|  |  |
| --- | --- |
| **PMP**  **Detailed Design Document** | |
| **Summary** | This is the Software Detailed Design Document for the PMP component of *DAIMLER MMA* Project. |

|  |  |  |
| --- | --- | --- |
| **Author** | **Review** | **Approval** |
| SW Dev: Septimiu-Darex C. Vintila | SW Arch: Andreea Negrea | SW PM: Andris Daniel |
|  |  |  |
| **Distribution** | | |
| See Project Master Document for the roles and Project Members List for the name of people | See Project Master Document for the roles and Project Members List for the name of people | See Project Master Document for the roles and Project Members List for the name of people |

# Table of content

1.1. Revision history \* 5

1.2. Purpose and Scope 5

1.3. Referenced documents 5

1.3.1. External documents 5

1.3.2. Internal Documents 5

1.3.3. Terminology and definitions 5

2.1. Overview 7

2.2. Traceability 7

3.1. Services 13

3.1.1. PMP\_Init 13

3.1.2. PMP\_runComputeDeficiencyLevel 14

3.1.3. PMP\_runGetFilteredTemperature 15

3.1.4. PMP\_runGetFilteredTemperature\_deg 16

3.1.5. PMP\_runSurveyBatteryVoltage 17

3.1.6. PMP\_runGetBatteryVoltage 19

3.1.7. PMP\_runGetBatteryVoltage\_2ms 19

3.1.8. PMP\_runGetBatteryVoltage\_10ms 20

3.1.9. PMP\_runGetDeficiencyLevel 21

3.1.10. PMP\_runGetMotorResistance 22

3.1.11. pmp\_GetCorrectedTemperature 22

3.1.12. pmp\_runGetBatteryVoltage 23

3.1.13. pmp\_runGetBatteryVoltage\_20ms 24

3.1.14. pmp\_runGetBatteryVoltage\_100ms 25

3.1.15. pmp\_runGetBatteryVoltage\_10ms 26

3.1.16. pmp\_runGetBatteryVoltage\_2ms 27

3.1.17. PMP\_runGetBoardResistance 28

3.1.18. PMP\_runGetMotorVn 29

3.1.19. PMP\_runGetMotorVn\_20ms 30

3.1.20. PMP\_runGetMotorVn\_100ms 30

3.1.21. PMP\_runGetMotorVp 31

3.1.22. PMP\_runGetMotorVp\_20ms 32

3.1.23. PMP\_runGetMotorVp\_100ms 33

3.1.24. PMP\_runGetHS1Level 34

3.1.25. PMP\_runGetBatteryVoltageSurveyStatus 34

3.2. Auto-Tests 35

3.2.1. PMP\_Autotest\_CheckTemperatureSensor 35

3.2.2. PMP\_Autotest\_CheckPowerSupplyOV 36

3.2.3. PMP\_Autotest\_CheckPowerSupplyUV 37

3.2.4. PMP\_Autotest\_CheckPowerSupplyUVDuringActivation 38

3.2.5. PMP\_Autotest\_CheckPowerSupplyOVDuringActivation 39

3.2.6. PMP\_Autotest\_CheckSystemUV 40

3.2.7. PMP\_Autotest\_CheckSystemOV 41

3.3. Types 42

3.3.1. PMP\_stBatterySurveyParamType 42

3.4. Variables 43

3.4.1. PMP\_stBatterySurveyParam 43

