**RESTRICTED ACCESS**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **DISTRIBUTION :** | **Firm** | **To** | **Ref** | **Copies** | **1st page** | **e-mail** |
|  | AEE-C | Mihai Ianos, Andreea Negrea, Stefan Dominte, Mirela, Obada | 4 |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**SW Architecture Design & Interface Description :**

**DAIMLER MMA**

OBJECT: This document is the main document of the software design for *Audi Tr6*

SUMMARY: This document provides a high-level view of the *DAIMLER MMA* partition. The inputs of this document are provided by the *DAIMLER MMA* software requirement.

CONCLUSION: Applicable from R05.0 SW release

**THIS DOCUMENT CONTAINS HIDDEN TEXT**

EVOLUTION OF THE DOCUMENT

|  |  |  |  |
| --- | --- | --- | --- |
| **Issue** | **Date** | **Author** | **Motive and nature of the modifications** |
| Start extended description based on mainstream document | | | |
| 1.1 | 16/07/2018 | A. Vaché | First update for PP4G extended platform |
| 1.2 | 08/08/2018 | A. Vaché | Add traceability to DES requirements + add description of unconfirmed wake-up |
| Start DAI MMA description based on extended document | | | |
| 1.1 | 16/12/2021 | A. Negrea | First update for DAI MMA |
| 1.2 | 26/01/2022 | A. Negrea | Update architecture doc |
| 1.3 | 31/01/2022 | A. Negrea | Update OS tasks |
| 1.4 | 15/02/2022 | A. Negrea | Update after review |
| 1.5 | 17/03/2022 | A. Negrea | Update according to SRM |
| 1.6  1.7 | 05/04/2022  03/05/2022 | A. Negrea  M. Obada | Update traceability according to SRM  Update ECU state diagrams |
| 1.8 | 07/06/2022 | A. Negrea | Update architecture for 2.0 Release |
| 1.9 | 17/06/2022 | A. Negrea | Update diagram typo |
| 1.10 | 20/06/2022 | A. Negrea | Update diagram for task typo |
| 1.11 | 17/08/2022 | A. Negrea | Update architecture for R3.0 Release |
| 1.12 | 18/08/2022 | A. Negrea | Update diagram for Tasks |
| 1.13 | 09/11/2022 | A. Negrea | Update architecture for R4.0 release |
| 1.14 | 18/11/2022 | A. Negrea | Update after review |
| 1.15  1.16  1.17  1.18  1.19  1.20  1.21 | 06/02/2022  20/07/2023  23/08/2023  24/08/2023  25/08/2023  30/08/2023  16/01/2024 | A. Negrea  M. Obada  M. Obada  M. Obada  M. Obada  M. Obada  M. Obada | Update with ROE on OsTasks  Update with Dem behavior  Update architecture after SRM  Update after SRM  Update after review.  Update after review.  Update for R8.1 |

This document contains **32** pages.

**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 10

1.5. HW Datasheet 10

1.6. Other Documents 10

1.7. Glossary And Definition 11

2. Overview 12

3. DAIMLER MMA partition description 13

3.1. Static description 13

3.1.1. Overview 13

3.1.2. Interfaces with BSW 14

3.1.3. SW Components Definition 16

3.1.4. Mcu memory partitioning 19

3.2. Dynamic Behaviour 20

3.2.1. Overview 20

3.2.2. Managing the ECU states 21

3.2.3. SW scheduling / connection 24

3.2.4. MCU resources 25

4. SW units description 26

**TABLE OF FIGURES**

[**Figure 1: Static overview of the DAIMLER MMA SW partition** 13](#_Toc156309998)

[Figure 2 Memory regions 19](#_Toc156309999)

[**Figure 3: ECU - State machine** 20](#_Toc156310000)

[**Figure 4: ECU\_STARTUP sequence diagram** 21](#_Toc156310001)

[**Figure 5: ECU\_RUN sequence diagram** 23](#_Toc156310002)

[Figure 6: Os Alarms configuration 25](#_Toc156310003)

[**Figure 7: Os - Tasks priority definition** 25](#_Toc156310004)

[Figure 8: ApplInitTask\_Core0 25](#_Toc156310005)

[Figure 9: OsTask\_ALV\_BeltFunctionAlgo 26](#_Toc156310006)

[Figure 10: ApplTask\_Core0\_LowPrio 26](#_Toc156310007)

[**Figure 11: Os – Interrupt Functions priority definition** 26](#_Toc156310008)

# Documentation

## Upper Level Relevant Documents

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 | NA | 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 DAI MMA sandbox.

|  |  |  |  |
| --- | --- | --- | --- |
| Nb | **Document** | **Reference** | **Company** |
|  | Design document of AdcIf | N/A | RBE/FCE |
|  | Design document of Auto Tests Manager | [ATM - 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 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 | [BMM - Detailed Design Document.docx](file:///S:\Architectures\Application\Description\Associated_Documents\BFE%20-%20Design%20Interface%20Description%20.docx) | RBE/FCE |
|  | Design document of Belt Parking Algorithm | N/A | RBE/FCE |
|  | Design document of Basic Software Manager Interface | N/A | RBE/FCE |
|  | Design Document of Belt Slack Reduction | [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 | [DIA - Detailed Design Document.docx](file:///S:\Architectures\Application\Description\Associated_Documents\DIA%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design document of DiagOnCAN services management | [EOL - Detailed Design Document.docx](file:///S:\Architectures\Application\Description\Associated_Documents\BFS%20-%20Design%20Interface%20Description%20.docx) | RBE/FCE |
|  | Design document of End of life | [ERH - 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 | SBE\_4G\_NVP\_layout.xls | 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 | N/A | RBE/FCE |

## Tier2 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 DAI MMA sandbox.

|  |  |  |  |
| --- | --- | --- | --- |
| Nb | **Document** | **Reference** | **Company** |
|  | Reuse Strategy Definition | See PMD |  |
|  | Software Development Plan | See PMD |  |

## HW Datasheet

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

|  |  |  |  |
| --- | --- | --- | --- |
| Nb | **Document** | **Reference** | **Company** |
|  | 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** |
|  | Vector Technical reference | [PTC link](file:///S:\Tools\Vector\Installation\msr_aurix-vMB-BLuC-1.2.1.0) |  |

## 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 |
| API | Application Programming Interface |
| 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 | Inertial 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 |
| ROM | Read Only Memory |
| RSU | Remote Sensor Unit |
| RTE | Real Time Environment |
| RTOS | Real Time Operating System |
| SFR | Standard Function Recovery |
| 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 |

# Overview

This document provides a high-level view of the DAI MMA partition. The inputs of this document are provided by the DAIMLER MMA software requirements (see reference in the Upper Level Relevant Documents section).

It provides the overall architecture of the DAIMLER MMA partition as a set of modules/packages that are linked together through interaction relations.

In the following sections different structural elements are introduced:

* **PARTITION:** It represents an **independent executable unit** like the application and the bootloader. A partition is made of a set of **packages/modules**.
* **PACKAGE/MODULE:** It represents an **independent functional unit** like the EEPROM management, the power management. Each package is made of **components** and provides an **interface** that can be used by the other packages/modules.
* **COMPONENT:** It represents a group of functions/variables belonging to the same package/module.
* **INTERFACE:** It represents a set of functions/variables that can be used by other structural elements defined within the system. When the interface is provided by a package/module, it is used by other packages/modules to access its functions/information. When the interface is provided by a component of a specific package/module, it is used by other components of the same package/module to realize the package/module function. Note that the package/module interface is made of some functions/variables of its components’ interface.

# DAIMLER MMA partition description

## Static description

### Overview

The *Audi Tr6* SW partition is composed of several blocks of SW modules.

