**RESTRICTED ACCESS**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **DISTRIBUTION :** | **Firm** | **To** | **Ref** | **Copies** | **1st page** | **e-mail** |
|  | FCE | Xavier Christmann, Claude Redon, Adel Bassaid | 8 |  |  |  |
|  | FCE | Hamza Zetti, Lory Médas, Sabrine Bouazizi, Wail Amri, Kérima Adjadi, Mickael Pastor, Aki Saito | 10 |  |  |  |
|  | RBE | M. Ianos | 1 |  |  |  |
|  |  |  |  |  |  |  |
|  | FCE | Secretary ship | 1 |  |  |  |

**SW Architecture Design & Interface Description :**

**PMP sw UNIT**

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

SUMMARY: This document provides a high-level view of the *PMP* 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 R01.0 SW release

**THIS DOCUMENT CONTAINS HIDDEN TEXT**

EVOLUTION OF THE DOCUMENT

|  |  |  |  |
| --- | --- | --- | --- |
| **Issue** | **Date** | **Author** | **Motive and nature of the modifications** |
| A00 | 31/08/2016 | C. Redon | First release (extract from the full PP4G architecture document) |
| A01 | 23/09/2016 | J. Lacroix | Update of requirement reference |
| A02 | 14/11/2016 | P-O. Pilot | Rename all autotests runnable from PMP\_Cbk\_XXX to PMP\_Autotest\_XXX |
| A03 | 14/11/2016 | P-O. Pilot | Update filter value (10 -> 100ms) used for power UV/OV test |
| A04 | 18/11/2016 | C. Redon | Correction of traceability issues |
| Start extended description based on mainstream document | | | |
| 1.1.1.2 | 15/07/2019 | A. Vaché | Update traceability to match PP4G extended platform requirements IDs |
| 1.1.1.3  1.1.1.4 | 08/08/2019  19/11/2019 | A. Vaché  W. AMRI | Solve some traceability issues highlighted by reqtify  Add missing traceability, issue correction: 801127. |
| 1.1.1.5 | 26/11/2019 | W. AMRI | Update template information |
| Start DAI MMA description based on mainstream document | | | |
| 1.1.3.1 | 05/01/2022 | A. Negrea | First revision |
| 1.1.3.2 | 05/01/2022 | A. Negrea | Duplicate revision |
| 1.1.3.3 | 10/02/2022 | A. Negrea | Update adc scalling |
| 1.1.3.4 | 22/02/2022 | A. Negrea | Add fix after review |
| 1.1.3.5 | 22/02/2022 | A. Negrea | Add another fix after review |
| 1.1.3.6 | 23/03/2022 | A. Negrea | Update traceability according to SRM |
| 1.1.3.7 | 09/06/2022 | M. Obada | Update document traceability |
| 1.1.3.8  1.1.3.9  1.1.3.10  1.1.3.11 | 14/06/2022  16/06/2022  20/06/2022  21/06/2022 | M. Obada  M. Obada  M. Obada  M. Obada | Update document according to SRM  Add fix after review  Update chapter 1.5 and 1.6  Update reference A[7] |
| 1.1.3.12 | 21/06/2202 | M. Obada | Update revision history |
| 1.1.3.13 | 22/08/2202 | S. Dominte | Update for 3.0 release |
| 1.1.3.14 | 29/08/2202 | S. Dominte | Update after review |
| 1.1.3.15  1.1.3.16  1.1.3.17  1.1.3.18  1.1.3.19 | 29/08/2202  09/11/2022  16/11/2022  19/12/2022  20/12/2022 | S. Dominte  M. Obada  M. Obada  M. Obada  M. Obada | Static description diagram updated.  Update for 4.0 release  Update after SRM  Update for 5.0  Update for 5.0 after SRM |
| 1.1.3.20 | 20/12/2022 | A. Paval | Update for 5.0 |
| 1.1.3.22 | 14/02/2023 | A. Paval | Update after review |
| 1.1.3.23 | 15/02/2023 | A. Paval | Fixes related to traceability |
| 1.1.3.24 | 15/02/2023 | A. Paval | Deleted deprecated requirements |
| 1.1.3.25  1.1.3.26  1.1.3.27 | 15/02/2023  16/05/2023  04/10/2023 | A. Paval  M. Obada  M. Obada | Fixed duplicated requirement IDs  Update filter value (100 -> 10ms) used for power UV/OV autotests  Update static diagram |

This document contains **32** pages.

