**RESTRICTED ACCESS**

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

**SW Architecture Design & Interface Description :**

**ERH sw UNIT**

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

SUMMARY: This document provides a high-level view of the *ERH* 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 **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 | | 31.01.2017 | | Michael pastor | | Initial revision |
| Start DAI MMA description based on extended document | | | | | | | |
| 1.1.4.1 | | 23.08.2022 | | M. Obada | | Duplicate revision | |
| 1.1.4.2 | | 23.08.2022 | | M. Obada | | Initial revision | |
| 1.1.4.3 | | 23.08.2022 | | M. Obada | | Update revision history | |
| 1.1.4.4 | | 24.08.2022 | | M. Obada | | Fix findings from review | |
| 1.1.4.5 | | 25.08.2022 | | M. Obada | | Update after SRM | |
| 1.1.4.6 | | 25.08.2022 | | M. Obada | | Fix findings from review | |
| 1.1.4.7 | | 26.08.2022 | | M. Obada | | Update RAM/ROM size | |
| 1.1.4.8 | | 09.11.2022 | | M. Obada | | Update document for 4.0 | |
| 1.1.4.9 | | 19.12.2022 | | M. Obada | | Update document for 4.0 after SRM | |
| 1.1.4.10 | | 20.12.2022 | | M. Obada | | Update document for 5.0 | |
| 1.1.4.11 | | 20.12.2022 | | M. Obada | | Update after SRM | |
| 1.1.4.12 | | 08.02.2023 | | M. Obada | | Fix after review | |
| 1.1.4.13 | | 08.02.2023 | | M. Obada | | Fix evolution of document | |
| 1.1.4.14 | | 11.04.2023 | | M. Obada | | Update document for 6.0 | |
| 1.1.4.15 | | 04.05.2023 | | M. Obada | | Update traceabilty after SRM | |
| 1.1.4.16 | | 08.05.2023 | | M. Obada | | Update traceabilty for implausible data powertrain drv | |
| 1.1.4.17 | | 11.05.2023 | | M. Obada | | Fix after review | |
| 1.1.4.18 | | 11.05.2023 | | M. Obada | | Update name spelling | |
| 1.1.4.19 | | 21/06/2023 | | M. Serban | | Update 6.1 release | |
| 1.1.4.20 | | 22/06/2023 | | M. Serban | | Update static view | |
| 1.1.4.21 | | 21/07/2023 | | M. Obada | | Update traceabilty for BeltAdj timeout error | |
| 1.1.4.22 | | 23/08/2023 | | M. Obada | | Update traceabilty after SRM | |
| 1.1.4.23 | | 23/08/2023 | | M. Obada | | Update traceability | |
| 1.1.4.24 | | 25/08/2023 | | M. Obada | | Update document | |
| 1.1.4.25 | | 31/08/2023 | | M. Obada | | Update after review | |
| 1.1.4.26 | | 31/08/2023 | | M. Obada | | Update static view | |
| 1.1.4.27 | | 01/09/2023 | | M. Obada | | Update group naming | |
| 1.1.4.28 | | 01/09/2023 | | M. Obada | | Update static view | |
| 1.1.4.29 | | 01/09/2023 | | M. Obada | | Add more informations about enable of DEM Ign cycle | |
| 1.1.4.30 | | 16/01/2024 | | M. Obada | | Update arch for 8.1 | |
| 1.1.4.31 | | 16/01/2024 | | M. Obada | | Update arch for 8.1 | |
| 1.1.4.32 | | 18.01.2024 | | M. Obada | | Update after SRM | |
| 1.1.4.33 | | 23.01.2024 | | M. Obada | | Fix after review | |
| 1.1.4.34 | | 23/01/2024 | | M. Obada | | Update revision history | |
| 1.1.4.35 | | 02/04/2024 | | M. Obada | | Update for R9.0 | |
| 1.1.4.36 | | 02/04/2024 | | M. Obada | | Update traceability | |

This document contains **27** pages.

**CONTENTS**

1. Documentation 5

1.1. Upper Level Relevant Documents 5

1.2. Design interface description Documents 6

1.3. Design Specification Documents 9

1.4. Tier2 Documents 9

1.5. HW Datasheet 10

1.6. Other Documents 10

1.7. Glossary And Definition 11

2. Description 12

3. Technical functions 14

4. Runnables 22

4.1. ERH\_Init 22

4.1.1. Definition 22

4.1.1. Called functions 22

4.2. ERH\_cyclic 22

4.2.1. Definition 22

4.2.2. Called functions 23

4.3. ERH\_runGetAecStatus 24

4.3.1. Definition 24

4.3.1. Data flow / Parameters 24

4.4. ERH\_runGetAecGroupStatus 24

4.4.1. Definition 24

4.4.1. Data flow / Parameters 25

4.5. ERH\_runSetAecEvent 25

4.5.1. Definition 25

4.5.1. Called functions 26

4.6. ERH\_ResetAllAECStatus 26

4.6.1. Definition 26

4.6.2. Called functions 26

5. MCU resources 27

**TABLE OF FIGURES**

[Figure 1 ERH - Debounce Counter behavior 12](#_Toc156470920)

[Figure 2 ERH - Static Diagram 14](#_Toc156470921)

# 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](file:///S:\Architectures\Application\Description\Associated_Documents\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](file:///S:\Architectures\Application\Description\Associated_Documents\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](file:///S:\Architectures\Application\Description\Associated_Documents\BFS%20-%20Design%20Interface%20Description%20.docx) | 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 | N/A | RBE/FCE |
|  | Design document of Memory Integrity Control | [HWA - Detailed Design Document.docx](file:///S:\Architectures\Application\Description\Associated_Documents\BFS%20-%20Design%20Interface%20Description%20.docx) | RBE/FCE |
|  | Design document of Mode Management | N/A | RBE/FCE |
|  | Design document of Network Management Interface | [MMG - Detailed Design Document.docx](file:///S:\Architectures\Application\Description\Associated_Documents\MMG%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design document of Non-Volatile Memory Interface | N/A | RBE/FCE |
|  | Design document of Non-Volatile Parameters | N/A | RBE/FCE |
|  | Design document of Power Abstraction Layer | [NVP - Detailed Design Document.docx](file:///S:\Architectures\Application\Description\Associated_Documents\NVP%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design document of Physical Measures Provider | [PAL - Detailed Design Document.docx](file:///S:\Architectures\Application\Description\Associated_Documents\PAL%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design document of Port Interface | [PMP - Detailed Design Document.docx](file:///S:\Architectures\Application\Description\Associated_Documents\PMP%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design document of Production cycle function | N/A | RBE/FCE |
|  | Design document of Reset Cause Management | N/A | RBE/FCE |
|  | Design document of RTE If | [RCM - Detailed Design Document.docx](file:///S:\Architectures\Application\Description\Associated_Documents\PMP%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  | Design document of System Context Management | N/A | RBE/FCE |
|  | Design document of Standard Function Recovery (releasing function) | N/A | RBE/FCE |
|  | Design document of Serial Peripheral Interface Interface | [SFR - Detailed Design Document.docx](file:///S:\Architectures\Application\Description\Associated_Documents\SFR%20-%20Design%20Interface%20Description.docx) | RBE/FCE |
|  |  | N/A |  |

## 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

* ERH is a generic error handler developed by AUTOLIV. It is responsible to handle error notification transmitted by Dem, using a filter for event qualification/deskilling based on thresholds recorded in NVM.

The “Error” will be so called AEC: Autoliv Error Code.

Each AEC can be qualified using a counter based algorithm:

* An internal counter is managed for each AEC. The AEC is unqualified until the debounce counter will reach the qualification threshold value from NVM. The debounce counter will increase with 1 at every call of ERH\_runSetAecEvent function with status PREFAILED. In case of the occurrence of PREPASSED as the status, the debounce counter will decrease by 1.

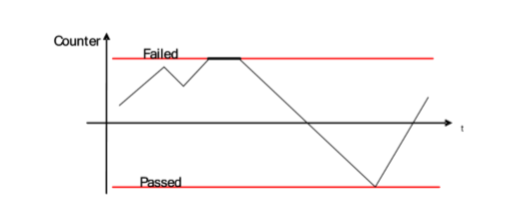


Figure 1 ERH - Debounce Counter behavior

Each AEC is associated to a group, and every group is represented by a DTC.