From the bottom to the top:

1. In yellow: the MCAL

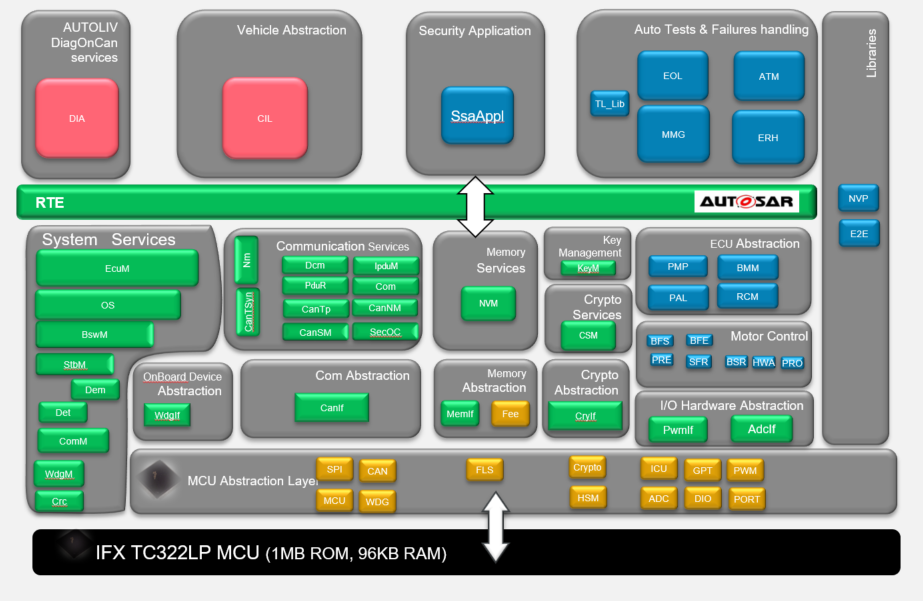
The purpose of the MCAL is to make the rest of the SW application independent from the MCU choice.

1. In green: The BSWs. These modules are delivered by Vector and allow to manage communication, memory, scheduling and RTE in compliance to AUTOSAR 4.4 standard.
2. In rose, the abstraction of the vehicle signals

This is fully car maker specific.

1. In blue: the AUTOLIV SW modules which are not car maker relevant

To sum-up, these components will implement the belt function management ,the auto tests, SSA handling,reset cause management , etc.



**Figure 1: Static overview of the DAIMLER MMA SW partition**

### Interfaces with BSW

#### 3.1.2.1 DEM specific:

The Diagnostic Event Manager (Dem) handles and stores the events detected by diagnostic monitors in both Software Components (SW-Cs) and Basic software (BSW) modules. The stored event information is available via an interface to other BSW modules or SW-Cs.

SW-Cs report faults to DEM via service ports and DEM gets the freeze frame information from other SW-Cs. The Dem module uses the EventId to manage the status of the ‘Diagnostic Event’ of a system and performs the required actions for individual test results, e.g. stores the freeze frame.

The software component responsible for reporting to Dem is ERH (Error handler module). As is shown figure 1: **Static overview of the DAIMLER MMA,** ERH reports to Dem trough RTE by setting Event to passed or failed.

DEM communicates with NVM to read and write fault information and event data to NV memory and communicates with DCM to send data to the tester.

**Fault storage:**

The ECU's fault management module shall evaluate the DTC-specific Storage Conditions and store a DTC. Storage Conditions are defined as system operating parameters that must be satisfied prior to allowing the retention of failures.

Depending on the Failure Level of a DTC, the ECU shall evaluate the corresponding Storage Conditions.

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_BSW\_0004 | All DTCs shall be reported from Diagnostic Chrono Stack, from the most recent to oldest. |  | DAI\_EXT\_TF\_I\_2751; |
| ARCH\_SW\_BSW\_0005 | In Diagnostic Chrono Stack if a DTC has priority 1, shall not be overwriten by other fault |  | DAI\_EXT\_TF\_I\_2753; |
| ARCH\_SW\_BSW\_0006 | All DTCs shall be stored to NVM on every shutdown |  | DAI\_EXT\_TF\_I\_2754; DAI\_EXT\_TF\_I\_2755 DAI\_EXT\_TF\_I\_2756; |
| ARCH\_SW\_BSW\_0007 | System shall handle qualification of CAN bus error |  | DAI\_EXT\_TF\_H\_1658; |
| ARCH\_SW\_BSW\_0008 | Dem module shall handle the following Storage conditions for DTCs:   * Storage Condition SC 1 = Control DTC * Storage Condition SC 2 = Ignition Status * Storage Condition SC 3 = Local Voltage * Storage Condition SC 4 = System Voltage * Storage Condition SC 5 = Transportation Mode * Storage Condition SC 8 = Communication Status * Storage Condition SC 10 = De-bounce Timer * Storage Condition SC 11 = Vehicle Startup * Storage Condition SC 12 = Production Mode |  |  |
| ARCH\_SW\_BSW\_0009 | If ALL of the following storage conditions are fulfilled, the ECU shall enable ECU Level fault storage else this group shall be disabled:  ⦁ SC1: Control DTC settings  ⦁ SC3: Local Voltage (KL30 input)  ⦁ SC11: Vehicle Startup |  | DAI\_EXT\_TF\_I\_2856 |
| ARCH\_SW\_BSW\_0010 | If ALL of the following storage conditions are fulfilled, the ECU shall enable ECU Level fault storage else this group shall be disabled:  ⦁ SC1: Control DTC settings  ⦁ SC2: Ignition Status  ⦁ SC4: System Voltage  ⦁ SC5: Transportation mode  ⦁ SC8: Communication Status  ⦁ SC10: De-bounce Timer network  ⦁ SC11: Vehicle Startup  ⦁ SC12: production mode |  | DAI\_EXT\_TF\_I\_2857 |
| ARCH\_SW\_BSW\_0011 | If ALL of the following storage conditions are fulfilled, the ECU shall enable ECU Level fault storage else this group shall be disabled:  ⦁ SC1: Control DTC settings  ⦁ SC4: System Voltage  ⦁ SC5: Transportation mode  ⦁ SC8: Communication Status  ⦁ SC10: De-bounce Timer Network w/o KL15  ⦁ SC11: Vehicle Startup  ⦁ SC12: production mode |  | DAI\_EXT\_TF\_I\_2858 |
| ARCH\_SW\_BSW\_0012 | If ALL of the following storage conditions are fulfilled, the ECU shall enable ECU Level fault storage else this group shall be disabled:  ⦁ SC1: Control DTC settings  ⦁ SC2: Ignition Status  ⦁ SC5: Transportation mode  ⦁ SC10: De-bounce Timer Power  ⦁ SC11: Vehicle Startup  ⦁ SC12: production mode |  | DAI\_EXT\_TF\_I\_2859 |
| ARCH\_SW\_BSW\_0013 | For the following DTCs ECU level storage condition shall be applicable:  ⦁ P1CA000  ⦁ B220400  ⦁ B228B49  ⦁ B228B4B  ⦁ B228B71  ⦁ B228B96  ⦁ B228B97  ⦁ B228BFA  ⦁ U112081  ⦁ U112083  ⦁ U112087 |  | DAI\_EXT\_TF\_I\_2851 |
| ARCH\_SW\_BSW\_0014 | For the following DTCs Network Communication storage condition shall be applicable:  ⦁ U011587  ⦁ U012287  ⦁ U015187  ⦁ U015587  ⦁ U020187  ⦁ U020287  ⦁ U041608  ⦁ U042708  ⦁ U044286  ⦁ U045208  ⦁ U188987  ⦁ U188A08 |  | DAI\_EXT\_TF\_I\_2852 |
| ARCH\_SW\_BSW\_0015 | For the following DTCs Network Communication without Clam 15 storage condition shall be applicable:  ⦁ P164456  ⦁ U016887 |  | DAI\_EXT\_TF\_I\_2853 |
| ARCH\_SW\_BSW\_0016 | For the following DTCs Power Distribution storage condition shall be applicable:  ⦁ B210A00  ⦁ B210B00  ⦁ B210D00  ⦁ B210E00  ⦁ B228B16  ⦁ B228B17  ⦁ U122488 |  | DAI\_EXT\_TF\_I\_2854 |
| ARCH\_SW\_BSW\_0017 | The following DTCs do not have any DTC storage condition group value:  ⦁ B1A6857  ⦁ B228B45  ⦁ B228B47 |  | DAI\_EXT\_TF\_I\_2855 |
| ARCH\_SW\_BSW\_0018 | The TestNotCompletedSinceLastClear information for each DTC shall be stored to non-volatile memory between power cycles including battery disconnect. |  | DAI\_EXT\_TF\_I\_2774 |
| ARCH\_SW\_BSW\_0019 | The DTC status shall contain the following bits:  Bit 0 – TestFailed  Bit 3 - Confirmed DTC  Bit 4 – TestNotCompletedSinceLastClear  Bit 5 - TestFailedSinceLastClear |  | DAI\_EXT\_TF\_I\_2759; DAI\_EXT\_TF\_I\_2760; DAI\_EXT\_TF\_I\_2762; DAI\_EXT\_TF\_I\_2763; DAI\_EXT\_TF\_I\_2766; DAI\_EXT\_TF\_I\_2767; DAI\_EXT\_TF\_I\_2769; DAI\_EXT\_TF\_I\_2770; |
| ARCH\_SW\_BSW\_0020 | Dem module shall handle the following Enable conditions for DTCs:   * Ignition ON, * Coding of API function, * Coding of RBTMFL "belt carrier present", Coding of RBTMFR "belt carrier present". |  | DAI\_EXT\_TF\_I\_2867; DAI\_EXT\_TF\_I\_2868; |
| ARCH\_SW\_BSW\_0021 | Only the following DTCs shall have specific Enable Conditions and the rest will have "None" = "ECU Power-up".   * U015187 - Ignition On * U042708 - Ignition On * U044286 - Ignition On * U045208 - Ignition On * U041608 - Ignition On * U188A08 - Ignition On and Coding of API function: yes * U188987 - Ignition On and Coding of API function: yes * U020187 - Coding of RBTMFL "belt carrier present": yes * U020287 - Coding of RBTMFR "belt carrier present": yes |  | DAI\_EXT\_TF\_I\_2870 |

