Document Title	Requirements on PORT Driver
Document Owner	AUTOSAR GbR
Document Responsibility	AUTOSAR GbR
Document Version	2.0.2
Document Status	Final
Part of Release	2.1
Revision	0014

Document Change History

Date	Version	Changed by	Change Description
24.01.2007	2.0.2	AUTOSAR Administration	• "Advice for users" revised • "Revision Information" added
28.11.2006	2.0.1	AUTOSAR Administration	Legal disclaimer revised
05.04.2006	2.0.0	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
02.06.2005	1.0.0	AUTOSAR Administration	Initial release as a part of the SRS SPAL V1.0.0

Page left intentionally blank

Disclaimer

Advice to users of AUTOSAR Specification Documents:

AUTOSAR Specification Documents may contain exemplary items (exemplary reference models, "use cases", and/or references to exemplary technical solutions, devices, processes or software).
Any such exemplary items are contained in the Specification Documents for illustration purposes only, and they themselves are not part of the AUTOSAR Standard. Neither their presence in such Specification Documents, nor any later documentation of AUTOSAR conformance of products actually implementing such exemplary items, imply that intellectual property rights covering such exemplary items are licensed under the same rules as applicable to the AUTOSAR Standard.

Table of Content

1 Scope of this 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 Requirements Guidelines

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

2.1 Conventions used

In requirements, the following specific semantics are used (taken from Request for Comment RFC 2119 from the Internet Engineering Task Force IETF)

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119. Note that the requirement level of the document in which they are used modifies the force of these words.

- MUST: This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.
- MUST NOT: This phrase, or the phrase „SHALL NOT“, means that the definition is an absolute prohibition of the specification.
- SHOULD: This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
- SHOULD NOT: This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
- MAY: This word, or the adjective „OPTIONAL“, means that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation, which does not include a particular option, MUST be prepared to interoperate with another implementation, which does include the option, though perhaps with reduced functionality. In the same vein an implementation, which does include a particular option, MUST be prepared to interoperate with another implementation, which does not include the option (except, of course, for the feature the option provides.)

2.2 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. These must appear in a local glossary.

Acronym:	***Description:***
ADC	Analogue to Digital Converter
DIO	Digital Input Output
HIS	Herstellerinitiative Software
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

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

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

4 Requirement Specification

4.1 Port Driver

4.1.1 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

4.1.2 Functional Requirements

4.1.2.1 Configuration

4.1.2.1.1 [BSW12001] Configuration of port pin properties

Initiator:	WP4.2.2.1.12
Date:	25.11.2004
Short Description:	Configuration of port pin properties
Type:	Changed (configuration of pin direction changeability)
Importance:	High
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 [BSW12405] Set port pin direction and [BSW12406] Refresh port direction.
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.2.1.2 [BSW12302] Configuration of symbolic names

Initiator:	WP4.2.2.1.12
Date:	28.09.2004
Short Description:	Configuration of symbolic names
Type:	New
Importance:	High
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:	--
Conflicts:	--
Supporting Material:	--

4.1.2.2 Normal Operation

4.1.2.2.1 [BSW12405] Set port pin direction

Initiator:	WP4.2.2.1.12
Date:	25.11.2004
Short Description:li	Set port pin direction
Type:	Changed (04.04.2005 switch direction → set direction)
Importance:	High
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:	[BSW12001] Configuration of port pin properties
Conflicts:	--
Supporting Material:	--

4.1.2.2.2 [BSW12406] Refresh port direction

Initiator:	WP4.2.2.1.12
Date:	25.11.2004
Short Description:	Refresh port direction
Type:	New
Importance:	High
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:	[BSW12001] Configuration of port pin properties
Conflicts:	--
Supporting Material:	--

4.1.3 Non-Functional Requirements (Qualities)

4.1.3.1 [BSW12423] Provide atomicity of port access

Initiator:	WP4.2.2.1.12
Date:	07.12.2004
Short Description:	Provide atomicity of port access
Type:	New

Importance:	High
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:	--
Conflicts:	--
Supporting Material:	--

4.1.4 Process Requirements

4.1.4.1 [BSW12300] Configuration of unused port pins

Initiator:	CAS
Date:	27.09.2004
Short Description:	Configuration of unused port pins
Type:	Changed
Importance:	High
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:	--
Conflicts:	--
Supporting Material:	--

5 References

5.1 Deliverables of AUTOSAR

[DOC_LAYERED_ARCH] Layered Software Architecture,
https://svn.autosar.org/repos/10Releases/
AUTOSAR_LayeredSoftwareArchitecture.pdf

[AUTOSAR_GLOSSARY] Glossary,
https://svn.autosar.org/repos/10Releases/
AUTOSAR_Glossary.pdf

[SRS_BSW_GENERAL] General Requirements on Basic Software Modules,
https://svn.autosar.org/repos/10Releases/
AUTOSAR_SRS_General.pdf

[SRS_BSW_SPAL] General Requirements on SPAL,
https://svn.autosar.org/repos/10Releases/
AUTOSAR_SRS_SPAL_General.pdf

[STD_HIS_IO_DRIVER] HIS API IO Driver, V2.1.3, April 29th, 2004,
http://www.automotive-his.de/download
API_IODriver_2_1_3.pdf

5.2 Related standards and norms

HIS IO Driver Specification [STD_HIS_IO_DRIVER]