3.4.2. PMP\_u16OldADCTemperature\_LSB 43

3.4.3. PMP\_u16OldCorrectedTemperature\_LSB 43

3.5. Macros 43

3.5.1. PMP\_KU8\_MOTOR\_RESISTANCE\_LINEAR\_INTERPOLATION\_NB\_POINTS 43

3.5.2. PMP\_RIGHT\_ALIGNED\_ADC 43

3.5.3. PMP\_LEFT\_ALIGNED\_ADC 44

3.5.4. PMP\_ALIGNMENT\_ADC\_CHANNELS 44

3.5.5. PMP\_KU16\_ADC\_10BIT\_RANGE 44

3.5.6. PMP\_KU16\_ADC\_8BIT\_RANGE 44

3.5.7. PMP\_KU16\_ADC\_FULL\_RANGE 44

3.5.8. PMP\_KU16\_ADC\_REFERENCE\_MV 44

3.5.9. PMP\_KU32\_HVI\_INTERNAL\_FACTOR 45

3.5.10. PMP\_KU32\_KL30\_RAW\_MUL\_FACTOR 45

3.5.11. PMP\_KU32\_KL30\_RAW\_DIV\_FACTOR 45

3.5.12. PMP\_KU32\_KL30\_MUL\_FACTOR 45

3.5.13. PMP\_KU32\_KL30\_DIV\_FACTOR 45

3.5.14. PMP\_KU32\_VN\_MUL\_FACTOR 46

3.5.15. PMP\_KU32\_VN\_DIV\_FACTOR 46

3.5.16. PMP\_KU32\_VP\_MUL\_FACTOR 46

3.5.17. PMP\_KU32\_VP\_DIV\_FACTOR 46

3.5.18. PMP\_KU32\_HS\_MUL\_FACTOR 46

3.5.19. PMP\_KU32\_HS\_DIV\_FACTOR 47

3.5.20. PMP\_KU32\_REFLOW\_MUL\_FACTOR 47

3.5.21. PMP\_KU32\_REFLOW\_DIV\_FACTOR 47

3.5.22. PMP\_KU32\_REFMID\_MUL\_FACTOR 47

3.5.23. PMP\_KU32\_REFMID\_DIV\_FACTOR 47

3.5.24. PMP\_KU32\_REFHIGH\_MUL\_FACTOR 47

3.5.25. PMP\_KU32\_REFHIGH\_DIV\_FACTOR 48

3.5.26. PMP\_KU32\_REFCPUREG\_MUL\_FACTOR 48

3.5.27. PMP\_KU32\_REFCPUREG\_DIV\_FACTOR 48

3.5.28. PMP\_KU16\_ADC\_RANGE\_MV 48

3.5.29. PMP\_KU16\_ADC\_OUT\_SCALE 48

3.5.30. PMP\_KU8\_NB\_INTERP\_PTS\_TEMP\_CALIB 49

3.5.31. PMP\_KU8\_DEG\_RANGE 49

3.5.32. PMP\_KU8\_DEG\_LOW\_RANGE 49

3.5.33. PMP\_KU32\_TYP\_H\_SLOPE\_UV\_PER\_DEG 49

3.5.34. PMP\_KU16\_ADC\_OUT\_H\_SCALE 49

3.5.35. PMP\_KU32\_ADC\_RANGE\_UV 49

3.5.36. PMP\_KU32\_ADC\_H\_RANGE\_UV 50

3.5.37. PMP\_KU32\_REF\_TEMP\_UV 50

3.5.38. PMP\_KU32\_ADC\_INPUT\_UV 50

3.5.39. PMP\_KU32\_ADC\_OUTPUT 50

3.5.40. PMP\_KU32\_REF\_TEMP\_ADC 50

3.5.41. PMP\_KS16\_ADC\_TO\_DEG 51

3.5.42. PMP\_KU16\_DEG\_TO\_ADC 51

3.5.43. PMP\_KU16\_COMPUTE\_TEMP\_SLOP 51

3.5.44. PMP\_KU16\_FIX\_TEMP\_ADC 51

3.5.45. PMP\_KU32\_TYP\_SLOPE\_UV\_PER\_DEG 52

3.5.46. PMP\_KU16\_REF\_TEMP\_DEG 52

3.5.47. PMP\_KU16\_REF\_TEMP\_MV 52

3.5.48. PMP\_KS16\_TEMP\_HIGH 52

3.5.49. PMP\_KU32\_TWO 52

3.5.50. PMP\_KU32\_ONE\_HUNDRED 53

3.5.51. PMP\_KU16\_ADC\_LEFT\_ALIGNED\_BIT\_SHIFT 53

3.5.52. PMP\_KU16\_FIVE\_HUNDRED 53

3.5.53. PMP\_KU8\_NO\_REQUEST 53

# Tables of figures

[Figure 1: PMP\_Init diagram 14](#_Toc134024446)