**CONTENTS**

1. Documentation 6

1.1. Upper Level Relevant Documents 6

1.2. Design interface description Documents 7

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 13

3.1. To get the vehicle signals (battery, logical supply, motor voltages…) 14

3.2. To get the temperature 16

3.3. To provide the SW self-protection 17

3.3.1. To manage the deficiency level 17

4. Runnables – Physical measurements 18

4.1. PMP\_Init 18

4.1.1. Definition 18

4.2. PMP\_runGetBatteryVoltage\_(Ext)\* 18

4.2.1. Definition 18

4.2.2. Called functions 19

4.3. PMP\_runGetMotorVn\_(Ext)\* 19

4.3.1. Definition 19

4.3.2. Called functions 19

4.4. PMP\_runGetMotorVp\_(Ext)\* 19

4.4.1. Definition 19

4.4.2. Called functions 20

4.5. PMP\_GetCorrectedTemperature() 20

4.5.1. Definition 20

4.5.2. Called functions 20

4.6. PMP\_ runGetHS1Level() 20

4.6.1. Definition 20

4.6.2. Called functions 21

4.7. PMP\_runGetBatteryVoltageSurveyStatus () 21

4.7.1. Definition 21

4.7.2. Called functions 21

4.8. PMP\_runSurveyBatteryVoltage 22

4.8.1. Definition 22

4.8.2. Data flow / Parameters 22

4.8.3. Called functions 22

4.9. PMP\_runGetFilteredTemperature 23

4.9.1. Definition 23

4.9.2. Called functions 23

4.10. PMP\_runGetFilteredTemperature\_deg 23

4.10.1. Definition 23

4.10.2. Called functions 24

4.11. PMP\_runGetDeficiencyLevel 24

4.11.1. Definition 24

4.12. PMP\_runComputeDeficiencyLevel 24

4.12.1. Definition 24

4.12.2. Data flow / Parameters 25

4.12.3. Called functions 25

4.13. PMP\_ runGetBoardResistance 25

4.13.1. Definition 25

4.13.2. Data flow / Parameters 26

4.1. PMP\_ runGetMotorResistance 26

4.1.1. Definition 26

4.1.2. Data flow / Parameters 26

5. Runnables – Auto-tests 27

5.1. PMP\_Autotest\_CheckPowerSupplyOV 27

5.1.1. Definition 27

5.1.2. Data flow 27

5.1.3. Called functions 27

5.2. PMP\_Autotest\_CheckPowerSupplyUV 27

5.2.1. Definition 27

5.2.2. Data flow 28

5.2.3. Called functions 28

5.3. PMP\_Autotest\_CheckPowerSupplyUVDuringActivation 28

5.3.1. Definition 28

5.3.2. Data flow 29

5.3.3. Called functions 29

5.4. PMP\_Autotest\_CheckPowerSupplyOVDuringActivation 29

5.4.1. Definition 29

5.4.2. Data flow 30

5.4.3. Called functions 30

5.5. PMP\_Autotest\_CheckTemperatureSensor 30

5.5.1. Definition 30

5.5.2. Data flow 30

5.5.3. Called functions 31

5.6. PMP\_Autotest\_CheckSystemOV 31

5.6.1. Definition 31

5.6.2. Data flow 31

5.6.3. Called functions 31

5.7. PMP\_Autotest\_CheckSystemUV 32

5.7.1. Definition 32

5.7.2. Data flow 32

5.7.3. Called functions 32

6. MCU resources 32

**TABLE OF FIGURES**

[**Figure 1: Pmp - Static description** 12](#_Toc127360477)

[Figure 2: Pmp - Dynamic description to read raw and filtered voltage signals 15](#_Toc127360478)

[**Figure 3: Pmp – To read the temperature** 16](#_Toc127360479)

# 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](../../../../Components/Application/Autoliv/NVP/Config/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 | [BMM - Design Interface Description.docx](BMM%20-%20Design%20Interface%20Description.docx) | 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-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 | N/A | 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 | [ATM - Detailed Design Document.docx](../../../../Components/Application/Autoliv/ATM/Design/ATM%20-%20Detailed%20Design%20Document.docx) | RBE/FCE |
|  | Design document of Belt Function Decision | N/A | RBE/FCE |
|  | Design document of Belt Function Execution | [BFE - Detailed Design Document.docx](../../../../Components/Application/Autoliv/BFE/Design/BFE%20-%20Detailed%20Design%20Document.docx) | RBE/FCE |
|  | Design document of Belt Function Selection | [BFS - Detailed Design Document.docx](../../../../Components/Application/Autoliv/BFS/Design/BFS%20-%20Detailed%20Design%20Document.docx) | RBE/FCE |
|  | Design document of Belt Movement Monitoring | [BMM - Detailed Design Document.docx](file:///C:\Users\alexandru.paval\Documents\Sandboxes\Daimler_MMA_ReVAS\Phase_02\View_Development\Components\Application\Autoliv\BMM\Design\BMM%20-%20Detailed%20Design%20Document.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](../../../../Components/Application/Autoliv/BSR/Design/BSR%20-%20Detailed%20Design%20Document.docx) |  |
|  | Design document of Communication Interaction Layer | [CIL - Detailed Design Document.docx](../../../../Components/Application/Autoliv/CIL/Design/CIL%20-%20Detailed%20Design%20Document.docx) | RBE/FCE |
|  | Design document of Diagnostic Communication Manager Interface | N/A | RBE/FCE |
|  | Design document of Diagnostic Event Manager Interface | N/A | RBE/FCE |
|  | Design document of DiagOnCAN services management | [DIA - Detailed Design Document.docx](../../../../Components/Application/Autoliv/DIA/Design/DIA%20-%20Detailed%20Design%20Document.docx) | RBE/FCE |
|  | Design document of End of life | [EOL - Detailed Design Document.docx](../../../../Components/Application/Autoliv/EOL/Design/EOL%20-%20Detailed%20Design%20Document.docx) | RBE/FCE |
|  | Design document of Error Handler | [ERH - Detailed Design Document.docx](../../../../Components/Application/Autoliv/ERH/Design/ERH%20-%20Detailed%20Design%20Document.docx) | RBE/FCE |
|  | Design document of Haptic Warning | [HWA - Detailed Design Document.docx](../../../../Components/Application/Autoliv/HWA/Design/HWA%20-%20Detailed%20Design%20Document.docx) | RBE/FCE |
|  | Design document of Memory Integrity Control | N/A | RBE/FCE |
|  | Design document of Mode Management | [MMG - Detailed Design Document.docx](../../../../Components/Application/Autoliv/MMG/Design/MMG%20-%20Detailed%20Design%20Document.docx) | RBE/FCE |
|  | Design document of Network Management Interface | N/A | RBE/FCE |
|  | Design document of Non-Volatile Memory Interface | N/A | RBE/FCE |
|  | Design document of Non-Volatile Parameters | N/A | RBE/FCE |
|  | Design document of Power Abstraction Layer | [PAL - Detailed Design Document.docx](../../../../Components/Application/Autoliv/PAL/Design/PAL%20-%20Detailed%20Design%20Document.docx) | RBE/FCE |
|  | Design document of Physical Measures Provider | [PMP - Detailed Design Document.docx](../../../../Components/Application/Autoliv/PMP/Design/PMP%20-%20Detailed%20Design%20Document.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 | N/A | 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](../../../../Components/Application/Autoliv/SFR/Design/SFR%20-%20Detailed%20Design%20Document.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.

|  |  |  |  |
| --- | --- | --- | --- |
| 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

The PMP (Physical Measures Provider) component is intended to provide the value of all physical signals (sampled by the ADC component) including the associated filtered values (issued from an IIR filter at different periods: 2ms, 10ms, 20ms and 100ms for the rest of the application).

Note:

The motor current shall not be handled by PMP because too power stage relevant (refer to PAL).

The following context diagram shows the static description of the module.

