



MCAL: SW Prod. Spec

Document Version : 54
Document Owner : Texas Instruments
Document Status : Published
Last Approval Date : Mar 15, 2022

TI Confidential - NDA Restrictions
Copyright ©2022 Texas Instruments Incorporated



Table Of Contents








- [Revision History](#)
 - [Terms and Abbreviations](#)
- [Document References](#)
- [Introduction](#)
 - [Overview](#)
 - [AUTOSAR \(MCAL drivers / modules\)](#)
 - [Purpose and Scope](#)
 - [Users of this Document](#)
 - [Approvers](#)
 - [Audience](#)
 - [Overall Description](#)
 - [Product Perspective](#)
 - [Product Function](#)
 - [MCAL](#)
 - [Constraints](#)
 - [Assumptions and Dependencies](#)
- [Requirement Verification Criteria](#)
- [Requirement Prioritization](#)
- [Detailed Requirements](#)
 - [Baselined Requirements:](#)
 - [Functional](#)
 - [Safety Functional](#)
 - [Runtime Environment](#)
 - [Interface](#)



- Performance and Resources
- Compliance and Compatibility
- Distribution
- Security
- Fault Handling
- Non-Functional Requirements
- Non-Requirement
- Testing
- Requirements not addressed (Is Not)
- Safety Diagnostic Requirements (HSI):
- Records Retention
- Appendix
- Template Revision



1 Revision History

Version	Date	Author	Comments
0.1	 02 Nov 2020	Texas Instruments	Initial Version
0.2	 20 Nov 2020	Texas Instruments	Review Comments Addressed
1.0	 20 Nov 2020	Texas Instruments	Upload to Galileo
2.0	 25 Nov 2020	Texas Instruments	Added missing Baseline Link, and Upload to Galileo
2.1		Texas Instruments	Updated for CR JACINTOREQ-1221
2.2	 16 Jul 2021	Texas Instruments	Updated JIRA Queries
2.3	 29 Jul 2021	Texas Instruments	Review Comments Addressed
2.4	 18 Feb 2022	Texas Instruments	JACINTOREQ-1870



Version	Date	Author	Comments
v21	 04 Mar 2022	Texas Instruments	Review Comments Addressed
v54	 15 Mar 2022	Texas Instruments	Minor update to Revision Table

1.1 Terms and Abbreviations

Abbreviation /Term	Meaning / Explanation
CDD	Complex Device Driver
MCU SS	MicroController Unit Sub System
AUTOSAR	AUTomotive Open System ARchitecture
Det	Development error tracer, used to report error in a development environment. MCAL module / drivers provide these traces

Abbreviation /Term	Meaning / Explanation
Dem	Diagnostic event manager, used to report diagnostic event's that could be logged/used to debug ECU in vehicle/field, MCAL module / drivers provides these events
ECUC	ECU Configuration, a GUI that could be used to configure drivers / modules of AUTOSAR stack
CSP	Compliance Support Package
FSQ	Functional Safety Quality
SoC	System on a Chip
ASIL	Automotive Safety Integrity Level
CSL	Chip Support Library
OSAL	Operation System Abstraction Layer
MCU	Micro Controller Unit

2 Document References

Reference	Description / Comments
MCAL MRD	Assess Phase: MCAL: Marketing Requirements
Introduction to AUTOSAR	https://en.wikipedia.org/wiki/AUTOSAR
Describes AUTOSAR Software Architecture	https://www.autosar.org/fileadmin/user_upload/standards/classic/4-3/AUTOSAR_EXP_LayeredSoftwareArchitecture.pdf
QRAS AP00216 Rev D	QRAS AP00216 Rev. D EP Software New Product Development
Galileo Project	https://galileo.itg.ti.com/cgi-bin/Galileo_UI/galileoHome.cgi?LevelId=11&MenuItemId=52&ItemId=62126
Product Proposal	Assess Phase: MCAL: Product Proposal
Jira	JIRA: LINK

