Document Title	Requirements on Function Inhibition Manager
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	081

Document Status	Draft
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	4.3.1

Document Change History			
Date	**Release**	**Changed by**	**Description**
2016-11-30	4.3.0	AUTOSAR Release Management	• Editorial Changes
2015-07-31	4.2.2	AUTOSAR Release Management	• Fim considers EventAvailbilty/ EventSuppression
2014-10-31	4.2.1	AUTOSAR Release Management	• Editorial changes
2013-10-31	4.1.2	AUTOSAR Release Management	• Editorial changes
2013-03-15	4.1.1	AUTOSAR Administration	• Editorial changes
2010-09-30	3.1.5	AUTOSAR Administration	• Add traceability to features • Apply new template ([1, TPS_STDT_00078]) for each SRS requirement
2010-09-30	3.1.5	AUTOSAR Administration	• Document structure reworked and extended • Added requirement for RTE API • Legal disclaimer revised
2008-08-13	3.1.1	AUTOSAR Administration	• Legal disclaimer revised

2008-08-13	3.1.1	AUTOSAR Administration	• [SRS_Fim_04713] added for OBD • Added expression "'diagnostic"' to requirement description in [SRS_Fim_04713] • Added definition for IUMPR • 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	• Initial release

Disclaimer

Table of Contents

1 Scope of document

The goal of AUTOSAR in particular working on the Function Inhibition Manager and this document is to define requirements on the functionality of the FIM. The focus is on the scope of the FIM but also the distinctions to other control mechanisms in AUTOSAR, such as RTE, and also to what extent elements of it have to be configurable and what preliminaries they shall comply with to meet the tailoring requirements.

If such the definition of these new elements is not part of this work package. Nevertheless the information about basic software elements additionally required shall be given to related work groups.

Constraints

First scope for specification of requirements on basic software modules are systems which are not safety relevant. For implementation of the basic software modules in safety relevant systems, it shall be checked if additional requirements are necessary.

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 [1, TPS_STDT_00078].
- 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 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)
- 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 and abbreviations

Abbreviation / Acronym:	**Description**:
Activity state	The activity state is the status of a software component being executed. The activity state results from the permission state as a precondition and also physical enable conditions. It is not calculated by the FIM and not available as a status variable. It could only be derived from local information within a software component.
API	Application Programming Interface
BSW	Basic Software
DEM	Diagnostic Event Manager
ECU	Electronic Control Unit
EOL	End Of Line
ESD	Electro Static Disturbance
ESP	Electronic Stability Program
FID	Function Identifier
FIM	Function Inhibition Manager
Functionality	Functionality comprises User-visible and User-non-visible functional aspects of a system (AUTOSAR_Glossary.pdf [2]). In addition to that - in the FIM context - a functionality can be built up of the contents of one, several or parts of runnable entities with the same set of permission / inhibit conditions. By means of the FIM, the inhibition of these functionalities can be configured and even modified by calibration. Each functionality is represented by a unique function ID. A functionality is featured by a specific set of inhibit condition in contrast to runnable entities having specific scheduling conditions.
HW	Hardware
ID	Identification/Identifier
ISO	International Standardization Organization
IUMPR	In Use Monitoring Performance Ratio: The In-Use-Monitor Performance Ratio (IUMPR) indicates how often the OBD system monitors, particular components, compared to the amount of the vehicle operation. It is defined as the number of times a fault could have been found (=numerator) divided by the number of times the vehicle operation has been fulfilled (=denominator) as defined in the respective OBD regulations.
MIL	Malfunction Indication Light
Monitoring function	• Part of the Software Component. • Mechanism to monitor and finally to detect a fault of a certain sensor, actuator or could be a plausibility check • Reports states about events from internal processing of a SW-C or from further processing of return values of other basic software modules. • See also AUTOSAR_SWS_DEM
NVRAM	Non volatile Memory
OBD	Onboard Diagnostics
OEM	Original Equipment Manufacturer
OS	Operating System
Permission state	The permission state contains the information whether a functionality, represented by its FID, can be executed or whether it shall not run. The state is controlled by the FIM based on reported events.
RAM	Random Access Memory
ROM	Read-only Memory
RTE	Runtime Environment