A diagram of a company

Description automatically generated

**Figure 1: Pmp - Static description**

# Technical functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Linked Runnable** | **Source** |
| ARCH\_SW\_PMP\_0005 | PMP sw unit shall be initialized. voltage thresholds are aligned with the ADC format | PMP\_Init() |  |
| ARCH\_SW\_PMP\_0010 | PMP shall be providing raw battery voltage value. | PMP\_runGetBatteryVolatge\_(EXT)\*  Ext\* = None |  |
| ARCH\_SW\_PMP\_0011 | PMP shall be providing filtered battery voltage value every 2ms. | PMP\_runGetBatteryVolatge\_(EXT)\*  Ext\* = 2ms |  |
| ARCH\_SW\_PMP\_0012 | PMP shall be providing filtered battery voltage value every 10ms. | PMP\_runGetBatteryVolatge\_(EXT)\*  Ext\* = 10ms |  |
| ARCH\_SW\_PMP\_0014 | PMP shall be providing filtered battery voltage value every 100ms. | PMP\_runGetBatteryVolatge\_(EXT)\*  Ext\* = 100ms |  |
| ARCH\_SW\_PMP\_9014 | PMP shall provide the corrected temperature of the ECU. | PMP\_GetCorrectedTemperature |  |
| ARCH\_SW\_PMP\_0200 | PMP shall monitor the stability of the battery voltage and update a battery status variable for the rest of application | PMP\_runSurveyBatteryVoltage |  |
| ARCH\_SW\_PMP\_0260 | PMP shall provide the filtered temperature to the rest of the application | PMP\_runGetFilteredTemperature |  |
| ARCH\_SW\_PMP\_0280 | PMP shall provide the temperature in Celsius degrees | PMP\_runGetFilteredTemperature\_deg |  |
| **ARCH\_SW\_PMP\_0320** | PMP shall provide the estimation of the energy stored in the motor | PMP\_runGetDeficiencyLevel |  |
| **ARCH\_SW\_PMP\_0340** | PMP shall compute the estimation of the energy stored in the motor | PMP\_runComputeDeficiencyLevel |  |
| ARCH\_SW\_PMP\_1000 | PMP shall check if an overvoltage is detected on the power stage of the battery voltage | PMP\_Autotest\_CheckPowerSupplyOV | ALV\_EXT\_TF\_H\_341; ALV\_EXT\_TF\_H\_345; ALV\_EXT\_TF\_H\_346; DAI\_EXT\_TF\_H\_2418; DAI\_EXT\_TF\_H\_2419; |
| ARCH\_SW\_PMP\_1050 | PMP shall check if an undervoltage is detected on the power stage of the battery voltage | PMP\_Autotest\_CheckPowerSupplyUV | ALV\_EXT\_TF\_H\_381; ALV\_EXT\_TF\_H\_385; ALV\_EXT\_TF\_H\_386; DAI\_EXT\_TF\_H\_2418; DAI\_EXT\_TF\_H\_2419; |
| ARCH\_SW\_PMP\_1060 | PMP shall check if an undervoltage is detected on the power stage of the battery voltage during activation | PMP\_Autotest\_CheckPowerSupplyUVDuringActivation | ALV\_EXT\_TF\_H\_419; ALV\_EXT\_TF\_H\_422; DAI\_EXT\_TF\_H\_2421; DAI\_EXT\_TF\_H\_2422; ALV\_EXT\_TF\_H\_2426; ALV\_EXT\_TF\_H\_2427; |
| ARCH\_SW\_PMP\_1070 | PMP shall check if an overrvoltage is detected on the power stage of the battery voltage during activation | PMP\_Autotest\_CheckPowerSupplyOVDuringActivation | DAI\_EXT\_TF\_H\_2421; DAI\_EXT\_TF\_H\_2422; DAI\_EXT\_TF\_H\_2423; DAI\_EXT\_TF\_H\_2424; DAI\_EXT\_TF\_H\_2244; |
| ARCH\_SW\_PMP\_1080 | PMP shall check if an undervoltage is detected on the power stage of the battery voltage at trigger of request | PMP\_Autotest\_CheckPowerSupplyUVDuringActivation | DAI\_EXT\_TF\_H\_2418; DAI\_EXT\_TF\_H\_2419; ALV\_EXT\_TF\_H\_2434; ALV\_EXT\_TF\_H\_2435; |
| ARCH\_SW\_PMP\_1090 | PMP shall check if an overrvoltage is detected on the power stage of the battery voltage at trigger of request | PMP\_Autotest\_CheckPowerSupplyOVDuringActivation | DAI\_EXT\_TF\_H\_2418; DAI\_EXT\_TF\_H\_2419; DAI\_EXT\_TF\_H\_2433; ALV\_EXT\_TF\_H\_2436; |
| ARCH\_SW\_PMP\_1100 | PMP shall check if an undervoltage is detected in the system, signal PN14\_SupBat\_Volt\_ST3 | PMP\_Autotest\_CheckSystemUV | DAI\_EXT\_TF\_H\_2418; DAI\_EXT\_TF\_H\_2419; DAI\_EXT\_TF\_H\_2394; DAI\_EXT\_TF\_H\_2395; |
| ARCH\_SW\_PMP\_1110 | PMP shall check if an undervoltage is detected in the system, signal PN14\_SupBat\_Volt\_ST3 | PMP\_Autotest\_CheckSystemOV | DAI\_EXT\_TF\_H\_2418; DAI\_EXT\_TF\_H\_2419; DAI\_EXT\_TF\_H\_2380; DAI\_EXT\_TF\_H\_2389; |
| ARCH\_SW\_PMP\_1120 | PMP shall convert the signal PN14\_SupBat\_Volt\_ST3 in V by using 0.025 V units |  | DAI\_EXT\_TF\_B\_2335; DAI\_EXT\_TF\_B\_2336; |
| ARCH\_SW\_PMP\_0383 | This function shall provide the motor resistance compensated in temperature. | PMP\_runGetMotorResistance |  |
| ARCH\_SW\_PMP\_0381 | This function shall return the NVP Parameter that indicates the ECU resistance. | PMP\_runGetBoardResistance |  |
| ARCH\_SW\_PMP\_0021 | This function shall get ADC value of the Negative Motor Voltage. | PMP\_runGetMotorVn\_(Ext)\* |  |
| ARCH\_SW\_PMP\_0023 | This function shall get ADC value of the Positive Motor Voltage. | PMP\_runGetMotorVp\_(Ext)\* |  |
| ARCH\_SW\_PMP\_0203 | This function shall provide the value of HighSide Switch Voltage from the ADC (not FILTERED). | PMP\_runGetHS1Level |  |
| ARCH\_SW\_PMP\_0205 | This function shall provide the actual battery state. | PMP\_runGetBatteryVoltageSurveyStatus |  |

## To get the vehicle signals (battery, logical supply, motor voltages…)

The diagram below describes the sequence to get the battery, auxiliary and motor voltages in mV.