If a group is qualified, then the associated DTC is qualified. If the group is deskilled, then the associated DTC shall be deskilled.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Group Number** | **Group Name** | **Group Mask** | **AECs of the group** | **DTC** |
| 0 | HARDWARE | KU32\_AEC\_GROUP\_MASK\_HARDWARE | ERH\_KU8\_HIGH\_SIDE\_SWITCH\_AEC\_CFG, ERH\_KU8\_MOSFET\_HIGH\_SC\_AEC\_CFG,  ERH\_KU8\_MOSFET\_LOW\_SC\_AEC\_CFG,  ERH\_KU8\_MOSFET\_OC\_AEC\_CFG,  ERH\_KU8\_HALL\_SENSOR\_AEC\_CFG,  ERH\_KU8\_ENABLE\_OUT\_OF\_ORDER\_AEC\_CFG, ERH\_KU8\_TEMPERATURE\_SENSOR\_FAILURE\_AEC\_CFG, | B228B49 |
| 1 | MOTOR | KU32\_AEC\_GROUP\_MASK\_MOTOR | ERH\_KU8\_MOTOR\_DISCONNECTED\_AEC\_CFG, ERH\_KU8\_MOTOR\_ORDER\_AEC\_CFG,  ERH\_KU8\_MOTOR\_BLOCKED\_AEC\_CFG,  ERH\_KU8\_MOTOR\_IN\_SC\_AEC\_CFG,  ERH\_KU8\_MOTOR\_CURRENT\_AEC\_CFG, | B228B71 |
| 2 | OVER\_VO | KU32\_AEC\_GROUP\_MASK\_OVER\_VO | ERH\_KU8\_OVO\_POWER\_AEC\_CFG | B210E00 |
| 3 | UNDER\_VO | KU32\_AEC\_GROUP\_MASK\_UNDER\_VO | ERH\_KU8\_UVO\_POWER\_AEC\_CFG | B210D00 |
| 4 | OVER\_VO\_TENS | KU32\_AEC\_GROUP\_MASK\_OVER\_VO\_TENS | ERH\_KU8\_OVO\_TENSIONING\_POWER\_AEC\_CFG | B228B17 |
| 5 | UNDER\_VO\_TENS | KU32\_AEC\_GROUP\_MASK\_UNDER\_VO\_TENS | ERH\_KU8\_UVO\_TENSIONING\_POWER\_AEC\_CFG | B228B16 |
| 6 | NVM | KU32\_AEC\_GROUP\_MASK\_NVM | NA | 0x28b45 |
| 7 | EOL\_1 | KU32\_AEC\_GROUP\_MASK\_EOL\_LOW\_HIGH | ERH\_KU8\_END\_OF\_LIFE\_LOW\_AEC\_CFG ERH\_KU8\_END\_OF\_LIFE\_HIGH\_AEC\_CFG | B228B96 |
| 8 | EOL\_2 | KU32\_AEC\_GROUP\_MASK\_EOL\_COMFORT | ERH\_KU8\_END\_OF\_LIFE\_COMFORT\_AEC\_CFG | B228B97 |
| 9 | EOL\_3 | KU32\_AEC\_GROUP\_MASK\_EOL\_MAX | ERH\_KU8\_END\_OF\_LIFE\_MAX\_FORCE\_AEC\_CFG | B228BFA |
| 10 | WARM RESET | KU32\_AEC\_GROUP\_MASK\_WARM\_RESET | ERH\_KU8\_WARM\_RESET\_AEC\_CFG | NA |
| 11 | SELF\_PROTECTION | KU32\_AEC\_GROUP\_MASK\_SELF\_PROTECTION | ERH\_KU8\_SW\_SELF\_PROTECTION\_AEC\_CFG ERH\_KU8\_HW\_SELF\_PROTECTION\_AEC\_CFG | B228B4B |
| 12 | TIMEOUT\_PRESAFE | KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_PRESAFE | ERH\_KU8\_TIMEOUT\_PRESAFE\_AEC\_CFG | U012287 |
| 13 | TIMEOUT\_BUCKLE | KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_BUCKLE | ERH\_KU8\_TIMEOUT\_BUCKLE\_AEC\_CFG | U015187 |
| 14 | IMPLAUSIBLE\_DATA\_PRESAFE | KU32\_AEC\_GROUP\_MASK\_IMPLAUSIBLE\_DATA\_PRESAFE | ERH\_KU8\_IMPLAUSIBLE\_DATA\_PRESAFE\_AEC\_CFG | U041608 |
| 15 | IMPLAUSIBLE\_DATA\_IGNITION | KU32\_AEC\_GROUP\_MASK\_IMPLAUSIBLE\_DATA\_IGNITION | ERH\_KU8\_IMPLAUSIBLE\_DATA\_IGNITION\_AEC\_CFG | U042708 |
| 16 | IMPLAUSIBLE\_DATA\_ORC\_BCKL | KU32\_AEC\_GROUP\_MASK\_IMPLAUSIBLE\_DATA\_BUCKLE\_ORC | ERH\_KU8\_IMPLAUSIBLE\_DATA\_BUCKLE\_AEC\_CFG ERH\_KU8\_IMPLAUSIBLE\_DATA\_ORC\_AEC\_CFG | U045208 |
| 17 | TIMEOUT\_IGNITION | KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_IGNITION | ERH\_KU8\_TIMEOUT\_IGNITION\_AEC\_CFG | U016887 |
| 18 | TIMEOUT\_POWERTRAIN | KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_POWERTRAIN | ERH\_KU8\_TIMEOUT\_POWERTRAIN\_AEC\_CFG | U011587 |
| 19 | IMPLAUSIBLE\_DATA\_POWERTRAIN | KU32\_AEC\_GROUP\_MASK\_IMPLAUSIBLE\_DATA\_POWERTRAIN | ERH\_KU8\_IMPLAUSIBLE\_DATA\_POWERTRAIN\_RDY\_AEC\_CFG  ERH\_KU8\_IMPLAUSIBLE\_DATA\_POWERTRAIN\_DRV\_AEC\_CFG | U044286 |
| 20 | TIMEOUT\_API | KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_API | ERH\_KU8\_TIMEOUT\_API\_AEC\_CFG | D88987 |
| 22 | TIMEOUT\_BELT\_HOL | KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_BH\_L | ERH\_KU8\_TIMEOUT\_BLTHD\_L\_AEC\_CFG | U020187 |
| 21 | TIMEOUT\_BELT\_HOR | KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_BH\_R | ERH\_KU8\_TIMEOUT\_BLTHD\_R\_AEC\_CFG | U020287 |
| 25 | BAT\_PWR\_OV\_SYSTEM | KU32\_AEC\_GROUP\_MASK\_SYSTEM\_OV | ERH\_KU8\_SYSTEM\_OV\_AEC\_CFG | B210E00 |
| 26 | BAT\_PWR\_UV\_SYSTEM | KU32\_AEC\_GROUP\_MASK\_SYSTEM\_UV | ERH\_KU8\_SYSTEM\_UV\_AEC\_CFG | B210D00 |
| 24 | STEERING\_CONFIG | KU32\_AEC\_GROUP\_MASK\_STEERING\_CONFIG | ERH\_KU8\_STEERING\_CONFIG\_AEC\_CFG | B220400 |
| 23 | TIMEOUT\_POWERTRAIN | KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_POWERTRAIN | ERH\_KU8\_TIMEOUT\_POWERTRAIN\_AEC\_CFG | U011587 |
| 27 | TIMEOUT\_ODOSPEEDOMETER | KU32\_AEC\_GROUP\_MASK\_ODOSPEEDOMETER | ERH\_KU8\_TIMEOUT\_ODOSPEEDOMETER\_AEC\_CFG | U015587 |
| 28 | IMPLAUSIBLE\_DATA\_API | KU32\_AEC\_GROUP\_MASK\_IMPLAUSIBLE\_DATA\_API | ERH\_KU8\_IMPLAUSIBLE\_DATA\_API\_AEC\_CFG | U188A08 |
| 29 | VARIANT\_CODING | KU32\_AEC\_GROUP\_MASK\_VARIANT\_CODING | ERH\_KU8\_VARIANT\_CODING\_AEC\_CFG | P1CA000 |
| 30 | TIMEOUT\_BELTADJ | KU32\_AEC\_GROUP\_MASK\_ODOSPEEDOMETER | ERH\_KU8\_TIMEOUT\_BELTADJ\_AEC\_CFG | U015587 |

