**RESTRICTED ACCESS**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **DISTRIBUTION :** | **Firm** | **To** | **Ref** | **Copies** | **1st page** | **e-mail** |
|  | RBE | Mihai Ianos, Daniel Andris, | 2 |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**SW Architecture Design & Interface Description :**

**MMG sw UNIT**

OBJECT: This document is the description of the design & interfaces for *MMG* SW unit.

SUMMARY: This document provides a high-level view of the *MMG* SW unit. The inputs of this document are provided by the software requirement. It is linked to the *DAIMLER\_MMA*\_SWarchitectureDesignInterfaceDescription document.

CONCLUSION: Applicable from **R7.0** SW release

**THIS DOCUMENT CONTAINS HIDDEN TEXT**

EVOLUTION OF THE DOCUMENT

|  |  |  |  |
| --- | --- | --- | --- |
| **Issue** | **Date** | **Author** | **Motive and nature of the modifications** |
| A00 | 31/08/2016 | C. Redon | First release (extract from the full PP4G architecture document) |
| A01 | 23/09/2016 | J. Lacroix | Update of requirement reference |
| Start extended description based on mainstream document | | | | |
| 1.1.1.2 | 12/07/2019 | A. Vaché | Update traceability to match PP4G extended platform requirements IDs |
| 1.1.1.3 | 08/08/2019 | A. Vaché | Solve some traceability issues highlighted by reqtify |
| Start DAI MMA Project Specific Documentation | | | | |
| 1.1.4.1 | 06/01/2022 | A. Negrea | First revision |
| 1.1.4.2 | 06/01/2022 | A. Negrea | Duplicate revision |
| 1.1.4.3 | 31/01/2022 | A. Negrea | Add DTC qualification |
| 1.1.4.4 | 03/02/2022 | A. Negrea | Update traceability |
| 1.1.4.5 | 21/02/2022 | A. Negrea | Update after review |
| 1.1.4.6 | 08/03/2022 | A. Negrea | Add cycle validity check |
| 1.1.4.7 | 17/03/2022 | A. Negrea | Update according to SRM |
| 1.1.4.8 | 09/06/2022 | M. Obada | Update traceability |
| 1.1.4.9 | 14/06/2022 | M. Obada | Update according SRM |
| 1.1.4.10 | 16/06/2022 | M. Obada | Add fix after review |
| 1.1.4.11 | 16/06/2022 | M. Obada | Update chapter 1.5 and 1.6 |
| 1.1.4.12 | 11/08/2022 | S. Dominte | Update for 3.0 release. |
| 1.1.4.13 | 18/08/2022 | M. Obada | Update static diagram |
| 1.1.4.14 | 24/08/2022 | M. Obada | Fix findings from review |
| 1.1.4.15 | 26/08/2022 | M. Obada | Fix findings from review |
| 1.1.4.16 | 26/08/2022 | M. Obada | Update first page |
| 1.1.4.16 | 29/08/2022 | S. Dominte | Added missing traceability |
| 1.1.4.17 | 29/08/2022 | S. Dominte | Added missing traceability |
| 1.1.4.18 | 08/11/2022 | A. Negrea | Update for 4.0 release |
| 1.1.4.19 | 08/11/2022 | A. Negrea | Update after reqtify reports |
| 1.1.4.20 | 18/11/2022 | A. Negrea | Update after review |
| 1.1.4.21 | 13/12/2022 | S. Dominte | Update for 5.0 release |
| 1.1.4.22 | 06/02/2023 | S. Dominte | Update the called functions list |
| 1.1.4.23 | 07/02/2023 | T. Gligor | Update the called functions list |
| 1.1.4.24 | 13/02/2023 | T.Gligor | Update after review |
| 1.1.4.25 | 13/02/2023 | S. Dominte | Update requirements ID’s |
| 1.1.4.26 | 02/03/2023 | A. Negrea | Update context for EOL LOW FORCE |
| 1.1.4.27 | 06/03/2023 | S. Dominte | Update context for ECU DEFECTIVE |
| 1.1.4.28 | 07/03/2023 | S. Dominte | Update static view |
| 1.1.4.29 | 08/03/2023 | S. Dominte | Re-update static view |
| 1.1.4.30 | 19/06/2023 | M. Serban | Update for 6.1 release |
| 1.1.4.31 | 21/06/2023 | M. Serban | Update after review |
| 1.1.4.32 | 17/07/2023 | A. Negrea | Add cycle validity |
| 1.1.4.33 | 10/01/2024 | M. Obada | Update enable conditions(inhibition) for autotests. |

This document contains **23** pages.

Peer Review associated to this document: [MMG - Design Interface Description Peer Review Workbook](http://ALVA-MKS01.alv.autoliv.int:7001/siro/viewrevision?projectName=%23p=e:/MKSProjects/SBE/PP/AUDI_TR6/AUDI_TR6.pj%23Phase_01/View_Development/Architectures/Application%23Quality_Assurance/Peer_Review&revision=1.1&selection=MMG%20-%20Design%20Interface%20Description%20Peer%20Review%20Workbook.xls)

**CONTENTS**

1. Documentation 5

1.1. Upper Level Relevant Documents 5

1.2. Design interface description Documents 6

1.3. Design Specification Documents 8

1.4. Tier2 Documents 8

1.5. HW Datasheet 9

1.6. Other Documents 9

1.7. Glossary And Definition 10

2. Description 11

3. Technical functions 11

4. Runnables 14

4.1. MMG\_runUpdateModeStatus 14

4.1.1. Definition 14

4.1.2. Data flow / Parameters 15

4.1.3. Called functions 18

4.2. MMG\_Init 19

4.2.1. Definition 19

4.2.2. Data flor / Parameters 19

4.2.3. Called functions 20

4.3. MMG\_runCheckModeStatus 20

4.3.1. Definition 20

4.4. MMG\_runManageModeDelays 20

4.4.1. Definition 20

4.4.2. Called functions 20

4.5. MMG\_runPreSafeRecorder 21

4.5.1. Definition 21

4.5.2. Called functions 21

4.6. MMG\_runUpdateNVMBlocks 22

4.6.1. Definition 22

4.6.2. Called functions 22

5. MCU resources 22

**TABLE OF FIGURES**

[Figure 1: MMG Static Description 11](#_Toc129161995)

# Documentation

## Upper Level Relevant Documents

This section presents all the documents needed to write the software architecture design document.