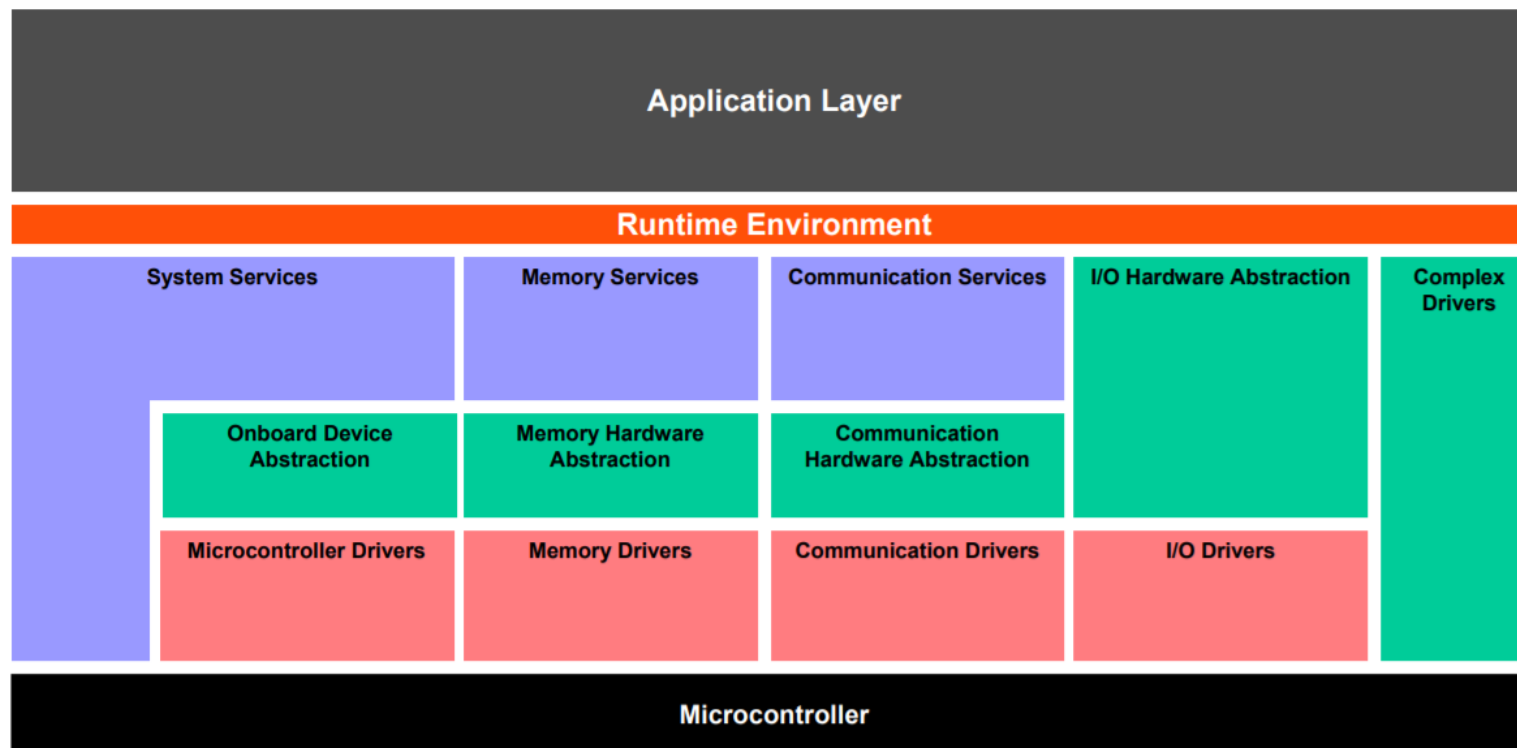


3 Introduction

3.1 Overview

3.1.1 AUTOSAR (MCAL drivers / modules)

As detailed by the AUTOSAR specification, **Micro**Controller **A**bstractio**n L**ayer is the lowest software layer of the Basic Software. This primarily contains drivers, which provide direct access to SoCs internal peripheral. The primary objective of these drivers is to provide standardized interface to other software layers of AUTOSAR stack.





More information on AUTOSAR https://www.autosar.org/fileadmin/user_upload/standards/classic/4-3/AUTOSAR_EXP_LayeredSoftwareArchitecture.pdf

3.2 Purpose and Scope

AUTOSAR specification for MCAL layer is specified as list of requirements, categorized by functionality of the module/drivers. These STANDARD requirements are included in this document as requirements. TI SoCs / Peripherals provide advanced / additional features and these features might not be called out as requirement by AUTOSAR, these features are listed under appropriate module as requirement in this document.

Some of these STANDARD requirements cannot be met (due to constraints such as lack-of-hardware-support, feature-not-required-for-usecase, feature-to-be-added-by-customers, etc...), these would be listed in section "Requirements Not Addressed"

3.2.1 Users of this Document

The users of the document shall include, architects, software engineers, test engineers, SQA Lead, Application Team and Marketing team.

3.2.2 Approvers

The approval authority of the document shall comply with the approval matrix defined as per AP216.

3.2.3 Audience

Earlier stated approvers and users of this document.



Product Line Marketing Managers

Customer Program Managers

3.3 Overall Description

TI SoCs are expected to be used in various automotive and industrial ECUs (such as front-camera ADAS, SRV, Gate Way, Head-Unit, etc...). AUTOSAR is the industry standard software architecture that allows OEM's software applications to scale to different vehicles, different SoCs, while maintaining safety and functional requirements. TI SoCs dedicates MCU SS to host AUTOSAR stack and provide a isolated processing entity.