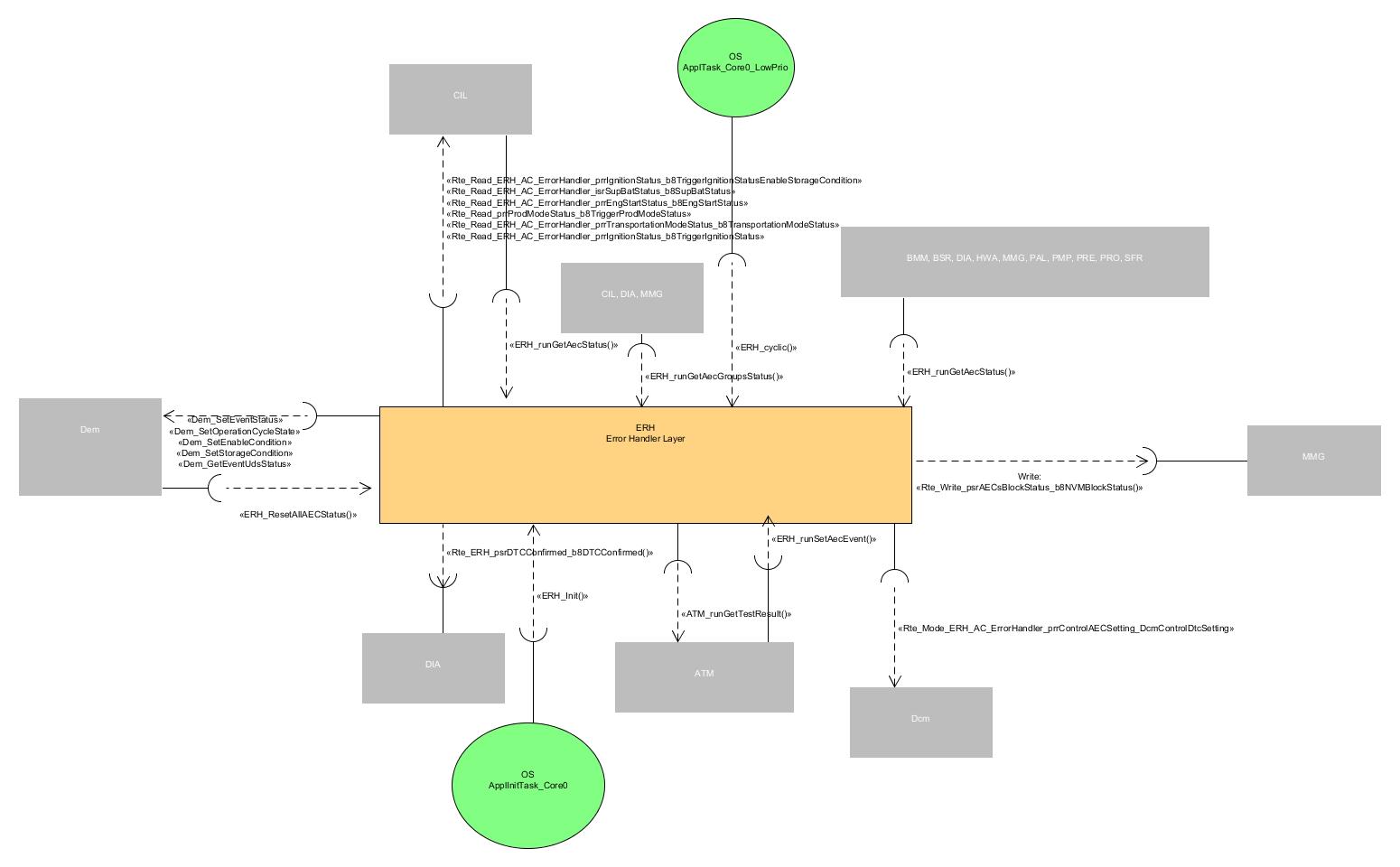


Figure ERH - Static Diagram

# Technical functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Linked Runnable** | **Source** |
| ARCH\_SW\_ERH\_0001 | The AEC ERH\_KU8\_HW\_SELF\_PROTECTION\_AEC\_CFG shall be defined as a Counter based AEC, with the following parameters: - qualification time: 1200ms - deskill time: 60000ms  - associated group: KU32\_AEC\_GROUP\_MASK\_SELF\_PROTECTION | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_256; DAI\_EXT\_TF\_I\_257; DAI\_EXT\_TF\_I\_269; |
| ARCH\_SW\_ERH\_0004 | The AEC ERH\_KU8\_MOTOR\_DISCONNECTED\_AEC\_CFG shall be defined as a Counter based AEC, with the following parameters:  - qualification count: 50  - deskill count: 1  - associated group: KU32\_AEC\_GROUP\_MASK\_MOTOR | ERH\_Init();  ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_437;  DAI\_EXT\_TF\_I\_438; DAI\_EXT\_TF\_I\_450; |
| ARCH\_SW\_ERH\_0006 | The AEC ERH\_KU8\_SW\_SELF\_PROTECTION\_AEC\_CFG shall be defined as a Counter based AEC, with the following parameters:  - qualification count: 1  - deskill count: 1  - associated group: KU32\_AEC\_GROUP\_MASK\_SELF\_PROTECTION | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_473; DAI\_EXT\_TF\_I\_474; DAI\_EXT\_TF\_I\_485; |
| ARCH\_SW\_ERH\_0008 | The AEC ERH\_KU8\_OVO\_POWER\_AEC\_CFG shall be defined as a Counter based AEC, with the following parameters:  - qualification time: 150s  - deskill time: 15s  - associated group: KU32\_AEC\_GROUP\_MASK\_OVER\_VO | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_1772; DAI\_EXT\_TF\_I\_1773; DAI\_EXT\_TF\_I\_925; |
| ARCH\_SW\_ERH\_0010 | The AEC ERH\_KU8\_UVO\_POWER\_AEC\_CFG shall be defined as a Counter based AEC, with the following parameters:  - qualification time: 15s  - deskill time: 150s  - associated group: KU32\_AEC\_GROUP\_MASK\_UNDER\_VO | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_1774; DAI\_EXT\_TF\_I\_1775; DAI\_EXT\_TF\_I\_942; |
| ARCH\_SW\_ERH\_0012 | The AEC ERH\_KU8\_UVO\_TENSIONING\_POWER\_AEC\_CFG  shall be defined as a Counter based AEC, with the following parameters:  - qualification time: 30ms  - deskill time: 30ms  - associated group: KU32\_AEC\_GROUP\_MASK\_UNDER\_VO\_TENS | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_1776; DAI\_EXT\_TF\_I\_1777; DAI\_EXT\_TF\_I\_960; |
| ARCH\_SW\_ERH\_0014 | The AEC ERH\_KU8\_UVO\_TENSIONING\_POWER\_AEC\_CFG  shall be defined as a Counter based AEC, with the following parameters:  - qualification time: 30ms  - deskill time: 30ms  - associated group: KU32\_AEC\_GROUP\_MASK\_OVER\_VO\_TENS | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_1778; DAI\_EXT\_TF\_I\_1779; DAI\_EXT\_TF\_I\_1714; |
| ARCH\_SW\_ERH\_0016 | The AEC ERH\_KU8\_PWM\_ORDER\_AEC\_CFG shall be defined as a Counter based AEC, with the following parameters: - qualification time: instantaneously - deskill time: instantaneously  - associated group: KU32\_AEC\_GROUP\_MASK\_MOTOR | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_455; DAI\_EXT\_TF\_I\_456; DAI\_EXT\_TF\_I\_467; |
| ARCH\_SW\_ERH\_0018 | The AEC ERH\_KU8\_HIGH\_SIDE\_SWITCH\_AEC\_CFG  shall be defined as a Counter based AEC, with the following parameters:  - qualification time: instantaneously  - deskill time: 1 (shall be deskilled even after reset to recheck the fault situation)  - associated group: KU32\_AEC\_GROUP\_MASK\_HARDWARE | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_274; DAI\_EXT\_TF\_I\_275; DAI\_EXT\_TF\_I\_286; |
| ARCH\_SW\_ERH\_0019 | The ERH\_KU8\_MOSFET\_HIGH\_SC\_AEC\_CFG should have the following properties:  ⦁ Qualification Count = 50;  ⦁ Qualification\_time = Qualification\_count \* Autotest\_Periodicity = 5000ms  ⦁ Deskill Count = 100;  ⦁ Deskill\_time = Deskill\_count \* Autotest\_Periodicity = 10000ms  SC"  - associated group: KU32\_AEC\_GROUP\_MASK\_HARDWARE | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_166; DAI\_EXT\_TF\_I\_167; DAI\_EXT\_TF\_I\_170; |
| ARCH\_SW\_ERH\_0020 | The ERH\_KU8\_MOSFET\_LOW\_SC\_AEC\_CFG should have the following properties:  ⦁ Qualification Count = 50;  ⦁ Qualification\_time = Qualification\_count \* Autotest\_Periodicity = 5000ms  ⦁ Deskill Count = 100;  ⦁ Deskill\_time = Deskill\_count \* Autotest\_Periodicity = 10000ms  - associated group: KU32\_AEC\_GROUP\_MASK\_HARDWARE | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_187; DAI\_EXT\_TF\_I\_188; DAI\_EXT\_TF\_I\_197; |
| ARCH\_SW\_ERH\_0021 | The ERH\_KU8\_MOSFET\_OC\_AEC\_CFG should have the following properties:  ⦁ Qualification\_time = instantaneously  ⦁ Deskill\_time = 1 (after each reset)  - associated group: KU32\_AEC\_GROUP\_MASK\_HARDWARE | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_202; DAI\_EXT\_TF\_I\_203;  DAI\_EXT\_TF\_I\_215; |
| ARCH\_SW\_ERH\_0022 | The ERH\_KU8\_HALL\_SENSOR\_AEC\_CFG should have the following properties:  ⦁ Qualification Count = 1;  ⦁ Qualification\_time = Qualification\_count \* Autotest\_Periodicity = 100ms  ⦁ Deskill Count = 1;  ⦁ Deskill\_time = Deskill\_count \* Autotest\_Periodicity = 100ms  - associated group: KU32\_AEC\_GROUP\_MASK\_HARDWARE | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_220; DAI\_EXT\_TF\_I\_221; DAI\_EXT\_TF\_I\_233; |
| ARCH\_SW\_ERH\_0023 | The ERH\_KU8\_ENABLE\_OUT\_OF\_ORDER\_AEC\_CFG should have the following properties:  ⦁ Qualification Count = 2;  ⦁ Qualification\_time = Qualification\_count \* Autotest\_Periodicity = 200ms  ⦁ Deskill Count = 60;  ⦁ Deskill\_time = Deskill\_count \* Autotest\_Periodicity = 6000ms  - associated group: KU32\_AEC\_GROUP\_MASK\_HARDWARE | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_238; DAI\_EXT\_TF\_I\_239; DAI\_EXT\_TF\_I\_250; |
| ARCH\_SW\_ERH\_0024 | The ERH\_KU8\_WARM\_RESET\_AEC\_CFG should have the following properties:  ⦁ Qualification Count = Instantaneously;  ⦁ Deskill Count = None (shall never be deskilled even after reset);  ⦁ Linked Auto-test: "TF\_H12 - To Handle reste causes"  - associated group: ERH\_OTHER\_GROUP | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_328; DAI\_EXT\_TF\_I\_329; DAI\_EXT\_TF\_I\_340; |
