Document Title	Requirements on I/O Hardware Abstraction
Document Owner	AUTOSAR GbR
Document Responsibility	AUTOSAR GbR
Document Version	1.0.1
Document Status	Final

Document Change History

Date	Version	Changed by	Change Description
28.11.2006	1.0.1	AUTOSAR Administration	Legal disclaimer revised
27.03.2006	1.0.0	AUTOSAR Administration	Initial release

Table of Contents

1 Scope of this document

This document defines general rules and formats for requirements specification within AUTOSAR. It shall be used as a basis for each requirements document.

The requirements are structured in the following way:

- General requirements on Basic Software Modules (other document)
- General requirements which apply to all modules of the Microcontroller and ECU Abstraction Layers (this document)
- Module specific requirements (this document)

Constraints

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

2 How to read this document

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

2.1 Conventions used

In requirements, the following specific semantics shall be used (based on 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:

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

2.2 Requirement structure

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

Functional Requirements:
- Configuration (which elements of the module need to be configurable)
- Initialization
- 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, abbrevations and expressions

3.1 Expressions - general

Expression	***Description***	***Example***
Class	A class represents a kind of electrical connection to the ECU. It could be for example an analogue, a discrete,…	Analogue Class, Discrete Class…
Electrical Signal	It is the electrical signal on the pin of the ECU	Physical input voltage at an ECU-Pin
ECU pin	It is an hardware electrical connection of the ECU with the rest of the electronic system	
ECU Signals	It is the **software representation** of an electrical signal. A signal has attributes and a symbolic name	Input voltage,Discrete Output, PWM Input …
ECU Signal group	It is the **software representation** of a group of electrical signals from the same Class	Only for discrete Inputs and discrete Outputs
Attributes	Characteristics that can be Software (SW) and Hardware (HW) for each kind of Signals existing in a ECU	Range, Lifetime / delay, …
Symbolic name	The symbolic name of a signal is used by the IO Hardware Abstraction module to make a link (function, pin)	

3.2 Expressions - signal attributes

Expression	***Description***	***Example***
Data Type	Analogue: Datatype of the signal Discrete: either bool or AUTOSAR defined type (BoolType)	(VoltageType, CurrentType, ResistanceType, BoolType) Each DataType has a given size: 16 bits or 32 bits
Range	This is a functional range and not an electrical range.) For analogue signals [lowerLimit...upperLimit] (Voltage, current), [0...upperLimit] (resistance) For discrete signals [0,1] For timing signals [0…upperLimit] (period), [-100…100%] (Duty Cycle)	[-12Volts...+12Volts] (voltage)
Resolution	This attribute for many Classes is dependent on the range and the Data Type. Example: (upperLimit - lowerLimit) / $(2^{datatypelength} -1)$ For the others is known and defined.	$Voltage_{min}$ = -12 Volts $Voltage_{max}$ = 12 Volts Data Type : 16 bits Resolution => 24 / 65535
Hardware Resolution	This is the maximum possible resolution of the hardware (ADC)	ADC converter could have a 8/10/12/16 bits resolution
Hardware Accuracy	This is the accuracy of Hardware. It depends on hardware peripheral used for acquisition and/or generation	ADC converter could have an accuracy of +-3LSB
Accuracy	It depends of hardware peripheral used for acquisition and/or generation.	ADC converter could be a a 8/10/12/16 bits converter