3.4 Product Perspective

TI as SoC vendor is expected to provide MCAL layer for these devices and AUTOSAR stack vendor would integrate provided MCAL driver to complete the AUTOSAR stack on TI SoCs. Typically provided MCAL drivers / modules would be used in AUTOSAR stack with no modifications (the MCAL drivers/modules could be re-configured to suite ECU/ System needs, but no source code changes).

This mandates that provided MCAL modules provides best-possible performance, features and quality, the requirements listed in sections below highlight required MCAL module, it's functional needs and collateral's that demonstrate quality of drivers/modules.

As part of AUTOSAR specification, each of the MCAL driver should be configurable using an GUI. Each MCAL module will be accompanied with GUI based configurator (GUI would be a third party PC based software).



3.5 Product Function

Primary objective of the product is to provide drivers / modules & configurators required for AUTOSAR stack to be functional on TI SoCs.

3.5.1 MCAL

- Function
 - MCAL SW Provides MCU HW abstraction for customer SW development
 - MCAL SW delivery from TI is integrated directly into customer SW, hence MCAL module safety collateral can be leveraged by Customers for their SW stack safety process
 - MCAL SW component supports the safety diagnostics for the various supported IP, per the definition in the safety manual
- Development and Target Environment
 - MCAL executed in context of no Operating system
- Interaction
 - MCU Demos interface MCAL drivers and Complex Device Drivers to perform use-case demonstrations.
 - FreeRTOS and BootLoader is used during testing of MCAL Drivers on TI EVMs.

3.6 Constraints

MCAL is component in AUTOSAR stack and AUTOSAR requirements detail the required interfaces for each module. These interfaces, including ECUC parameters shall be adhered to expedite integration of MCAL modules into third party AUTOSAR stack.

3.7 Assumptions and Dependencies

Some key **assumptions** are as listed below

- **Version of AUTOSAR:** MCAL module specification are based on AUTOSAR version 4.3.1
- **Compiler:** MCAL drivers/modules provided shall be compiled using TI compiler (i.e. no third party compilers shall be employed, any exceptions shall be adequately documented).

Some key **dependencies** are as listed below

- **SDK Utilities:** Some of the utility (such as interrupt hooks, routines to display text on console, take inputs from console) shall be re-used from SDK. These utilities shall only be used to demonstrate functionality of MCAL drivers/modules.
- **SDK Libraries:** Some MCAL drivers may rely on SoC's internal peripheral (such as movement of data to/from peripheral, power mode configurations of peripheral, etc...) SDK provides libraries that enable use/configuration of these internal peripherals, MCAL drivers shall rely on PDK libraries to meet performance/configuration requirements.
- **Configurator:** Shall rely on a third party to tool configure MCAL modules/drivers, such as Elektrobit's Studio (EB)



4 Requirement Verification Criteria

All functional and testable requirements are verified by development of Qualification Test cases. Each requirement has a verification methods and verification description added in JIRA.

All other type of requirements like documentation, licensing, packaging, build etc. are verified using review or by Jenkins automated builds.

Safety Certification related requirements will also be verified with Audits and Inspection Reviews.



5 Requirement Prioritization

The priority for each of the requirement for the component is determined by the priority field in the JIRA type. The priority for requirements are set based on two considerations:

1. Delivery need for customers and to complete the whole product offering
2. Dependency between requirements and to ensure the development happens efficiently with minimal risk to execution

6 Detailed Requirements

Detailed driver and demo application requirements are listed in the following sections. Requirements are baselined with Jira/R4J and listed as query in section below

6.1 Baselined Requirements:

MCAL FR Baseline for PPR: <https://jira.itg.ti.com/plugins/servlet/com.easesolutions.jira.plugins.requirements/baseline?op=details&baselineId=7804&prj=MCAL>

Baseline Name: FSQ_MCAL_SPS_Baseline_v0

Note: Below filters are dynamic filters and shows the current requirements (and state) from the JIRA database. This is provided to aid in easily review of requirements in a tabular format. The baselined (approved) requirement list is provided in R4J link above.