| ARCH\_SW\_ERH\_0025 | The ERH\_KU8\_TEMPERATURE\_SENSOR\_FAILURE\_AEC\_CFG should have the following properties:  ⦁ Qualification Count = 1;  ⦁ Qualification\_time = Qualification\_count \* Autotest\_Periodicity = 5000ms  ⦁ Deskill Count = 1;  ⦁ Deskill\_time = Deskill\_count \* Autotest\_Periodicity = 5000ms  - associated group: KU32\_AEC\_GROUP\_MASK\_HARDWARE | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_343; DAI\_EXT\_TF\_I\_346; DAI\_EXT\_TF\_I\_347; |
| ARCH\_SW\_ERH\_0026 | The ERH\_KU8\_MOTOR\_BLOCKED\_AEC\_CFG should have the following properties:  ⦁ Qualification Count = 5;  ⦁ Qualification\_time = Qualification\_count \* Autotest\_Periodicity = 50ms  ⦁ Deskill Count = 1;  ⦁ Deskill\_time = Deskill\_count \* Autotest\_Periodicity = 10ms  - associated group: KU32\_AEC\_GROUP\_MASK\_MOTOR | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_383; DAI\_EXT\_TF\_I\_384; DAI\_EXT\_TF\_I\_395; |
| ARCH\_SW\_ERH\_0027 | The ERH\_KU8\_MOTOR\_IN\_SC\_AEC\_CFG should have the following properties:  ⦁ Qualification Count = 6;  ⦁ Qualification\_time = Qualification\_count \* Autotest\_Periodicity = 60ms  ⦁ Deskill Count = 1 (shall be deskilled after reset);  ⦁ associated group: KU32\_AEC\_GROUP\_MASK\_MOTOR | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_401; DAI\_EXT\_TF\_I\_402; DAI\_EXT\_TF\_I\_414; |
| ARCH\_SW\_ERH\_0028 | The ERH\_KU8\_MOTOR\_CURRENT\_AEC\_CFG should have the following properties:  ⦁ Qualification Count:  - Medium Current threshold: 1200 ms  - High Current threshold: 120000 ms;  ⦁ Deskill Count = 60000 ms;  - associated group: KU32\_AEC\_GROUP\_MASK\_MOTOR | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_419; DAI\_EXT\_TF\_I\_420; DAI\_EXT\_TF\_I\_431; |
| ARCH\_SW\_ERH\_0029 | The ERH\_KU8\_END\_OF\_LIFE\_HIGH\_AEC\_CFG (NGU should have the following properties:  ⦁ Qualification Count: Instantaneously  (i.e. : NVP\_u16CounterToValidateEOLHighForce = 1;  ⦁ Deskill Count: None (shall never be deskilled even after reset)  (i.e. : NVP\_u16CounterToDeskillEOLHighForce = 0);  - associated group: KU32\_AEC\_GROUP\_MASK\_EOL\_LOW\_HIGH | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_1058;  DAI\_EXT\_TF\_I\_1059; DAI\_EXT\_TF\_I\_1071; |
| ARCH\_SW\_ERH\_0030 | The ERH\_KU8\_END\_OF\_LIFE\_LOW\_AEC\_CFG should have the following properties:  ⦁ Qualification Count: Instantaneously  (i.e. : NVP\_u16CounterToValidateEOLLowForce = 1;  ⦁ Deskill Count: None (shall never be deskilled even after reset)  (i.e. : NVP\_u16CounterToDeskillEOLLowForce = 0);  - associated group: KU32\_AEC\_GROUP\_MASK\_EOL\_LOW\_HIGH | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_1076; DAI\_EXT\_TF\_I\_1077; DAI\_EXT\_TF\_I\_1089; |
| ARCH\_SW\_ERH\_0031 | The ERH\_KU8\_END\_OF\_LIFE\_COMFORT\_AEC\_CFG should have the following properties:  ⦁ Qualification Count: Instantaneously  (i.e. : NVP\_u16CounterToValidateEOLComfort = 1;  ⦁ Deskill Count: None (shall never be deskilled even after reset)  (i.e. : NVP\_u16CounterToDeskillEOLComfort = 0);  - associated group: KU32\_AEC\_GROUP\_MASK\_EOL\_COMFORT | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_1094; DAI\_EXT\_TF\_I\_1095; DAI\_EXT\_TF\_I\_1107; |
| ARCH\_SW\_ERH\_0032 | The ERH\_KU8\_TIMEOUT\_PRESAFE\_AEC\_CFG should have the following properties:  ⦁ Qualification Count = Instantaneously  ⦁ Deskill Count = 3;  - associated group: KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_PRESAFE | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_1788; DAI\_EXT\_TF\_I\_1912; DAI\_EXT\_TF\_I\_1790; |
| ARCH\_SW\_ERH\_0033 | The ERH\_KU8\_TIMEOUT\_BUCKLE\_AEC\_CFG should have the following properties:  ⦁ Qualification Count = Instantaneously  ⦁ Deskill Count = 3;  - associated group: KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_BUCKLE | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_1795; DAI\_EXT\_TF\_I\_1913; DAI\_EXT\_TF\_I\_1798; |
| ARCH\_SW\_ERH\_0034 | The ERH\_KU8\_IMPLAUSIBLE\_DATA\_PRESAFE\_AEC\_CFG should have the following properties:  ⦁ Qualification Count = 10  ⦁ Deskill Count = 3;  - associated group: KU32\_AEC\_GROUP\_MASK\_IMPLAUSIBLE\_DATA\_PRESAFE | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_1914; DAI\_EXT\_TF\_I\_1884; DAI\_EXT\_TF\_I\_1805; |
| ARCH\_SW\_ERH\_0035 | The ERH\_KU8\_IMPLAUSIBLE\_DATA\_IGNITION\_AEC\_CFG should have the following properties:  ⦁ Qualification Count = 10  ⦁ Deskill Count = 3;  - associated group: KU32\_AEC\_GROUP\_MASK\_IMPLAUSIBLE\_DATA\_IGNITION | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_1918; DAI\_EXT\_TF\_I\_1917; DAI\_EXT\_TF\_I\_1819; |
| ARCH\_SW\_ERH\_0036 | The ERH\_KU8\_IMPLAUSIBLE\_DATA\_BUCKLE\_AEC\_CFG should have the following properties:  ⦁ Qualification Count = 10  ⦁ Deskill Count = 3;  - associated group: KU32\_AEC\_GROUP\_MASK\_IMPLAUSIBLE\_DATA\_BUCKLE\_ORC | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_1915; DAI\_EXT\_TF\_I\_1916; DAI\_EXT\_TF\_I\_1833; |
| ARCH\_SW\_ERH\_0037 | The ERH\_KU8\_SYSTEM\_OV\_AEC\_CFG should have the following properties:  ⦁ Qualification\_time = 15s  ⦁ Deskill\_time = 15sec  - associated group: KU32\_AEC\_GROUP\_MASK\_SYSTEM\_OV | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_1865; DAI\_EXT\_TF\_I\_1866; DAI\_EXT\_TF\_I\_1868 |
| ARCH\_SW\_ERH\_0038 | The ERH\_KU8\_SYSTEM\_UV\_AEC\_CFG should have the following properties:  ⦁ Qualification\_time = 15s  ⦁ Deskill\_time = 15sec  - associated group: KU32\_AEC\_GROUP\_MASK\_SYSTEM\_UV | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_1873; DAI\_EXT\_TF\_I\_1874; DAI\_EXT\_TF\_I\_1875; |
| ARCH\_SW\_ERH\_0039 | The ERH\_KU8\_TIMEOUT\_POWERTRAIN\_AEC\_CFG should have the following properties:  ⦁ Qualification Count = Instantaneously  ⦁ Deskill Count = 3;  - associated group: KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_POWERTRAIN | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_1879; DAI\_EXT\_TF\_I\_1880; DAI\_EXT\_TF\_I\_1882; |
| ARCH\_SW\_ERH\_0040 | The ERH\_KU8\_IMPLAUSIBLE\_DATA\_POWERTRAIN\_RDY\_AEC\_CFG should have the following properties:  ⦁ Qualification Count =10  ⦁ Deskill Count = 3;  - associated group: KU32\_AEC\_GROUP\_MASK\_IMPLAUSIBLE\_DATA\_POWERTRAIN | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_1892; DAI\_EXT\_TF\_I\_2061; DAI\_EXT\_TF\_I\_2062; |
