Document Title	Requirements on Firmware Over-The-Air
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	944

Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R19-11

Document Change History

Date	Release	Changed by	Description
2019-11-28	R19-11	AUTOSAR Release Management	• Initial release

Disclaimer

This work (specification and/or software implementation) and the material contained in it, as released by AUTOSAR, is for the purpose of information only. AUTOSAR and the companies that have contributed to it shall not be liable for any use of the work.

The material contained in this work is protected by copyright and other types of intellectual property rights. The commercial exploitation of the material contained in this work requires a license to such intellectual property rights.

This work may be utilized or reproduced without any modification, in any form or by any means, for informational purposes only. For any other purpose, no part of the work may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The work has been developed for automotive applications only. It has neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Table of Contents

1 Scope of Document

This document specifies requirements on the AUTOSAR FOTA (Firmware Over-The-Air) Target module realized as CDDs.

The FOTA Master mentioned in this document is analogous to the UCM-Master, which is planned to be specified in upcoming AUTOSAR releases.

2 How to read this document

2.1 Conventions used

The representation of requirements in AUTOSAR documents follows the table specified in [1, TPS_STDT_00078].

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.)

3 Acronyms and abbreviations

The glossary below includes acronyms and abbreviations relevant to [2] that are not included in the AUTOSAR Glossary [3].

Abbreviation / Acronym:	**Description:**
FOTA	(AUTOSAR) Firmware Over-The-Air

Table 3.1: Acronyms and Abbreviations

4 Requirements Specification

The following tasks of the entire FOTA procedure shall be allocated to the FOTA Target ECU:

- Realization of FOTA related diagnostics communication services
- Guarding of the FOTA sequence, which is usually implementer specific
- Buffering of received data chunks and appropriate packetizing of those to meet the requirements of the memory stack, especially with respect to minimum page size of the used flash
- Providing data chunks to memory stack and reporting the result of write access to diagnostic communication manager (Dcm), which finally provides a diagnsotic response to previously received request
- Realization of update interruption and resuming, i.e. finding the last possible point to proceed with an update and, if possible, not to restart entirely.
- Triggering ECU-internal rollback mechanisms to recover ECU's previous image based on request from FOTA Master

4.1 Functional Overview

4.2 Functional Requirements

4.2.1 General FOTA Requirements on FOTA Target ECUs

[RS_FOTA_CONSTR_00001] Memory capabilities of FOTA Target ECU for background installation. ⌈

Type:	draft
Description:	FOTA Target ECU shall be capable to install, i.e. receive and store, a new SW image while the current image on the ECU is executed in its normal operating mode.
Rationale:	Reduce the vehicle downtime caused by an update.
Dependencies:	
Use Case:	Dependent on the image size, vehicle bus topology and other factors the transfer of data from a FOTA Master ECU to FOTA Target ECUs as well as storage of those data in the FOTA Target ECU can take a long time. Realizing these steps in the normal operational mode, i.e. while driving, will significantly reduce the time, where the vehicle is blocked or functions not available due to an ongoing update.
Supporting Material:	Note: for the activation of new image the vehicle shall be still put into a safe-state, e.g. standstill, engine off etc.

⌋*()*

[RS_FOTA_CONSTR_00002] Memory capabilities of FOTA Target ECU for roll-back. ⌈

Type:	draft
Description:	FOTA Target ECUs shall be capable to internally recover the SW image, being active before last (FOTA) activation.
Rationale:	In case of failed activation the FOTA Target ECU can fall back to the previous state. It is more reliable in comparison to a re-installation trial by the FOTA Master ECU and also contributes to a reduced vehicle downtime in case of a rollback need.
Dependencies:	
Use Case:	In case the new image can not be properly activated, e.g. due to not given consistency, detection of integrity violation during activation etc., the FOTA Target ECU will recover its previously installed image. In case the activation was successful on a particular FOTA Target ECU, but an incompatibility arises on the vehicle level, a rollback of all affected FOTA Target ECUs may be required.
Supporting Material:	

⌋*()*

[RS_FOTA_00003] Support of FOTA related diagnostic services in normal operating mode. ⌈

Type:	draft

▽

△

Description:	FOTA Target ECU shall support diagnostic services dedicated for image updates in normal application mode. *This is mainly about securely accessing the ECU, erasing dedicated memory parts, requesting to start the installation of the new image, transfer of data and triggering the verification procedure at the end of the transfer.*
Rationale:	In non FOTA capable ECUs the update relevant diagnostic services are usually supported in reprogramming or bootmode only. As the installation on FOTA Target ECUs shall now happen during normal operating mode, e.g. driving, these services shall be supported in the normal operating mode too.
Dependencies:	–
Use Case:	The FOTA Master ECU is able to establish an update related diagnostic communication with FOTA Target ECU in normal operating mode.
Supporting Material:	