As depicted by this sequence diagram, the client is getting the voltages in mV thanks to a synchronous call of the PMP\_runGetBatteryVoltage\_(Ext), PMP\_runGetAuxiliaryBatterySignal\_(Ext) and the PMP\_runGetMotorVp services\_(Ext) knowing that the signal can be raw or filtered among 2ms, 10ms, 20ms and 100ms.

This synchronous call is done in the real time execution context of the client.

The principle is the same for the other voltage signals.

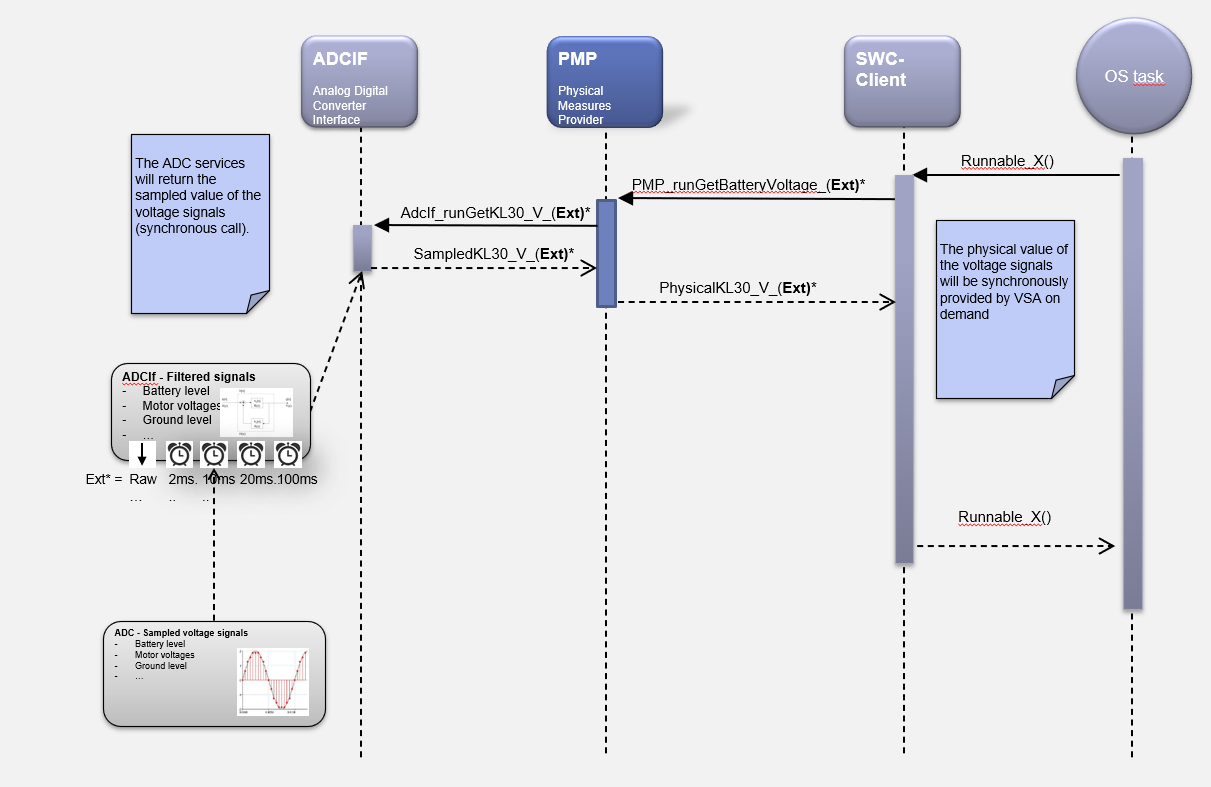


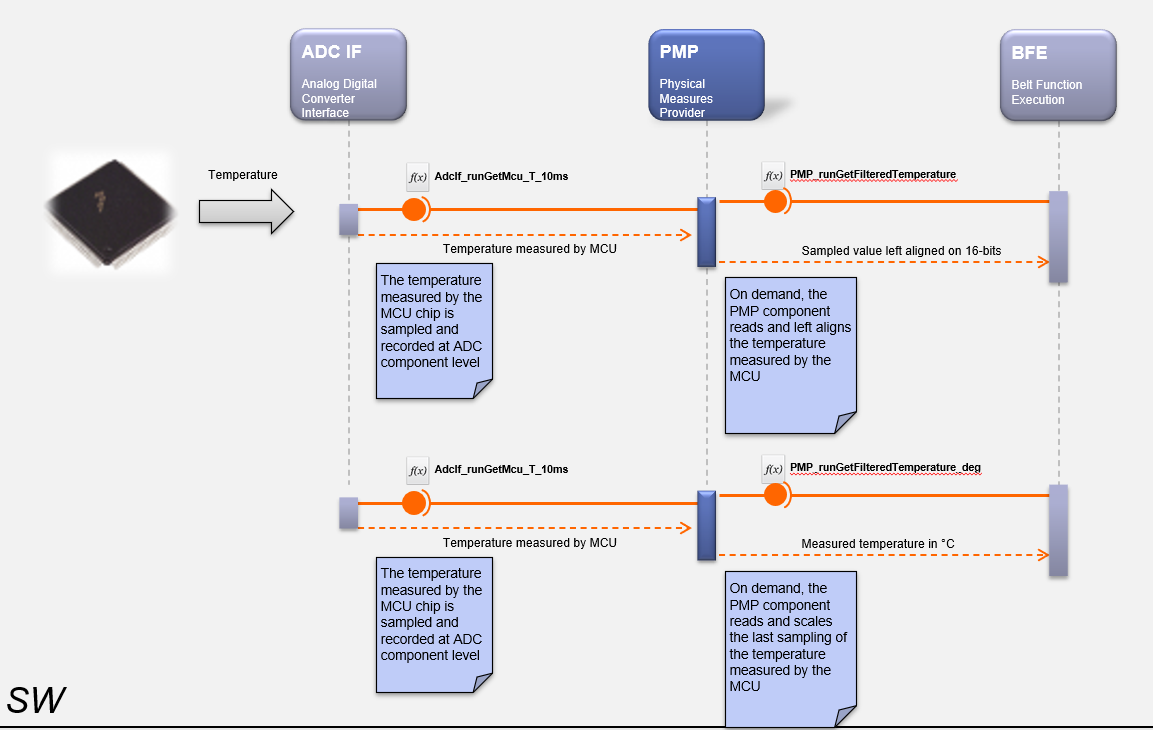
Figure : Pmp - Dynamic description to read raw and filtered voltage signals

## To get the temperature

The Hearst MCU is equipped with a sensor providing the temperature through the A/D converter.