[Figure 2: PMP\_runComputeDeficiencyLevel diagram 15](#_Toc134024447)

[Figure 3: PMP\_runGetFilteredTemperature diagram 16](#_Toc134024448)

[Figure 4: PMP\_runGetFilteredTemperature\_deg diagram 17](#_Toc134024449)

[Figure 5: PMP\_runSurveyBatteryVoltage diagram 18](#_Toc134024450)

[Figure 6: PMP\_runGetBatteryVoltage diagram 19](#_Toc134024451)

[Figure 7: PMP\_runGetBatteryVoltage\_2ms diagram 20](#_Toc134024452)

[Figure 8: PMP\_runGetBatteryVoltage\_10ms diagram 21](#_Toc134024453)

[Figure 9: PMP\_runGetDeficiencyLevel diagram 22](#_Toc134024454)

[Figure 10: PMP\_runGetMotorResistance diagram 22](#_Toc134024455)

[Figure 11 pmp\_GetCorrectedTemperature 23](#_Toc134024456)

[Figure 12 PMP\_runGetBatteryVoltage 24](#_Toc134024457)

[Figure 13 PMP\_runGetBatteryVoltage\_20ms 25](#_Toc134024458)

[Figure 14 PMP\_runGetBatteryVoltage\_100ms 26](#_Toc134024459)

[Figure 15 PMP\_runGetBatteryVoltage\_10ms 27](#_Toc134024460)

[Figure 16 PMP\_runGetBatteryVoltage\_2ms 28](#_Toc134024461)

[Figure 17 PMP\_runGetBoardResistance 29](#_Toc134024462)

[Figure 18 PMP\_runGetMotorVn 30](#_Toc134024463)

[Figure 19 PMP\_runGetMotorVn\_20ms 30](#_Toc134024464)

[Figure 20 PMP\_runGetMotorVn\_100ms 31](#_Toc134024465)

[Figure 21 PMP\_runGetMotorVp 32](#_Toc134024466)

[Figure 22 PMP\_runGetMotorVp\_20ms 33](#_Toc134024467)

[Figure 23 PMP\_runGetMotorVp\_100ms 33](#_Toc134024468)

[Figure 24 PMP\_runGetHS1Level 34](#_Toc134024469)

[Figure 25 PMP\_runGetBatteryVoltageSurveyStatus 35](#_Toc134024470)

[Figure 26 PMP\_Autotest\_CheckTemperatureSensor 36](#_Toc134024471)

[Figure 27 PMP\_Autotest\_CheckPowerSupplyOV 37](#_Toc134024472)

[Figure 28 PMP\_Autotest\_CheckPowerSupplyUV 38](#_Toc134024473)

[Figure 29 PMP\_Autotest\_CheckPowerSupplyUVDuringActivation 39](#_Toc134024474)

[Figure 30 PMP\_Autotest\_CheckPowerSupplyOVDuringActivation 40](#_Toc134024475)

[Figure 31 PMP\_Autotest\_CheckSystemUV 41](#_Toc134024476)

[Figure 32 PMP\_Autotest\_CheckSystemOV 42](#_Toc134024477)

