals to digital I/O ports	[SRS_Port_12001] [SRS_Port_12300] [SRS_Port_12302] [SRS_Port_12405] [SRS_Port_12406] [SRS_Port_12423]

Table 5.1: RequirementsTracing

6 References

References

[1] Standardization Template
AUTOSAR_TPS_StandardizationTemplate

[2] Glossary
AUTOSAR_TR_Glossary

[3] Requirements on AUTOSAR Features
AUTOSAR_RS_Features