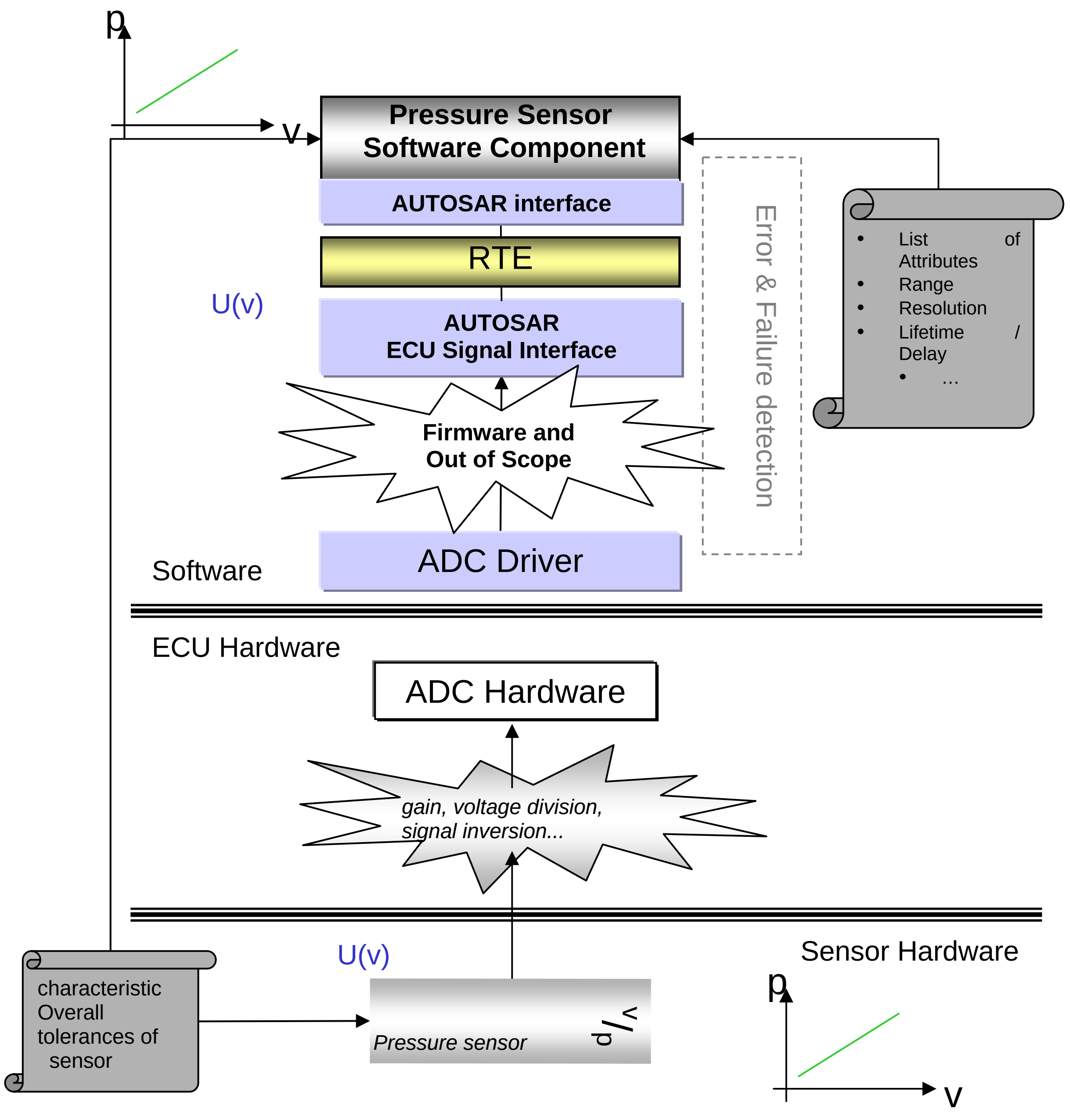
Expression	***Description***	***Example***
Diagnosis	Diagnosis capability of the functionality	Diagnosis Not Supported (could be a static check) No valid information available Short to Power Supply Short to Ground Open Load Over Temperature Diagnosis OK
Synchronization	A signal could be synchronize with another signal or with an event like a trigger	If a discrete signal is “TRUE”, acquire an analogue signal
Access	Defines if the Signal is attached to a Get(Read) / Set(Write) feature.	
Inversion	Inversion between the physical value and the logical value. This attribute is not visible and not configurable by users of IO Hardware Abstraction.	Physical HighState → (Signal=False) Physical LowState → (Signal=True)
Lifetime	Only for **Inputs**: It is the maximum allowed age of the data (time is in microseconds). If Lifetime is 0, then the signal is directly get from the register.	Lifetime = 0 is a direct access Lifetime = 1000µs the value read is at maximum 1ms older
Delay	Only for **Outputs**: It is the maximum allowed time until an output is actually set (time is in microseconds) If Delay is 0, then the signal is set immediately	Delay = 0 is a direct access Delay = 100µs the command is set until 100µs elapse
Filtering / Debouncing	It defines if the Signal is provided as a raw value or if a filtering/debouncing method is included in the IO Hardware Abstraction module for this Signal.	Raw, Debounce 3 Samples, Wait 10ms,
Sampling Rate	Time period required to get a Signal value.	Sampling rate for a sampling windows (burst)
Report Changes	This attribute is only applicable to Discrete Inputs. It defines the capability (or not) of reporting level changes.	Enable or Disable
Pulse Test	This attribute means that the output shall be tested thanks a dedicated pulse. If this attribute is not set, diagnosis will be done while using the output.	Available or non available