⌋*()*

[RS_FOTA_00004] Impact of ongoing installation during normal operating mode. ⌈

Type:	draft
Description:	An ongoing installation shall not disturb or reduce the functional scope of the respective FOTA Target ECU.
Rationale:	The support of background installation shall not lead to functional restrictions on a particular FOTA Target ECU or the entire vehicle, especially in terms of functional safety. In case a functional reduction due to an ongoing installation is necessary, the functional safety aspects of both, particular FOTA Target ECU and vehicle as a system, shall be respected.
Dependencies:	–
Use Case:	Functional safety aspects are not affected and will be fulfilled also in case of an ongoing FOTA update procedure.
Supporting Material:	

⌋*()*

[RS_FOTA_00005] Activation of new software in vehicle's safe-state only. ⌈

Type:	draft
Description:	FOTA Target ECU shall accept activation of new software in vehicle safe-state only. *Note: The definiton of the "vehicle safe state" is up to the implementation, but needs to ensure that the functional safety requirements are still fulfilled.*

▽

△

Rationale:	Activation of new software mostly includes an ECU reboot, which will affect the functional safety requirements and general availability of the addressed FOTA Target ECU.
Dependencies:	
Use Case:	After successful installation and verification of the new software, this software is activated in a vehicle safe-state only.
Supporting Material:	

⌋*()*

[RS_FOTA_00006] Dependencies of software and configuration data. ⌈

Type:	draft
Description:	FOTA Target ECU shall support mechanisms to resolve potential incompatibilities between different software revisions and corresponding configuration data.
Rationale:	Software and configuration data shall be in line to each other.
Dependencies:	
Use Case:	In case of a FOTA update, configuration data (variant information, learnt values, offsets, etc.) is to be handled properly. In case configuration data shall be adapted after installation of a new software revision, the corresponding mechanisms, either FOTA Target ECU internal or with involvement of the FOTA Master ECU, are provided.
Supporting Material:	

⌋*()*

[RS_FOTA_00007] Implementation of a FOTA Handler on FOTA Target ECUs. ⌈

Type:	draft
Description:	The FOTA Handler is a SW implementation on FOTA Target ECUs, which shall be realized as a complex device driver (CDD) in the current FOTA iteration.
Rationale:	Since the FOTA Handler module will interface the memory stack as well as the DCM and potential application SW, it is realized as CDD in order to keep the "layered-SW-architecuter" approach of AUTOSAR
Dependencies:	
Use Case:	The FOTA Handler provides over-the-air update capability to an ECU according to FOTA requirements and mechanisms, as long as all mentioned memory requirements can be fulfilled.
Supporting Material:	

⌋*()*

[RS_FOTA_CONSTR_00008] Reception of image data using diagnostic services ⌈

Type:	draft
Description:	The FOTA image data is provided to the *FOTA Handler* by the DCM module located in the *FOTA Target ECU*.
Rationale:	Avoid downtime of the vehicle due to ongoing installation by supporting corresponding UDS diagnostic communication in normal operating mode.
Dependencies:	
Use Case:	Reception and processing of software happens in the background without causing functional restrictions in FOTA Target ECU.
Supporting Material:	

⌋*()*

4.2.2 Requirements on FOTA Handler

[RS_FOTA_00009] Data processing and forwarding to the memory stack on the FOTA Target ECU. ⌈

Type:	draft
Description:	FOTA Handler shall be able to receive software chunks of different length, buffer and process them internally and provide them to the memory stack, which is finally writing the data to dedicated memory location.
Rationale:	The amount of received data can differ from the amount, that can be written directly into the memory, e.g. the received data doesn't fit to the page size requirements of the used flash driver instance. The FOTA Handler shall buffer the data that cannot be processed by the used flash driver instance at once.
Dependencies:	
Use Case:	Data transfer from the FOTA Master ECU to a FOTA Target ECU is processed using UDS as diagnostic protocol and different transport protocols, e.g. CanTp, FrTp, TCP/UDP, to disassemble the whole software image into several chunks. Since most transport protocols allow flexible payload length, the size of FOTA image chunks may vary, e.g. to accommodate with varying bus loads, FOTA Handler is able to process data chunks of variable length.
Supporting Material:	