This section presents all the documents needed to write the software architecture design document.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nb** | **Document** | **Reference** | **Company** |
|  | TF-A: To Manage the power supply | /RevAS/30\_DES\_Requirements/Technical Functions/  DES\_TF\_A\_To\_Manage\_The\_Power\_Supply | RBE/FCE |
|  | TF-B: To Manage the communication | /RevAS/30\_DES\_Requirements/Technical Functions/  DES\_TF\_B\_To\_Manage\_The\_Communication | RBE/FCE |
|  | TF-C: To Secure PP ECU functioning using Pictus MCU | /RevAS/30\_DES\_Requirements/Technical Functions/  DES\_TF\_C\_To\_Secure\_PP\_ECU\_Functioning\_Pictus | RBE/FCE |
|  | TF-D: To Program MCU | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_D\_To\_Program\_MCU | RBE/FCE |
|  | TF-E: To Manage Diagnostic Requests | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_E\_To\_Manage\_Diagnostic\_Requests | RBE/FCE |
|  | TF-F: To Perform Measurements | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_F\_To\_Perform\_Measurements | RBE/FCE |
|  | TF-G: To Drive the Motor | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_G\_To\_Drive\_the\_Motor | RBE/FCE |
|  | TF-H: To Perform Autotests | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_H\_To\_Perform\_Autotests | RBE/FCE |
|  | TF-I: To Manage the Failure | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_I\_To\_Manage\_The\_Failure | RBE/FCE |
|  | TF-J: To Manage NVM - NVP (Non Volatile Parameters) | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_J\_To\_Manage\_NVM | RBE/FCE |
|  | TF-K: To Ensure ECU Protection and Integration | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_K\_To\_Ensure\_ECU\_Protection\_And\_Integration | RBE/FCE |
|  | TF-L: To Ensure ECU Integration in Environment EMC ESD | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_L\_To\_Ensure\_ECU\_Integration\_In\_Environment\_EMC\_ESD | RBE/FCE |
|  | TF-M: To generate time base | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_M\_To\_Generate\_Time\_Base | RBE/FCE |
|  | TF-N: To evaluate belt data | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_N\_To\_Evaluate\_Belt\_Data | RBE/FCE |
|  | TF-O: To schedule the SW | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_O\_To\_Run\_SW | RBE/FCE |
|  | TF-P: To handle network management | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_P\_To Handle\_Network\_Management | RBE/FCE |
|  | TF-Q: To Provide Data For Expertise | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_Q\_To\_Provide\_Data\_For\_Expertise | RBE/FCE |
|  | TF-R: To Decide Belt Function Execution | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_R\_To\_Decide\_Belt\_Function\_Execution | RBE/FCE |
|  | TF-S: To drive the boost | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_S\_To\_Drive\_Boost | RBE/FCE |
|  | TF-X: To generate time base | /RevAS/30\_DES\_Requirements/Technical Functions/DES\_TF\_M\_To\_Generate\_Time\_Base | RBE/FCE |

## Design interface description Documents

This section presents all the documents that are linked to this software architecture design document.

Note: All links are related to S:\drive, to have them functional, please mount the S:\drive on your DAI MMA sandbox.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nb** | **Document** | **Reference** | **Company** |
|  | EEPROM parameters | SBE\_4G\_NVP\_layout.xls | RBE/FCE |
|  | Design Interface description of AdcIf | N/A | RBE/FCE |
|  | Design Interface Description of Auto Tests Manager | [ATM – Design Interface Description.docx](ATM%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design Interface Description of Belt Function Decision | N/A | RBE/FCE |
|  | Design Interface Description of Belt Function Execution | [BFE - Design Interface Description.docx](file:///S:\Architectures\Application\Description\Associated_Documents\BFE%20-%20Design%20Interface%20Description%20.docx) | RBE/FCE |
|  | Design Interface Description of Belt Function Selection | [BFS - Design Interface Description.docx](file:///S:\Architectures\Application\Description\Associated_Documents\BFS%20-%20Design%20Interface%20Description%20.docx) | RBE/FCE |
|  | Design Interface Description of Belt Movement Monitoring | N/A | RBE/FCE |
|  | Design Interface Description of Belt Parking Algorithm | N/A | RBE/FCE |
|  | Design Interface Description of Belt Slack Reduction | [BSR - Design Interface Description.docx](file:///S:\Architectures\Application\Description\Associated_Documents\BFS%20-%20Design%20Interface%20Description%20.docx) | RBE/FCE |
|  | Design Interface Description of Basic Software Manager | N/A | RBE/FCE |
|  | Design Interface Description of Basic Software Manager Interface | N/A | RBE/FCE |
|  | Design Interface Description of Can Tranceiver Interface | N/A | RBE/FCE |
|  | Design Interface Description of Communication Interaction Layer | [CIL - Design Interface Description.docx](file:///S:\Architectures\Application\Description\Associated_Documents\CIL%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design Interface Description of Diagnostic Communication Manager Interface | N/A | RBE/FCE |
|  | Design Interface Description of Diagnostic Event Manager Interface | N/A | RBE/FCE |
|  | Design Interface Description of DiagOnCAN services management | [DIA - Design Interface Description.docx](file:///S:\Architectures\Application\Description\Associated_Documents\DIA%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design Interface Description of Electronic Control Unit Manager | N/A | RBE/FCE |
|  | Design Interface Description of Electronic Control Unit Manager Interface | N/A | RBE/FCE |
|  | Design Interface Description of End of life | [EOL - Design Interface Description.docx](file:///S:\Architectures\Application\Description\Associated_Documents\DIA%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design Interface Description of Error Handler | [ERH – Design Interface Description.docx](ERH%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design Interface Description of Haptic Warning | [HWA - Design Interface Description.docx](file:///S:\Architectures\Application\Description\Associated_Documents\DIA%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design Interface Description of Memory Integrity Control | N/A | RBE/FCE |
|  | Design Interface Description of Mode Management | [MMG - Design Interface Description.docx](file:///S:\Architectures\Application\Description\Associated_Documents\MMG%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design Interface Description of Network Management Interface | N/A | RBE/FCE |
|  | Design Interface Description of Non-Volatile Memory Interface | N/A | RBE/FCE |
|  | Design Interface Description of Non-Volatile Parameters | [NVP - Design Interface Description.docx](file:///S:\Architectures\Application\Description\Associated_Documents\NVP%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design Interface Description of Operating System Interface | N/A | RBE/FCE |
|  | Design Interface Description of Power Abstraction Layer | [PAL - Design Interface Description.docx](file:///S:\Architectures\Application\Description\Associated_Documents\PAL%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design Interface Description of Pre-Crash Master | N/A | RBE/FCE |
|  | Design Interface Description of Physical Measures Provider | [PMP - Design Interface Description.docx](file:///S:\Architectures\Application\Description\Associated_Documents\PMP%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design Interface Description of Port Interface | N/A | RBE/FCE |
|  | Design Interface Description of Pre Pre-Tensioning | [PRE - Design Interface Description.docx](file:///S:\Architectures\Application\Description\Associated_Documents\PMP%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design Interface Description of Production cycle function | N/A | RBE/FCE |
|  | Design Interface Description of Pulse Width Modulation Interface | N/A | RBE/FCE |
|  | Design Interface Description of Reset Cause Management | [RCM - Design Interface Description.docx](file:///S:\Architectures\Application\Description\Associated_Documents\DIA%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design Interface Description of SBC | N/A | RBE/FCE |
|  | Design Interface Description of System Context Management | N/A | RBE/FCE |
|  | Design Interface Description of Standard Function Recovery (releasing function) | [SFR - Design Interface Description.docx](file:///S:\Architectures\Application\Description\Associated_Documents\SFR%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design Interface Description of Serial Peripheral Interface Interface | N/A | RBE/FCE |
|  | Design Interface Description of Startup | N/A | RBE/FCE |
|  | Design Interface Description of System Time Management | N/A | RBE/FCE |
|  | Design Interface Description of Vehicle Dynamics algorithm | N/A | RBE/FCE |

## Design Specification Documents

This section presents all the documents that complete this software architecture design document.