These requirements are classified using guidelines present here:

<https://confluence.itg.ti.com/display/SWDevelopment/JIRA+Requirement+Types+and+their+use+by+SW+Development>

6.2 Functional

These are marked as 'Functional'

Key	Summary	Priori ty	Verification Description	Verificati on Method	Compo nent/s	Description
MCAL -8071	MCAL Modules shall adhere to appropriate build requirements.	P3- Mediu m	Verify Builds	Tests	Comm on	MCAL modules shall have makefile based builds, and shall support below: <ul style="list-style-type: none"> 1. C++ Builds 2. Debug and Release Profile builds 3. Module Per Driver builds
MCAL -7697	PDK: SCIClient Library	P3- Mediu m	Run the example applications.	Tests	Comm on	MCUSW demo application shall depend on SCICLIENT to instruct DMSEC to power up & allocate the peripheral to MCUSS.
MCAL -7686	MCAL Modules can be hosted on MCU1_0 and MCU2_1	P3- Mediu m	Qualification Tests on both cores	Tests	Comm on	AUTOSAR stack could be hosted on MCU 1 Core 0, which is MCU domain. MCAL modules shall support this implementation. AUTOSAR Stack could potentially be hosted on MCU 2 Core 1, which is in main domain. MCAL modules shall executable from MCU 2 Core 1

Key	Summary	Priori ty	Verification Description	Verificati on Method	Compo nent/s	Description
						An user guide shall be provided that details the steps require to support MCAL on other R5F cores (MCU 20, MCU 30 MCU 31). The user guide can choose one of the cores as reference/example to detail the steps
MCAL-7681	PDK: Utility for Interrupt registration & prints on console	P3-Mediu m	Running the example applications for drivers.	Tests	Comm on	MCUSW demo application and driver demo applications requires to register ISR to interrupts, MCUSW shall depend on PDK utilities for interrupt registration and display result/status on console via print utilities
MCAL-7676	PDK: UDMA Library	P3-Mediu m	UDMA will be used from SDK. Check SPI and ETH drivers.	Tests	Comm on	MCUSW driver and demo application will depend on UDMA to move data from & to peripheral to & from memory. MCUSW shall depend on PDK to provide UDMA libraries that could be used to program UDMA for data movement

6.3 Safety Functional

These are marked as 'Safety-Functional'

Key	Summary	Pri ty	Verifica tion Description	Verifica tion Method	Compo nent/s	Description
MCAL -8071	MCAL Modules shall adhere to appropriate build requirements.	P3- Mediu m	Verify Builds	Tests	Comm on	MCAL modules shall have makefile based builds, and shall support below: 1. C++ Builds 2. Debug and Release Profile builds 3. Module Per Driver builds
MCAL -7697	PDK: SCIClient Library	P3- Mediu m	Run the example applications.	Tests	Comm on	MCUSW demo application shall depend on SCICLIENT to instruct DMSEC to power up & allocate the peripheral to MCUSS.
MCAL -7686	MCAL Modules can be hosted on MCU1_0 and MCU2_1	P3- Mediu m	Qualification Tests on both cores	Tests	Comm on	AUTOSAR stack could be hosted on MCU 1 Core 0, which is MCU domain. MCAL modules shall support this implementation. AUTOSAR Stack could potentially be hosted on MCU 2 Core 1, which is in main domain. MCAL modules shall executable from MCU 2 Core 1

Key	Summary	Priori ty	Verification Description	Verificati on Method	Compo nent/s	Description
						An user guide shall be provided that details the steps require to support MCAL on other R5F cores (MCU 20, MCU 30 MCU 31). The user guide can choose one of the cores as reference/example to detail the steps
MCAL-7681	PDK: Utility for Interrupt registration & prints on console	P3-Mediu m	Running the example applications for drivers.	Tests	Comm on	MCUSW demo application and driver demo applications requires to register ISR to interrupts, MCUSW shall depend on PDK utilities for interrupt registration and display result/status on console via print utilities
MCAL-7676	PDK: UDMA Library	P3-Mediu m	UDMA will be used from SDK. Check SPI and ETH drivers.	Tests	Comm on	MCUSW driver and demo application will depend on UDMA to move data from & to peripheral to & from memory. MCUSW shall depend on PDK to provide UDMA libraries that could be used to program UDMA for data movement

6.4 Runtime Environment

Note, following are the requirements that pertain to the runtime environment. These are replicated from the consolidated list above ONLY to call out specific requirements that apply to these requirement type

Key	Summary	Priority	Verification Description	Verification Method	Component/s	Description
MCAL-8071	MCAL Modules shall adhere to appropriate build requirements.	P3-Medium	Verify Builds	Tests	Common	MCAL modules shall have makefile based builds, and shall support below: <ol style="list-style-type: none"> 1. C++ Builds 2. Debug and Release Profile builds 3. Module Per Driver builds
MCAL-7697	PDK: SCIClient Library	P3-Medium	Run the example applications.	Tests	Common	MCUSW demo application shall depend on SCICLIENT to instruct DMSEC to power up & allocate the peripheral to MCUSS.