⌋*()*

[RS_FOTA_00010] Abstraction of FOTA Target ECU internal memory layout. ⌈

Type:	draft
Description:	The FOTA Handler shall abstract the FOTA Target ECU internal memory layout from the FOTA Master ECU.
Rationale:	It shall be possible for the FOTA Master ECU to always operate on identical memory addresses or logically represented memory blocks, when installing a new software, regardless on which physical addresses in FOTA Target ECU data will be finally written.
Dependencies:	
Use Case:	Reduce the complexity of FOTA Master ECU by abstraction of internal memory layout of FOTA Target ECUs.
Supporting Material:	

⌋*()*

[RS_FOTA_00030] Completeness of new software. ⌈

Type:	draft
Description:	The FOTA Handler shall verify the completeness of received data as it was indicated by the FOTA Master ECU at the beginning of installation as prerequisite for activation. *Note: Verification features and techniques are implementation specific and are therefore not part ot this document.*
Rationale:	In order to ensure that the newly installed SW image is complete, which is a major prerequisite for booting the SW, a mechanism to provide this feature must be in place.
Dependencies:	
Use Case:	The completeness of the newly installed SW is a major prerequisite in order to indicate a successful installation procedure.
Supporting Material:	

⌋*()*

[RS_FOTA_00011] Activation of new software. ⌈

Type:	draft
Description:	The activation procedure shall only be triggered FOTA Master ECU.
Rationale:	A Request from the FOTA Master ECU to activate new the software shall be accepted by the FOTA Target ECU under reasonable conditions only. *Note: according to the upcoming Cyber Security standard ISO-21434 the integrity and authenticity of the new software shall be ensured before activation. As definition of Cyber Security requirements is out of scope in this document this note shall be treated as a hint rather than as a requirement.*

▽

△

Dependencies:	
Use Case:	As several FOTA Target ECUs can be affected by a functionally distributed software update the installation will be finished at different times on particular FOTA Target ECUs. Hence, the request to activate the new software is orchestrated by the FOTA Master ECU for compatibility reasons between different FOTA Target ECUs.
Supporting Material:	

⌋*()*

[RS_FOTA_00012] Rollback to previous software. ⌈

Type:	draft
Description:	The FOTA Handler shall be able to restore the previously active software in case of errors during the installation procedure.
Rationale:	The *FOTA Target ECU* internal rollback provides more reliability and downtime reduction. As multiple FOTA Target ECUs can be requested to execute a rollback at once, it reduces the overall vehicle downtime due to rollback parallelization.
Dependencies:	
Use Case:	In case of an error during or after activation of the new software, there are reliable measures in place to prevent bricked ECUs caused by those failed update.
Supporting Material:	

⌋*()*

[RS_FOTA_00013] FOTA Target ECU triggered Rollback ⌈

Type:	draft
Description:	The FOTA Target ECU shall be capable to trigger a local rollback in case of an ECU independent update procedure which raised errors during processing.
Rationale:	In case of errors during the installation, the FOTA Target ECU can trigger the Rollback procedure when no other ECUs were affected with the update. However, this will require the FOTA Handler to provide this information within the *Installation* procedure to be accessable in the final *Activation* phase *Note: The implicit rollback is executed by the FOTA Target ECU itself.*
Dependencies:	
Use Case:	A new software was installed only one FOTA Target ECU but cannot be booted successfully. If all (implemenation specific) conditions are met, the FOTA Target ECU will trigger the internal *Rollback* procedure.

▽

△

Supporting Material:	

⌋()

[RS_FOTA_00031] FOTA Master ECU triggered Rollback ⌈

Type:	draft
Description:	The FOTA Target ECU shall be able to receive and execute a rollback instruction received by the Dcm.
Rationale:	In case of errors during the update of one or more ECUs within one update campaign a central instance needs to trigger a rollback of the whole update campaign on all affected ECUs. *Note: The explicit rollback is triggered by the FOTA Master ECU.*
Dependencies:	
Use Case:	An update campaign is executed but causes errors on several affected ECUS, which requires a rollback of all those ECUs.
Supporting Material:	

⌋()

[RS_FOTA_00014] Update progress status. ⌈

Type:	draft
Description:	The FOTA Target ECU shall provide current update progress status on request from FOTA Master ECU.
Rationale:	Optimization of the overall update procedure, e.g. jump over several diagnostic services by FOTA Master ECU based on the update progress status communicated by FOTA Target ECU.
Dependencies:	
Use Case:	The FOTA Master ECU may request the update progress status from a FOTA Target ECU to correspondingly adjust the set of FOTA related diagnostic services to successfully proceed with the update. The definition of the intermediate update progress states, e.g. memory erased, part of data received, as well as the status granularity can be project specific, but shall be aligned between FOTA Target ECUs and FOTA Master ECU specifications.
Supporting Material:	

⌋()