4 Requirement Specification

4.1 IO Hardware Abstraction

4.1.1 Functional Overview

The IO Hardware Abstraction module abstracts from the signal path of the ECU hardware (Layout, Microcontroller Pins, Microcontroller external devices like IO ASIC). It provides a signal based interface to the upper software layer. It performs static abstraction and inversion (if needed) of values according to their physical representation at the inputs/outputs of the ECU hardware (compensation of static influences caused within the path between ECU IO and Microcontroller pin, e.g. voltage divider, hardware inversion).


p
v
Pressure Sensor Software Component
AUTOSAR interface
RTE
U(v)
AUTOSAR ECU Signal Interface
Firmware and Out of Scope
Error & Failure detection
List of Attributes
Range
Resolution
Lifetime / Delay
…
Software
ADC Driver
ECU Hardware
ADC Hardware
gain, voltage division, signal inversion...
U(v)
Sensor Hardware
characteristic Overall tolerances of sensor
Pressure sensor
p/v
p
v

The IO Hardware Abstraction module allows configuring each signal according to an attributes list. Interfaces are AUTOSAR Standard.

4.1.2 Overview of Attributes to qualify Signals

Signal \ Attributes	Signal Data Type	Access	BSW-Range	Unit	BSW-Resolution	Failure Monitoring	Age (Lifetime/Delay)	Filtering / Debouncing	Sampling Rate	Report Feature	Pulse Test	Wakeup
Analogue$_{in}$	X	X	X	X	X	-	Xl	X	X	O	-	-
Analogue$_{out}$	X	X	X	X	X	O	Xd	-	-	-	O	-
Discrete$_{in}$	X	X	F	-	-	-	Xl	X	X	O	-	O
Discrete$_{Status}$	X	-	F	-	-	-	Xl	X	-	-	-	O
Discrete$_{pow}$	X	X	F	-	-	O	Xd	-	-	-	O	-
PWx Period$_{in}$	X	X	X	X	X	-	Xl	-	-	-	-	O
PWx Period$_{out}$	X	X	X	X	X	O	Xd	-	-	-	O	-
PWx Duty Cycle$_{in}$	X	X	F	F	X	-	Xl	-	-	-	-	O
PWx Duty Cycle$_{out}$	X	X	F	F	X	O	Xd	-	-	-	O	-

Table legend

- **X** means the Attribute is applicable to this Class of ECU Signal and shall be configured.
  - **Xl** means the Age Attribute applicable is the Lifetime.
  - **Xd** means the Age Attribute applicable is the Delay.
- **F** means the Attribute is applicable to this Class of ECU Signal but it is a fixed standard value.
- **O** means the Attribute is optional to this Class of ECU Signal and depends on a static configuration (disable/enable).
- **-** means the Attribute is not applicable or has no meaning to this Class of ECU Signal.

4.1.3 Functional Requirements

4.1.3.1 General

4.1.3.1.1 [BSW12409] Values within one static range for each signal

Initiator:	WP4.2.2.1.12
Date:	07.12.2004
Short Description:	Values within one static range for each signal

Type:	New
Importance:	High
Description:	The IO Hardware Abstraction module shall provide values within one static range for each Signal. This range is independent of the basic software driver scaling factor. Examples: Analogue Signals => [lowerLimit...upperLimit] with (lowerLimit = - UpperLimit or with (lowerLimit = 0)
Rationale:	Support of a wide range with a high resolution
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.1.2 [BSW12410] Measurement of input voltage

Initiator:	WP4.2.2.1.12
Date:	07.12.2004
Short Description:	Measurement of input voltage
Type:	New
Importance:	High
Description:	The IO Hardware Abstraction module shall provide a service to read an input voltage with these attributes: • DataType: VoltageType, • Range: [lowerLimit...upperLimit], lowerLimit and upperLimit can be negative ([-5Volts, -3Volts] • Resolution: VoltageType / (upperLimit - lowerLimit) • Accuracy: HW delivers • Synchronization: Yes / No • Lifetime: x = delayed of x microseconds • Filtering/Debouncing: raw, filtered (bandwith, corner frequency) • Sampling Rate: x: sample every x µs
Rationale:	Basic functionality of IO Hardware Abstraction
Use Case:	To control a component / sensor responding with voltage.
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.1.3 [BSW12411] Control of output voltage