| ARCH\_SW\_ERH\_0041 | The ERH\_KU8\_IMPLAUSIBLE\_DATA\_ORC\_AEC\_CFG should have the following properties:  ⦁ Qualification Count =10  ⦁ Deskill Count = 3;  - associated group: KU32\_AEC\_GROUP\_MASK\_IMPLAUSIBLE\_DATA\_BUCKLE\_ORC | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_1896; DAI\_EXT\_TF\_I\_1897; DAI\_EXT\_TF\_I\_1898; |
| ARCH\_SW\_ERH\_0042 | The ERH\_KU8\_TIMEOUT\_BLTHD\_L\_AEC\_CFG should have the following properties:  ⦁ Qualification Count = Instantaneously  ⦁ Deskill Count = 3;  - associated group: KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_BH\_L | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_1902; DAI\_EXT\_TF\_I\_1903; DAI\_EXT\_TF\_I\_1905; |
| ARCH\_SW\_ERH\_0043 | The ERH\_KU8\_TIMEOUT\_BLTHD\_R\_AEC\_CFG should have the following properties:  ⦁ Qualification Count = Instantaneously  ⦁ Deskill Count = 3;  - associated group: KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_BH\_R | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_1909; DAI\_EXT\_TF\_I\_1910; DAI\_EXT\_TF\_I\_1911; |
| ARCH\_SW\_ERH\_0044 | The ERH\_KU8\_TIMEOUT\_API should have the following properties:  ⦁ Qualification Count = Instantaneously  ⦁ Deskill Count = 3;  - associated group: KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_API | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_1933; DAI\_EXT\_TF\_I\_1934; DAI\_EXT\_TF\_I\_2027; |
| ARCH\_SW\_ERH\_0045 | The ERH\_KU8\_END\_OF\_LIFE\_MAX\_FORCE\_AEC\_CFG (NGU should have the following properties:  ⦁ Qualification Count: Instantaneously  (i.e. : NVP\_u16CounterToValidateEOLMaxForce = 1;  ⦁ Deskill Count: None (shall never be deskilled even after reset)  (i.e. : NVP\_u16CounterToDeskillEOLMaxForce = 0);  - associated group: KU32\_AEC\_GROUP\_MASK\_EOL\_MAX | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_1719; DAI\_EXT\_TF\_I\_1720; DAI\_EXT\_TF\_I\_1722; |
| ARCH\_SW\_ERH\_0046 | The ERH\_KU8\_STEERING\_CONFIG\_AEC\_CFG should have the following properties:  ⦁ Qualification Count = Instantaneously  ⦁ Deskill Count = Instantaneously;  -associated group: KU32\_AEC\_GROUP\_MASK\_STEERING\_CONFIG | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_1925; DAI\_EXT\_TF\_I\_1927; DAI\_EXT\_TF\_I\_1929; |
| ARCH\_SW\_ERH\_0047 | The ERH\_KU8\_TIMEOUT\_IGNITION\_AEC\_CFG should have the following properties:  ⦁ Qualification Count = Instantaneously  ⦁ Deskill Count = 1;  - associated group: KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_IGNITION | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_1845; DAI\_EXT\_TF\_I\_2077; DAI\_EXT\_TF\_I\_1848; |
| ARCH\_SW\_ERH\_0048 | The CAN BUS ERROR shall qualify the U122488 DTC.  Qulification time: 3s  Deskill time: Instantaneously |  | DAI\_EXT\_TF\_I\_1039; DAI\_EXT\_TF\_I\_1040; DAI\_EXT\_TF\_I\_1052; |
| ARCH\_SW\_ERH\_0050 | All AEC shall have an identifier and should be linked to one Autotest (see table from chapter2, description) | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_253; DAI\_EXT\_TF\_I\_434;  DAI\_EXT\_TF\_I\_470; DAI\_EXT\_TF\_I\_909; DAI\_EXT\_TF\_I\_927; DAI\_EXT\_TF\_I\_945; DAI\_EXT\_TF\_I\_452;  DAI\_EXT\_TF\_I\_271; DAI\_EXT\_TF\_I\_163; DAI\_EXT\_TF\_I\_181; DAI\_EXT\_TF\_I\_343;  DAI\_EXT\_TF\_I\_380;  DAI\_EXT\_TF\_I\_181; DAI\_EXT\_TF\_I\_199; DAI\_EXT\_TF\_I\_217; DAI\_EXT\_TF\_I\_235; DAI\_EXT\_TF\_I\_325; DAI\_EXT\_TF\_I\_398; DAI\_EXT\_TF\_I\_416; DAI\_EXT\_TF\_I\_1055; DAI\_EXT\_TF\_I\_1073; DAI\_EXT\_TF\_I\_1091;  DAI\_EXT\_TF\_I\_1708; DAI\_EXT\_TF\_I\_1828; DAI\_EXT\_TF\_I\_1814; DAI\_EXT\_TF\_I\_1800;  DAI\_EXT\_TF\_I\_1793; DAI\_EXT\_TF\_I\_1786; DAI\_EXT\_TF\_I\_1923; DAI\_EXT\_TF\_I\_1706; DAI\_EXT\_TF\_I\_1931; DAI\_EXT\_TF\_I\_1907; DAI\_EXT\_TF\_I\_1900; DAI\_EXT\_TF\_I\_1894; DAI\_EXT\_TF\_I\_1887; DAI\_EXT\_TF\_I\_1877; DAI\_EXT\_TF\_I\_1843; DAI\_EXT\_TF\_I\_1870; DAI\_EXT\_TF\_I\_1860; |
| ARCH\_SW\_ERH\_0060 | All AECs status and counters shall be initialized with 0x00. | ERH\_Init () |  |
| ARCH\_SW\_ERH\_0070 | All AECs status shall be reset if the function is called. | ERH\_ResetAllAECStatus () |  |
| ARCH\_SW\_ERH\_0080 | This prototype shall manage the Qualification status and Deskill status of the AECs | ERH\_runSetAecEvent () |  |
| ARCH\_SW\_ERH\_0083 | This prototype shall manage the Qualification status and Deskill status of the DTCs by AECs group status | ERH\_runSetAecEvent () |  |
| ARCH\_SW\_ERH\_0084 | A DTC shall be qualified if a group of AECs is qualified (minim one AEC from group is qualified). | ERH\_Init () ERH\_ResetAllAECStatus () |  |
| ARCH\_SW\_ERH\_0085 | A DTC shall be deskilled if all AECs from the associated group are deskilled. | ERH\_Init () ERH\_ResetAllAECStatus () |  |
| ARCH\_SW\_ERH\_0086 | The ERH\_KU8\_IMPLAUSIBLE\_DATA\_POWERTRAIN\_DRV\_AEC\_CFG should have the following properties:  ⦁ Qualification Count = 10  ⦁ Deskill Count = 3;  - associated group: KU32\_AEC\_GROUP\_MASK\_IMPLAUSIBLE\_DATA\_POWERTRAIN | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_2067; DAI\_EXT\_TF\_I\_2066; DAI\_EXT\_TF\_I\_2065; DAI\_EXT\_TF\_I\_2068 |
| ARCH\_SW\_ERH\_0087 | The ERH\_KU8\_TIMEOUT\_ODOSPEEDOMETER should have the following properties:  ⦁ Qualification Count = 10;  ⦁ Deskill Count = 1;  - associated group: KU32\_AEC\_GROUP\_MASK\_ODOSPEEDOMETER | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_2029; DAI\_EXT\_TF\_I\_2031; DAI\_EXT\_TF\_I\_2032; DAI\_EXT\_TF\_I\_2034; |
| ARCH\_SW\_ERH\_0088 | The ERH\_KU8\_TIMEOUT\_SECTICKCOUNT shall qualify the following DTC: U112087 | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_2051; DAI\_EXT\_TF\_I\_2052; DAI\_EXT\_TF\_I\_2053; |
| ARCH\_SW\_ERH\_0089 | The ERH\_KU8\_IMPLAUSIBLE\_DATA\_API\_AEC\_CFG should have the following properties:  ⦁ Qualification Count =10  ⦁ Deskill Count = 3;  - associated group: KU32\_AEC\_GROUP\_MASK\_IMPLAUSIBLE\_DATA\_API | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_2043; DAI\_EXT\_TF\_I\_2045; DAI\_EXT\_TF\_I\_2046; DAI\_EXT\_TF\_I\_2047; |