[RS_FOTA_00015] Diagnostic communication between FOTA Master ECU and FOTA Target ECUs ⌈

Type:	draft
Description:	The FOTA related diagnostic communication protocol between FOTA Master ECU and FOTA Target ECUs shall be UDS as described in [4].
Rationale:	Communication between FOTA Target and UCM Master
Dependencies:	
Use Case:	All information between FOTA Target and FOTA Master is exchanged according to the UDS diagnostic protocol (see [4] for details).
Supporting Material:	

⌋*()*

[RS_FOTA_00032] Persit FOTA related data in the non-voaltive memory ⌈

Type:	draft
Description:	During the FOTA update procedure, process related (user) data shall be stored in the non-volatile memory (NvM) in order to exchange these information after an interruption with the FOTA Master ECU. *Note: For details about status and other FOTA Handler specific user data see the specification documents for the NvM module [5] and memory services [6]*
Dependencies:	
Use Case:	An interrupted FOTA installation, shall be capable for resumption. Resumption is initiated based on update progress status persisted during the previous installation phase of the same update sequence.
Supporting Material:	

⌋*()*

[RS_FOTA_00016] Interruptions and resumption during installation. ⌈

Type:	draft
Description:	The FOTA Handler shall be able to resume an interrupted installation.
Rationale:	In order to increase update efficiency, an installation shall rather be resumed, than entirely restarted in case of installation interruptions, e.g. due to switching ignition off.
Dependencies:	
Use Case:	An interrupted FOTA installation, regardless if preempted by a higher priority diagnostic job or interrupted by a system event (e.g. power loss, ignition off, etc.), is capable for resumption. Resumption is initiated based on update progress status, which has been achieved in the last valid installation step.
Supporting Material:	

⌋*()*

[RS_FOTA_00017] Cancel and restart of installation. ⌈

Type:	draft
Description:	The FOTA Handler shall support active cancellation and restart during installation.
Rationale:	There might be a need to actively cancel an ongoing installation by the FOTA Master ECU.
Dependencies:	
Use Case:	In case of necessity to cancel an ongoing installation and to potentially restart it, e.g. in case a newer update package is indicated to the FOTA Master ECU, while the previous one is still under installation, the FOTA Handler shall accept and execute the cancellation.
Supporting Material:	

⌋*()*

[RS_FOTA_00018] Interaction with Crypto Services provided by AUTOSAR ⌈

Type:	draft
Description:	The FOTA Handler shall be able to access Crypto Services using the interfaces as defined by AUTOSAR. *Note: The use of Crypto Services is optional wihtin the FOTA process. Features and mechanisms are implementaton speicific and are therefore not part of this document.*
Rationale:	Secure FOTA updates.
Dependencies:	
Use Case:	As integrity and authenticity of new contents (software, data) on FOTA Target ECUs is ensured the FOTA Handler is able to use the defined Crypto Service Manager (CSM) APIs.
Supporting Material:	

⌋*()*

4.2.3 Memory Stack Requirements

Note: The term memory stack shall substitute the memory driver (FlashDriver) and memory management modules (e.g. Fee) regardless if they are realized as internal or external driver, since no decisions on architectural solutions have been made yet.

[RS_FOTA_00033] Provide interfaces to the memory memory stack ⌈

Type:	draft
Description:	The memory stack shall provide interfaces to be used by the FOTA Handler module to program the new image data to the flash memory.
Rationale:	Transfer of FOTA chunk data from the FOTA Handler module to the memory stack. *The definition of the interfaces to the memory stack are not yet specified and therefore not listed in this document.*
Dependencies:	
Use Case:	In order to finally write the received data from the FOTA image chunk to the low-level memory stack (flash procedure).
Supporting Material:	

⌋*()*

[RS_FOTA_00022] Memory stack shall handle the targeted types of a flash device(s) ⌈

Type:	draft
Description:	The memory stack shall handle the for the FOTA procedure used types of flash devices.
Rationale:	Ensure standardized access to different types of flash devices.
Dependencies:	
Use Case:	The type, location and HW specific features of each physical flash driver instance is irrelevant for the implementer, since he uses a standardized API.
Supporting Material:	

⌋*()*

[RS_FOTA_00034] Memory stack shall be able to handle program flash device(s) ⌈

Type:	draft
Description:	The memory stack shall be able handle the program flash.
Rationale:	The memory stack must write (flash) new program data into the expected program flash sections.
Dependencies:	
Use Case:	In order to realize the FOTA procedure, the program flash sections must be accessible by the related driver for writing (and reading).
Supporting Material:	