Note: All links are related to S:\drive, to have them functional, please mount the S:\drive on your sandbox.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nb** | **Document** | **Reference** | **Company** |
|  | Design document of AdcIf | N/A | RBE/FCE |
|  | Design document of Auto Tests Manager | N/A | RBE/FCE |
|  | Design document of Belt Function Decision | N/A | RBE/FCE |
|  | Design document of Belt Function Execution | [BFE - Detailed Design Document.docx](file:///S:\Architectures\Application\Description\Associated_Documents\BFE%20-%20Design%20Interface%20Description%20.docx) | RBE/FCE |
|  | Design document of Belt Function Selection | [BFS - Detailed Design Document.docx](file:///S:\Architectures\Application\Description\Associated_Documents\BFS%20-%20Design%20Interface%20Description%20.docx) | RBE/FCE |
|  | Design document of Belt Movement Monitoring | N/A | RBE/FCE |
|  | Design document of Belt Parking Algorithm | N/A | RBE/FCE |
|  | Design document of Basic Software Manager Interface | [BSR - Detailed Design Document.docx](file:///S:\Architectures\Application\Description\Associated_Documents\BFS%20-%20Design%20Interface%20Description%20.docx) | RBE/FCE |
|  | Design document of Communication Interaction Layer | [CIL - Detailed Design Document.docx](file:///S:\Architectures\Application\Description\Associated_Documents\CIL%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design document of Diagnostic Communication Manager Interface | N/A | RBE/FCE |
|  | Design document of Diagnostic Event Manager Interface | N/A | RBE/FCE |
|  | Design document of DiagOnCAN services management | [DIA - Detailed Design Document.docx](file:///S:\Architectures\Application\Description\Associated_Documents\DIA%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design document of End of life | [EOL - Detailed Design Document.docx](file:///S:\Architectures\Application\Description\Associated_Documents\BFS%20-%20Design%20Interface%20Description%20.docx) | RBE/FCE |
|  | Design document of Error Handler | N/A | RBE/FCE |
|  | Design document of Haptic Warning | [HWA - Detailed Design Document.docx](file:///S:\Architectures\Application\Description\Associated_Documents\BFS%20-%20Design%20Interface%20Description%20.docx) | RBE/FCE |
|  | Design document of Memory Integrity Control | N/A | RBE/FCE |
|  | Design document of Mode Management | [MMG - Detailed Design Document.docx](file:///S:\Architectures\Application\Description\Associated_Documents\MMG%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design document of Network Management Interface | N/A | RBE/FCE |
|  | Design document of Non-Volatile Memory Interface | N/A | RBE/FCE |
|  | Design document of Non-Volatile Parameters | [NVP - Detailed Design Document.docx](file:///S:\Architectures\Application\Description\Associated_Documents\NVP%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design document of Power Abstraction Layer | [PAL - Detailed Design Document.docx](file:///S:\Architectures\Application\Description\Associated_Documents\PAL%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design document of Physical Measures Provider | [PMP - Detailed Design Document.docx](file:///S:\Architectures\Application\Description\Associated_Documents\PMP%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design document of Port Interface | N/A | RBE/FCE |
|  | Design document of Production cycle function | N/A | RBE/FCE |
|  | Design document of Reset Cause Management | [RCM - Detailed Design Document.docx](file:///S:\Architectures\Application\Description\Associated_Documents\PMP%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design document of RTE If | N/A | RBE/FCE |
|  | Design document of System Context Management | N/A | RBE/FCE |
|  | Design document of Standard Function Recovery (releasing function) | [SFR - Detailed Design Document.docx](file:///S:\Architectures\Application\Description\Associated_Documents\SFR%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design document of Serial Peripheral Interface Interface |  | RBE/FCE |

## Tier2 Documents

This section presents all the documents that complete this software architecture design document.

|  |  |  |  |
| --- | --- | --- | --- |
| Nb | **Document** | **Reference** | **Company** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## HW Datasheet

This section presents all the documents related to the HW components that complete this software architecture design document.

|  |  |  |  |
| --- | --- | --- | --- |
| Nb | **Document** | **Reference** | **Company** |
| [G1] | Infineon-TLE9471-3ES datasheet | TLE9461-3ES-Infineon.pdf | Infineon |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Other Documents

This section presents all the documents that also have been needed to write this software architecture design document.

|  |  |  |  |
| --- | --- | --- | --- |
| Nb | **Document** | **Reference** | **Company** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Glossary And Definition

This section presents all the definitions and/or abbreviations used in this document.

*List of terms in alphabetical order:*

|  |  |
| --- | --- |
| ***Term*** | ***Meaning*** |
| ADC | Analog Digital Converter |
| AEC | Autoliv Error Code |
| API | Application Programming Interface |
| ASDM | Active Safety Domain Master |
| ASIC | Application Specific Integrated Circuit |
| ASY | Active SafetY |
| BSW | Basic SW modules |
| CAN | Controller Area Network |
| C/S | Chip Select |
| COP | Computer Operating Properly |
| eCPL | Electronic Crash Pole Locking |
| DART | Ditch - Airborne - Rough Terrain |
| DFLASH | Data FLASH |
| ECC | Error Code Correction |
| ECU | Electronic Control Unit |
| EOL | End of Life |
| EEPROM | Electric Erasable and Programmable Read only Memory |
| HFPP | High Force Pre-Pre-Tensioning belt function |
| HF-PRE | High Force PRE pre-tensioning |
| HR | Hard Releasing |
| I/O | Input/Output |
| IMU | Inartial Measurements Unit |
| ISS | Integrated Safing System |
| LFPP | Low Force Pre-Pre-Tensioning belt function |
| MSA | Motor Start/Stop Automatic |
| MCAL | Micro-Controller Abstraction Layer |
| MCU | Micro-controller Unit |
| NMG | Mode ManaGement |
| NVM | Non-Volatile Memory |
| OS | Operating System |
| PCM | Pre-Crash Master |
| PFLASH | Program FLASH |
| PIT | Periodic Interrupt Timer |
| PLL | Phase-locked loop |
| RAM | Random Access Memory |
| RCWM | Rear Collision Warning and Mitigation |
| RML | Left PP ECU |
| RMR | Right PP ECU |
| RMx | Both PP ECU |
| ROM | Read Only Memory |
| RSU | Remote Sensor Unit |
| RTE | Real Time Environment |
| RTOS | Real Time Operating System |
| SFR | Standard Function Recovery |
| SODL | Side Obstacle Detection Left |
| SPI | Serial Peripheral Interface |
| SRS | Supplementary Restraint System |
| TBC | To be confirmed |
| TBD | To be defined |
| TF | Technical Function |
| TFLASH | Test FLASH of the Pictus MCU (“one time programmable” memory) |
| W/D | Watchdog |

# Description

MMG is intended to factorize the monitoring of some specific system/product situations (like “under voltage”, OVER-voltage, etc.) which will lead to the inhibition of some product functionalities (like auto-tests execution, belt functions execution).

These situations will be so called “Mode”.

Pratically, SW units will check if modes (aka situations) are active (check will be done in one call, see the runnable section below).