As depicted by the figure below, the purpose of the PMP component is to provide server operations:

* To get the temperature sampled by the Adc unit but left aligned on 16-bits
* To get the temperature in Celcius degrees (for diagnosis purpose for instance)



**Figure 3: Pmp – To read the temperature**

## To provide the SW self-protection

Despite the HW self-protection capability, the complete system is not fully protected against over-temperature (especially the motor).

Therefore, the HW self-protection is completed by a SW self-protection to mostly protect the motor against over-temperature.

### To manage the deficiency level

The management of the SW self-protection is based on the computation of a so called deficiency level (estimation of the energy stacked in the motor).

The management of the deficiency level will consist in:

* Calculating the deficiency level based on the square of motor current measurement and the temperature value.

This will be done periodically thank to PMP\_runComputeDeficiencyLevel

# Runnables – Physical measurements

## PMP\_Init

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| void **PMP\_Init** (void) | | | |
| **Object** | | | |
| This function shall initialize the PMP SW unit. Minimum and maximum voltage thresholds are aligned with the ADC format. | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| NA | NA | NA | NA |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Synchronous server operation  Reentrant | | | |
| **Covered requirements** | | | |
| ARCH\_SW\_PMP\_0005 | | | |

The following table shows the filtering period value for getting the different voltage signals. All the associated runnables will be prefixed with (Ext)\* standing for the filtering period value issued from IIR filter depending on use for the rest of application.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Ext\* = Filtering period** | | | | |
| **RUNNABLES** | **None**  **(Raw)** | **2ms**  **(Filtered)** | **10ms**  **(Filtered)** | **20ms**  **(Filtered)** | **100ms**  **(Filtered)** |
| PMP\_runGetBatteryVoltage\_(Ext)\* | X | X | X |  | X |
| PMP\_runGetMotorVn\_(Ext)\* | X |  |  | X | X |
| PMP\_runGetMotorVp\_(Ext)\* | X |  |  | X | X |

## PMP\_runGetBatteryVoltage\_(Ext)\*

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| void **PMP\_runGetBatteryVoltage\_(Ext)\*** (u16VoltageType \* pu16Voltage) | | | |
| **Object** | | | |
| This function shall provide the physical value of the KL30 signal voltage in mV filtered or not depending on use. | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| pu16Voltage | u16VoltageType | Out | Battery voltage in mV (1LSB = 1mV) |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Synchronous server operation  Reentrant | | | |
| **Covered requirements** | | | |
| ARCH\_SW\_PMP\_0010; ARCH\_SW\_PMP\_0011; ARCH\_SW\_PMP\_0012; ARCH\_SW\_PMP\_0013; ARCH\_SW\_PMP\_0014 | | | |

### Called functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| **ARCH\_SW\_PMP\_0020** | The AdcIf\_runGetKL30\_V\_(Ext)\* function shall be called to get the sampled value of the KL30 signal. | (Ext)\* from ADC and PMP services are the same |  |

## PMP\_runGetMotorVn\_(Ext)\*

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| void **PMP\_runGetMotorVn\_(Ext)\*** (u16VoltageType \* pu16Voltage) | | | |
| **Object** | | | |
| This function shall get ADC value of the Negative Motor Voltage. | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| pu16Voltage | u16VoltageType | Out | Negative motor voltage in mV (1LSB = 1mV) |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Synchronous server operation  Reentrant | | | |
| **Covered requirements** | | | |
| ARCH\_SW\_PMP\_0021; ARCH\_SW\_PMP\_0022; | | | |

### Called functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| **ARCH\_SW\_PMP\_0022** | The AdcIf\_runGetMotor\_Vn\_(Ext)\* function shall be called to get the sampled value. | (Ext)\* from ADC and PMP services are the same |  |

## PMP\_runGetMotorVp\_(Ext)\*

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| void **PMP\_runGetMotorVp\_(Ext)\*** (u16VoltageType \* pu16Voltage) | | | |
| **Object** | | | |
| This function shall get ADC value of the Positive Motor Voltage. | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| pu16Voltage | u16VoltageType | Out | Positive motor voltage in mV (1LSB = 1mV) |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Synchronous server operation  Reentrant | | | |
| **Covered requirements** | | | |
| ARCH\_SW\_PMP\_0023; ARCH\_SW\_PMP\_0024; | | | |

### Called functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| **ARCH\_SW\_PMP\_0024** | The AdcIf\_runGetMotor\_Vp\_(Ext)\* function shall be called to get the sampled value. | (Ext)\* from ADC and PMP services are the same |  |

## PMP\_GetCorrectedTemperature()

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| void **PMP\_GetCorrectedTemperature** (u16InternalTemperatureType \* const pu16Temperature) | | | |
| **Object** | | | |
| * This function shall provide the CORRECTED Temperature of the ECU in LSB (not FILTERED). | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| pu16Temperature | u16InternalTemperatureType | Out | Value of the temperature directly taken out from the ADC |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Synchronous server operation  Reentrant | | | |
| **Covered requirements** | | | |
| ARCH\_SW\_PMP\_9014; ARCH\_SW\_PMP\_9009 | | | |

### Called functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| **ARCH\_SW\_PMP\_9009** | The AdcIf\_runGetMcu\_T\_10ms function shall be called to get the sampled value of the unfiltered ECU temperature from the ADC. |  |  |