Initiator:	WP4.2.2.1.12
Date:	07.12.2004
Short Description:	Control of output voltage
Type:	New
Importance:	High
Description:	The IO Hardware Abstraction module shall provide a service to control a output voltage with these attributes: • DataType: VoltageType • Range: [lowerLimit...upperLimit], lowerLimit can be negative • Resolution: VoltageType / (upperLimit - lowerLimit) • Accuracy: HW delivers • Diagnosis: The IO Hardware Abstraction module is able to detect the

	following failures: o Diagnosis Not Supported (could be a static check) o No valid information available o Short to Power Supply o Short to Ground o Open Load o Over Temperature o Diagnosis OK • Synchronization: Yes / No • Delay: x = delayed of x microseconds
Rationale:	Basic functionality
Use Case:	ECU supply Abstract the generation of an analogue signal by usage of PWM
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.1.4 [BSW12413] Measurement of input current

Initiator:	WP4.2.2.1.12
Date:	07.12.2004
Short Description:	Measurement of input Current
Type:	New
Importance:	High
Description:	The IO Hardware Abstraction module shall provide a service to read an input current with these attributes: • DataType: CurrentType • Range: [lowerLimit…upperLimit], lowerLimit can be negative • Resolution: CurrentType / (upperLimit - lowerLimit) • Accuracy: HW delivers • Synchronization: Yes / No • Lifetime: x = delayed of x microseconds • Filtering/Debouncing: raw, filtered (bandwith, corner frequency) • Sampling Rate: x: sample every x µs
Rationale:	Basic functionality of IO Hardware Abstraction
Use Case:	To control a component / sensor driven by current.
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.1.5 [BSW12415] Measurement of connected resistance

Initiator:	WP4.2.2.1.12
Date:	07.12.2004
Short Description:	Measurement of connected resistance
Type:	New
Importance:	High
Description:	The IO Hardware Abstraction module shall provide a service to measure a connected resistance using these attributes: • DataType: ResistanceType • Range: [lowerLimit…upperLimit], lowerLimit is null or positive • Resolution: ResistanceType / (upperLimit - lowerLimit) • Accuracy: HW delivers

	• Synchronization: Yes / No • Lifetime: x = delayed of x microseconds • Filtering/Debouncing: raw, filtered (bandwith, corner frequency) • Sampling Rate: x: sample every x µs
Rationale:	Basic functionality of IO Hardware Abstraction
Use Case:	Measurement of temperature sensor
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.1.6 [BSW12412] Get / Read a discrete input

Initiator:	WP4.2.2.1.12
Date:	07.12.2004
Short Description:	Get / Read a discrete input
Type:	New
Importance:	High
Description:	The IO Hardware Abstraction module shall provide a service to get/read a discrete input with these attributes: • DataType: boolean • Range: 0 or 1 • Resolution: Logical State • Accuracy: 1 bit • Synchronization: Yes / No • Lifetime: x = delayed of x microseconds • Filtering/Debouncing: raw, filtered (bandwith, corner frequency) • Sampling Rate: x: sample every x µs • Change Reporting: Enable or Disable
Rationale:	Basic functionality of IO Hardware Abstraction
Use Case:	Get/Read a logical value (0 / 1) from a ECU pin
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.1.7 [BSW12324] Simultaneous Get / Read of several discrete Inputs

Initiator:	VALEO
Date:	20.09.2004
Short Description:	Simultaneous Get / Read of several discrete Inputs
Type:	New
Importance:	High
Description:	The AUTOSAR IO Hardware Abstraction module shall offer a service to Get / Read simultaneously several discrete inputs. The number of inputs shall be configurable. This is limited to a physical portt.
Rationale:	All inputs belong to the same functionality or the same enhanced onboard chip.
Use Case:	For motor control but also for runtime optimization
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.1.8 [BSW12450] Report discrete input changes

Initiator:	Valeo
Date:	12.01.2005
Short Description:	Report discrete input changes
Type:	New
Importance:	High
Description:	The IO Hardware Abstraction module shall provide a mechanism to report to the Signal Client, when a discrete input changes. This functionality is only available in case of Change Reporting attribute is enabled for this Signal.
Rationale:	To ensure a real time behavior
Use Case:	To detect a sensor activity and react in time for example about wiping and brake pedal.
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.1.9 [BSW12419] Failure monitoring