Abbreviation / Acronym:	**Description**:
Runnable entity	A Runnable Entity is a part of an Atomic Software-Component which can be executed and scheduled independently from the other Runnable Entities of this Atomic Software-Component. It is described by a sequence of instructions that can be started by the RTE. Each runnable entity is associated with exactly one EntryPoint.
SW-C	Software Components
Xxx_	Placeholder for an API provider

4 Requirement Specification

4.1 Functional Overview

The Function Inhibition Manager is responsible for providing a control mechanism for software components and the functionality therein. In this context, a functionality can be built up of the contents of one, several or parts of runnable entities with the same set of permission / inhibit conditions. By means of the FIM, the inhibiting of these functionalities can be configured and even modified by calibration. Therefore, the adaptation of a functionality into a new system context with modified physical boundary conditions and influences is significantly enhanced.

A functionality in the sense of the FIM and a runnable entity are different and independent types of classifications. Runnable entities are mainly featured by their scheduling requirements. In contrast to that, functionalities are classified by their inhibit conditions. The services of the FIM focus on applications in the SW-Cs, however, they are not limited to them. Functionalities of the BSW can also use the FIM services.

Note, there is no functional relationship between RTE and FIM. The RTE only provides communication in the sense that it connects the required ports of the SW components with the provided port(s) of the FIM. But the RTE does not implement any functionality of the FIM. In contrast to that, the FIM deals with inhibit conditions and provides supporting mechanisms for controlling functionalities within runnables via respective identifiers (FID). Therefore, the FIM and RTE concepts do not interfere with each other.

4.2 Functional Requirements

4.2.1 Configuration

4.2.1.1 [SRS_Fim_04701] The Functionalities supervised by the FIM shall be defined by static configuration

[SRS_Fim_04701] The Functionalities supervised by the FIM shall be defined by static configuration ⌈

Type:	Valid
Description:	The set of functionalities which should be supervised by the Function Inhibition Manager (FIM) shall be defined by static configuration.
Rationale:	Only functionalities being supervised via FID can make use of the FIM functionality/services (configurable permission state). The FIM has to deal with the FIDs of the functionalities to provide the automatic checking-mechanism for permission of execution on the demanded sections.
Use Case:	The number of FIDs to be handled by the FIM strongly depends on the application. Therefore, the list of FIDs shall be defined by configuration.
Supporting Material:	–

⌋(*RS_BRF_02216*)

4.2.1.2 [SRS_Fim_04702] The FIM shall support different inhibit options

[SRS_Fim_04702] The FIM shall support different inhibit options ⌈

Type:	Valid
Description:	The FIM shall support different inhibit options. The possible inhibit options are based on Dem_EventStatusExtendedType (TestFailed, Passed, ...) being provided by the DEM. The FIM shall at least support inhibition due to event state "failed". The exchange of information between DEM and FIM is ensured by forwarding the extended event status. The reactions of the FIM can only be based on that.
Rationale:	The most common reaction upon detected failure is to deactivate affected functionalities. Therefore, the FIM shall support inhibit due to "failed".
Use Case:	If an important sensor fails, e.g. adaptation functionality shall be stopped in order to prevent wrong adaptation values.
Supporting Material:	AUTOSAR_SWS_DEM

⌋(*RS_BRF_02216*)

4.2.1.3 [SRS_Fim_04719] Mechanism for summarized diagnostic event states shall be provided

[SRS_Fim_04719] Mechanism for summarized diagnostic event states shall be provided ⌈

