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

Document Version	1.0.3
Document Status	Final
Part of Release	3.0
Revision	0001

Document Change History

Date	Version	Changed by	Change Description
31.10.2007	1.0.3	AUTOSAR Administration	• Document meta information extended • Small layout adaptations made
24.01.2007	1.0.2	AUTOSAR Administration	• “Advice for users” revised • “Revision Information” added
28.11.2006	1.0.1	AUTOSAR Administration	Legal Disclaimer revised
18.04.2006	1.0.0	AUTOSAR Administration	Initial release

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

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

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)
- 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). 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
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
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 General Requirements

4.2 Function Inhibition

4.2.1 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.

4.2.2 Functional Requirements

4.2.2.1 Configuration

4.2.2.1.1 [BSW04701] Functionality supervised by the FIM

Initiator:	FIM-group
Date:	10.05.2005
Short Description:	Functionality supervised by the FIM
Type:	New
Importance:	High
Description:	The set of functionalities which should be supervised by the Function Inhibition Manager (FIM) shall be defined by statical 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.

Dependencies:	Contribution to Software Component Template is necessary.
Conflicts:	--
Supporting Material:	--

4.2.2.1.2 [BSW04702] Support of inhibit options

Initiator:	FIM-group
Date:	10.05.2005
Short Description:	Support of inhibit options
Type:	New
Importance:	High
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. an adaptation functionality shall be stopped in order to prevent wrong adaptation values.
Dependencies:	--
Conflicts:	--
Supporting Material:	AUTOSAR_SWS_DEM

4.2.2.1.3 [BSW04719] Mechanism for summarized diagnostic event states

Initiator:	FIM-group
Date:	25.07.2005
Short Description:	Mechanism for summarized diagnostic event states
Type:	New
Importance:	High
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.
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.2.2.1.4 [BSW04706] Individual configuration of inhibit conditions of functionalities

Initiator:	FIM-group
Date:	10.05.2005
Short Description:	Individual configuration of inhibit conditions of functionalities

Type:	New
Importance:	High
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 (BSW04719). The FIM shall allow a matrix based configuration of dependencies between functionalities and inhibit conditions. This configuration determines for every inhibit condition the enabling / disabling of each available functionality.
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.
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.2.2.2 Initialization

4.2.2.2.1 [BSW04712] Initialization of the permission states at start up

Initiator:	FIM-group
Date:	07.06.2004
Short Description:	Initialization of the permission states at start up
Type:	New
Importance:	High
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. In order to have a time-efficient initialization the FIM needs direct access to event information (structure) rather than using the API Dem_GetEventStatus.
Rationale:	Necessity for the FIM to get notified of events which may affect the permission of FIDs.
Use Case:	--
Dependencies:	FIM-initialization process has to be integrated after DEM re-storage.
Conflicts:	--
Supporting Material:	--

4.2.2.3 Normal Operation

4.2.2.3.1 [BSW04700] Interface for querying the FID permission status

Initiator:	FIM-group
Date:	10.05.2005
Short Description:	Interface for permission status of FID
Type:	New
Importance:	High
Description:	The FIM shall provide an interface to SW-components and/or BSW modules (e.g. IUMPR calculation in the DEM) so that a software component is able to

	query its 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:	The modules shall be independent of 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.
Dependencies:	The FIDs have to be unique per FIM (ECU).
Conflicts:	--
Supporting Material:	--

4.2.2.3.2 [BSW04709] Evaluation of permission state before executing functionalities

Initiator:	WP 4.2.2.1.9 (session 2nd/3rd June 2004)
Date:	07.06.2004
Short Description:	Evaluation of permission state before executing functionalities
Type:	New
Importance:	High
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.
Dependencies:	Design guideline for applications, external requirement that cannot be fulfilled by the FIM on its own but by SW components and BSW as well.
Conflicts:	--
Change Requests:	--
Supporting Material:	--

4.2.2.3.3 [BSW04713] Methods for the computation of permission states

Initiator:	FIM-group
Date:	15.06.2005
Short Description:	Methods for the computation of permission states
Type:	New
Importance:	High
Description:	The FIM shall provide methods for the computation of permission status of an individual FID. The permission status yields from the event states related to the FID. These event states are reported to the DEM and then forwarded to the FIM (BSW04700).
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.
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.2.2.3.4 [BSW04717] Updating the permission states

Initiator:	WP 4.2.2.1.9 (session 2nd/3rd June 2004)
Date:	07.06.2004
Short Description:	Updating the permission states
Type:	New
Importance:	High
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:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.2.2.4 ShutDown Operation

No requirement

4.2.2.5 Fault Operation

No requirement

4.2.3 Non-Functional Requirements

4.2.3.1 Timing Requirements

No requirement

4.2.3.2 Resource Usage

No special requirement. Usage depends on implementation and hardware.

5 References

5.1 Deliverables of AUTOSAR

[DOC_MOD_LIST] List of Basic Software Modules
https:/svn2.autosar.org/repos2/22_Releases
AUTOSAR_BasicSoftwareModules.pdf

[DOC_LAYERED_ARCH] Layered Software Architecture
https:/svn2.autosar.org/repos2/22_Releases
AUTOSAR_LayeredSoftwareArchitecture.pdf

[DOC_VFB] Specification of the Virtual Functional Bus
https:/svn2.autosar.org/repos2/22_Releases
AUTOSAR_VirtualFunctionBus.pdf

5.2 Related standards and norms

5.2.1 OSEK

[STD_OSEK_OS]
http://www.osek-vdx.org

[STD_OSEK_ORTI]
http://www.osek-vdx.org

[STD_OSEK_OIL]
http://www.osek-vdx.org

[STD_OSEKTIME_OS]
http://www.osek-vdx.org

[STD_OSEKTIME_OIL]
http://www.osek-vdx.org

[STD_OSEKTIME_FTCOM]
http://www.osek-vdx.org

5.2.2 HIS

[STD_HIS_PROTECTED_OS]
http://www.automotive-his.de/his-ergebnisse.htm

[STD_HIS_IODRIVER]
http://www.automotive-his.de/his-ergebnisse.htm

5.2.3 ITEA-EAST

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

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

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