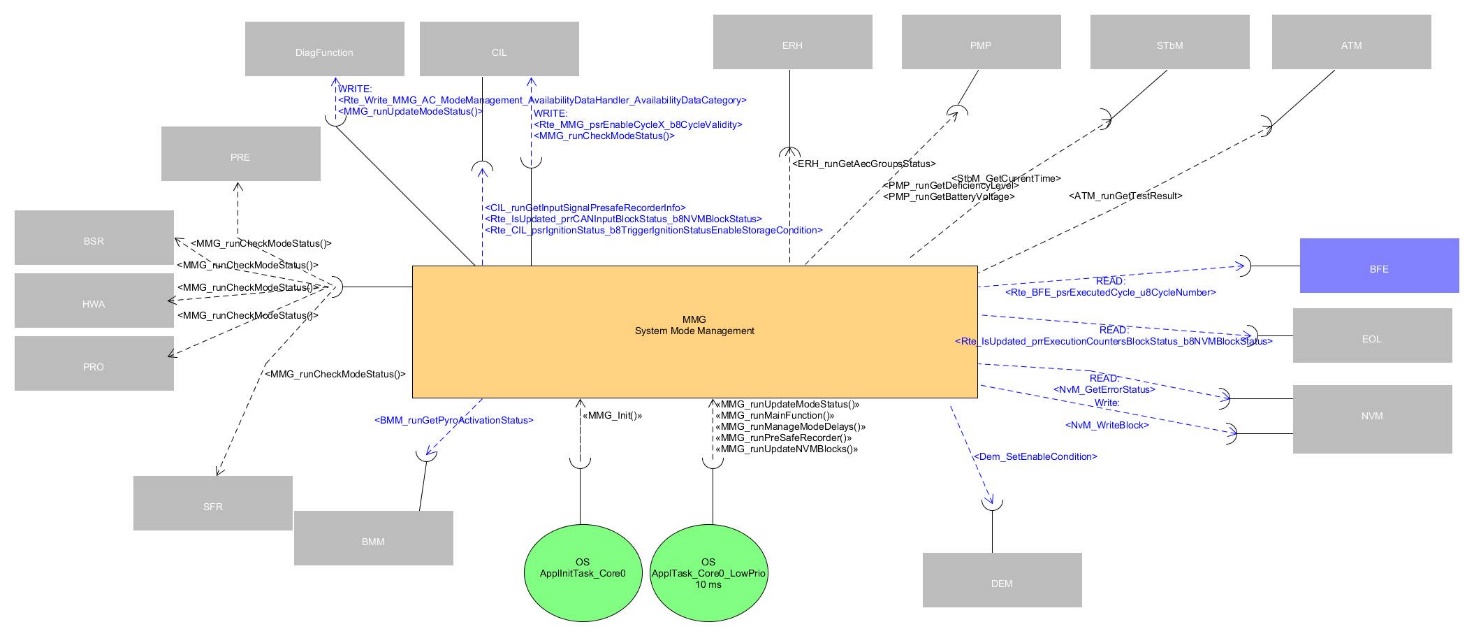


Figure : MMG Static Description

# Technical functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Linked Runnable** | **Source** |
| **ARCH\_SW\_MMG\_0004** | Context should be periodically updated in order to inhibit/abort cycles. | MMG\_runUpdateModeStatus | DAI\_EXT\_TF\_R\_2665; DAI\_EXT\_TF\_R\_2667; DAI\_EXT\_TF\_R\_2668; DAI\_EXT\_TF\_R\_2677; DAI\_EXT\_TF\_R\_2696; DAI\_EXT\_TF\_R\_2581; DAI\_EXT\_TF\_R\_2440; DAI\_EXT\_TF\_R\_2482 |
| **ARCH\_SW\_MMG\_0006** | This module shall inform other modules about the requested mode status. | MMG\_runCheckModeStatus | DAI\_EXT\_TF\_R\_2666; DAI\_EXT\_TF\_R\_2526; DAI\_EXT\_TF\_R\_2678; DAI\_EXT\_TF\_R\_2696; DAI\_EXT\_TF\_R\_2582 |
| **ARCH\_SW\_MMG\_0007** | Modes for delayed executed cycle should be managed by this module | MMG\_runManageModeDelays |  |
| **ARCH\_SW\_MMG\_0008** | All cycles shall be inhibitted if they or their steps are not valid . | MMG\_Init | DAI\_EXT\_TF\_R\_2458 |
| **ARCH\_SW\_MMG\_0009** | The **Start Block** of the PRE-SAFE Recorder Datashall be manipulated and saved in RAM when a requested profile has started regardless of status of execution: inhibited or started successful. | MMG\_runPreSafeRecorder | DAI\_EXT\_TF\_Q\_149; DAI\_EXT\_TF\_Q\_138; DAI\_EXT\_TF\_Q\_143; DAI\_EXT\_TF\_Q\_147; DAI\_EXT\_TF\_Q\_195; |
| **ARCH\_SW\_MMG\_0010** | The **Stop Block** of the PRE-SAFE Recorder Datashall be manipulated and saved in RAM when a requested profile has ended regardless of status of execution: ended successful, aborted or inhibited. | MMG\_runPreSafeRecorder | DAI\_EXT\_TF\_Q\_121; DAI\_EXT\_TF\_Q\_139; DAI\_EXT\_TF\_Q\_144; DAI\_EXT\_TF\_Q\_148; |
| **ARCH\_SW\_MMG\_0011** | The **State Block** of the PRE-SAFE Recorder Datashall be manipulated and saved in RAM whenever a start or stop block is saved. | MMG\_runPreSafeRecorder | DAI\_EXT\_TF\_Q\_122; DAI\_EXT\_TF\_Q\_137; DAI\_EXT\_TF\_Q\_142; DAI\_EXT\_TF\_Q\_146; |
| **ARCH\_SW\_MMG\_0012** | The PRE-SAFE Recorder 1/2 Data shall be saved in NVM as a rolling buffer containing the data of the last two profiles requested when the ECU goes to sleep. | MMG\_runPreSafeRecorder | DAI\_EXT\_TF\_Q\_182; DAI\_EXT\_TF\_Q\_183; DAI\_EXT\_TF\_Q\_184 |
| **ARCH\_SW\_MMG\_0013** | The writing of a PRE-SAFE Recorder data should be tiggered if a PreSafe profile(PRE-SAFE ≥ 1 and PRE-SAFE ≤ 8 except PRE-SAFE = 2) is requested regardless of status of execution: inhibited, aborted or executed successful. | MMG\_runPreSafeRecorder | DAI\_EXT\_TF\_Q\_128; DAI\_EXT\_TF\_Q\_129; |
| **ARCH\_SW\_MMG\_0014** | The writing of a PRE-SAFE Recorder data should be tiggered if a PreSafe profile (PRE-SAFE ≥ 1 and PRE-SAFE ≤ 8 except PRE-SAFE = 2) is requested regardless of type of request: PreSafe, API CAN frames or a UDS diagnostic request. | MMG\_runPreSafeRecorder | DAI\_EXT\_TF\_Q\_128; DAI\_EXT\_TF\_Q\_130;  DAI\_EXT\_TF\_Q\_206; |
| **ARCH\_SW\_MMG\_0016** | **Buckle Switch State** shall represent the **0-1** BitsofByte **0** from **Start** and **Stop Blocks** and shall be saved with the following values:  “STAT\_OK = 0” if the buckle state is buckled  "STAT\_NOT = 1” if the buckle state is unbuckled  "STAT\_FLT = 2” if the buckle state is fault  "STAT\_SNA = 3” if the signal is not available | MMG\_runPreSafeRecorder | DAI\_EXT\_TF\_Q\_154; DAI\_EXT\_TF\_Q\_163; |
| **ARCH\_SW\_MMG\_0017** | **Belt Hand Over State** shall represent the **2-3** BitsofByte **0** from **Start** and **Stop Blocks** and shall be saved with the following values:  “RETRACT = 0” if the belt hand over is retracted  "OUT\_OF\_END = 1” if the belt hand over is out of end position  "EXTEND = 2” if the belt hand over is extended  "SNA = 3” if the signal is not available | MMG\_runPreSafeRecorder | DAI\_EXT\_TF\_Q\_155; DAI\_EXT\_TF\_Q\_164; |
| **ARCH\_SW\_MMG\_0018** | **Pre-Safe Level** shall represent the **0-3** BitsofByte**1** from **Start** and **Stop Blocks** and shall be saved with the following values:  “None = 0” No pre-safe requested  "PRE-SAFE 1 = 1” pre-safe 1 requested  "PRE-SAFE 3 = 3” pre-safe 3 requested  "PRE-SAFE 4 = 4” pre-safe 4 requested  "PRE-SAFE 5 = 5” pre-safe 5 requested  "PRE-SAFE 6 = 6” pre-safe 6 requested  "PRE-SAFE 7 = 7” pre-safe 7 requested  "PRE-SAFE 8 = 8” pre-safe 8 requested  “Not Defined 9-13” pre-safe not defined  "SNA = 15” if the signal is not available | MMG\_runPreSafeRecorder | DAI\_EXT\_TF\_Q\_156; DAI\_EXT\_TF\_Q\_165; |
| **ARCH\_SW\_MMG\_0019** | **Impact X** shall represent the **4** BitofByte **1** from **Start** and **Stop Blocks** and shall be saved with the following values:  “No = 0” if no Impact X Crash conditions detected  "Yes = 1” if Impact X Crash conditions detected | MMG\_runPreSafeRecorder | DAI\_EXT\_TF\_Q\_157; DAI\_EXT\_TF\_Q\_166; |
| **ARCH\_SW\_MMG\_0020** | **Type 1 Roll Over Event** shall represent the **6** BitofByte **1** from **Start** and **Stop Blocks** and shall be saved with the following values:  “No = 0” if no Type 1 Roll Over Crash conditions detected  "Yes = 1” if Type 1 Roll Over Crash conditions detected | MMG\_runPreSafeRecorder | DAI\_EXT\_TF\_Q\_158; DAI\_EXT\_TF\_Q\_167; |