Initiator:	WP4.2.2.1.12
Date:	07.12.2004
Short Description:	Failure monitoring
Type:	New
Importance:	High
Description:	The IO Hardware Abstraction module shall provide a service to monitor hardware failure and to set a status: • DataType: StatusType • Range: o No valid information available o Short to Power Supply o Short to Ground o Open Load o Over Temperature o Diagnosis OK • Synchronization: Yes / No • Lifetime: x = delayed of x microseconds • Filtering/Debouncing: raw, filtered (bandwith, corner frequency)
Rationale:	Basic functionality
Use Case:	Know the actual relay/lamp output state
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.1.10 [BSW12418] Control of discrete powered outputs

Initiator:	WP4.2.2.1.12
Date:	07.12.2004
Short Description:	Control of discrete powered outputs
Type:	New
Importance:	High
Description:	The IO Hardware Abstraction module shall provide a service to control a discrete powered output with these attributes: • DataType: boolean

	• Range: 0 or 1 • Resolution: Logical State • Accuracy: 1 bit • Diagnosis: The IO Hardware Abstraction module is able to detect the following failures: o Diagnosis Not Supported (could be a static check) o No valid information available o Short to Power Supply o Short to Ground o Open Load o Over Temperature o Diagnosis OK • Synchronization: Yes / No • Delay: x = delayed of x microseconds A simple Output (without powered) is a subset of Powered Outputs with Diagnosis attribute always as “Diagnosis Not Supported (could be a static check)”
Rationale:	Basic functionality
Use Case:	Relay control, lamp control
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.1.11 [BSW12323] Simultaneous update of several discrete outputs

Initiator:	VALEO
Date:	20.09.2004
Short Description:	Simultaneous update of several discrete outputs
Type:	New
Importance:	High
Description:	The AUTOSAR IO Hardware Abstraction module shall offer a service to update simultaneously several discrete outputs. The number of outputs shall be configurable. This is limited to a physical port.
Rationale:	All outputs belong to the same functionality or the same enhanced onboard chip.
Use Case:	For synchronization and for runtime optimization
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.1.12 [BSW12417] Measurement of the period time of signals

Initiator:	WP4.2.2.1.12
Date:	07.12.2004
Short Description:	Measurement of the period time of signals
Type:	New
Importance:	High
Description:	The IO Hardware Abstraction module shall provide a service to measure the period time between two falling or rising edges on a Signal using these attributes: • DataType: PeriodType • Range: [lowerLimit…upperLimit], lowerLimit is null or positive

	• Resolution: PeriodType / (upperLimit - lowerLimit) • Accuracy: HW delivers • Lifetime: x = delayed of x microseconds
Rationale:	Basic functionality of IO Hardware Abstraction
Use Case:	Measure the period of a PWM sensor
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.1.13 [BSW12416] Control the period time of a signal

Initiator:	WP4.2.2.1.12
Date:	07.12.2004
Short Description:	Control the period time of a signal
Type:	New
Importance:	High
Description:	The IO Hardware Abstraction module shall provide a service to control the period time between two falling or rising edges on a Signal using these attributes: • DataType: PeriodType • Range: [lowerLimit…upperLimit], lowerLimit is null or positive • Resolution: PeriodType / (upperLimit - lowerLimit) • Accuracy: HW delivers • Diagnosis: o Diagnosis Not Supported (could be a static check) o No valid information available o Short to Power Supply o Short to Ground o Open Load o Over Temperature o Diagnosis OK • Synchronization: Yes / No • Delay: x = delayed of x microseconds
Rationale:	Basic functionality of IO Hardware Abstraction
Use Case:	Control the period of a PWM output signal
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.1.14 [BSW12414] Control of Duty Cycle for a periodic Signal

Initiator:	WP4.2.2.1.12
Date:	07.12.2004
Short Description:	Control of Duty Cycle for a periodic Signal
Type:	New
Importance:	High
Description:	The IO Hardware Abstraction module shall provide a service to control the ratio between the active level and the inactive level of a periodic output Signal using these attributes: • DataType: DutyCycleType • Range: [-100%…+100%] • Resolution: to be defined • Accuracy: HW delivers