### SW Components Definition

The following table extracted from **Error! Reference source not found.**, lists all the modules included in the Application and their ownership.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id** | **Prefix** | | **Description** | **Functional cluster** |
| **1** | **Adc** | | **Analog to Digital Convertor** | **Microcontroller Abstraction Layer** |
| **2** | **ATM** | | **Autotest Management** | **Autotest and error handling** |
| **3** | **AdcIf** | | **Analog to Digital Convertor Interface** | **Microcontroller Abstraction Layer** |
| **4** | **BFE** | | **Belt Function Execution** | **Motor control** |
| **5** | **BFS** | | **Belt Function Selection** | **Motor control** |
| **6** | **BSR** | | **Belt Slack Reduction** | **Motor control** |
| **7** | **BMM** | | **Belt Movement Monitoring** | **ECU Abstraction Layer** |
| **8** | **BswM** | | **Basic Software Mode Manager** | **System Services Layer** |
| **9** | **Can** | | **Controller Area Network** | **Microcontroller Abstraction Layer** |
| **10** | **CanIf** | | **CAN Interface** | **Com Abstraction Layer** |
| **11** | **CanNm** | | **CAN Network Management** | **Communication Services Layer** |
| **12** | **CanSM** | | **CAN State Manager** | **Communication Services Layer** |
| **13** | **CanTp** | | **CAN Transport** | **ECU Abstraction Layer** |
| **14** | **CanTSyn** | | **Time Syncronization over Can** | **Communication Services Layer** |
| **15** | **CIL** | | **Communication Interaction Layer** | **Application Layer -Vechicle Abstraction** |
| **16** | **Com** | | **Communication Module** | **Communication Services Layer** |
| **17** | **ComM** | | **Communication Manager** | **System Services Layer** |
| **18** | **Crc** | | **Cyclic Redundancy Check** | **System Services Layer** |
| **19** | **Crypto** | | **Crypto Driver** | **Crypto Abstraction** |
| **20** | **CryIf** | | **Crypto Driver Interface** | **Crypto Abstraction** |
| **21** | **CSM** | | **Crypto Service Manager** | **Crypto Services** |
| **22** | **Dcm** | | **Diagnostic Communication Manager** | **Communication Services Layer** |
| **23** | **Dem** | | **Diagnostic Event Manager** | **System Services Layer** |
| **24** | **Det** | | **Development Error Trace** | **System Services Layer** |
| **25** | **DIA** | | **DiagOnCAN** | **Application Layer - Diagnostics** |
| **26** | **Dio** | | **Digital Input Output** | **Microcontroller Abstraction Layer** |
| **27** | **EcuM** | | **Ecu Manager** | **System Services Layer** |
| **28** | **EOL** | | **End of Life module** | **Application Layer - Auto-tests and Error-handling** |
| **29** | **ERH** | | **Error Handler module** | **Application Layer - Auto-tests and Error-handling** |
| **30** | **E2E** | | **End to End protection library** | **Libraries** |
| **31** | **Fee** | | **Flash Emulated EEPROM** | **ECU Abstraction Layer** |
| **32** | **Fls** | | **FLash Driver** | **ECU Abstraction Layer** |
| **33** | **GPT** | | **General Purpose Timer** | **Microcontroller Abstraction Layer** |
| **34** | **HSM** | | **High Security Module** | **Microcontroller Abstraction Layer** |
| **35** | **HWA** | | **Haptic Warning** | **Motor control** |
| **36** | **IpduM** | | **I-PDU Multiplexer** | **Communication Services Layer** |
| **37** | **ICU** | | **Input Capture Unit** | **Microntroller Abstraction Layer** |
| **38** | **KeyM** | | **Key Management module** | **Key Management** |
| **39** | **Mcu** | | **Microcontroller Unit** | **Microcontroller Abstraction Layer** |
| **40** | **MemIf** | | **Memory Interface** | **ECU Abstraction Layer** |
| **41** | **MMG** | | **System Mode Management** | **Application Layer - Auto-tests and Error-handling** |
| **42** | **Nm** | | **Network Management Module** | **Communication Services Layer** |
| **43** | **NvM** | | **Non-Volatile Memory** | **Memory Services Layer** |
| **44** | **NVP** | | **Non-Volatile Parameters** | **Application Layer - Parameters** |
| **45** | **Os** | | **Operating System** | **System Services Layer** |
| **46** | **PAL** | | **Power Abstraction Layer** | **ECU Abstraction Layer** |
| **47** | **PduR** | | **PDU Router** | **Communication Services Layer** |
| **48** | **PMP** | | **Physical Measures Provider** | **ECU Abstraction Layer** |
| **49** | **Port** | | **AUTOSAR implementation of the Port module** | **Microcontroller Abstraction Layer** |
| **50** | **PRE** | | **Pre-Pretensioner** | **Motor control** |
| **51** | **PRO** | | **Production Cycles** | **Motor control** |
| **52** | **Pwm** | | **Pulse Width Modulator** | **Microcontroller Abstraction Layer** |
| **53** | **PwmIf** | | **Pulse Width Modulator Interface** | **Microcontroller Abstraction Layer** |
| **54** | **Rte** | | **Runtime Environment** | **Runtime Environment** |
| **55** | **RCM** | | **Reset Cause Management** | **ECU Abstraction Layer** |
| **56** | **SFR** | | **Standard Function Recovery** | **Motor control** |
| **57** | **SSAppl** | | **Standard Security Architecture Application module** | **Security Application** |
| **58** | **Spi** | | **Serial Port Interface** | **Microcontroller Abstraction Layer** |
| **59** | **SecOC** | | **Secure Onboard Communication** | **Communication Services Layer** |
| **60** | **StbM** | | **Syncronized Time Based module** | **System Services Layer** |
| **61** | **TL\_Lib** | | **Helper functions for MBD modules** | **Application Layer** |
| **62** | **WDG** | **Watchdog** | | **Microcontroller Abstraction Layer** |
| **63** | **WDGIf** | **Watchdog Interface** | | **OnBoard Device Abstraction** |
| **64** | **WdgM** | **Watchdog Manager** | | **System Services Layer** |

Color map: Blue - Application Components; Gray - Component plugins provided by Vector.