| **ARCH\_SW\_MMG\_0021** | **Type 2 Roll Over Event** shall represent the **7** BitofByte **1** from **Start** and **Stop Blocks** and shall be saved with the following values:  “No = 0” if no Type 2 Roll Over Crash conditions detected  "Yes = 1” if Type 2 Roll Over Crash conditions detected | MMG\_runPreSafeRecorder | DAI\_EXT\_TF\_Q\_159; DAI\_EXT\_TF\_Q\_168; |
| **ARCH\_SW\_MMG\_0022** | **KL30 Voltage** shall represent the Byte **2** from **Start** and **Stop Blocks** and shall be saved with values between 0x00 and 0xFF. | MMG\_runPreSafeRecorder | DAI\_EXT\_TF\_Q\_160; DAI\_EXT\_TF\_Q\_169; |
| **ARCH\_SW\_MMG\_0023** | **ECU Temperature** shall represent the Bytes **3-4** from **Start** and **Stop Blocks** and shall be saved with the following values:  **Byte 3:**  “Low-temperature – 0x00” if deficiency level < 1200  “Middle-temperature – 0x01” if deficiency level >= 1200 and deficiency level < 32000  “High-temperature – 0x02” if deficiency level >= 32000 and deficiency level <150000  “Hot-temperature – 0x03” if deficiency level >= 150000 and deficiency level <160000  “Over-temperature – 0x04” if deficiency level >= 160000  “Invalid – 0xFF” Value not available  **Byte 4**: will always be 0 | MMG\_runPreSafeRecorder | DAI\_EXT\_TF\_Q\_199; DAI\_EXT\_TF\_Q\_201; |
| **ARCH\_SW\_MMG\_0024** | **System Timpe** shall represent the Bytes **5-9** from **Start** and **Stop Blocks** and shall be saved with the following values:  **Byte 5-8:** will represent the current operating time represented in seconds  **Byte 9**: will always be 0 | MMG\_runPreSafeRecorder | DAI\_EXT\_TF\_Q\_203; DAI\_EXT\_TF\_Q\_204; |
| **ARCH\_SW\_MMG\_0025** | **Reserve** bits from **Start**, **Stop** and **State** blocks shall always be 0. | MMG\_runPreSafeRecorder | DAI\_EXT\_TF\_Q\_189; DAI\_EXT\_TF\_Q\_191; DAI\_EXT\_TF\_Q\_192; DAI\_EXT\_TF\_Q\_193; DAI\_EXT\_TF\_Q\_194; DAI\_EXT\_TF\_Q\_202; DAI\_EXT\_TF\_Q\_200 |
| **ARCH\_SW\_MMG\_0026** | **Tensioning State** shall represent the **0-1** BitsofByte **0** from **State Block** and shall be saved with the following values:  “Successfull = 0” if a tensioning was started or ended successfull  "Not Started = 1” if a tensioning was not started  “Aborted = 2” if a tensioning was aborted  "Not Defined = 3” not defined | MMG\_runPreSafeRecorder | DAI\_EXT\_TF\_Q\_172 |
| **ARCH\_SW\_MMG\_0027** | **Pre-Safe cycles are disabled** shall represent the **2** BitofByte **0** from **State Block** and shall be saved with the following values:  “No = 0” if the tensioning was not inhibited/aborted by disabled cycles  "Yes = 1” if the tensioning was inhibited/ aborted by disabled cycles | MMG\_runPreSafeRecorder | DAI\_EXT\_TF\_Q\_173 |
| **ARCH\_SW\_MMG\_0028** | **Suppresion of PresfAct\_TensSupp\_Rq\_ST3** shall represent the **3** BitofByte **0** from **State Block** and shall be saved with the following values:  “No = 0” if the tensioning was not inhibited/aborted by PresfAct\_TensSupp\_Rq\_ST3 signal  "Yes = 1” if the tensioning was inhibited/ aborted by PresfAct\_TensSupp\_Rq\_ST3 signal | MMG\_runPreSafeRecorder | DAI\_EXT\_TF\_Q\_174 |
| **ARCH\_SW\_MMG\_0029** | **Failure Message on the bus** shall represent the **4** BitofByte **0** from **State Block** and shall be saved with the following values:  “No = 0” if the tensioning was not inhibited/aborted by a failure message on buss  "Yes = 1” if the tensioning was inhibited/ aborted by a failure message on buss | MMG\_runPreSafeRecorder | DAI\_EXT\_TF\_Q\_175 |
| **ARCH\_SW\_MMG\_0030** | **Ignition rejection** shall represent the **5** BitofByte **0** from **State Block** and shall be saved with the following values:  “No = 0” if the tensioning was not inhibited/aborted by ignition signal  "Yes = 1” if the tensioning was inhibited/ aborted by ignition signal | MMG\_runPreSafeRecorder | DAI\_EXT\_TF\_Q\_176 |
| **ARCH\_SW\_MMG\_0031** | **Undervoltage / ovelvoltage situation** shall represent the **6** BitofByte **0** from **State Block** and shall be saved with the following values:  “No = 0” if the tensioning was not inhibited/aborted by Ov or UV  "Yes = 1” if the tensioning was inhibited/ aborted by Uv or Ov | MMG\_runPreSafeRecorder | DAI\_EXT\_TF\_Q\_177 |
| **ARCH\_SW\_MMG\_0032** | **OverTemperature** shall represent the **0** BitofByte **1** from **State Block** and shall be saved with the following values:  “No = 0” if the tensioning was not inhibited/aborted by overtemperature  "Yes = 1” if the tensioning was inhibited/ aborted by overtemperature | MMG\_runPreSafeRecorder | DAI\_EXT\_TF\_Q\_178 |
| **ARCH\_SW\_MMG\_0033** | **EOL counter rejection** shall represent the **1** BitofByte **1** from **State Block** and shall be saved with the following values:  “No = 0” if the tensioning was not inhibited/aborted by EOL counter limit  "Yes = 1” if the tensioning was inhibited/ aborted by EOL counter limit | MMG\_runPreSafeRecorder | DAI\_EXT\_TF\_Q\_179 |
| **ARCH\_SW\_MMG\_0034** | **System failure** shall represent the **2** BitofByte **1** from **State Block** and shall be saved with the following values:  “No = 0” if the tensioning was not inhibited/aborted by system failure  "Yes = 1” if the tensioning was inhibited/ aborted by system failure | MMG\_runPreSafeRecorder | DAI\_EXT\_TF\_Q\_180 |