# General Information

## Revision history \*

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Author(s)** | **Description/comment** |
| 1.1.3.1 | 07.01.2022 | Septimiu-Darex C. Vintila | Initial revision. |
| 1.1.3.2 | 07.01.2022 | Septimiu-Darex C. Vintila | Services added. |
| 1.1.3.3 | 09.02.2022 | Septimiu-Darex C. Vintila | Corrections after review. |
| 1.1.3.4 | 24.03.2022 | Septimiu-Darex C. Vintila | Add new requirements. |
| 1.1.3.5 | 29.03.2022 | Septimiu-Darex C. Vintila | Typo in requirements source corrected. |
| 1.1.3.6 | 11.04.2022 | Septimiu-Darex C. Vintila | Distribution fields updated. |
| 1.1.3.7 | 15.06.2022 | Obada Mirela | New autotests and traceability added |
| 1.1.3.8 | 17.06.2022 | Obada Mirela | Update after review |
| 1.1.3.9 | 22.06.2022 | Obada Mirela | Update fnctions template |
| 1.1.3.10 | 06.09.2022 | Madalina Serban | Added new functions and traceability |
| 1.1.3.11 | 08.09.2022 | Madalina Serban | Fixed review findings |
| 1.1.3.12 | 10.02.2023 | Mirela Obada | Update traceability for 5.0 |
| 1.1.3.13 | 15.02.2023 | Mirela Obada | Update for 5.0 |
| 1.1.3.14 | 16.02.2023 | Mirela Obada | Update for 5.0 after new corrections of architecture |
| 1.1.3.15 | 03.05.2023 | Mirela Obada | Update for 6.0 |
| 1.1.3.16 | 03.05.2023 | Mirela Obada | Update after review |
| 1.1.3.17 | 04.10.2023 | Mirela Obada | Update UVT & OVT autotests diagrams |

*\* Template history is found in the CM tool used for templates*

## Purpose and Scope

The purpose of this document is to establish the functionality and behavior of the PMP component

## Referenced documents

### External documents

|  |  |  |
| --- | --- | --- |
| **Id** | **Title** | **Reference** |
|  |  |  |
|  |  |  |
|  |  |  |

### Internal Documents

|  |  |  |
| --- | --- | --- |
| **Id** | **Title** | **Reference** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

### Terminology and definitions

The generic acronyms are available in the [AEM process and method wiki](https://alvteams.alv.autoliv.int/sites/aeuaeequalityassurance/AEM%20Process%20wiki/acronyms.aspx)

|  |  |
| --- | --- |
| **Terminology** | **Meaning** |
| AAU | Atomic architectural unit |
| SW | software |
| PMP | Physical Measures Provider |

# SW atomic architectural unit design

## Overview

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).