Key	Summary	Priori ty	Verification Description	Verificati on Method	Compo nent/s	Description
MCAL -7686	MCAL Modules can be hosted on MCU1_0 and MCU2_1	P3-Mediu m	Qualification Tests on both cores	Tests	Comm on	<p>AUTOSAR stack could be hosted on MCU 1 Core 0, which is MCU domain. MCAL modules shall support this implementation.</p> <p>AUTOSAR Stack could potentially be hosted on MCU 2 Core 1, which is in main domain. MCAL modules shall executable from MCU 2 Core 1</p> <p>An user guide shall be provided that details the steps require to support MCAL on other R5F cores (MCU 20, MCU 30 MCU 31). The user guide can choose one of the cores as reference/example to detail the steps</p>
MCAL -7681	PDK: Utility for Interrupt registration & prints on console	P3-Mediu m	Running the example applications for drivers.	Tests	Comm on	MCUSW demo application and driver demo applications requires to register ISR to interrupts, MCUSW shall depend on PDK utilities for interrupt registration and display result/status on console via print utilities
MCAL -7676	PDK: UDMA Library	P3-Mediu m	UDMA will be used from SDK.	Tests	Comm on	MCUSW driver and demo application will depend on UDMA to move data from & to peripheral to & from memory. MCUSW shall depend on PDK to

Key	Summary	Priority	Verification Description	Verification Method	Component/s	Description
			Check SPI and ETH drivers.			provide UDMA libraries that could be used to program UDMA for data movement

6.5 Interface

Key	Summary	Priority	Verification Description	Verification Method	Component/s	Description
MCAL-8071	MCAL Modules shall adhere to appropriate build requirements.	P3-Medium	Verify Builds	Tests	Common	<p>MCAL modules shall have makefile based builds, and shall support below:</p> <ol style="list-style-type: none"> 1. C++ Builds 2. Debug and Release Profile builds 3. Module Per Driver builds

Key	Summary	Priority	Verification Description	Verification Method	Component/s	Description
MCAL-7697	PDK: SCIClient Library	P3-Medium	Run the example applications.	Tests	Common	MCUSW demo application shall depend on SCICLIENT to instruct DMSEC to power up & allocate the peripheral to MCUSS.
MCAL-7686	MCAL Modules can be hosted on MCU1_0 and MCU2_1	P3-Medium	Qualification Tests on both cores	Tests	Common	<p>AUTOSAR stack could be hosted on MCU 1 Core 0, which is MCU domain. MCAL modules shall support this implementation.</p> <p>AUTOSAR Stack could potentially be hosted on MCU 2 Core 1, which is in main domain. MCAL modules shall executable from MCU 2 Core 1</p> <p>An user guide shall be provided that details the steps require to support MCAL on other R5F cores (MCU 20, MCU 30 MCU 31). The user guide can choose one of the cores as reference/example to detail the steps</p>
MCAL-7681	PDK: Utility for Interrupt registration & prints on console	P3-Medium	Running the example applications for drivers.	Tests	Common	MCUSW demo application and driver demo applications requires to register ISR to interrupts, MCUSW shall depend on PDK utilities for interrupt registration and display result/status on console via print utilities

Key	Summary	Priori ty	Verification Description	Verificati on Method	Compo nent/s	Description
MCAL -7676	PDK: UDMA Library	P3- Mediu m	UDMA will be used from SDK. Check SPI and ETH drivers.	Tests	Comm on	MCUSW driver and demo application will depend on UDMA to move data from & to peripheral to & from memory. MCUSW shall depend on PDK to provide UDMA libraries that could be used to program UDMA for data movement

6.6 Performance and Resources

Note, following are the requirements that pertain to the performance and resources. These are replicated from the consolidated list above ONLY to call out specific requirements that apply to Performance and Resources

Key	Summary	Priori ty	Verification Description	Verificati on Method	Compo nent/s	Description
MCAL -8071	MCAL Modules shall adhere to appropriate build requirements.	P3- Mediu m	Verify Builds	Tests	Comm on	MCAL modules shall have makefile based builds, and shall support below: 1. C++ Builds 2. Debug and Release Profile builds 3. Module Per Driver builds
MCAL -7697	PDK: SCIClient Library	P3- Mediu m	Run the example applications.	Tests	Comm on	MCUSW demo application shall depend on SCICLIENT to instruct DMSEC to power up & allocate the peripheral to MCUSS.
MCAL -7686	MCAL Modules can be hosted on MCU1_0 and MCU2_1	P3- Mediu m	Qualification Tests on both cores	Tests	Comm on	AUTOSAR stack could be hosted on MCU 1 Core 0, which is MCU domain. MCAL modules shall support this implementation. AUTOSAR Stack could potentially be hosted on MCU 2 Core 1, which is in main domain. MCAL modules shall executable from MCU 2 Core 1