	• Synchronization: Yes / No • Delay: x = delayed of x microseconds
Rationale:	Basic functionality of IO Hardware Abstraction
Use Case:	A negative range is justified to allow for instance a motor direction control on the level of the hardware (negative duty cycle = to the left, positive duty cycle = to the right)
Dependencies:	Command of a Period Time signal
Conflicts:	--
Supporting Material:	--

4.1.3.1.15 [BSW12445] Measurement of Duty Cycle of a periodic Signal

Initiator:	WP4.2.2.1.12
Date:	07.12.2004
Short Description:	Measurement of Duty Cycle of a periodic Signal
Type:	New
Importance:	High
Description:	The IO Hardware Abstraction module shall provide a service to measure the ratio between the active level and the inactive level of a periodic input Signal using these attributes: • DataType: DutyCycleType • Range: [-100%…+100%] • Resolution: 100 / M • Accuracy: HW delivers • Lifetime: x = delayed of x microseconds
Rationale:	Basic functionality of IO Hardware Abstraction
Use Case:	ICU requirement
Dependencies:	Measurement of Period Time Signal
Conflicts:	--
Supporting Material:	--

4.1.3.1.16 [BSW12338] Synchronous interface for signal access

Initiator:	WP4.2.2.1.12
Date:	27.09.2004
Short Description:	Synchronous interface for signal access
Type:	New
Importance:	High
Description:	The IO Hardware Abstraction shall provide synchronous signal access functions (signal access) even if the sampling is asynchronous.
Rationale:	Abstraction of different mechanisms.
Use Case:	Access to the buffer of a cyclic ADC conversion.
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.1.17 [BSW13905] Uniqueness of the IO Hardware Abstraction module

Initiator:	WP4.2.2.1.12
Date:	05.05.2005
Short Description:	Uniqueness of the IO Hardware Abstraction module

Type:	New
Importance:	High
Description:	The IO Hardware Abstraction module shall be provided only once for each ECU
Rationale:	Only one instance of the IO Hardware Abstraction, avoid multiplicity of IO Hardware Abstraction modules, avoid conflict with multiple declaration.
Use Case:	The IO Hardware Abstraction module shall be generated one time according to the generation process. Any change (new input, new external peripheriphal required cause a new generation of the module
Dependencies:	--
Conflicts:	--
Supporting Material:	AUTOSAR methodology, WP4.1.1.2 requirements

4.1.3.2 Configuration

4.1.3.2.1 [BSW12232] Symbolic Name for each Signal

Initiator:	Hella
Date:	02.08.2004
Short Description:	Symbolic Name for each Signal
Type:	New
Importance:	High
Description:	The IO Hardware Abstraction shall allow to configure statically an unique Symbolic Name for each Signal.
Rationale:	No change in the upper layers when changing the hardware allocation.
Use Case:	Flexible ECU-Design
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.2.2 [BSW12319] Independency between physical and logical Level

Initiator:	VALEO
Date:	20.09.2004
Short Description:	Independency between physical and logical Level
Type:	New
Importance:	High
Description:	The AUTOSAR IO Hardware Abstraction module shall be independent of physical level (i.e. : output command active at low state or high state) and provide only logical level to the upper layer for digital IO.
Rationale:	Independency between users and hardware design
Use Case:	For instance, doors could be OPEN / CLOSE and theses states are independent of real hardware input status (0v, 5v, 12v)
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.2.3 [BSW12449] Signal groups

Initiator:	WP4.2.2.1.12
Date:	19.01.2005
Short Description:	Signal groups
Type:	New
Importance:	High
Description:	The IO Hardware Abstraction shall be able to handle more than one electrical signal simultaneously. The definition of a group is done during the configuration step. Signals belonging to a group have always the same type.
Rationale:	Control without time delay a group of signal
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--