### Mcu memory partitioning

The figure below depicts the memory regions of DAIMLER MMA project.

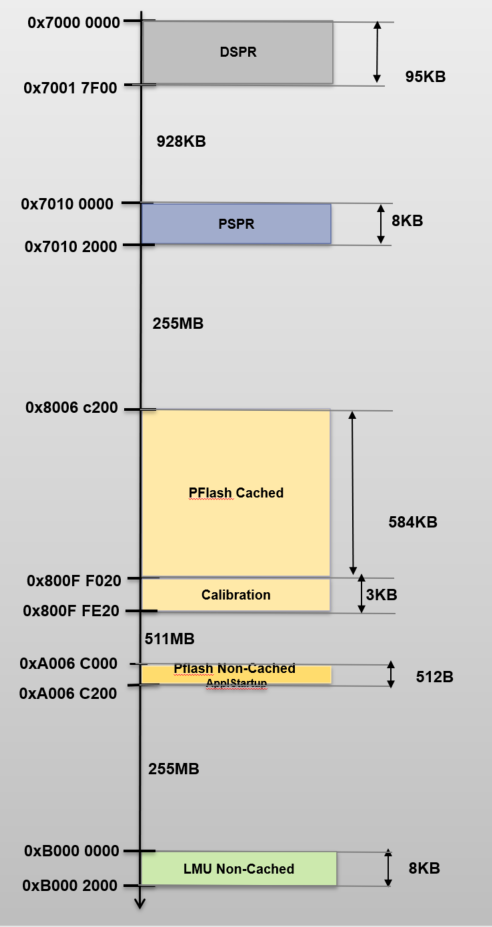


Figure 2 Memory regions

## Dynamic Behaviour

### Overview

This section aims at giving an overview of the main scenarios achieved by the SW.

ECU STATES

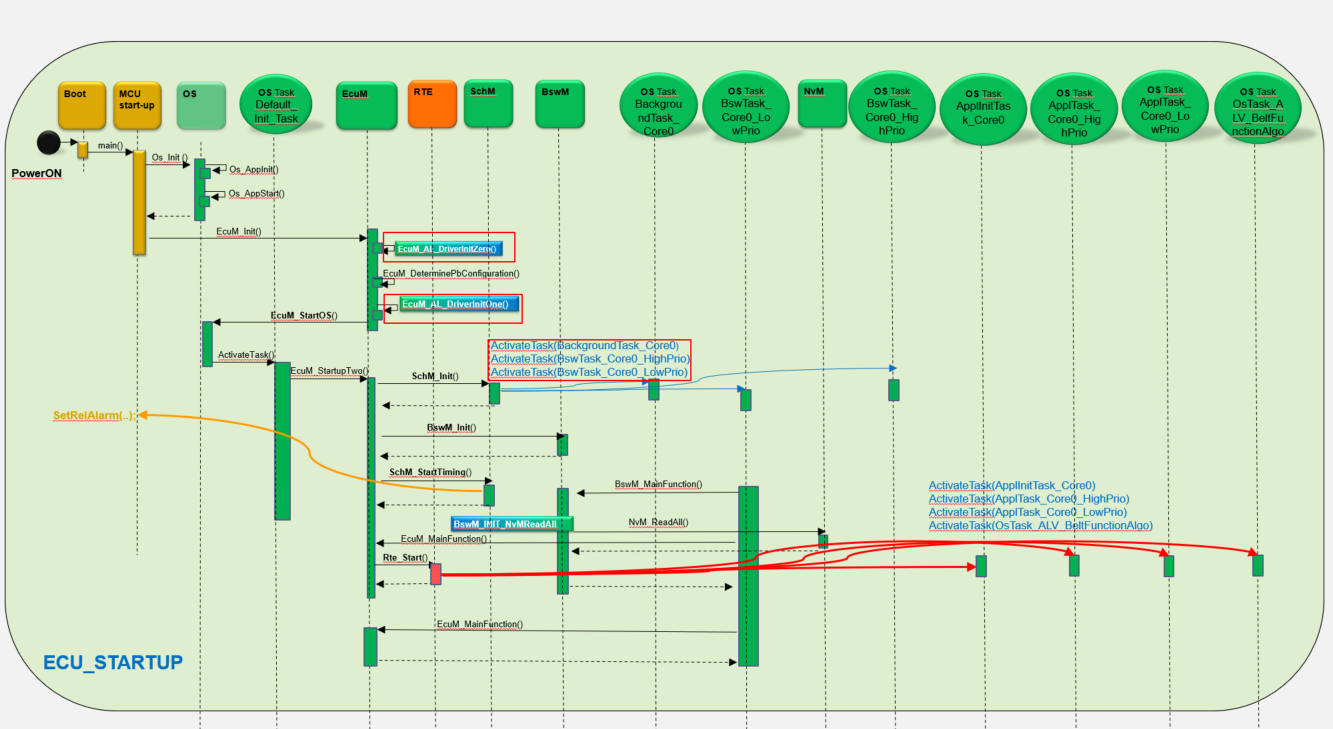
Description automatically generated

**Figure 3: ECU - State machine**

### Managing the ECU states

#### ECU\_STARTUP: Startup sequence description

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| **ARCH\_SW\_MAIN\_0000** | The ECU\_STARTUP state shall be reach if the ECU is supplied, Powered ON. | *See transition 1, Figure 3* | DAI\_EXT\_TF\_B\_1894 |
| **ARCH\_SW\_MAIN\_0001** | Software initialization is done in ECU\_Startup State using AUTOSAR specifications. | *See Figure 4* | DAI\_EXT\_TF\_B\_1894 |
| **ARCH\_SW\_MAIN\_0002** | To Leave ECU\_STARTUP state, Wake -up Validation notification is expected to go ECU\_RUN State. | *See transition 3, Figure 3* | DAI\_EXT\_TF\_B\_1894; DAI\_EXT\_TF\_B\_1895 |
| **ARCH\_SW\_MAIN\_0003** | In this state if POWER OFF occurred the ecu switch to ECU\_OFF state. *See transition 4* | *See transition 2, Figure 3* | DAI\_EXT\_TF\_B\_1894 |