Key	Summary	Priori ty	Verification Description	Verificati on Method	Compo nent/s	Description
						An user guide shall be provided that details the steps require to support MCAL on other R5F cores (MCU 20, MCU 30 MCU 31). The user guide can choose one of the cores as reference/example to detail the steps
MCAL-7681	PDK: Utility for Interrupt registration & prints on console	P3-Mediu m	Running the example applications for drivers.	Tests	Comm on	MCUSW demo application and driver demo applications requires to register ISR to interrupts, MCUSW shall depend on PDK utilities for interrupt registration and display result/status on console via print utilities
MCAL-7676	PDK: UDMA Library	P3-Mediu m	UDMA will be used from SDK. Check SPI and ETH drivers.	Tests	Comm on	MCUSW driver and demo application will depend on UDMA to move data from & to peripheral to & from memory. MCUSW shall depend on PDK to provide UDMA libraries that could be used to program UDMA for data movement

6.7 Compliance and Compatibility

Key	Summary	Priority	Verification Description	Verification Method	Component/s	Description
MCAL-8071	MCAL Modules shall adhere to appropriate build requirements.	P3-Medium	Verify Builds	Tests	Common	MCAL modules shall have makefile based builds, and shall support below: <ol style="list-style-type: none"> 1. C++ Builds 2. Debug and Release Profile builds 3. Module Per Driver builds
MCAL-7697	PDK: SCIClient Library	P3-Medium	Run the example applications.	Tests	Common	MCUSW demo application shall depend on SCICLIENT to instruct DMSEC to power up & allocate the peripheral to MCUSS.
MCAL-7686	MCAL Modules can be hosted on MCU1_0 and MCU2_1	P3-Medium	Qualification Tests on both cores	Tests	Common	AUTOSAR stack could be hosted on MCU 1 Core 0, which is MCU domain. MCAL modules shall support this implementation. AUTOSAR Stack could potentially be hosted on MCU 2 Core 1, which is in main domain. MCAL modules shall executable from MCU 2 Core 1

Key	Summary	Priori ty	Verification Description	Verificati on Method	Compo nent/s	Description
						An user guide shall be provided that details the steps require to support MCAL on other R5F cores (MCU 20, MCU 30 MCU 31). The user guide can choose one of the cores as reference/example to detail the steps
MCAL-7681	PDK: Utility for Interrupt registration & prints on console	P3-Mediu m	Running the example applications for drivers.	Tests	Comm on	MCUSW demo application and driver demo applications requires to register ISR to interrupts, MCUSW shall depend on PDK utilities for interrupt registration and display result/status on console via print utilities
MCAL-7676	PDK: UDMA Library	P3-Mediu m	UDMA will be used from SDK. Check SPI and ETH drivers.	Tests	Comm on	MCUSW driver and demo application will depend on UDMA to move data from & to peripheral to & from memory. MCUSW shall depend on PDK to provide UDMA libraries that could be used to program UDMA for data movement

6.8 Distribution

Key	Summary	Priority	Verification Description	Verification Method	Component/s	Description
MCAL-8071	MCAL Modules shall adhere to appropriate build requirements.	P3-Medium	Verify Builds	Tests	Common	MCAL modules shall have makefile based builds, and shall support below: <ol style="list-style-type: none"> 1. C++ Builds 2. Debug and Release Profile builds 3. Module Per Driver builds
MCAL-7697	PDK: SCIClient Library	P3-Medium	Run the example applications.	Tests	Common	MCUSW demo application shall depend on SCICLIENT to instruct DMSEC to power up & allocate the peripheral to MCUSS.
MCAL-7686	MCAL Modules can be hosted on MCU1_0 and MCU2_1	P3-Medium	Qualification Tests on both cores	Tests	Common	AUTOSAR stack could be hosted on MCU 1 Core 0, which is MCU domain. MCAL modules shall support this implementation. AUTOSAR Stack could potentially be hosted on MCU 2 Core 1, which is in main domain. MCAL modules shall executable from MCU 2 Core 1

Key	Summary	Priori ty	Verification Description	Verificati on Method	Compo nent/s	Description
						An user guide shall be provided that details the steps require to support MCAL on other R5F cores (MCU 20, MCU 30 MCU 31). The user guide can choose one of the cores as reference/example to detail the steps
MCAL-7681	PDK: Utility for Interrupt registration & prints on console	P3-Mediu m	Running the example applications for drivers.	Tests	Comm on	MCUSW demo application and driver demo applications requires to register ISR to interrupts, MCUSW shall depend on PDK utilities for interrupt registration and display result/status on console via print utilities
MCAL-7676	PDK: UDMA Library	P3-Mediu m	UDMA will be used from SDK. Check SPI and ETH drivers.	Tests	Comm on	MCUSW driver and demo application will depend on UDMA to move data from & to peripheral to & from memory. MCUSW shall depend on PDK to provide UDMA libraries that could be used to program UDMA for data movement