# Runnables

## MMG\_runUpdateModeStatus

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| void **MMG\_runUpdateModeStatus** (void) | | | |
| **Object** | | | |
| This function shall periodically update the status of the monitored modes, based on internal and external information. | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| NA | NA | NA | NA |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Periodic 10ms (refer to [RUN schedule table)](#_Hlk411349502" \s "1,600796,600815,5,,RUN schedule table)  Non-reentrant | | | |
| **Covered requirements** | | | |
| ARCH\_SW\_MMG\_0004; ARCH\_SW\_MMG\_0133; ARCH\_SW\_MMG\_0135; ARCH\_SW\_MMG\_0138; ARCH\_SW\_MMG\_0140; ARCH\_SW\_MMG\_0150; ARCH\_SW\_MMG\_0151; ARCH\_SW\_MMG\_0153; ARCH\_SW\_MMG\_0152; ARCH\_SW\_MMG\_0157; ARCH\_SW\_MMG\_0162; ARCH\_SW\_MMG\_0200; ARCH\_SW\_MMG\_0201; ARCH\_SW\_MMG\_0202; ARCH\_SW\_MMG\_0203; ARCH\_SW\_MMG\_0204; ARCH\_SW\_MMG\_0205; ARCH\_SW\_MMG\_0206; ARCH\_SW\_MMG\_0207; ARCH\_SW\_MMG\_0208;  ARCH\_SW\_MMG\_0209; ARCH\_SW\_MMG\_0250; ARCH\_SW\_MMG\_0251; ARCH\_SW\_MMG\_0252; ARCH\_SW\_MMG\_0253; ARCH\_SW\_MMG\_0254; ARCH\_SW\_MMG\_0154; ARCH\_SW\_MMG\_0301; ARCH\_SW\_MMG\_0302 | | | |

### Data flow / Parameters

* **Context MMG\_KU32\_MASK\_CRITICAL\_AT\_NOK**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Linked Runnable** | **Source** |
| **ARCH\_SW\_MMG\_0250** | The startup Autotest status shall be an input for MMG\_KU32\_MASK\_CRITICAL\_AT\_NOK  context qualification. | MMG\_runUpdateModeStatus |  |

* **Context MMG\_KU32\_MASK\_POWER\_VOLTAGE\_UNSTABLE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Linked Runnable** | **Source** |
| **ARCH\_SW\_MMG\_0251** | The Battery voltage survey status shall be an input for MMG\_KU32\_MASK\_POWER\_VOLTAGE\_UNSTABLE context qualification | MMG\_runUpdateModeStatus |  |

* **Context MMG\_KU32\_MASK\_NO\_BELTFUNCTIONS\_DELAY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Linked Runnable** | **Source** |
| **ARCH\_SW\_MMG\_0252** | The executed cycle delay value shall be an input for MMG\_KU32\_MASK\_NO\_BELTFUNCTIONS\_DELAY context qualification | MMG\_runUpdateModeStatus |  |

* **Context MMG\_KU32\_MASK\_NO\_BELTFUNCTIONS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Linked Runnable** | **Source** |
| **ARCH\_SW\_MMG\_0253** | The current execute cycle and pyro activation status shall be an input for MMG\_KU32\_MASK\_NO\_BELTFUNCTIONS context qualification | MMG\_runUpdateModeStatus |  |

* **Context MMG\_KU32\_MASK\_NO\_TENSIONING**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Linked Runnable** | **Source** |
| **ARCH\_SW\_MMG\_0254** | The executed cycle shall be an input for MMG\_KU32\_MASK\_NO\_TENSIONING context qualification | MMG\_runUpdateModeStatus |  |

* **Context MMG\_KU32\_MASK\_NO\_HALL\_EFFECT\_SENSOR**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Linked Runnable** | **Source** |
| ARCH\_SW\_MMG\_0140 | The “To check Hall Efect Sensor" AEC status from ERH shall be an input for MMG\_KU32\_MASK\_NO\_HALL\_EFFECT\_SENSOR context qualification | MMG\_runUpdateModeStatus | DAI\_EXT\_TF\_R\_2615 |

* **Context MMG\_KU32\_MASK\_AEC\_INHIB\_ALL\_CYCLES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Linked Runnable** | **Source** |
| ARCH\_SW\_MMG\_0150 | The “KU32\_AEC\_GROUP\_MASK\_OVER\_VO – To check Over Voltage" AEC group status from ERH shall be an input for MMG\_KU32\_MASK\_AEC\_INHIB\_ALL\_CYCLES context qualification | MMG\_runUpdateModeStatus | DAI\_EXT\_TF\_R\_2668 |
| ARCH\_SW\_MMG\_0151 | The “KU32\_AEC\_GROUP\_MASK\_UNDER\_VO – To check Under Volatge" AEC group status from ERH shall be an input for MMG\_KU32\_MASK\_AEC\_INHIB\_ALL\_CYCLES context qualification | MMG\_runUpdateModeStatus | DAI\_EXT\_TF\_R\_2667; |

* **Context MMG\_KU32\_MASK\_AEC\_ECU\_DEFECT\_ALL\_CYCLES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Linked Runnable** | **Source** |
| ARCH\_SW\_MMG\_0152 | The “KU32\_AEC\_GROUP\_MASK\_HARDWARE – To check ECU is defective" AEC group status from ERH shall be an input for MMG\_KU32\_MASK\_AEC\_ECU\_DEFECT\_ALL\_CYCLES context qualification | MMG\_runUpdateModeStatus | DAI\_EXT\_TF\_R\_2665; DAI\_EXT\_TF\_R\_2674 |
| ARCH\_SW\_MMG\_0153 | The “KU32\_AEC\_GROUP\_MASK\_ACTUATOR\_BLOCKED – To check if the actuator is blocked" AEC group status from ERH shall be an input for MMG\_KU32\_MASK\_AEC\_ECU\_DEFECT\_ALL\_CYCLES context qualification | MMG\_runUpdateModeStatus | DAI\_EXT\_TF\_R\_2738; DAI\_EXT\_TF\_R\_2754 |
| ARCH\_SW\_MMG\_0154 | The Pyro activation status from BMM shall be an input for MMG\_KU32\_MASK\_AEC\_INHIB\_ALL\_CYCLES context qualification | MMG\_runUpdateModeStatus | DAI\_EXT\_TF\_R\_2846;  DAI\_EXT\_TF\_R\_2847 |

* **Context MMG\_KU32\_MASK\_** **MMG\_KU32\_MASK\_EOL\_LOW\_FORCE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Linked Runnable** | **Source** |
| ARCH\_SW\_MMG\_0138 | The “KU32\_AEC\_GROUP\_MASK\_EOL\_LOW\_HIGH – to check counters" AEC group status from ERH shall be an input for \_ MMG\_KU32\_MASK\_EOL\_LOW\_FORCE context qualification | MMG\_runUpdateModeStatus | DAI\_EXT\_TF\_R\_2526 |

* **Context MMG\_KU32\_MASK\_AEC\_INHIB\_TENSIONING\_CYCLES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Linked Runnable** | **Source** |
| ARCH\_SW\_MMG\_0133 | The “KU32\_AEC\_GROUP\_MASK\_SELF\_PROTECTION – SW/HW Self protection status" AEC group status from ERH shall be an input for MMG\_KU32\_MASK\_AEC\_INHIB\_TENSIONING\_CYCLES context qualification | MMG\_runUpdateModeStatus | DAI\_EXT\_TF\_R\_2666; DAI\_EXT\_TF\_R\_2672 |
| ARCH\_SW\_MMG\_0135 | The “KU32\_AEC\_GROUP\_MASK\_UNDER\_VO\_TENS – to check undervoltage " AEC group status from ERH shall be an input for MMG\_KU32\_MASK\_AEC\_INHIB\_TENSIONING\_CYCLES context qualification | MMG\_runUpdateModeStatus | DAI\_EXT\_TF\_R\_2675 |
| ARCH\_SW\_MMG\_0157 | The “KU32\_AEC\_GROUP\_MASK\_OVER\_VO\_TENS – to check overvoltage during tensioning" AEC group status from ERH shall be an input for MMG\_KU32\_MASK\_AEC\_INHIB\_TENSIONING\_CYCLES context qualification | MMG\_runUpdateModeStatus | DAI\_EXT\_TF\_R\_2676 |
| ARCH\_SW\_MMG\_0158 | The “KU32\_AEC\_GROUP\_MASK\_IMPLAUSIBLE\_DATA\_BUCKLE – to check implausible data for buckle signal" AEC group status from ERH shall be an input for MMG\_KU32\_MASK\_AEC\_INHIB\_TENSIONING\_CYCLES context qualification | MMG\_runUpdateModeStatus | DAI\_EXT\_TF\_R\_2748; DAI\_EXT\_TF\_R\_2760 |
| ARCH\_SW\_MMG\_0159 | The “KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_BUCKLE – to check timout for buckle signal" AEC group status from ERH shall be an input for MMG\_KU32\_MASK\_AEC\_INHIB\_TENSIONING\_CYCLES context qualification | MMG\_runUpdateModeStatus | DAI\_EXT\_TF\_R\_2749; DAI\_EXT\_TF\_R\_2761 |
| ARCH\_SW\_MMG\_0160 | The “KU32\_AEC\_GROUP\_MASK\_STEERING\_CONFIGURATION – to check control unit configuration " AEC group status from ERH shall be an input for MMG\_KU32\_MASK\_AEC\_INHIB\_TENSIONING\_CYCLES context qualification | MMG\_runUpdateModeStatus | DAI\_EXT\_TF\_R\_2746 |