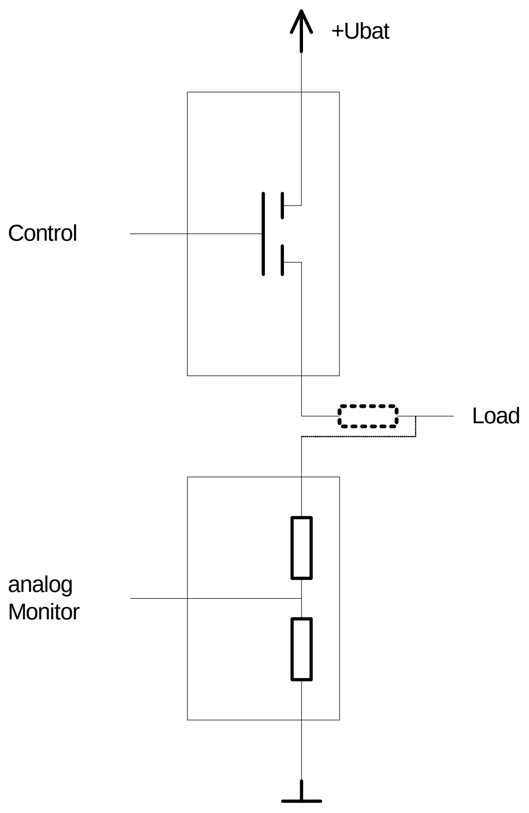
4.1.3.3 Normal Operation

4.1.3.3.1 [BSW12242] Onboard peripherals abstraction

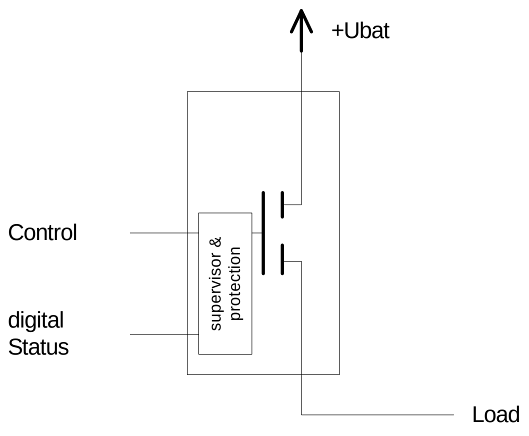
Initiator:	Hella
Date:	02.08.2004
Short Description:	Onboard peripherals abstraction
Type:	New
Importance:	High
Description:	The IO Hardware Abstraction shall hide any communication over ECU internal onboard peripherals to access Signals. The Signal routing on the ECU is abstracted by this interface.
Rationale:	It shall be no difference for the upper layers, if a port is connected directly to the Microcontroller or an onboard ASIC.
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.4 Diagnostic Functions

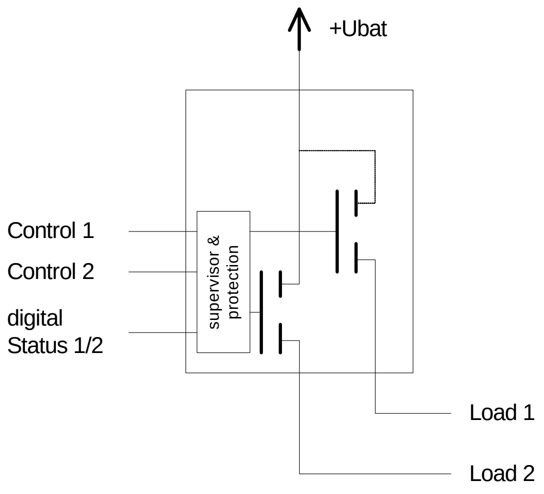
The following schematics show output driver stages and their diagnostic capability. They shall help to understand the following requirements.



Concept scheme for diagnostic of digital output by analogue monitor line



Concept scheme for diagnostic of digital output by digital monitor line of a single driver stage



Concept scheme for diagnostic of digital output by digital monitor line of a n-channel driver stage.

4.1.3.4.1 [BSW13900] Diagnostic of output signal, detection of short-circuit to the ground

Initiator:	WP4.2.2.1.12
Date:	30.03.2005
Short Description:	Diagnostic of output signal, detection of short circuit to the ground
Type:	Changed
Importance:	High
Description:	The IO Hardware Abstraction shall detect a failure short circuit to the ground, according to the hardware capabilities
Rationale:	Basic functionality
Use Case:	Analog monitoring of a discrete load signal, Fault detection for diagnosis and/or software driven driver stage protection
Dependencies:	--
Conflicts:	--
Supporting Material:	Specific Hardware Design : existing diagnostic monitoring.