Type:	Valid
Description:	The FIM shall provide a mechanism to handle summarized diagnostic event states. By a summarized diagnostic event state the calculation of a combined fault out of several individual faults in the software component is meant. However, it is not outlined whether this requirement shall be achieved by means of configuration process or by implementation in the FIM.
Rationale:	Easier calibration, robust against changes in the diagnostic package and reduced resources.
Use Case:	All faults that indicate a failed sensor.
Supporting Material:	–

⌋*(RS_BRF_02216)*

4.2.1.4 [SRS_Fim_04706] Individual configuration of inhibit conditions of functionalities shall be available

[SRS_Fim_04706] Individual configuration of inhibit conditions of functionalities shall be available ⌈

Type:	Valid
Description:	The FIM shall be configured per FID to relate events to it in a flexible way. The event - FID (inhibit) relation shall be changeable by calibration within configured limits, e.g. number of FIDs, supported inhibit masks, etc. Note, that summarized events could also be considered here ([SRS_Fim_04719] Mechanism for summarized diagnostic event states shall be provided).
Rationale:	The result of a fault is the reduction of available functionality. This must be configured by the related information of faults and SW-components.
Use Case:	Fault of oxygen sensor will lead to the reporting of a respective event and then to a reduced functionality of the catalyst diagnostics.
Supporting Material:	–

⌋*(RS_BRF_02216)*

4.2.2 Initialization

4.2.2.1 [SRS_Fim_04712] The permission states at start up shall be initialized

[SRS_Fim_04712] The permission states at start up shall be initialized ⌈

Type:	Valid

Description:	Based on all restored event status information (not only events stored in the fault memory) of the DEM, the FIM needs to compute the permission state for all FIDs at the initialization.
Rationale:	Necessity for the FIM to get notified of events which may affect the permission of FIDs.
Use Case:	–
Supporting Material:	–

⌋(*RS_BRF_01136*, *RS_BRF_02216*)

4.2.3 Normal Operation

4.2.3.1 [SRS_Fim_04700] An Interface for querying the FID permission status shall be provided

[SRS_Fim_04700] An Interface for querying the FID permission status shall be provided ⌈

Type:	Valid
Description:	The FIM shall provide an interface to SW-components and/or BSW modules (e.g. IUMPR calculation in the DEM) so that they are able to query their permission status. The FID has to be handed over as a parameter and the return value is either permitted or inhibited (permission yes/no).
Rationale:	Other BSW modules and software components shall be independent from the implementation of the FIM. The only relevant information is the permission status. Therefore, the release status shall be queried via interface function with the FID as parameter.
Use Case:	The catalyst monitoring function shall not be executed if the oxygen sensor was detected as failed. If the catalyst monitoring function is controlled via FID the reported malfunction of the sensor shall cause the FID to be inhibited.
Supporting Material:	–

⌋(*RS_BRF_02216*, *RS_BRF_01440*)

4.2.3.2 [SRS_Fim_04709] The permission state shall be evaluated before executing functionalities

[SRS_Fim_04709] The permission state shall be evaluated before executing functionalities ⌈

Type:	Valid
Description:	A functionality which is under supervision of the Function Inhibition Manager by using an FID shall query the FIM for its permission. If the FID is released, the functionality may be executed if all other enable conditions are met. On the other hand, if the FID is inhibited, the functionality must not be executed.
Rationale:	Main functionality

Use Case:	A functionality which is inactive must be prevented from executing. Since specification of FIM aims at notification mechanism, the permission is queried within the application SW. There, all enable conditions need to be checked.
Supporting Material:	–

⌋(*RS_BRF_02216*)

4.2.3.3 [SRS_Fim_04713] Methods for the computation of permission states shall be provided

[SRS_Fim_04713] Methods for the computation of permission states shall be provided ⌈