⌋*()*

[RS_FOTA_00023] Memory stack shall provide a queuing mechanism ⌈

Type:	draft
Description:	In order to accept jobs from multiple users, the memory stack shall provide a queuing mechanism.
Rationale:	Queue user requests to the memory stack.
Dependencies:	
Use Case:	The memory stack and FOTA want to read and write data at the same time. No requests are rejected in case the addressed flash driver is already busy, but will be queued and processed as soon as possible.
Supporting Material:	

⌋*()*

[RS_FOTA_00024] Memory stack shall process multiple requests in parallel ⌈

Type:	draft
Description:	The memory stack shall handle several Flash devices. Those devices may or may not be accessed in parallel. Depending on the managed devices the memory stack shall process multiple requests in parallel.
Rationale:	Process multiple requests in parallel.
Dependencies:	
Use Case:	Program and data flash work independently but may be handled by one single memory driver. The driver forwards one request to each of the devices, if available, instead of processing them subsequently.
Supporting Material:	

⌋*()*

[RS_FOTA_00025] Memory stack shall prioritize accesses to flash devices ⌈

Type:	draft
Description:	In order to keep all non-FOTA related services and applications available, all incoming memory requests shall be prioritized.
Rationale:	Prioritization of concurrent memory accesses.
Dependencies:	
Use Case:	FOTA and system or application related requests are received at the same time (logically). The memory stack applies the lowest processing priority to the FOTA request and system or application related requests are always executed first.
Supporting Material:	

⌋*()*

[RS_FOTA_00026] Memory stack shall be able to preempt active jobs ⌈

Type:	draft
Description:	The memory stack shall preempt an ongoing job, if a higher priority job was requested. The preempted job shall be resumed as soon as the interrupting, higher priority jobs were processed. The preemption shall not be recognized by the user.
Rationale:	Prioritization of concurrent memory accesses.
Dependencies:	
Use Case:	Assuming FOTA writes multiple data chunks, the write job may be preempted, if a memory stack job requests immediate action. After the memory stack job was processed, the processing of the FOTA jobs is resumed. During the preemption time, the FOTA job(s) remain as pending and the FOTA Target waits for the job(s) to finish.
Supporting Material:	

⌋*()*

[RS_FOTA_00027] Memory stack shall provide an interface to access hardware specific information ⌈

Type:	draft
Description:	Memory stack shall provide an interface to access hardware specific information.
Rationale:	Program flash can behave different to data flash, provide other information, other restrictions.
Dependencies:	
Use Case:	In FOTA use case, for example the active partition or other hardware specific information may be important for the FOTA Target or even the master. Since this information depend on the used hardware, an overall interface is specified and implemented.
Supporting Material:	

⌋*()*

[RS_FOTA_00028] Memory stack for program flash shall handle all program flash related differences to the data flash. ⌈

Type:	draft
Description:	Memory stack for program flash shall handle all program flash related differences to the data flash.
Rationale:	Program flash can behave different to the data flash. The memory driver shall provide enough information to handle/validate the program flash related errors.
Dependencies:	

▽

△

Use Case:	Because of hardware restriction the program flash might provide other/ more information about failures/ states than the data flash. To be able to react to these failures/ states they are propagated to the upper layer, e.g. the FOTA Target ECU.
Supporting Material:	

⌋*()*

4.2.4 Additional Requirements

[RS_FOTA_00029] Only one specific SW image shall be provided to the FOTA Target ECU. ⌈

Type:	draft
Description:	Independent of the used HW and memory layout, only one generic image shall be provided to the FOTA Target ECU.
Rationale:	Regardless if running on HW with fixed or flexible runtime address mapping, only one specific image will be provided to the FOTA Target ECU. Offsets and recalculations are implementation specific but must be considered.
Dependencies:	
Use Case:	Since complete SW images can be quite big, it shall be avoided to provide one distinct image per memory partition, which may save a lot of space.
Supporting Material:	

⌋*()*

4.3 Non-Functional Requirements (Qualities)

5 Requirements Tracing

The following table references the features specified in [7] and links to the fulfillments of these.

Feature	Description	Satisfied by

6 References

[1] Standardization Template
AUTOSAR_TPS_StandardizationTemplate

[2] Explanation of Firmware Over-The-Air
AUTOSAR_EXP_FirmwareOverTheAir

[3] Glossary
AUTOSAR_TR_Glossary

[4] Specification of Diagnostic Communication Manager
AUTOSAR_SWS_DiagnosticCommunicationManager

[5] NV Data Handling Guideline
AUTOSAR_EXP_NVDataHandling

[6] Requirements on Memory Services
AUTOSAR_SRS_MemoryServices

[7] Requirements on AUTOSAR Features
AUTOSAR_RS_Features