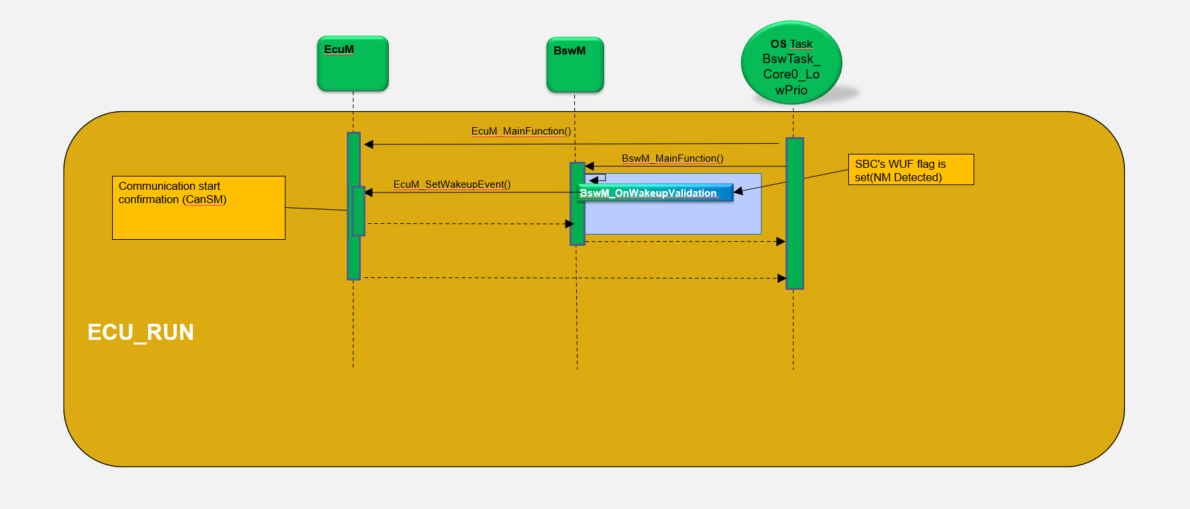


**Figure 4: ECU\_STARTUP sequence diagram**

* The startup sequence is fully specified by AUTOSAR.
* The content of init functions (red boxes above) can be tuned thanks to Vector Configurator tool. It will allow calling the SW initialization functions in accordance with the specified order and real time context.
* The phase #0 of the SW initialization is performed by the EcuM\_AL\_DriverInitZero function. (Note: The side allocation will be identified in this function. It will allow initializing the rest of the SW according to ECU side allocation (left or right / front or rear)).
* The phase #1 of the SW initialization is performed by the EcuM\_AL\_DriverInitOne function. The purpose of this function is to initialize the low level layers of the SW (drivers, CDD, BSW modules…) before starting the OS.
* The phase #2 of the SW initialization is performed in the context of the EcuM. EcuM\_StartupTwo is executed and afterwards Bsw Task are activated , this will allow execution of BswM\_mainFunction.
* As reminder, the phase #2 of the SW initialization sequence is the place to initialize the drivers which need OS service support but not the NVM stack initialization.
* The phase #3 of the SW initialization is performed in the context of the BswM\_MainFunction, which executes the NVMReadAll action list. This is the place in the SW initialization sequence to initialize the drivers which need OS service support and the restoration of the non-volatile parameters.

#### ECU\_RUN : Running sequence description

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| **ARCH\_SW\_MAIN\_0004** | After entering ECU\_RUN the following actions shall be considered:  - Rx CAN communication frames shall be received  - Tx CAN communication frames shall be transmitted  - ECU shall respond to diagnostic frames request  - all autotests shall be periodically executed and their status shall be reported | *See Figure 5* | DAI\_EXT\_TF\_B\_1894; DAI\_EXT\_TF\_B\_1895 |
| **ARCH\_SW\_MAIN\_0005** | If:  -CanSM notifies that no communication is needed anymore(TTimeout = 4 seconds),  -no diagnostics available,  -no belt function execution,  -no bus communication, and  then ECU goes in ECU\_SLEEP state. | *See transition 5, Figure 3* | DAI\_EXT\_TF\_B\_1895; DAI\_EXT\_TF\_B\_1897; DAI\_EXT\_TF\_B\_1896 |
| **ARCH\_SW\_MAIN\_0006** | In this state if POWER OFF occurred the ecu switch to ECU\_OFF state. | *See transition 4, Figure 3* | DAI\_EXT\_TF\_B\_1895 |

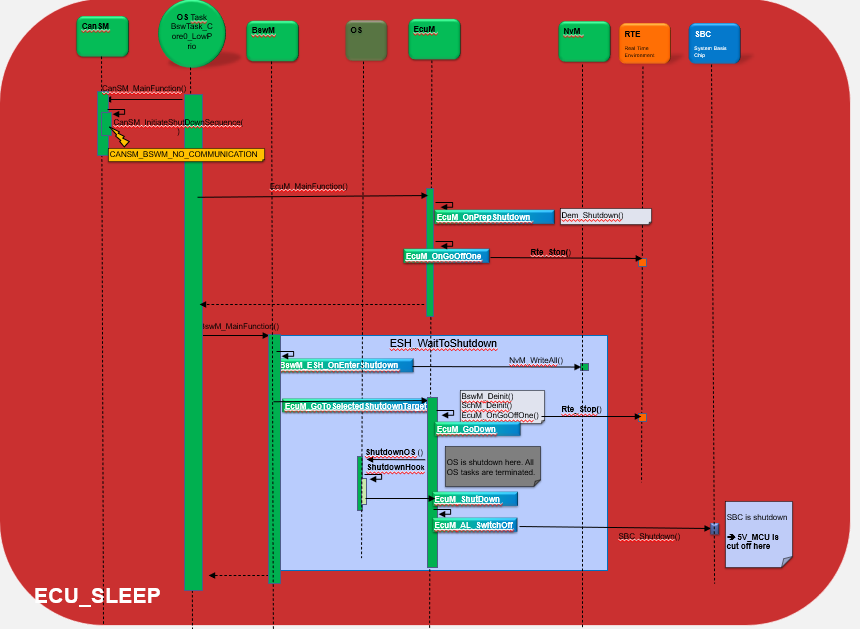


**Figure 5: ECU\_RUN sequence diagram**

* In the context of the EcuM\_MainFunction and BswM\_MainFunction, which executes the BswMActionList\_BswMRule\_WakeUpValidated action list when mode wake-up has been validated () ( first NM frame is received ). This is the place to authorize communication.
* After this, the BswM shall periodically evaluate if the CAN communication stay active.

**ECU\_SLEEP: Shutdown sequence description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| **ARCH\_SW\_MAIN\_0007** | In ECU\_SLEPP state the following actions shall be considered:  - monitor bus activity but no communication shall be ensured. | *See Figure 6* | DAI\_EXT\_TF\_B\_1896 |
| **ARCH\_SW\_MAIN\_0008** | In this state if nm frame is detected then ECU shall switch to ECU\_RUN state. | *See transition 6, Figure 3* | DAI\_EXT\_TF\_B\_1896; DAI\_EXT\_TF\_B\_1895 |
| **ARCH\_SW\_MAIN\_0009** | In this state if bus activity is detected or reset is received then ECU shall switch to ECU\_STARTUP state. | *See transition 8, Figure 3* | DAI\_EXT\_TF\_B\_1896; DAI\_EXT\_TF\_B\_1894 |
| **ARCH\_SW\_MAIN\_0010** | In this stare if POWER OFF occurred the ecu switch to ECU\_OFF state. | *See transition 7, Figure 3* | DAI\_EXT\_TF\_B\_1896 |



**Figure 6: ECU\_SLEEP sequence diagram**

* The diagram above illustrates the function calls flow of the shutdown sequence according to which BswM/EcuM state is executed.
* The content of blue boxes functions can be tuned thanks to Davinci Configurator tool. It will allow calling functions needed to save data and shutdown the ECU.
* The SHUTDOWN sequence performs the following jobs:
* To shut down the RTE
* To write the NvM blocks in DFLASH memory (the ones configured in WriteAll mode)
* To shut down the OS, once all NvM blocks are written in DFLASH memory
* To switch off the MCU power supply once the OS is shutdown
* Before performing these jobs, some actions have to be completed in order to prepare future recording of some data during the NvM\_WriteAll operation of the shutdown sequence.
* The BswM module is in charge of detecting that shutdown sequence shall be started, to perform some actions of this sequence, and then to give back control to EcuM module for the end of the sequence.
* The EcuM module has two roles during the shutdown sequence of the SW:
* Notify BswM module that NvM\_writeAll operation is finished
* Finalize the shutdown sequence (shutdown of scheduler, Os, and requesting MCU power off).
* A large part of the EcuM actions to finalize the shutdown sequence is implemented by Vector EcuM plugin in accordance to the Autosar specification.