## PMP\_ runGetHS1Level()

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| void **PMP\_runGetHS1Level** (u16VoltageType \* pu16Voltage) | | | |
| **Object** | | | |
| * This function shall provide the value of HighSide Switch Voltage from the ADC (not FILTERED). | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| pu16Voltage | u16VoltageType | Out | Get the value of HighSide Switch Voltage from the ADC |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Synchronous server operation  Reentrant | | | |
| **Covered requirements** | | | |
| ARCH\_SW\_PMP\_0203; ARCH\_SW\_PMP\_0204; | | | |

### Called functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| **ARCH\_SW\_PMP\_0204** | The AdcIf\_runGetHS1\_V function shall be called to get the sampled value of the HighSide Switch from the ADC. |  |  |

## PMP\_runGetBatteryVoltageSurveyStatus ()

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| void **PMP\_runGetBatteryVoltageSurveyStatus** (u16VoltageType \* pu16Voltage) | | | |
| **Object** | | | |
| * This function shall provide the actual battery state. | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| pu8BatteryVoltageSurveyStatus | u8BatteryVoltageSurveyStatusType | Out | The battery state value. |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Synchronous server operation  Reentrant | | | |
| **Covered requirements** | | | |
| ARCH\_SW\_PMP\_0205; ARCH\_SW\_PMP\_0206; | | | |

### Called functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| **ARCH\_SW\_PMP\_0206** | The PMP\_stBatterySurveyParam.u8BatteryVoltageSurveyState value shall be sent externally. |  |  |

## PMP\_runSurveyBatteryVoltage

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| void **PMP\_runSurveyBatteryVoltage** (void) | | | |
| **Object** | | | |
| This function shall monitor the stability of the battery voltage and update a battery status variable for the rest of application. | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| NA | NA | NA | NA |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Synchronous server operation  Reentrant | | | |
| **Covered requirements** | | | |
| ARCH\_SW\_PMP\_0200; ARCH\_SW\_PMP\_0201; ARCH\_SW\_PMP\_0202; ARCH\_SW\_PMP\_0210 | | | |

### Data flow / Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| **ARCH\_SW\_PMP\_0201** | The NVP\_u16KL30MinimumThr parameter shall be read to detect unstable battery voltage situation |  |  |
| **ARCH\_SW\_PMP\_0202** | The NVP\_u16KL30MaximumThr parameter shall be read to detect unstable battery voltage situation |  |  |

The stability of battery voltage is tested within a voltage range defined in NVP parameters. If the measured battery voltage is outside the voltage range, a variable shall be set to BATTERY\_UNSTABLE otherwise BATTERY\_STABLE.

### Called functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| **ARCH\_SW\_PMP\_0210** | The AdcIf\_runGetKL30\_V function shall be called to get the sampled value of the KL30 signal as a raw value  function shall be called to get the sampled value of the Cpu regulator signal |  |  |

## PMP\_runGetFilteredTemperature

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| void **PMP\_runGetFilteredTemperature** (u16InternalTemperatureType \* pu16Temperature) | | | |
| **Object** | | | |
| This function shall provide the temperature sampled by the Adc unit but right aligned on 16-bits long and saturated.  Refer to the interface definition for more details. | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| pu16Temperature | u16InternalTemperatureType | Out | Temperature measured by the MCU, rescaled in °C, right aligned and saturated in a real physical range  Look-up table is used to scale the sampled value of the temperature in °C |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Synchronous server operation  Reentrant | | | |
| **Covered requirements** | | | |
| ARCH\_SW\_PMP\_0260; ARCH\_SW\_PMP\_0265 | | | |

### Called functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_PMP\_0265 | The PMP\_GetCorrectedTemperature function shall be called to get the sampled value of the temperature (measured at MCU level) |  |  |

## PMP\_runGetFilteredTemperature\_deg

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| void **PMP\_runGetFilteredTemperature\_deg** (s16InternalTemperature\_deg\_Type \* s16FilteredTemperature\_Deg) | | | |
| **Object** | | | |
| This function shall provide the temperature in Celsius degrees (based on the temperature sampled by the Adc unit) | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| s16FilteredTemperature | s16InternalTemperature\_deg\_Type | Out | Temperature measured by the MCU, rescaled in °C, right aligned |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Synchronous server operation  Reentrant | | | |
| **Covered requirements** | | | |
| ARCH\_SW\_PMP\_0280; ARCH\_SW\_PMP\_0285 | | | |

### Called functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_PMP\_0285 | The PMP\_runGetFilteredTemperature function shall be called to get the sampled value of the temperature (measured at MCU level) |  |  |

## PMP\_runGetDeficiencyLevel

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| void **PMP\_runGetDeficiencyLevel** (u32DeficiencyLevelType \* pu32DeficiencyLevel) | | | |
| **Object** | | | |
| This function shall provide the estimation of the energy stored in the motor. | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| pu32DeficiencyLevel | u32DeficiencyLevelType | Out | Energy stored in the motor |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Synchronous server operation  Reentrant | | | |
| **Covered requirements** | | | |
| ARCH\_SW\_PMP\_0320 | | | |

## PMP\_runComputeDeficiencyLevel

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| void **PMP\_runComputeDeficiencyLevel** (void) | | | |
| **Object** | | | |
| This function shall compute the estimation of the energy stored in the motor, also called deficiency level. | | | |
|  | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| NA | NA | NA | NA |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Periodic – 100ms, called on task  Non-reentrant | | | |
| **Covered requirements** | | | |
| **ARCH\_SW\_PMP\_0340; ARCH\_SW\_PMP\_9006;** **ARCH\_SW\_PMP\_9007; ARCH\_SW\_PMP\_9008; ARCH\_SW\_PMP\_9004; ARCH\_SW\_PMP\_9005; ARCH\_SW\_PMP\_9001; ARCH\_SW\_PMP\_9002; ARCH\_SW\_PMP\_9003; ARCH\_SW\_PMP\_0341** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| **ARCH\_SW\_PMP\_9006** | If the current consumption is different than 0 and temperature is below 35 degrees, the new deficiency level shall be computed by adding the square current to the previous deficiency level. |  |  |
| **ARCH\_SW\_PMP\_9007** | If the current consumption is different than 0 and the temperature is above 35 degrees, the new deficiency level shall be computed using the following formula:  new def level= previous def level+ square current \* (1+2\*(temperature-35)/100). |  |  |
| **ARCH\_SW\_PMP\_9008** | If the current consumption is 0, the deficiency level shall be decreased with the value of NVP\_u8SWThermalProtecDecStep. |  |  |