* **Context MMG\_KU32\_MASK\_EOL\_COMFORT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Linked Runnable** | **Source** |
| ARCH\_SW\_MMG\_0162 | The “KU32\_AEC\_GROUP\_MASK\_EOL\_COMFORT – To check EOL comfort counter" AEC group status from ERH shall be an input for MMG\_KU32\_ MASK\_EOL\_COMFORT context qualification | MMG\_runUpdateModeStatus | DAI\_EXT\_TF\_R\_2704 |

* **Context MMG\_KU32\_MASK\_SIGNAL\_RELEASE\_INHIBITION**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Linked Runnable** | **Source** |
| ARCH\_SW\_MMG\_0301 | The Release Inhibition signal from CIL shall be an input for MMG\_KU32\_MASK\_SIGNAL\_RELEASE\_INHIBITION context qualification | MMG\_runUpdateModeStatus |  |

* **Context MMG\_KU32\_MASK\_ENABLE\_IGNITION\_ON**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Linked Runnable** | **Source** |
| ARCH\_SW\_MMG\_0320 | The Ignition signal from CIL shall be an input for MMG\_KU32\_MASK\_ENABLE\_IGNITION\_ON context qualification | MMG\_runUpdateModeStatus | DAI\_EXT\_TF\_I\_2870; |

* **Context MMG\_KU32\_MASK\_ENABLE\_CODING\_API**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Linked Runnable** | **Source** |
| ARCH\_SW\_MMG\_0321 | The VehicleEquipemnt parameter for API activation be an input for MMG\_KU32\_MASK\_ENABLE\_CODING\_API context qualification | MMG\_runUpdateModeStatus | DAI\_EXT\_TF\_I\_2870; |

* **Context MMG\_KU32\_MASK\_ENABLE\_CODING\_RBTMFL**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Linked Runnable** | **Source** |
| ARCH\_SW\_MMG\_0322 | The Vehicle Equipemnt parameter for coding RBTMFL/RBTMFR be an input for MMG\_KU32\_MASK\_ENABLE\_CODING\_RBTMFL context qualification | MMG\_runUpdateModeStatus | DAI\_EXT\_TF\_I\_2870; |

* **Context MMG\_KU32\_MASK\_ENABLE\_CODING\_RBTMFR**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Linked Runnable** | **Source** |
| ARCH\_SW\_MMG\_0323 | The Vehicle Equipemnt parameter coding RBTMFL/RBTMFR be an input for MMG\_KU32\_MASK\_ENABLE\_CODING\_RBTMFr context qualification | MMG\_runUpdateModeStatus | DAI\_EXT\_TF\_I\_2870; |

### Called functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Linked Runnable** | **Source** |
| ARCH\_SW\_MMG\_0200 | ATM\_runGetLastCriticalAutotestId shall be called to get last critical autotest ID | MMG\_runUpdateModeStatus |  |
| ARCH\_SW\_MMG\_0201 | ATM\_runGetTestResult shall be called to get the status of the last executed autotest | MMG\_runUpdateModeStatus |  |
| ARCH\_SW\_MMG\_0202 | PMP\_runGetBatteryVoltageSurveyStatus shall be called to get the battery voltage | MMG\_runUpdateModeStatus |  |
| ARCH\_SW\_MMG\_0203 | Rte\_Read\_prrExecutedCycle\_u8CycleNumber shall be called to get the executed cycle for different contexts | MMG\_runUpdateModeStatus |  |
| ARCH\_SW\_MMG\_0204 | BMM\_runGetPyroActivationStatus shall be called to get pyro activation status | MMG\_runUpdateModeStatus |  |
| ARCH\_SW\_MMG\_0205 | ERH\_runGetAecGroupsStatus shall be called to get de status of the AEC for different contexts | MMG\_runUpdateModeStatus |  |
| ARCH\_SW\_MMG\_0206 | BMM\_runEnableHBSMonitoring shall be called to notify BMM about restoring the puls counting | MMG\_runUpdateModeStatus |  |
| ARCH\_SW\_MMG\_0207 | PMP\_runGetBatteryVoltage\_10ms shall be called to read the KL30 battery voltage to check if measured battery voltage is outside Extended Operating range | MMG\_runUpdateModeStatus | DAI\_EXT\_TF\_R\_2755 |
| ARCH\_SW\_MMG\_0208 | PMP\_runGetBatteryVoltage\_10ms shall be called to read the KL30 battery voltage to check if measured battery voltage is outside Normal Operating range | MMG\_runUpdateModeStatus | DAI\_EXT\_TF\_R\_2747 |
| ARCH\_SW\_MMG\_0209 | Rte\_Write\_MMG\_AC\_ModeManagement\_AvailabilityDataHandler\_AvailabilityDataCategory shall be called to send the AvailabilityDataCategory towards DiagFunction | MMG\_runUpdateModeStatus | DAI\_EXT\_TF\_B\_2082 |
| ARCH\_SW\_MMG\_0302 | Rte\_Read\_prrCustomerSpecific\_b8SignalRelInhibition shall be called to read release inhibition for costumer signals | MMG\_runUpdateModeStatus |  |

## MMG\_Init

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| void **MMG\_Init** (void) | | | |
| **Object** | | | |
| This function shall initialize MMG. | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| NA | NA | NA | NA |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Synchronous server operation  Non-reentrant | | | |
| **Covered requirements** | | | |
| **ARCH\_SW\_MMG\_0008; ARCH\_SW\_MMG\_0300;** | | | |

### Data flor / Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Linked Runnable** | **Source** |
| **ARCH\_SW\_MMG\_0310** | If the first step from a cycle is 0xFF then cycle is invalid | MMG\_Init() | ALV\_EXT\_TF\_G\_67 |
| **ARCH\_SW\_MMG\_0311** | If Cycle number to execute is not in the permitted range (35 cycles) then cycle is invalid | MMG\_Init() | ALV\_EXT\_TF\_G\_68 |
| **ARCH\_SW\_MMG\_0312** | If a cycle has Week or YEAR equal to 0xFF then cycle is invalid | MMG\_Init() | ALV\_EXT\_TF\_G\_68 |
| **ARCH\_SW\_MMG\_0313** | If step number is not in the permitted range (145 steps) then step is invalid | MMG\_Init() | ALV\_EXT\_TF\_G\_70 |
| **ARCH\_SW\_MMG\_0314** | If step duration is over 655350 ms then step is invalid | MMG\_Init() | ALV\_EXT\_TF\_G\_72 |
| **ARCH\_SW\_MMG\_0315** | If Step order is not a valid one (not 0x00: PWM order ; 0x01: Current order ; 0x02: Velocity Control order type; 0x03: Velocity Control + backup PWM order type; 0x04: Voltage order type; 0x05: Velocity Control + backup Current order type; 0x06: Velocity Control + backup Voltage order type ) then step is invalid | MMG\_Init() | ALV\_EXT\_TF\_G\_73 |