### SW scheduling / connection

#### OS tasks

Task monitoring performed by the Operating System is a mechanism performed by the OS is linked to the number of task activation. This error will be set if the activation task counter of a task reaches the maximal value configured for the corresponding task. This counter is incremented each time an “ActivateTask” request is performed for the corresponding task, and decremented each time this task ends (“TerminateTask” request called).

This mechanism allows detecting too long tasks activation, or a blocking of tasks.

This paragraph describes the configuration choices to detect an unexpected behavior of tasks activations.

The maximum number of allowed task activation per task, shall also consider tasks priority and preemption configuration.

The maximum number of activations for each periodic task executed in RUN mode shall be configured to 1. (Even if the tasks have different priorities they should not be blocked/last more than their periods(2ms/5ms/10ms)

The present section will specify the needs in term of OS tasks definition:

* The IdleTask\_OsCore0 task shall be defined at OS configuration level. he idle task has the virtual priority 0xFFFFFFFF to differentiate it from regular tasks. It will be generated to have the lowest priority, even if there are tasks configured with priority 0.
* The Default\_Init\_Task\_Trusted task shall be defined at OS configuration level. The task is started once by the OS. It is separated from Default\_Init\_Task() to allow the SafeContext partitioning UseCase.
* The Default\_Init\_Task task shall be defined at OS configuration level. InitTask to call EcuM\_StartupTwo().
* The ApplInitTask\_Core0 task shall be defined at OS configuration level. This tasks initialise applications modules and ssa.
* The BswTask\_Core0\_HighPrio task shall be defined at OS configuration level. This task shall activate CanTp main functions ,Com, DCM , Sbc main functions
* The ApplTask\_Core0\_HighPrio task shall be defined at OS configuration level. This task shall call SSA main functions
* The BswTask\_Core0\_LowPrio task shall be defined at OS configuration level. This task shall call Nvm, Fee , Fls main functions.
* The OsTask\_ALV\_BeltFunctionAlgo task shall be defined at OS configuration level. This task shall call the Belt Functions modules main functions.
* The ApplTask\_Core0\_LowPrio task shall be defined at OS configuration level. This task shall rest of the application modules main functions.
* The BackgroundTask\_Core0 task shall be defined at OS configuration level.This task shall call KeyM\_MainBackgroundFunction.
* The ApplBackgroundTask\_Core0 task shall be defined at OS configuration level. This task shall call Ssa\_ProcCtrl\_ServiceDispatcher.

#### OS alarm

An OsAlarm may be used to asynchronously inform or activate a specific task

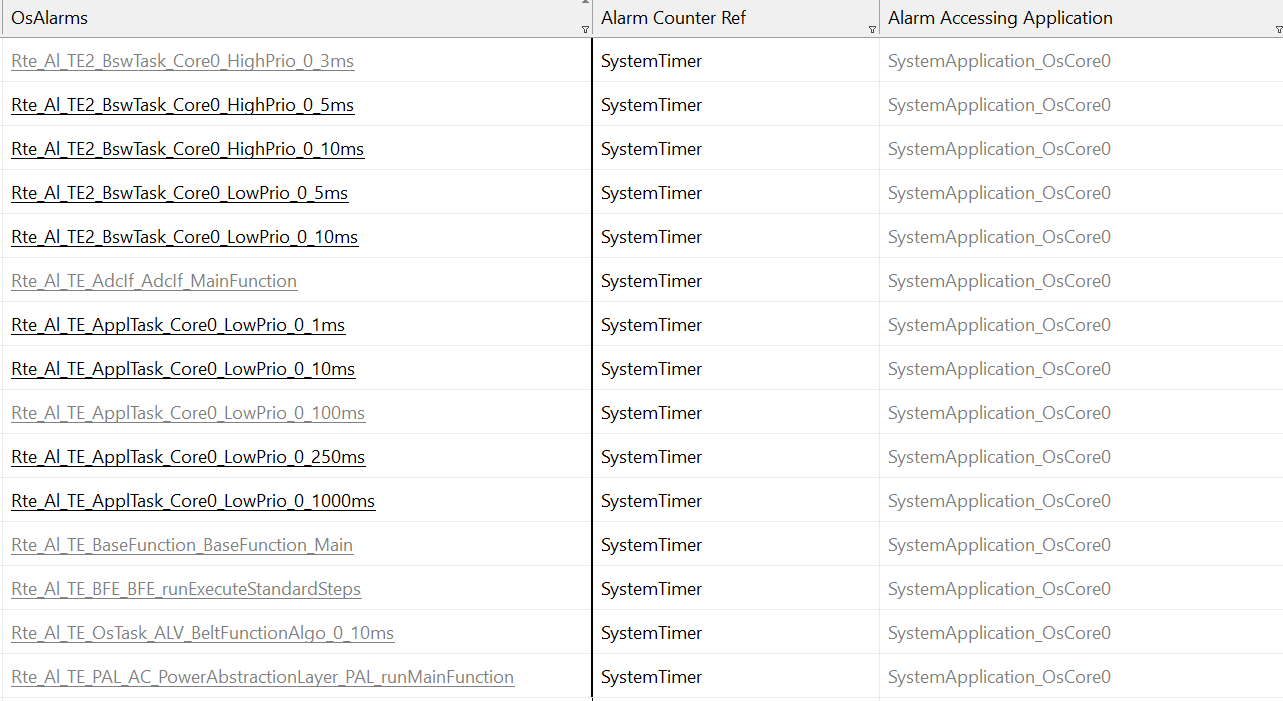
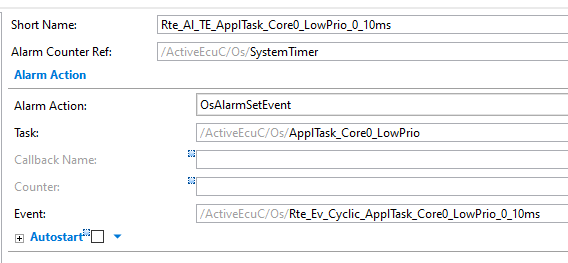


Figure 6: Os Alarms configuration

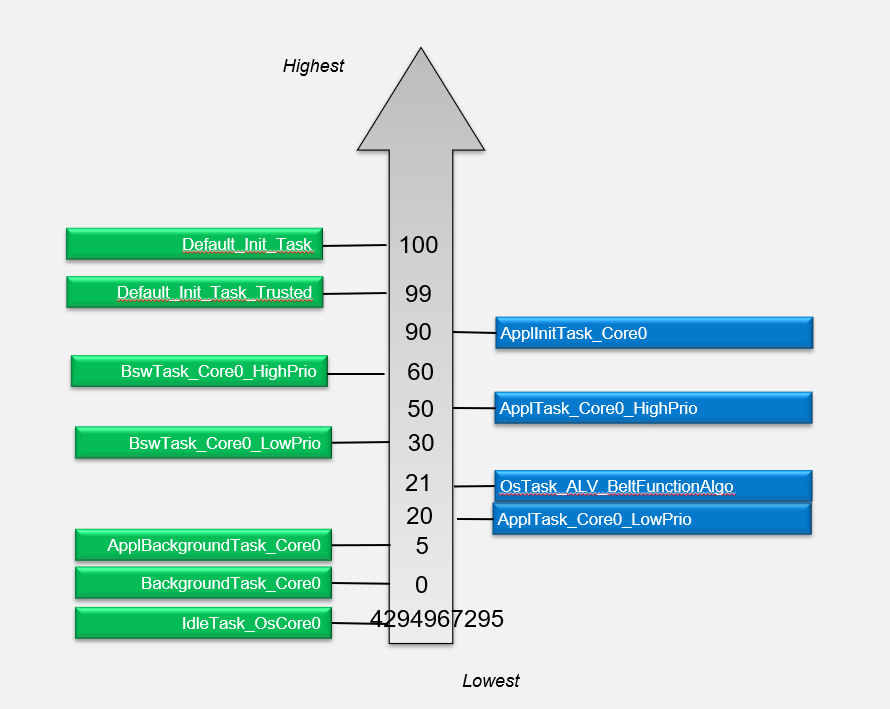
Each alarm has an event.