4.1.3.4.2 [BSW13901] Diagnostic of output signal, detection of short-circuit to +Ubat

Initiator:	WP4.2.2.1.12
Date:	30.03.2005
Short Description:	Diagnostic of output signal, detection of short circuit to the +Ubat
Type:	Changed
Importance:	High
Description:	The IO Hardware Abstraction shall detect a failure short circuit to +UBat, according to the hardware capabilities
Rationale:	Basic functionality
Use Case:	Analog monitoring of a discrete load signal, Fault detection for diagnosis and / or error reaction
Dependencies:	--
Conflicts:	--
Supporting Material:	Specific Hardware Design : existing diagnostic monitoring.

4.1.3.4.3 [BSW13902] Diagnostic of output signal, detection of open circuit

Initiator:	WP4.2.2.1.12
Date:	30.03.2005
Short Description:	Diagnostic of output signal, detection of open circuit
Type:	Changed
Importance:	High
Description:	The IO Hardware Abstraction shall detect an open circuit, according to the hardware capabilities
Rationale:	Basic functionality
Use Case:	Analog monitoring of a discrete load signal, Fault detection for diagnosis and / or error reaction
Dependencies:	The safe detection of this failure is depended from the load and hardware circuit, resistance of driver stage
Conflicts:	--
Supporting Material:	Specific Hardware Design : existing diagnostic monitoring.

4.1.3.4.4 [BSW13903] Diagnostic of output signal, detection of overload

Initiator:	WP4.2.2.1.12
Date:	30.03.2005
Short Description:	Diagnostic of output signal, detection of over load
Type:	Changed
Importance:	High
Description:	The IO Hardware Abstraction shall detect a failure over load. This detection is done if controlled Output Signal is activated.
Rationale:	Basic functionality
Use Case:	Analog monitoring of a discrete load signal, Fault detection for diagnosis and / or error reaction
Dependencies:	The safe detection of this failure is depended from the load and hardware circuit, resistance of driver stage
Conflicts:	--
Supporting Material:	Specific Hardware Design : Diagnostic monitoring.

4.1.3.4.5 [BSW13904] Diagnostic of output signal, detection of over temperature

Initiator:	WP4.2.2.1.12
Date:	30.03.2005
Short Description:	Diagnostic of output signal, detection of overtemperature
Type:	Changed
Importance:	High
Description:	The IO Hardware Abstraction shall detect an over temperature. This detection is done if controlled Output Signal is activated.
Rationale:	Basic functionality
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	Specific Hardware Design : Diagnostic monitoring.

4.1.3.5 Fault operation

4.1.3.5.1 [BSW12248] ECU Hardware protection

Initiator:	WP4.2.2.1.12
Date:	20.01.2005
Short Description:	ECU Hardware protection
Type:	Changed
Importance:	High
Description:	The IO Hardware Abstraction module shall keep the ECU hardware safe. The IO Hardware Abstraction shall be able to cut off an output signal when a failure is detected on this output. This will be done in order to protect the hardware.
Rationale:	Protection against ECU deterioration.
Use Case:	Short circuit to the ground, short circuit to the supply, over temperature, overload. Deactivation of the output after three orders.
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.5.2 [BSW12451] No hardware failure recovery

Initiator:	WP4.2.2.1.12
Date:	20.01.2005
Short Description:	No hardware failure recovery
Type:	New
Importance:	High
Description:	The IO Hardware Abstraction module shall not decide on its own to switch an output on again that has been switched off for hardware protection reasons. Such a strategy to recover a failure shall be defined in a Software Component.
Rationale:	Strategies are included in SW-C
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.5.3 [BSW12452] Failure management: test pulse

Initiator:	WP4.2.2.1.12
Date:	20.01.2005
Short Description:	Failure management: test pulse
Type:	New
Importance:	High
Description:	The IO Hardware Abstraction module shall offer an interface to trigger an output after an output has been cut off.
Rationale:	Detect electrical defaults
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.4 Non-Functional Requirements (Qualities)

4.1.4.1 [BSW12339] Guarantee worst case delay times

Initiator:	WP4.2.2.1.12
Date:	27.09.2004
Short Description:	Guarantee worst case delay times
Type:	New
Importance:	High
Description:	The IO Hardware Abstraction shall guarantee a given worst case delay time for each signal. The time can be used for the evaluation of the time constraints.
Rationale:	Fulfill the request reaction times.
Use Case:	A lot of time constraints e.g. reaction after a pushed button
Dependencies:	--
Conflicts:	--
Supporting Material:	--

5 References

5.1 Deliverables of AUTOSAR

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

5.2 Related standards and norms

[2] HIS API IO Driver Specification V2.1.3