### Called functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Linked Runnable** | **Source** |
| **ARCH\_SW\_MMG\_0300** | Write Rte\_MMG\_psrEnableCycleX\_ | MMG\_Init() | DAI\_EXT\_TF\_R\_2458; DAI\_EXT\_TF\_R\_2459 |

## MMG\_runCheckModeStatus

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| void **MMG\_runCheckModeStatus** (u32ModeMaskType u32ModesToCheck, u8ModeStatusType \*bModeStatus) | | | |
| **Object** | | | |
| This function shall indicate if the mode (passed in argument to the function) is active or not.  Remark: This service allows to check modes status in one call | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| u32ModesToCheck | u32ModeMaskType | In | Modes to be checked (bits field) |
| bModeStatus | u8ModeStatusType | Out | Status of modes passed in argument |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Synchronous server operation  Reentrant | | | |
| **Covered requirements** | | | |
| ARCH\_SW\_MMG\_0006 | | | |

## MMG\_runManageModeDelays

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| void **MMG\_runManageModeDelays** (void) | | | |
| **Object** | | | |
| This function shall manage delays used to update mode status after specified time. | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| NA | NA | NA | NA |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Periodic – 10ms  Non-reentrant | | | |
| **Covered requirements** | | | |
| ARCH\_SW\_MMG\_0400; ARCH\_SW\_MMG\_0401; ARCH\_SW\_MMG\_0402; | | | |

### Called functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Linked Runnable** | **Source** |
| ARCH\_SW\_MMG\_0400 | RCM\_runGetResetCause shall be called to get the reset cause | MMG\_runManageModeDelays () |  |
| ARCH\_SW\_MMG\_0402 | BMM\_runGetPyroActivationStatus shall be called to get the pyro activation status | MMG\_runManageModeDelays () |  |
| ARCH\_SW\_MMG\_0401 | Rte\_BFE\_psrExecutedCycle\_u8CycleNumber shall be called to get cycle number. | MMG\_runManageModeDelays () |  |

## MMG\_runPreSafeRecorder

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| q**Prototype** | | | |
| void MMG\_runPreSafeRecorder(void) | | | |
| **Object** | | | |
| Function is in charge handling the mode management of Presafe recorder | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| NA | NA | NA | NA |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Periodic – 10ms  Non-reentrant | | | |
| **Covered requirements** | | | |
| ARCH\_SW\_MMG\_0408; ARCH\_SW\_MMG\_0403; ARCH\_SW\_MMG\_0404; ARCH\_SW\_MMG\_0405; ARCH\_SW\_MMG\_0406; ARCH\_SW\_MMG\_0009; ARCH\_SW\_MMG\_0010; ARCH\_SW\_MMG\_0011; ARCH\_SW\_MMG\_0012; ARCH\_SW\_MMG\_0013; ARCH\_SW\_MMG\_0014; ARCH\_SW\_MMG\_0015; ARCH\_SW\_MMG\_0016; ARCH\_SW\_MMG\_0017; ARCH\_SW\_MMG\_0018; ARCH\_SW\_MMG\_0019; ARCH\_SW\_MMG\_0020; ARCH\_SW\_MMG\_0021; ARCH\_SW\_MMG\_0022; ARCH\_SW\_MMG\_0023; ARCH\_SW\_MMG\_0024; ARCH\_SW\_MMG\_0025; ARCH\_SW\_MMG\_0026; ARCH\_SW\_MMG\_0027; ARCH\_SW\_MMG\_0028; ARCH\_SW\_MMG\_0029; ARCH\_SW\_MMG\_0030; ARCH\_SW\_MMG\_0031; ARCH\_SW\_MMG\_0032; ARCH\_SW\_MMG\_0033; ARCH\_SW\_MMG\_0034; ARCH\_SW\_MMG\_0407; | | | |

### Called functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Linked Runnable** | **Source** |
| ARCH\_SW\_MMG\_0403 | ERH\_runGetAecGroupsStatus shall be called in order to get the AEC group status | MMG\_runPreSafeRecorder() |  |
| ARCH\_SW\_MMG\_0404 | CIL\_runGetInputSignalPresafeRecorderInfo shall be called in order to get PreSafe Recorder Info data from CIL module | MMG\_runPreSafeRecorder() |  |
| ARCH\_SW\_MMG\_0405 | PMP\_runGetDeficiencyLevel shall be called in order to get the Deficiency level input | MMG\_runPreSafeRecorder() |  |
| ARCH\_SW\_MMG\_0406 | PMP\_runGetBatteryVoltage shall be called in order to get the battery voltage | MMG\_runPreSafeRecorder() |  |
| ARCH\_SW\_MMG\_0407 | StbM\_GetCurrentTime shall be called in order to get the system time | MMG\_runPreSafeRecorder() |  |
| ARCH\_SW\_MMG\_0408 | ATM\_runGetTestResult shall be called in order to get the autotest result for EOL counters. | MMG\_runPreSafeRecorder() |  |

## MMG\_runUpdateNVMBlocks

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| FUNC(void, MMG\_AC\_ModeManagement\_CODE) **MMG\_runUpdateNVMBlocks**(void) | | | |
| **Object** | | | |
| The function checks if a specific NVM block needs written and send a request write for the specific block | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| NA | NA | NA | NA |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Periodic – 10ms  Non-reentrant | | | |
| **Covered requirements** | | | |
| ARCH\_SW\_MMG\_0303; ARCH\_SW\_MMG\_0304; ARCH\_SW\_MMG\_0305; ARCH\_SW\_MMG\_0306; ARCH\_SW\_MMG\_0307; ARCH\_SW\_MMG\_0308; ARCH\_SW\_MMG\_0309 | | | |

### Called functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Linked Runnable** | **Source** |
| ARCH\_SW\_MMG\_0303 | NvM\_GetErrorStatus shall be called in order to get the error status for each block | MMG\_runUpdateNVMBlocks() |  |
| ARCH\_SW\_MMG\_0304 | NvM\_WriteBlock shall be called in order to write updated blocks | MMG\_runUpdateNVMBlocks() |  |
| ARCH\_SW\_MMG\_0305 | Rte\_IsUpdated\_prrExecutionCountersBlockStatus\_b8NVMBlockStatus shall be called to get the updated block status for execution counters | MMG\_runUpdateNVMBlocks() | DAI\_EXT\_TF\_J\_161 |
| ARCH\_SW\_MMG\_0306 | Rte\_IsUpdated\_prrCANInputBlockStatus\_b8NVMBlockStatus shall be called to get the updated block status for CAN input signals | MMG\_runUpdateNVMBlocks() | DAI\_EXT\_TF\_B\_2308;  DAI\_EXT\_TF\_B\_2310; |
| ARCH\_SW\_MMG\_0307 | Rte\_IsUpdated\_prrAECsBlockStatus\_b8NVMBlockStatus shall be called to get the updated block status for AECs | MMG\_runUpdateNVMBlocks() |  |
| ARCH\_SW\_MMG\_0308 | Rte\_IrvRead\_MMG\_runUpdateNVMBlocks\_b8PreSafeRecorder1Update shall be called to get the updated block status for PreSafeRecorder1 | MMG\_runUpdateNVMBlocks() |  |
| ARCH\_SW\_MMG\_0309 | Rte\_IrvRead\_MMG\_runUpdateNVMBlocks\_b8PreSafeRecorder2Update shall be called to get the updated block status for PreSafeRecorder2 | MMG\_runUpdateNVMBlocks() |  |

# MCU resources

The following requirements on resource consumption objectives apply to the module/package:

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_MMG\_9997 | The ROM size consumed by this component shall not exceed 2.7K bytes. |  |  |
| ARCH\_SW\_MMG\_9998 | The heap size consumed by this component shall be 4.5k bytes. |  |  |