Each main functions mapped to a task is implemented by an event.

#### Task priority

The next table will specify the needs in term of task priority.



**Figure 7: Os - Tasks priority definition**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_OS\_0050 | The Default\_Init\_Task task shall have the highest priority regarding periodic tasks.  Justification:  Priority is given to the power management. | The task priority shall be 100. |  |
| ARCH\_SW\_OS\_0051 | The priority of the Default\_Init\_Task\_Trusted task shall also have a high priority for init functions. | The task priority shall be 99. |  |
| ARCH\_SW\_OS\_0052 | The priority of the ApplInitTask\_Core0 shall be also higher sincer the applications init functions are called here, | The task priority shall be 90. |  |
| ARCH\_SW\_OS\_0053 | The priority of the BswTask\_Core0\_HighPrio task shall be lower than the init tasks. | The task priority shall be 60. |  |
| ARCH\_SW\_OS\_0054 | The priority of the ApplTask\_Core0\_HighPrio task shall be lower than the BswTask\_Core0\_HighPrio task one. | The task priority shall be 50. |  |
| ARCH\_SW\_OS\_0057 | The priority of the BswTask\_Core0\_LowPrio shall be lower than higher priorirty tasks. | The task priority shall be 30. |  |
| ARCH\_SW\_OS\_0058 | The priority of the OsTask\_ALV\_BeltFunctionAlgo task shall be bigger than ApplTask\_Core0\_LowPrio task | The task priority shall be 21. |  |
| ARCH\_SW\_OS\_0065 | The priority of the ApplTask\_Core0\_LowPrio task shall be lower than all perioric tasks, except wdg monitoring one  Justification:  The priority should be lower because is called several times | The task priority shall be 20. |  |
| ARCH\_SW\_OS\_0059 | The priority of the ApplBackgroundTask\_Core0 shall be low. | The task priority shall be 5. |  |
| ARCH\_SW\_OS\_0060 | The priority of the BackgroundTask\_Core0 task shall be low. | The task priority shall be 0. |  |
| ARCH\_SW\_OS\_0061 | The priority of the IdleTask\_OsCore0 idle task has the virtual priority 0xFFFFFFFF to differentiate it from regular tasks. It will be generated to have the lowest priority, even if there are tasks configured with priority 0. | The task priority shall be 4,294,967,295. |  |

The following detailed os tasks are presented in order to see detailes only about application (AUTOLIV components) main functions periodicity and order.