| ARCH\_SW\_ERH\_0090 | The ERH\_KU8\_VARIANT\_CODING\_AEC\_CFG should have the following properties:  ⦁ Qualification Count =1;  ⦁ Deskill Count = 1;  - associated group: KU32\_AEC\_GROUP\_MASK\_VARIANT\_CODING | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_2055; DAI\_EXT\_TF\_I\_2057; DAI\_EXT\_TF\_I\_2058; DAI\_EXT\_TF\_I\_2060; |
| ARCH\_SW\_ERH\_0091 | All DTCs shall have some defined properties like: priority, self-healing counter, storage condition grup, enable condition, and warning indicator. |  | DAI\_EXT\_TF\_I\_2070; DAI\_EXT\_TF\_I\_2073; DAI\_EXT\_TF\_I\_2074; DAI\_EXT\_TF\_I\_2075; DAI\_EXT\_TF\_I\_2076; |
| ARCH\_SW\_ERH\_0092 | The ERH\_KU8\_TIMEOUT\_BELTADJ\_AEC\_CFG should have the following properties:  ⦁ Qualification Count = 10  ⦁ Deskill Count = 1;  - associated group: KU32\_AEC\_GROUP\_MASK\_ODOSPEEDOMETER | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_2093; DAI\_EXT\_TF\_I\_2095; DAI\_EXT\_TF\_I\_2096; DAI\_EXT\_TF\_I\_2097; |
| ARCH\_SW\_ERH\_0093 | The ERH component shall be able to manage a 10s delay in order to enable the start of a DEM *Ignition Cycle.* | ERH\_cyclic() | DAI\_EXT\_TF\_I\_2747; |
| ARCH\_SW\_ERH\_0094 | Transition from "Ignition off" to "Ignition on" shall start the delay timer. Status of Ignition signal is get through RTE(Rte\_Read\_ERH\_AC\_ErrorHandler\_prrIgnitionStatus\_b8TriggerIgnitionStatus) interface from CIL module | ERH\_cyclic() | DAI\_EXT\_TF\_I\_2747; DAI\_EXT\_TF\_I\_2748; |
| ARCH\_SW\_ERH\_0095 | Transition from "Ignition on" to "Ignition off" shall stop the delay timer. Status of Ignition signal is get through RTE(Rte\_Read\_ERH\_AC\_ErrorHandler\_prrIgnitionStatus\_b8TriggerIgnitionStatus) interface from CIL module | ERH\_cyclic() | DAI\_EXT\_TF\_I\_2747; DAI\_EXT\_TF\_I\_2749; |
| ARCH\_SW\_ERH\_0096 | For the following AECs ECU level storage condition shall be applicable:  ERH\_KU8\_ENABLE\_OUT\_OF\_ORDER\_AEC\_CFG, ERH\_KU8\_MOSFET\_HIGH\_SC\_AEC\_CFG, ERH\_KU8\_MOSFET\_LOW\_SC\_AEC\_CFG, ERH\_KU8\_MOTOR\_DISCONNECTED\_AEC\_CFG, ERH\_KU8\_HALL\_SENSOR\_AEC\_CFG, ERH\_KU8\_TEMPERATURE\_SENSOR\_FAILURE\_AEC\_CFG, ERH\_KU8\_EXT\_WDG\_OUT\_OF\_ORDER\_AEC\_CFG, ERH\_KU8\_HW\_SELF\_PROTECTION\_AEC\_CFG, ERH\_KU8\_SW\_SELF\_PROTECTION\_AEC\_CFG, ERH\_KU8\_END\_OF\_LIFE\_COMFORT\_AEC\_CFG, ERH\_KU8\_END\_OF\_LIFE\_LOW\_AEC\_CFG, ERH\_KU8\_END\_OF\_LIFE\_HIGH\_AEC\_CFG, ERH\_KU8\_END\_OF\_LIFE\_MAX\_FORCE\_AEC\_CFG, ERH\_KU8\_STEERING\_CONFIG\_AEC\_CFG, ERH\_KU8\_VARIANT\_CODING\_AEC\_CFG | ERH\_runSetAecEvent() | DAI\_EXT\_TF\_I\_2851 |
| ARCH\_SW\_ERH\_0097 | For the following AECs Network Communication storage condition shall be applicable:  ERH\_KU8\_TIMEOUT\_PRESAFE\_AEC\_CFG, ERH\_KU8\_TIMEOUT\_BUCKLE\_AEC\_CFG, ERH\_KU8\_IMPLAUSIBLE\_DATA\_PRESAFE\_AEC\_CFG, ERH\_KU8\_IMPLAUSIBLE\_DATA\_IGNITION\_AEC\_CFG, ERH\_KU8\_IMPLAUSIBLE\_DATA\_BUCKLE\_AEC\_CFGERH\_KU8\_IMPLAUSIBLE\_DATA\_ORC\_AEC\_CFG, ERH\_KU8\_IMPLAUSIBLE\_DATA\_POWERTRAIN\_RDY\_AEC\_CFG, ERH\_KU8\_IMPLAUSIBLE\_DATA\_POWERTRAIN\_DRV\_AEC\_CFG,  ERH\_KU8\_TIMEOUT\_BLTHD\_R\_AEC\_CFG, ERH\_KU8\_TIMEOUT\_BLTHD\_L\_AEC\_CFG, ERH\_KU8\_TIMEOUT\_POWERTRAIN\_AEC\_CFG, ERH\_KU8\_TIMEOUT\_ODOSPEEDOMETER\_AEC\_CFG, ERH\_KU8\_IMPLAUSIBLE\_DATA\_API\_AEC\_CFG, ERH\_KU8\_TIMEOUT\_BELT\_ADJ\_AEC\_CFG, ERH\_KU8\_TIMEOUT\_API | ERH\_runSetAecEvent() | DAI\_EXT\_TF\_I\_2852 |
| ARCH\_SW\_ERH\_0098 | For the following AECs Network Communication without Clam 15 storage condition shall be applicable: ERH\_KU8\_TIMEOUT\_IGNITION\_AEC\_CFG | ERH\_runSetAecEvent() | DAI\_EXT\_TF\_I\_2853 |
| ARCH\_SW\_ERH\_0099 | For the following AECs Power Distribution storage condition shall be applicable:  ERH\_KU8\_OVO\_POWER\_AEC\_CFG,  ERH\_KU8\_UVO\_POWER\_AEC\_CFG, ERH\_KU8\_UVO\_TENSIONING\_POWER\_AEC\_CFG, ERH\_KU8\_OVO\_TENSIONING\_POWER\_AEC\_CFG, ERH\_KU8\_SYSTEM\_OV\_AEC\_CFG, ERH\_KU8\_SYSTEM\_UV\_AEC\_CFG | ERH\_runSetAecEvent() | DAI\_EXT\_TF\_I\_2854 |
| ARCH\_SW\_ERH\_0150 | The following AECs shall not have any storage condition applicable:  ERH\_KU8\_SPI\_ERROR\_AEC\_CFG, ERH\_KU8\_HIGH\_SIDE\_SWITCH\_AEC\_CFG, ERH\_KU8\_MOSFET\_OC\_AEC\_CFG, ERH\_KU8\_WARM\_RESET\_AEC\_CFG, ERH\_KU8\_MOTOR\_CURRENT\_AEC\_CFG, ERH\_KU8\_MOTOR\_BLOCKED\_AEC\_CFG, ERH\_KU8\_MOTOR\_ORDER\_AEC\_CFG, ERH\_KU8\_MOTOR\_IN\_SC\_AEC\_CFG | ERH\_runSetAecEvent() |  |
| ARCH\_SW\_ERH\_0151 | "ECU level" shall include the folowing storage conditions:   * SC1: Control DTC settings * SC3: Local Voltage (KL30 input) * SC11: Vehicle Startup | ERH\_runSetAecEvent() | DAI\_EXT\_TF\_I\_2856; |
| ARCH\_SW\_ERH\_0152 | "Network Communication" shall include the folowing storage conditions:   * 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 | ERH\_runSetAecEvent() | DAI\_EXT\_TF\_I\_2857; |
| ARCH\_SW\_ERH\_0153 | "Network Communication - Timeout monitoring without Clamp 15" shall include the folowing storage conditions:   * 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 | ERH\_runSetAecEvent() | DAI\_EXT\_TF\_I\_2858; |
| ARCH\_SW\_ERH\_0154 | "Power Distribution" shall include the folowing storage conditions:   * SC1: Control DTC settings * SC2: Ignition Status * SC5: Transportation mode * SC10: De-bounce Timer Power * SC11: Vehicle Startup * SC12: production mode | ERH\_runSetAecEvent() | DAI\_EXT\_TF\_I\_2859; |
| ARCH\_SW\_ERH\_0156 | A debounce timer of 5s shall be applicable for the following storage condition groups if all the SCs are fulfiled:  Power Distribution, Network Communication without Clam 15 and Network Communication |  | DAI\_EXT\_TF\_I\_2825; DAI\_EXT\_TF\_I\_2828; DAI\_EXT\_TF\_I\_2875; DAI\_EXT\_TF\_I\_2863; DAI\_EXT\_TF\_I\_2831; DAI\_EXT\_TF\_I\_2832; DAI\_EXT\_TF\_I\_2876; DAI\_EXT\_TF\_I\_2864; DAI\_EXT\_TF\_I\_2833; DAI\_EXT\_TF\_I\_2877; DAI\_EXT\_TF\_I\_2865; |
| ARCH\_SW\_ERH\_0157 | The ERH\_KU8\_SPI\_ERROR\_AEC\_CFG should have the following properties:  ⦁ Qualification Count =1;  ⦁ Deskill Count = 1; | ERH\_Init(); ERH\_runGetAecStatus() ERH\_runSetAecEvent(); ERH\_ResetAllAECStatus(); | DAI\_EXT\_TF\_I\_854; DAI\_EXT\_TF\_I\_857; DAI\_EXT\_TF\_I\_858; DAI\_EXT\_TF\_I\_870; |
| ARCH\_SW\_ERH\_0158 | ERH module shall be responsible to update all Groups at Init. | ERH\_Init(); |  |