Type:	Valid
Description:	The FIM shall provide methods for the computation of permission status of an individual FID. The permission status yields from the diagnostic event states related to the FID. These event states are reported to the DEM and then forwarded to the FIM (SRS_Fim_04700
Rationale:	The focus of this requirement is on providing the methods for the computation of the permission state. It shall not be explicitly required to store the permission state of an FID or to compute it upon request for permission.
Use Case:	Suppose FID_alpha shall be inhibited by event_1 or event_2, hence the permission state of FID_alpha depends on the status of event_1 and event_2. Upon request of permission of FID_alpha the states of event_1 and event_2 could be evaluated. Alternatively, the status information of FID_alpha could be provided which is updated whenever event_1 or event_2 is changed.
Supporting Material:	–

⌋(*RS_BRF_02216*)

4.2.3.4 [SRS_Fim_04717] The permission states shall be updated

[SRS_Fim_04717] The permission states shall be updated ⌈

Type:	Valid
Description:	The FIM shall provide an API to the DEM in order to get informed about relevant status changes of reported events. Then, the status of the relevant FIDs can be updated.
Rationale:	Necessity for the FIM to get notified of events which may affect the permission of FIDs.
Use Case:	–
Supporting Material:	–

⌋(*RS_BRF_02216*)

4.2.3.5 [SRS_Fim_04723] The FIM shall provide a boolean configuration option per FID.

[SRS_Fim_04723] The FIM shall provide a boolean configuration option per FID. ⌈

Type:	Valid
Description:	The FIM shall provide a boolean configuration option per FID.
Rationale:	Rationale: Use case-specific configuration of functionality, only required functionality may be executed in ECU.
Use Case:	Use Case: Variant coding.
Supporting Material:	–

⌋*()*

4.2.3.6 [SRS_Fim_04721] OBD Functionalities shall be supported

[SRS_Fim_04721] OBD Functionalities shall be supported ⌈

Type:	Valid
Description:	For OBD, the in-use-performance on monitors needs to be tracked. For that purpose, records are generated by the DEM. In order to consider the impact of inhibiting faults on the monitors, the FIM shall provide access on its configuration data to the DEM.
Rationale:	DEM needs access to inhibit relations for the handling of IUMPR data.
Use Case:	–
Supporting Material:	–

⌋*(RS_BRF_02216)*

4.2.4 ShutDown Operation

No requirement

4.2.5 Fault Operation

No requirement

4.3 Non-Functional Requirements

4.3.1 Timing Requirements

No requirements

4.3.2 Resource Usage

No special requirement. Usage depends on implementation and hardware.

5 Requirements Tracing

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

Feature	Description	Satisfied by
[RS_BRF_01136]	AUTOSAR shall support variants of configured BSW data resolved after system start-up	[SRS_Fim_04712]
[RS_BRF_01440]	AUTOSAR services shall support system diagnostic functionality	[SRS_Fim_04700]
[RS_BRF_02216]	AUTOSAR diagnostic shall allow runtime degradation of faulty functionality to maintain minimum ECU/vehicle operability	[SRS_Fim_04700] [SRS_Fim_04701] [SRS_Fim_04702] [SRS_Fim_04706] [SRS_Fim_04709] [SRS_Fim_04712] [SRS_Fim_04713] [SRS_Fim_04717] [SRS_Fim_04719] [SRS_Fim_04721]

6 References

6.1 Deliverables of AUTOSAR

[1] Standardization Template
AUTOSAR_TPS_StandardizationTemplate

[2] Glossary
AUTOSAR_TR_Glossary

[3] Requirements on AUTOSAR Features
AUTOSAR_RS_Features

6.2 Related standards and norms

6.2.1 ITEA-EAST

[10] D1.5-General Architecture; ITEA/EAST-EEA, Version 1.0; chapter 3, page 72 et seq.

[20] D2.1-Embedded Basic Software Structure Requirements; ITEA/EAST-EEA, Version 1.0 or higher

[30] D2.2-Description of existing solutions; ITEA/EAST-EEA, Version 1.0 or higher.