### Data flow / Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| **ARCH\_SW\_PMP\_9004** | NVP\_u8SWThermalProtecDecStep shall be used to get the decrement step used to decrease the deficiency level when current consumption is 0. |  |  |
| **ARCH\_SW\_PMP\_9005** | NVP\_u32SWThermalProtectionValidationThrs shall be used to get the maximum value of the deficiency level to which its value will be saturated. |  |  |

### Called functions

|  |  |  |
| --- | --- | --- |
| **Requirement ID** | **Decription** | **Source** |
| ARCH\_SW\_PMP\_0341 | PAL\_runReadMotorCurrentInA shall be called to get the measured current in amps. |  |
| ARCH\_SW\_PMP\_9001 | Rte\_IrvRead\_PMP\_runComputeDeficiencyLevel\_u32DeficiencyLevel shall be used to get the previous deficiency level. |  |
| ARCH\_SW\_PMP\_9002 | PMP\_runGetFilteredTemperature\_deg shall be called to get the temperature in degrees. |  |
| ARCH\_SW\_PMP\_9003 | Rte\_IrvWrite\_PMP\_runComputeDeficiencyLevel\_u32DeficiencyLevel shall be used to write the new deficiency level. |  |

## PMP\_ runGetBoardResistance

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| void **PMP\_runGetBoardResistance** (void) | | | |
| **Object** | | | |
| This function shall return the NVP Parameter that indicates the ECU resistance. | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| NA | NA | NA | NA |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Synchronous server operation  Reentrant | | | |
| **Covered requirements** | | | |
| ARCH\_SW\_PMP\_0381; ARCH\_SW\_PMP\_0382; | | | |

### Data flow / Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| **ARCH\_SW\_PMP\_0382** | The NVP\_u16EcuResistance parameter shall be read. |  |  |

## PMP\_ runGetMotorResistance

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| void **PMP\_runGetMotorResistance** (void) | | | |
| **Object** | | | |
| This function shall return the motor resistance in mOhm by using the NVP look-up table. | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| NA | NA | NA | NA |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Synchronous server operation  Reentrant | | | |
| **Covered requirements** | | | |
| ARCH\_SW\_PMP\_0383; ARCH\_SW\_PMP\_0384; ARCH\_SW\_PMP\_0385; | | | |

### Data flow / Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| **ARCH\_SW\_PMP\_0384** | The NVP\_au16MotorResistTempIndex parameter shall be read. |  |  |
| **ARCH\_SW\_PMP\_0385** | The NVP\_au8MotorResistReferences parameter shall be read. |  |  |

# Runnables – Auto-tests

## PMP\_Autotest\_CheckPowerSupplyOV

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| u8TestResultType **PMP\_Autotest\_CheckPowerSupplyOV** (void) | | | |
| **Object** | | | |
| This function shall check if an overvoltage is detected on the power stage of the battery voltage. | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| pu8TestResult | u8TestResultType | OUT | Status of the test |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Server operation  Non Reentrant | | | |
| **Covered requirements** | | | |
| ARCH\_SW\_PMP\_1000; ARCH\_SW\_PMP\_1001; ARCH\_SW\_PMP\_1005; ARCH\_SW\_PMP\_1007; ARCH\_SW\_PMP\_1008; | | | |

### Data flow

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_PMP\_1001 | NVP parameters shall be used by this auto-tests |  | ALV\_EXT\_TF\_H\_328; ALV\_EXT\_TF\_H\_331; |

### Called functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_PMP\_1005 | PMP\_runGetBatteryVoltage\_10ms shall be called to get the measured value on the power supply |  | ALV\_EXT\_TF\_H\_322; |
| ARCH\_SW\_PMP\_1007 | ATM\_runGetTestResult shall be called to get the status oft specific tests (inhibiting the execution of the present one if failed). |  | ALV\_EXT\_TF\_H\_339; |
| ARCH\_SW\_PMP\_1008 | ERH\_runGetAecStatus shall be called to get the qualification status of a specific AEC. |  |  |

## PMP\_Autotest\_CheckPowerSupplyUV

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| u8TestResultType **PMP\_Autotest\_CheckPowerSupplyUV** (void) | | | |
| **Object** | | | |
| This function shall check if an undervoltage is detected on the power stage of the battery voltage. | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| pu8TestResult | u8TestResultType | OUT | Status of the test |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Server operation  Non Reentrant | | | |
| **Covered requirements** | | | |
| ARCH\_SW\_PMP\_1050; ARCH\_SW\_PMP\_1051; ARCH\_SW\_PMP\_1055; ARCH\_SW\_PMP\_1057; ARCH\_SW\_PMP\_1058; ARCH\_SW\_PMP\_1059; ARCH\_SW\_PMP\_1054; | | | |

### Data flow

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_PMP\_1051 | NVP parameters shall be used by this auto-tests |  | ALV\_EXT\_TF\_H\_367; ALV\_EXT\_TF\_H\_370; |

### Called functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_PMP\_1055 | [PMP\_runGetBatteryVoltage\_10ms](#_Hlk411940329) shall be called to get the measured value on the power supply |  | ALV\_EXT\_TF\_H\_361; |
| ARCH\_SW\_PMP\_1057 | ATM\_runGetTestResult shall be called to get the status ot specific tests (inhibiting the execution of the present one if failed). |  | ALV\_EXT\_TF\_H\_379; |
| ARCH\_SW\_PMP\_1058 | MMG\_runCheckModeStatus shall be called to get the tensioning state. |  | ALV\_EXT\_TF\_H\_378; |
| ARCH\_SW\_PMP\_1059 | ERH\_runGetAecStatus shall be called to get the qualification status of a specific AEC. |  |  |

## PMP\_Autotest\_CheckPowerSupplyUVDuringActivation

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| u8TestResultType **PMP\_Autotest\_CheckPowerSupplyUVDuringActivation** (void) | | | |
| **Object** | | | |
| This function shall check if an undervoltage is detected on the power stage of the battery voltage during activation. | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| pu8TestResult | u8TestResultType | OUT | Status of the test |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Server operation  Non Reentrant | | | |
| **Covered requirements** | | | |
| ARCH\_SW\_PMP\_1060; ARCH\_SW\_PMP\_1061; ARCH\_SW\_PMP\_1065; ARCH\_SW\_PMP\_1067; ARCH\_SW\_PMP\_1080; ARCH\_SW\_PMP\_1068; ARCH\_SW\_PMP\_1069; ARCH\_SW\_PMP\_1064; | | | |