6.9 Security

Not applicable for current scope of project

6.10 Fault Handling

Key	Summary	Priority	Verification Description	Verification Method	Component/s	Description
MCAL-8071	MCAL Modules shall adhere to appropriate build requirements.	P3-Medium	Verify Builds	Tests	Common	MCAL modules shall have makefile based builds, and shall support below: <ol style="list-style-type: none"> 1. C++ Builds 2. Debug and Release Profile builds 3. Module Per Driver builds
MCAL-7697	PDK: SCIClient Library	P3-Medium	Run the example applications.	Tests	Common	MCUSW demo application shall depend on SCICLIENT to instruct DMSEC to power up & allocate the peripheral to MCUSS.
MCAL-7686	MCAL Modules can be hosted on MCU1_0 and MCU2_1	P3-Medium	Qualification Tests on both cores	Tests	Common	AUTOSAR stack could be hosted on MCU 1 Core 0, which is MCU domain. MCAL modules shall support this implementation. AUTOSAR Stack could potentially be hosted on MCU 2 Core 1, which is in main domain. MCAL modules shall executable from MCU 2 Core 1

Key	Summary	Priori ty	Verification Description	Verificati on Method	Compo nent/s	Description
						An user guide shall be provided that details the steps require to support MCAL on other R5F cores (MCU 20, MCU 30 MCU 31). The user guide can choose one of the cores as reference/example to detail the steps
MCAL-7681	PDK: Utility for Interrupt registration & prints on console	P3-Mediu m	Running the example applications for drivers.	Tests	Comm on	MCUSW demo application and driver demo applications requires to register ISR to interrupts, MCUSW shall depend on PDK utilities for interrupt registration and display result/status on console via print utilities
MCAL-7676	PDK: UDMA Library	P3-Mediu m	UDMA will be used from SDK. Check SPI and ETH drivers.	Tests	Comm on	MCUSW driver and demo application will depend on UDMA to move data from & to peripheral to & from memory. MCUSW shall depend on PDK to provide UDMA libraries that could be used to program UDMA for data movement

6.11 Non-Functional Requirements

Key	Summary	Priority	Verification Description	Verification Method	Component/s	Description
MCAL-8071	MCAL Modules shall adhere to appropriate build requirements.	P3-Medium	Verify Builds	Tests	Common	MCAL modules shall have makefile based builds, and shall support below: <ol style="list-style-type: none"> 1. C++ Builds 2. Debug and Release Profile builds 3. Module Per Driver builds
MCAL-7697	PDK: SCIClient Library	P3-Medium	Run the example applications.	Tests	Common	MCUSW demo application shall depend on SCICLIENT to instruct DMSEC to power up & allocate the peripheral to MCUSS.
MCAL-7686	MCAL Modules can be hosted on MCU1_0 and MCU2_1	P3-Medium	Qualification Tests on both cores	Tests	Common	AUTOSAR stack could be hosted on MCU 1 Core 0, which is MCU domain. MCAL modules shall support this implementation. AUTOSAR Stack could potentially be hosted on MCU 2 Core 1, which is in main domain. MCAL modules shall executable from MCU 2 Core 1

Key	Summary	Priori ty	Verification Description	Verificati on Method	Compo nent/s	Description
						An user guide shall be provided that details the steps require to support MCAL on other R5F cores (MCU 20, MCU 30 MCU 31). The user guide can choose one of the cores as reference/example to detail the steps
MCAL-7681	PDK: Utility for Interrupt registration & prints on console	P3-Mediu m	Running the example applications for drivers.	Tests	Comm on	MCUSW demo application and driver demo applications requires to register ISR to interrupts, MCUSW shall depend on PDK utilities for interrupt registration and display result/status on console via print utilities
MCAL-7676	PDK: UDMA Library	P3-Mediu m	UDMA will be used from SDK. Check SPI and ETH drivers.	Tests	Comm on	MCUSW driver and demo application will depend on UDMA to move data from & to peripheral to & from memory. MCUSW shall depend on PDK to provide UDMA libraries that could be used to program UDMA for data movement