#### ApplInitTask\_Core0

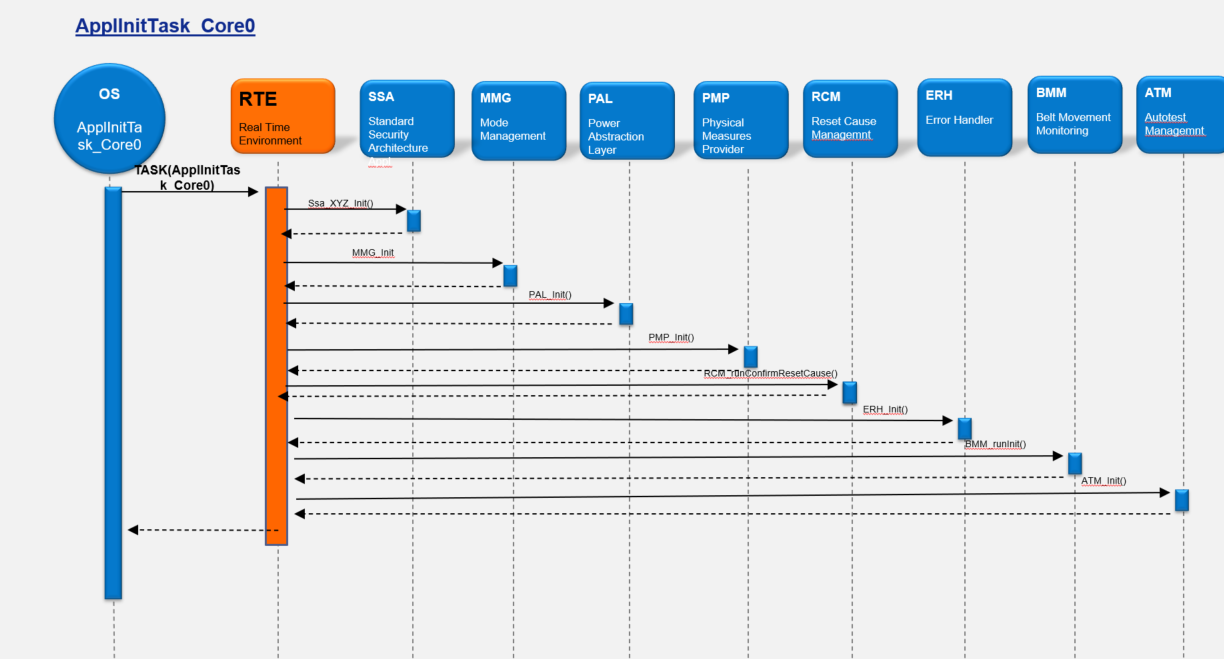


Figure 8: ApplInitTask\_Core0

#### OsTask\_ALV\_BeltFunctionAlgo

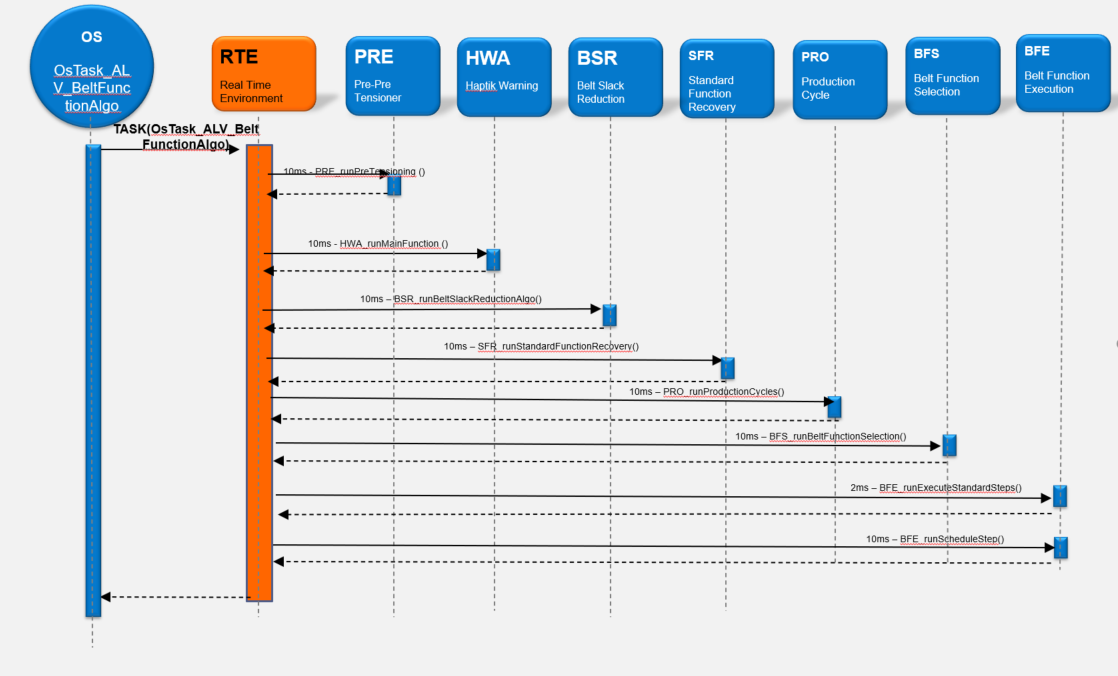


Figure 9: OsTask\_ALV\_BeltFunctionAlgo

#### ApplTask\_Core0\_LowPrio

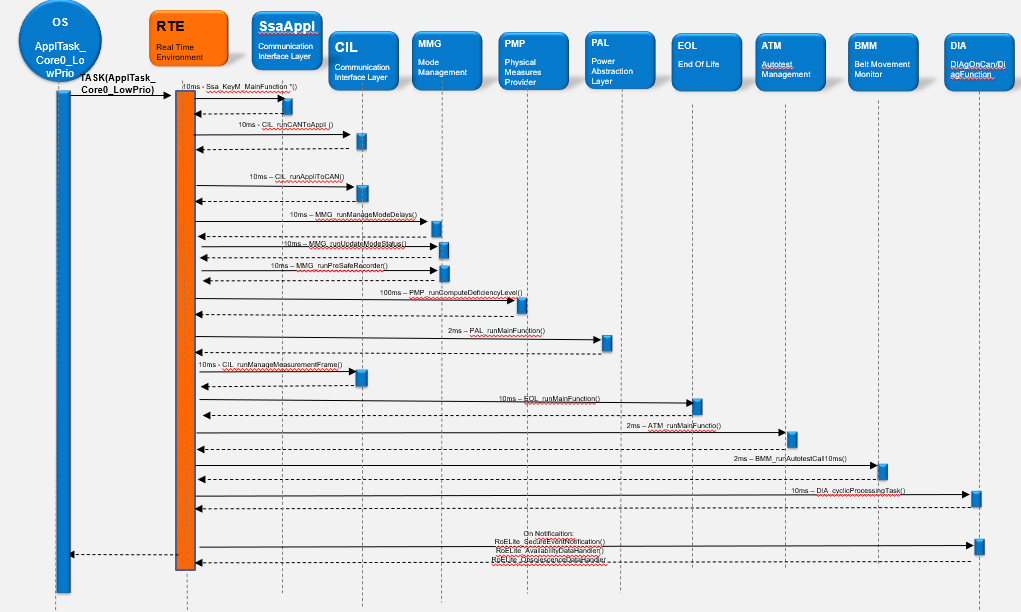


Figure 10: ApplTask\_Core0\_LowPrio

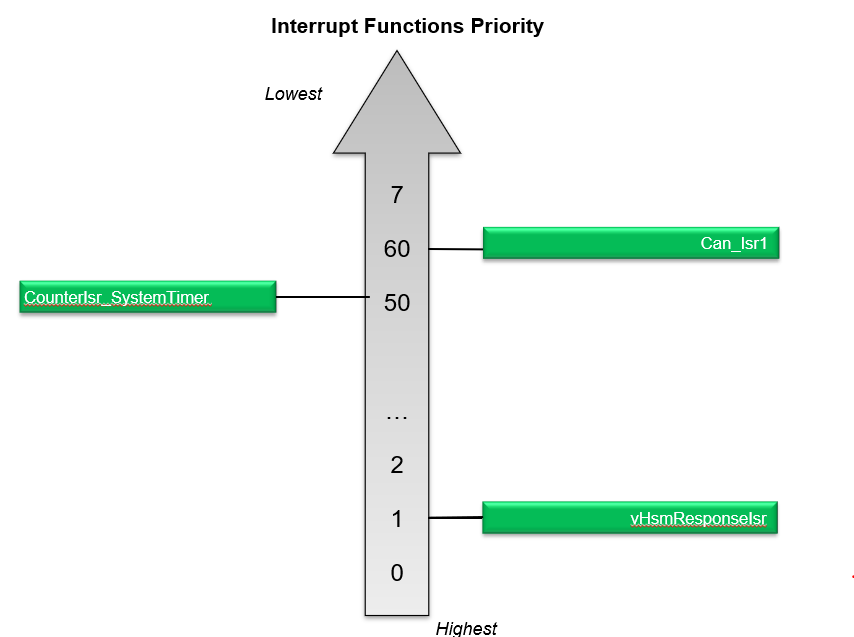
#### OS interrupt function

The present section will specify the needs in term of OS interrupt functions.

* The vHsmResponseIsr interrupt function shall be defined at OS configuration level.
* The CounterIsr\_SystemTimer interrupt function shall be configured.
* The CanIsr\_1 interrupt function shall be defined at OS configuration level.

**Interrupt functions priority**

The next table will specify the needs in term of interrupt functions priority.



**Figure 11: Os – Interrupt Functions priority definition**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| **ARCH\_SW\_OS\_00200** | The vHsmResponseIsr shall have the highest priority.  Justification:  Smaller numerical values denote a higher priority. | Interrupt functions priority shall be 1. |  |
| **ARCH\_SW\_OS\_00201** | The CounterIsr\_SystemTimer shall have the lowest priority.  Justification:  Smaller numerical values denote a higher priority. | Interrupt functions priority shall be 50. |  |
| **ARCH\_SW\_OS\_00202** | The CanIsr\_1 interrupt function shall be defined at OS configuration level. | Interrupt functions priority shall be 60. |  |

### MCU resources

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

###### CPU load requirements for periodic OS tasks

The table below specifies the maximum allowed CPU load for each periodic task.

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_OS\_0300 | The CPU load related to the Default\_Init\_Task task shall NOT exceed 1% in average. |  |  |
| ARCH\_SW\_OS\_0301 | The CPU load related to the Default\_Init\_Task\_Trusted task shall NOT exceed 1% in average. |  |  |
| ARCH\_SW\_OS\_0302 | The CPU load related to the ApplInitTask\_Core0 task shall NOT exceed 1% in average. |  |  |
| ARCH\_SW\_OS\_0303 | The CPU load related to the BswTask\_Core0\_HighPrio task shall NOT exceed 4% in average. |  |  |
| ARCH\_SW\_OS\_0304 | The CPU load related to the ApplTask\_Core0\_HighPrio task shall NOT exceed 3% in average. |  |  |
| ARCH\_SW\_OS\_0305 | The CPU load related to the BswTask\_Core0\_LowPrio task shall NOT exceed 5% in average. |  |  |
| ARCH\_SW\_OS\_0306 | The CPU load related to the OsTask\_ALV\_BeltFunctionAlgo task shall NOT exceed 5% in average. |  |  |
| ARCH\_SW\_OS\_0307 | The CPU load related to the ApplTask\_Core0\_LowPrio task shall NOT exceed 5% in average. |  |  |
| ARCH\_SW\_OS\_0308 | The CPU load related to the ApplBackgroundTask\_Core0 task shall NOT exceed 1% in average. |  |  |
| ARCH\_SW\_OS\_0309 | The CPU load related to the BackgroundTask\_Core0 task shall be 100% - overall CPU load. |  |  |
| ARCH\_SW\_OS\_0310 | The CPU load related to the IdleTask\_OsCore0 task shall NOT exceed 0% in average. |  |  |

###### CPU load requirements for periodic interrupt function

The table below specifies the maximum allowed CPU load for each periodic interrupt function.

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_OS\_00400 | The CPU load related to the vHsmResponseIsr interrupt function shall NOT exceed 5% in average. |  |  |
| ARCH\_SW\_OS\_00401 | The CPU load related to the CounterIsr\_SystemTimer interrupt function shall NOT exceed 5% in average. |  |  |
| ARCH\_SW\_OS\_00402 | The CPU load related to the CanIsr\_1 interrupt function shall NOT exceed 5% in average. |  |  |

# SW units description

Please refer to [B1] to [B42] for SW units design interface description and AUTOSAR Interfaces description.