### Data flow

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_PMP\_1061 | NVP parameters shall be used by this auto-tests |  | ALV\_EXT\_TF\_H\_406; ALV\_EXT\_TF\_H\_409; |

### Called functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_PMP\_1065 | [PMP\_runGetBatteryVoltage\_10ms](#_Hlk411940329) shall be called to get the measured value on the power supply |  | ALV\_EXT\_TF\_H\_400; |
| ARCH\_SW\_PMP\_1067 | ATM\_runGetTestResult shall be called to get the status ot specific tests (inhibiting the execution of the present one if failed). |  | DAI\_EXT\_TF\_H\_2278; |
| ARCH\_SW\_PMP\_1068 | MMG\_runCheckModeStatus shall be called to get the tensioning state. |  | ALV\_EXT\_TF\_H\_378; |
| ARCH\_SW\_PMP\_1069 | ERH\_runGetAecStatus shall be called to get the qualification status of a specific AEC. |  |  |
| ARCH\_SW\_PMP\_1064 | Rte\_CIL\_psrCustomerSpecific\_u8TriggerRequest shall be called to get the status of the request |  |  |

## PMP\_Autotest\_CheckPowerSupplyOVDuringActivation

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| u8TestResultType **PMP\_Autotest\_CheckPowerSupplyOVDuringActivation** (void) | | | |
| **Object** | | | |
| This function shall check if an overvoltage is detected on the power stage of the battery voltage during activation. | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| pu8TestResult | u8TestResultType | OUT | Status of the test |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Server operation  Non Reentrant | | | |
| **Covered requirements** | | | |
| ARCH\_SW\_PMP\_1070; ARCH\_SW\_PMP\_1071; ARCH\_SW\_PMP\_1075; ARCH\_SW\_PMP\_1077; ARCH\_SW\_PMP\_1078; ARCH\_SW\_PMP\_1079; | | | |

### Data flow

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_PMP\_1071 | NVP parameters shall be used by this auto-tests |  | DAI\_EXT\_TF\_H\_2255; DAI\_EXT\_TF\_H\_2256; |

### Called functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_PMP\_1075 | [PMP\_runGetBatteryVoltage\_10ms](#_Hlk411940329) shall be called to get the measured value on the power supply |  | DAI\_EXT\_TF\_H\_2242; |
| ARCH\_SW\_PMP\_1077 | ATM\_runGetTestResult shall be called to get the status ot specific tests (inhibiting the execution of the present one if failed). |  | ALV\_EXT\_TF\_H\_2279; |
| ARCH\_SW\_PMP\_1078 | MMG\_runCheckModeStatus shall be called to get the tensioning state. |  | ALV\_EXT\_TF\_H\_378; |
| ARCH\_SW\_PMP\_1079 | ERH\_runGetAecStatus shall be called to get the qualification status of a specific AEC. |  |  |

## PMP\_Autotest\_CheckTemperatureSensor

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| void PMP\_Autotest\_CheckTemperatureSensor(u8TestResultType \* pu8TestResult) | | | |
| **Object** | | | |
| This autotest aims at detecting a SBC temperature error. | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| pu8TestResult | u8TestResultType | OUT | Status of the test |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Server operation  Non Reentrant | | | |
| **Covered requirements** | | | |
| ARCH\_SW\_PMP\_1081; ARCH\_SW\_PMP\_1082; | | | |

### Data flow

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_PMP\_1081 | NVP parameters shall be used by this auto-tests |  | ALV\_EXT\_TF\_H\_1363; |

### Called functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_PMP\_1082 | ATM\_runGetTestResult shall be called to get the status ot specific tests (inhibiting the execution of the present one if failed). |  | DAI\_EXT\_TF\_H\_2240; |

## PMP\_Autotest\_CheckSystemOV

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| u8TestResultType **PMP\_Autotest\_CheckSystemOV** (void) | | | |
| **Object** | | | |
| This function shall check if an overvoltage is detected on the signal PN14\_SupBat\_Volt\_ST3. | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| pu8TestResult | u8TestResultType | OUT | Status of the test |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Server operation  Non Reentrant | | | |
| **Covered requirements** | | | |
| ARCH\_SW\_PMP\_1102; ARCH\_SW\_PMP\_1101; ARCH\_SW\_PMP\_1100; ARCH\_SW\_PMP\_1120; | | | |

### Data flow

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_PMP\_1101 | Value of signal PN14\_SupBat\_Volt\_ST3 shall be an input. |  |  |

### Called functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_PMP\_1102 | ERH\_runGetAecStatus shall be called to get the qualification status of a specific AEC. |  |  |

## PMP\_Autotest\_CheckSystemUV

### Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Prototype** | | | |
| u8TestResultType **PMP\_Autotest\_CheckSystemUV** (void) | | | |
| **Object** | | | |
| This function shall check if an undervoltage is detected on the signal PN14\_SupBat\_Volt\_ST3. | | | |
| **Parameters** | | | |
| Name | Type | Direction | Description |
| pu8TestResult | u8TestResultType | OUT | Status of the test |
| **Returned value** | | | |
| Name | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Server operation  Non Reentrant | | | |
| **Covered requirements** | | | |
| ARCH\_SW\_PMP\_1111; ARCH\_SW\_PMP\_1112; ARCH\_SW\_PMP\_1110; ARCH\_SW\_PMP\_1120; | | | |

### Data flow

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_PMP\_1111 | Value of signal PN14\_SupBat\_Volt\_ST3 shall be an input. |  |  |

### Called functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_PMP\_1112 | ERH\_runGetAecStatus shall be called to get the qualification status of a specific AEC. |  |  |

# MCU resources

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Criteria** | **Levels/Tolerances** | **Source** |
| ARCH\_SW\_PMP\_9997 | The ROM size consumed by this component shall not exceed 2.3K bytes. |  |  |
| ARCH\_SW\_PMP\_9998 | The heap size consumed by this component shall be 70 bytes. |  |  |