6.12 Non-Requirement

Key	Summary	Priority	Verification Description	Verification Method	Component/s	Description
MCAL-8071	MCAL Modules shall adhere to appropriate build requirements.	P3-Medium	Verify Builds	Tests	Common	MCAL modules shall have makefile based builds, and shall support below: <ol style="list-style-type: none"> 1. C++ Builds 2. Debug and Release Profile builds 3. Module Per Driver builds
MCAL-7697	PDK: SCIClient Library	P3-Medium	Run the example applications.	Tests	Common	MCUSW demo application shall depend on SCICLIENT to instruct DMSEC to power up & allocate the peripheral to MCUSS.
MCAL-7686	MCAL Modules can be hosted on MCU1_0 and MCU2_1	P3-Medium	Qualification Tests on both cores	Tests	Common	AUTOSAR stack could be hosted on MCU 1 Core 0, which is MCU domain. MCAL modules shall support this implementation. AUTOSAR Stack could potentially be hosted on MCU 2 Core 1, which is in main domain. MCAL modules shall executable from MCU 2 Core 1

Key	Summary	Priori ty	Verification Description	Verificati on Method	Compo nent/s	Description
						An user guide shall be provided that details the steps require to support MCAL on other R5F cores (MCU 20, MCU 30 MCU 31). The user guide can choose one of the cores as reference/example to detail the steps
MCAL-7681	PDK: Utility for Interrupt registration & prints on console	P3-Mediu m	Running the example applications for drivers.	Tests	Comm on	MCUSW demo application and driver demo applications requires to register ISR to interrupts, MCUSW shall depend on PDK utilities for interrupt registration and display result/status on console via print utilities
MCAL-7676	PDK: UDMA Library	P3-Mediu m	UDMA will be used from SDK. Check SPI and ETH drivers.	Tests	Comm on	MCUSW driver and demo application will depend on UDMA to move data from & to peripheral to & from memory. MCUSW shall depend on PDK to provide UDMA libraries that could be used to program UDMA for data movement

6.13 Testing

These are marked as 'Testing'

Key	Summary	Priori ty	Verification Description	Verificati on Method	Compo nent/s	Description
MCAL -8071	MCAL Modules shall adhere to appropriate build requirements.	P3- Mediu m	Verify Builds	Tests	Comm on	MCAL modules shall have makefile based builds, and shall support below: <ul style="list-style-type: none"> 1. C++ Builds 2. Debug and Release Profile builds 3. Module Per Driver builds
MCAL -7697	PDK: SCIClient Library	P3- Mediu m	Run the example applications.	Tests	Comm on	MCUSW demo application shall depend on SCICLIENT to instruct DMSEC to power up & allocate the peripheral to MCUSS.
MCAL -7686	MCAL Modules can be hosted on MCU1_0 and MCU2_1	P3- Mediu m	Qualification Tests on both cores	Tests	Comm on	AUTOSAR stack could be hosted on MCU 1 Core 0, which is MCU domain. MCAL modules shall support this implementation. AUTOSAR Stack could potentially be hosted on MCU 2 Core 1, which is in main domain. MCAL modules shall executable from MCU 2 Core 1

Key	Summary	Priori ty	Verification Description	Verificati on Method	Compo nent/s	Description
						An user guide shall be provided that details the steps require to support MCAL on other R5F cores (MCU 20, MCU 30 MCU 31). The user guide can choose one of the cores as reference/example to detail the steps
MCAL-7681	PDK: Utility for Interrupt registration & prints on console	P3-Mediu m	Running the example applications for drivers.	Tests	Comm on	MCUSW demo application and driver demo applications requires to register ISR to interrupts, MCUSW shall depend on PDK utilities for interrupt registration and display result/status on console via print utilities
MCAL-7676	PDK: UDMA Library	P3-Mediu m	UDMA will be used from SDK. Check SPI and ETH drivers.	Tests	Comm on	MCUSW driver and demo application will depend on UDMA to move data from & to peripheral to & from memory. MCUSW shall depend on PDK to provide UDMA libraries that could be used to program UDMA for data movement



7 Requirements not addressed (Is Not)

Requirements are dispositioned and rejected in JIRA. Please refer to JIRA project.



8 Safety Diagnostic Requirements (HSI):

The SoC Hardware Safety Manual serves as the HSI. The document lists the requirements towards SW, so that, the combination of HW safety features and the SW, is able to meet the functional safety requirements assumed by the SoC. These SW safety requirements were analyzed for the MCAL project. The applicable requirements from the HW safety manual, towards SW, are identified as Design Requirements, and listed in the Architecture and Design documents.







9 Records Retention

Records retention follows QSS 016-000.



10 Appendix

11 Template Revision

Author Name	Description	Version	Date
Asger Nielsen	Initial Version	A	 07 Jul 2018
Jon Nafziger	Updates to reflect the expected Requirement Type fields for Auto/FS programs	B	 27 Sep 2018
Jon Nafziger	EPSWDEVCCB-3	C	 27 Nov 2018
Jon Nafziger	EPSWDEVCCB-51 Updating Galileo Location line from EPSWDEVCCB-4 . Updated instructions EPSWDEVCCB-2 .	D	 28 May 2019