# Runnables

## ERH\_Init

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| void **ERH\_Init** (void) | | | |
| **Object** | | | |
| This function shall initialize all AEC statuses and associated counters to default values. The “EVER QUALIFIED” status should be updated according to its value from NVP. | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| NA | NA | NA | NA |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Called once at startup by BswM\_OnStartComEffective function  This function initializes the internal data to default values:  - set group counters to 0 value,  - set pending event indexes to 0 value,  - set all AEC statuses to 0 value,  - if Deskill/Qualification threshold exits the AEC counter should be set to its Deskill/Qualification threshold value,  - read the “EVER QUALIFIED” status from NVP and update the AEC status according its value. | | | |
| **Requirements** | | | |
| ARCH\_SW\_ERH\_0060; ARCH\_SW\_ERH\_0001; ARCH\_SW\_ERH\_0004; ARCH\_SW\_ERH\_0006; ARCH\_SW\_ERH\_0008; ARCH\_SW\_ERH\_0010; ARCH\_SW\_ERH\_0012; ARCH\_SW\_ERH\_0014; ARCH\_SW\_ERH\_0016; ARCH\_SW\_ERH\_0018; ARCH\_SW\_ERH\_0019; ARCH\_SW\_ERH\_0020; ARCH\_SW\_ERH\_0021; ARCH\_SW\_ERH\_0022; ARCH\_SW\_ERH\_0023; ARCH\_SW\_ERH\_0024; ARCH\_SW\_ERH\_0025; ARCH\_SW\_ERH\_0026; ARCH\_SW\_ERH\_0027; ARCH\_SW\_ERH\_0028; ARCH\_SW\_ERH\_0029; ARCH\_SW\_ERH\_0030; ARCH\_SW\_ERH\_0031; ARCH\_SW\_ERH\_0050; ARCH\_SW\_ERH\_0085; ARCH\_SW\_ERH\_0084; ARCH\_SW\_ERH\_0100; ARCH\_SW\_ERH\_0157; | | | |

### Called functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Linked Runnable** | **Source** |
| **ARCH\_SW\_ERH\_0100** | DemConf\_DemOperationCycle\_PowerCycle and Dem\_SetEnableCondition called to enable conditions for error handler mechanism. | ERH\_Init () |  |

## ERH\_cyclic

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| void **ERH\_cyclic** (void) | | | |
| **Object** | | | |
| This function is used to set a debounce of 10s before starting of Dem Ignition cycle | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| NA | NA | NA | NA |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Called periodically at 10ms | | | |
| **Requirements** | | | |
| ARCH\_SW\_ERH\_0110; ARCH\_SW\_ERH\_0093; ARCH\_SW\_ERH\_0094; ARCH\_SW\_ERH\_0095; ARCH\_SW\_ERH\_0111; ARCH\_SW\_ERH\_0112; ARCH\_SW\_ERH\_0113; ARCH\_SW\_ERH\_0114; ARCH\_SW\_ERH\_0115; ARCH\_SW\_ERH\_0116; ARCH\_SW\_ERH\_0117; ARCH\_SW\_ERH\_0118; ARCH\_SW\_ERH\_0119; ARCH\_SW\_ERH\_0120; ARCH\_SW\_ERH\_0121; | | | |

### Called functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Linked Runnable** | **Source** |
| **ARCH\_SW\_ERH\_0110** | *DemConf\_DemOperationCycle\_IgnitionCycle* is called to **enable** Ignition cycle, when Ignition status is TRUE, otherwise *DemConf\_DemOperationCycle\_IgnitionCycle* is called to **end** DEM ignition status. | ERH\_cyclic () |  |
| **ARCH\_SW\_ERH\_0111** | Read *Rte\_Read\_isrSupBatStatus\_b8SupBatStatus* to get status in order to manage storage conditions. | ERH\_cyclic () | DAI\_EXT\_TF\_I\_2807; DAI\_EXT\_TF\_I\_2808; DAI\_EXT\_TF\_I\_2812; |
| **ARCH\_SW\_ERH\_0112** | Read *Rte\_Call\_pcsGetComMStatus\_GetCurrentComMode* to get status in order to manage storage conditions. | ERH\_cyclic () | DAI\_EXT\_TF\_I\_2821; DAI\_EXT\_TF\_I\_2822; |
| **ARCH\_SW\_ERH\_0113** | Read *Rte\_Read\_prrEngStartStatus\_b8EngStartStatus* to get status in order to manage storage conditions. | ERH\_cyclic () | DAI\_EXT\_TF\_I\_2837; DAI\_EXT\_TF\_I\_2838; DAI\_EXT\_TF\_I\_2839; |
| **ARCH\_SW\_ERH\_0114** | Read *Rte\_Read\_prrProdModeStatus\_b8TriggerProdModeStatus* to get status in order to manage storage conditions. | ERH\_cyclic () | DAI\_EXT\_TF\_I\_2843; DAI\_EXT\_TF\_I\_2844; DAI\_EXT\_TF\_I\_2845; DAI\_EXT\_TF\_I\_2846; |
| **ARCH\_SW\_ERH\_0115** | Read *Rte\_Read\_prrTransportationModeStatus\_b8TransportationModeStatus* to get status in order to manage storage conditions. | ERH\_cyclic () | DAI\_EXT\_TF\_I\_2816; DAI\_EXT\_TF\_I\_2817; DAI\_EXT\_TF\_I\_2818; DAI\_EXT\_TF\_I\_2819; |
| **ARCH\_SW\_ERH\_0116** | Read *Rte\_Call\_pcsAutotestServices\_GetTestResult* to get status in order to manage storage conditions. | ERH\_cyclic () | DAI\_EXT\_TF\_I\_2798; DAI\_EXT\_TF\_I\_2802; DAI\_EXT\_TF\_I\_2804; |
| **ARCH\_SW\_ERH\_0117** | Read *Rte\_Read\_prrIgnitionStatus\_b8TriggerIgnitionStatusEnableStorageCondition*to get status in order to manage storage conditions. | ERH\_cyclic () | DAI\_EXT\_TF\_I\_2791; DAI\_EXT\_TF\_I\_2792; DAI\_EXT\_TF\_I\_2793; |
| **ARCH\_SW\_ERH\_0118** | Read *Rte\_Read\_prrIgnitionStatus\_b8TriggerIgnitionStatus*to get status in order to manage ignition power cycle. | ERH\_cyclic () |  |
| **ARCH\_SW\_ERH\_0119** | *Dem\_SetStorageCondition* is called to send information to DEM to manage storage conditions for DTCs | ERH\_cyclic () |  |
| **ARCH\_SW\_ERH\_0120** | Dem\_GetEventUdsStatus is called in order to check DTC status for DIA external tester present. | ERH\_cyclic () | DAI\_EXT\_TF\_I\_2433; |
| **ARCH\_SW\_ERH\_0121** | Write *Rte\_Write\_ERH\_AC\_ErrorHandler\_psrDTCConfirmed\_b8DTCConfirmed* for DIA external tester present. | ERH\_cyclic () | DAI\_EXT\_TF\_I\_2433; |

## ERH\_runGetAecStatus

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| void **ERH\_runGetAecStatus** (u8AecIdentifierType **u8AecIdentifier**, u8AecStatusType \* **pu8AecStatus**) | | | |
| **Object** | | | |
| This function shall provide the AEC status to Dem module if the AEC identifier is valid.  If the AEC identifier transmitted by the input parameter “u8AecIdentifier” is between 0x00 and 0x37 this service shall send by the output parameter “pu8AecStatus” the current AEC status, and if the AEC identifier is out of range the transmitted AEC status shall be set to "UNPLAUSIBLE\_STATUS". | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| **u8AecIdentifier** | u8AecIdentifierType | Input | AEC identifier |
| **pu8AecStatus** | u8AecStatusType | Output | The AEC statuses could be the following:  - AEC\_QUALIFIED\_STATUS - 01 value,  - AEC\_EVER\_QUALIFIED\_STATUS - 02 value,  - AEC\_EVER\_TESTED\_STATUS - 04 value,  - AEC\_IN\_PROGRESS\_STATUS - 08 value,  - AEC\_FAILED\_AT\_LEAST\_ONCE\_STATUS -16 value,  - AEC\_UNPLAUSIBLE\_STATUS - 251 value, |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Synchronous server operation  Non re-entrant | | | |
| **Requirements** | | | |
| ARCH\_SW\_ERH\_0064;ARCH\_SW\_ERH\_0001; ARCH\_SW\_ERH\_0004; ARCH\_SW\_ERH\_0006; ARCH\_SW\_ERH\_0008; ARCH\_SW\_ERH\_0010; ARCH\_SW\_ERH\_0012; ARCH\_SW\_ERH\_0014; ARCH\_SW\_ERH\_0016; ARCH\_SW\_ERH\_0018; ARCH\_SW\_ERH\_0019; ARCH\_SW\_ERH\_0020; ARCH\_SW\_ERH\_0021; ARCH\_SW\_ERH\_0022; ARCH\_SW\_ERH\_0023; ARCH\_SW\_ERH\_0024; ARCH\_SW\_ERH\_0025; ARCH\_SW\_ERH\_0026; ARCH\_SW\_ERH\_0027; ARCH\_SW\_ERH\_0028; ARCH\_SW\_ERH\_0029; ARCH\_SW\_ERH\_0030; ARCH\_SW\_ERH\_0031; ARCH\_SW\_ERH\_0050; ARCH\_SW\_ERH\_0085; ARCH\_SW\_ERH\_0084; ARCH\_SW\_ERH\_0086; ARCH\_SW\_ERH\_0087; ARCH\_SW\_ERH\_0088; ARCH\_SW\_ERH\_0089; ARCH\_SW\_ERH\_0090; ARCH\_SW\_ERH\_0157; | | | |

### Data flow / Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Linked Runnable** | **Source** |
| **ARCH\_SW\_ERH\_0064** | **The** AEC statusshall be an ouput for the ERH\_runGetAecStatus function | **ERH\_runGetAecStatus()** |  |

## ERH\_runGetAecGroupStatus

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| void **ERH\_runGetAecGroupStatus** (u32AecGroupsType \* pu32AecGroupsStatus) | | | |
| **Object** | | | |
| This function shall provide all current AEC group status to other modules. ( CIL, DIA, MMG) | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| **pu32AecGroupStatus** | u8AecStatusType | Output | Every bit of status will be set if associated group is set.  Bit0 -> group 0  Bit1 -> group 1  Bit2 -> group 2.etc. |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Synchronous server operation  Non re-entrant | | | |
| **Requirements** | | | |
| ARCH\_SW\_ERH\_0090; | | | |

### Data flow / Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Linked Runnable** | **Source** |
| **ARCH\_SW\_ERH\_0090** | **The** AEC group statusshall be an ouput for the ERH\_runGetAecGroupStatus function | **ERH\_runGetAecStatus()** |  |

## ERH\_runSetAecEvent

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| void **ERH\_runSetAecEvent** (u8AecIdentifierType **u8AecIdentifier**, u8AecCommandType **u8Command**) | | | |
| **Object** | | | |
| This function shall update the status of a received AEC applying the denouncing mechanism based on the previous status of this AEC. If no other event is already in progress the current one shall be processed. This function shall check the identifier validity and execute separately the 4 possible cases regarding the received request:  -KU8\_AEC\_FAILED (0x00)  -KU8\_AEC\_PASSED (0x01)  -KU8\_AEC\_RST\_COUNTER (0x02)  -KU8\_AEC\_RST\_ALL (0x03)  In each case the diagnose request and debouncing mechanism are performed calling the erh\_ManageAECCommand function. | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| **u8AecIdentifier** | u8AecIdentifierType | Input | AEC Identifier value |
| **u8Command** | u8AecCommandType | Input | Type of command:  - KU8\_AEC\_FAILED  - KU8\_AEC\_PASSED  - KU8\_AEC\_RST\_COUNTER  - KU8\_AEC\_RST\_ALL |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Synchronous server operation  Non re-entrant | | | |
| **Requirements** | | | |
| ARCH\_SW\_ERH\_0080; ARCH\_SW\_ERH\_0083; ARCH\_SW\_ERH\_0001; ARCH\_SW\_ERH\_0004; ARCH\_SW\_ERH\_0006; ARCH\_SW\_ERH\_0008; ARCH\_SW\_ERH\_0010; ARCH\_SW\_ERH\_0012; ARCH\_SW\_ERH\_0014; ARCH\_SW\_ERH\_0016; ARCH\_SW\_ERH\_0018; ARCH\_SW\_ERH\_0019; ARCH\_SW\_ERH\_0020; ARCH\_SW\_ERH\_0021; ARCH\_SW\_ERH\_0022; ARCH\_SW\_ERH\_0023; ARCH\_SW\_ERH\_0024; ARCH\_SW\_ERH\_0025; ARCH\_SW\_ERH\_0026; ARCH\_SW\_ERH\_0027; ARCH\_SW\_ERH\_0028; ARCH\_SW\_ERH\_0029; ARCH\_SW\_ERH\_0030; ARCH\_SW\_ERH\_0031; ARCH\_SW\_ERH\_0050; ARCH\_SW\_ERH\_0085; ARCH\_SW\_ERH\_0084; ARCH\_SW\_ERH\_0086; ARCH\_SW\_ERH\_0087; ARCH\_SW\_ERH\_0088; ARCH\_SW\_ERH\_0089; ARCH\_SW\_ERH\_0090; ARCH\_SW\_ERH\_0096; ARCH\_SW\_ERH\_0097; ARCH\_SW\_ERH\_0098; ARCH\_SW\_ERH\_0099; ARCH\_SW\_ERH\_0150; ARCH\_SW\_ERH\_0151; ARCH\_SW\_ERH\_0152; ARCH\_SW\_ERH\_0153;  ARCH\_SW\_ERH\_0154; ARCH\_SW\_ERH\_0157; | | | |

### Called functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Linked Runnable** | **Source** |
| **ARCH\_SW\_ERH\_0081** | DcmControlDtcSetting shall be called to get the status of DTCs setting Mode. | Rte\_Mode\_prrControlAECSetting\_DcmControlDtcSetting | DAI\_EXT\_TF\_I\_2784; DAI\_EXT\_TF\_I\_2785; DAI\_EXT\_TF\_I\_2786; |
| **ARCH\_SW\_ERH\_0159** | Rte\_Write\_ERH\_AC\_ErrorHandler\_psrSnapshotData\_u8SnapshotAecId shall be called in order to send information to Dem module about Qualification reason of a DTC | Rte\_Mode\_prrControlAECSetting\_DcmControlDtcSetting | DAI\_EXT\_TF\_E\_11536; |

## ERH\_ResetAllAECStatus

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| Std\_ReturnType **ERH\_ResetAllAECStatus** (void) | | | |
| **Object** | | | |
| This function shall reset all AEC status and store them in NVP structure, except the "AEC\_EVER\_TESTED\_STATUS" bit.  The possible status of the Autoliv Error Codes could be:  - QUALIFIED\_STATUS - value 01  - EVER\_QUALIFIED\_STATUS - value 02  - EVER\_TESTED\_STATUS - value 04  - IN\_PROGRESS\_STATUS - value 08  - FAILED\_AT\_LEAST\_ONCE\_STATUS - value 16  - UNPLAUSIBLE\_STATUS - value 251,  All AEC statuses except the "AEC\_EVER\_TESTED\_STATUS" bit shall be set to 0.  The all AEC statuses shall be written in NVP structure before software reset.  If the writing process is stiil in progress the return value shall be “DCM\_E\_PENDING”. | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| NA | NA | NA | NA |
| **Returned value** | | | |
| Name | Description | | |
| Std\_ReturnType Return | Returned values:  - DCM\_E\_OK - This value is a successful operation (0x00U)  - DCM\_E\_NOT\_OK - This value is an unsuccessful operation (0x01U)  - DCM\_E\_PENDING - The writing process in NVP is still in progress (0x0AU) | | |
| **Dynamic aspect** | | | |
| Asynchronous server operation  Non-Reentrant | | | |
| **Requirements** | | | |
| ARCH\_SW\_ERH\_0070; **ARCH\_SW\_ERH\_0073; ARCH\_SW\_ERH\_0074; ARCH\_SW\_ERH\_0075; ARCH\_SW\_ERH\_0076; ARCH\_SW\_ERH\_0001; ARCH\_SW\_ERH\_0004; ARCH\_SW\_ERH\_0006; ARCH\_SW\_ERH\_0008; ARCH\_SW\_ERH\_0010; ARCH\_SW\_ERH\_0012; ARCH\_SW\_ERH\_0014; ARCH\_SW\_ERH\_0016; ARCH\_SW\_ERH\_0018; ARCH\_SW\_ERH\_0019; ARCH\_SW\_ERH\_0020; ARCH\_SW\_ERH\_0021; ARCH\_SW\_ERH\_0022; ARCH\_SW\_ERH\_0023; ARCH\_SW\_ERH\_0024; ARCH\_SW\_ERH\_0025; ARCH\_SW\_ERH\_0026; ARCH\_SW\_ERH\_0027; ARCH\_SW\_ERH\_0028; ARCH\_SW\_ERH\_0029; ARCH\_SW\_ERH\_0030; ARCH\_SW\_ERH\_0031; ARCH\_SW\_ERH\_0050;** ARCH\_SW\_ERH\_0085; ARCH\_SW\_ERH\_0084; **ARCH\_SW\_ERH\_0077;** ARCH\_SW\_ERH\_0157; | | | |

### Called functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Linked Runnable** | **Source** |
| **ARCH\_SW\_ERH\_0073** | The erh\_UpdateGroup functionshall be called to update the group status after all errors of the group has just been cleared. | ERH\_ResetAllAECStatus () |  |
| **ARCH\_SW\_ERH\_0074** | The erh\_CheckAecId function shall be called to check the validity of all AEC identifiers and to get the counter index and the status array of each AEC | ERH\_ResetAllAECStatus () |  |
| **ARCH\_SW\_ERH\_0075** | The NvM\_SetRamBlockStatus interface shall be called to notify NVM that AEC status block shall be stored in NVM by the next call of NvM\_WriteAll function | ERH\_ResetAllAECStatus () |  |
| **ARCH\_SW\_ERH\_0076** | NvM\_WriteBlock shall be called to copy the data of the RAM block to its corresponding NVM block | ERH\_ResetAllAECStatus () |  |
| **ARCH\_SW\_ERH\_0077** | Rte\_Write\_psrAECsBlockStatus\_b8NVMBlockStatus shall be called to write the updated AECs block in NVM | ERH\_ResetAllAECStatus () |  |

# MCU resources

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_ERH\_9997 | The ROM size consumed by this component shall not exceed 4.5K bytes. |  |  |
| ARCH\_SW\_ERH\_9998 | The RAM size consumed by this component shall be 500 bytes. |  |  |