## Traceability

|  |  |  |  |
| --- | --- | --- | --- |
| Requirements | Criteria | Linked Runnable | Source |
| DSG\_PMP\_0001 | The default value for the previous result of battery voltage shall be 0. | PMP\_Init() | ARCH\_SW\_PMP\_0005 |
| DSG\_PMP\_0002 | Maximum threshold for KL30 voltage shall be initialize with a value read from non-volatile memory and aligned to the used ADC format. | PMP\_Init() | ARCH\_SW\_PMP\_0005 |
| DSG\_PMP\_0003 | Minimum threshold for KL30 voltage shall be initialize with a value read from non-volatile memory and aligned to the used ADC format. | PMP\_Init() | ARCH\_SW\_PMP\_0005 |
| DSG\_PMP\_0004 | The default value for the battery survey state shall be KU8\_BATTERY\_UNSTABLE. | PMP\_Init() | ARCH\_SW\_PMP\_0005 |
| DSG\_PMP\_0005 | The default value for the occurrences counter shall be 0. | PMP\_Init() | ARCH\_SW\_PMP\_0005 |
| DSG\_PMP\_0006 | The deficiency level shall be calculated based on previous deficiency level value, the absolute value of measured motor current and the internal temperature. | PMP\_runComputeDeficiencyLevel() | ARCH\_SW\_PMP\_0340  ARCH\_SW\_PMP\_9006  ARCH\_SW\_PMP\_9007  ARCH\_SW\_PMP\_9008  ARCH\_SW\_PMP\_0341  ARCH\_SW\_PMP\_9001  ARCH\_SW\_PMP\_9002  ARCH\_SW\_PMP\_9003 |
| DSG\_PMP\_0007 | In case the motor current is null and the previous deficiency level value is below the related threshold, the deficiency level shall become null. | PMP\_runComputeDeficiencyLevel() | ARCH\_SW\_PMP\_0340  ARCH\_SW\_PMP\_9004  ARCH\_SW\_PMP\_9001  ARCH\_SW\_PMP\_9003 |
| DSG\_PMP\_0008 | In case the motor current is null and the previous deficiency level value is over the related threshold, the deficiency level shall be the difference between the previous deficiency level value and the threshold. | PMP\_runComputeDeficiencyLevel() | ARCH\_SW\_PMP\_0340  ARCH\_SW\_PMP\_9004  ARCH\_SW\_PMP\_9001  ARCH\_SW\_PMP\_9003 |
| DSG\_PMP\_0009 | In case the calculated deficiency level exceeds the authorized range, the deficiency level value shall be saturated to the related threshold value. | PMP\_runComputeDeficiencyLevel() | ARCH\_SW\_PMP\_0340  ARCH\_SW\_PMP\_9005  ARCH\_SW\_PMP\_9003 |
| DSG\_PMP\_0010 | The filtered temperature (in degree) shall be made available to the rest of the application. | PMP\_runGetFilteredTemperature() PMP\_runGetFilteredTemperature\_deg() | ARCH\_SW\_PMP\_0260  ARCH\_SW\_PMP\_0280  ARCH\_SW\_PMP\_0285 |
| DSG\_PMP\_0011 | The corrected temperature (in degree) shall be made available to the rest of the application. | pmp\_GetCorrectedTemperature() | ARCH\_SW\_PMP\_9014  ARCH\_SW\_PMP\_0265 |
| DSG\_PMP\_0012 | The stability of the battery voltage shall be monitored based on consecutive measurements of the KL30 voltage. | PMP\_runSurveyBatteryVoltage() | ARCH\_SW\_PMP\_0200  ARCH\_SW\_PMP\_0201  ARCH\_SW\_PMP\_0202 |
| DSG\_PMP\_0013 | In case the measured battery voltage is outside the authorized range, the battery voltage status shall take the value KU8\_BATTERY\_UNSTABLE. | PMP\_runSurveyBatteryVoltage() | ARCH\_SW\_PMP\_0200  ARCH\_SW\_PMP\_0201  ARCH\_SW\_PMP\_0202 |
| DSG\_PMP\_0014 | In case the measured battery voltage is in the authorized range and the variation between two consecutive measurements is in range, the battery voltage status shall take the value KU8\_BATTERY\_STABLE. | PMP\_runSurveyBatteryVoltage() | ARCH\_SW\_PMP\_0200 |
| DSG\_PMP\_0015 | In case the measured battery voltage is in the authorized range and the variation between two consecutive measurements is outside range, the battery voltage status shall take the value KU8\_BATTERY\_UNSTABLE. | PMP\_runSurveyBatteryVoltage() | ARCH\_SW\_PMP\_0200 |
| DSG\_PMP\_0016 | The raw battery voltage shall be updated every 1ms. | PMP\_runGetBatteryVoltage() | ARCH\_SW\_PMP\_0010  ARCH\_SW\_PMP\_0020  ARCH\_SW\_PMP\_0210 |
| DSG\_PMP\_0017 | The raw battery voltage shall be updated every 2ms and made available to the system. | PMP\_runGetBatteryVoltage\_2ms() | ARCH\_SW\_PMP\_0011  ARCH\_SW\_PMP\_0020 |
| DSG\_PMP\_0018 | The raw battery voltage shall be updated every 10ms and made available to the system. | PMP\_runGetBatteryVoltage\_10ms() | ARCH\_SW\_PMP\_0012  ARCH\_SW\_PMP\_0020 |
| DSG\_PMP\_0019 | The raw battery voltage shall be updated every 100ms and made available to the system. | PMP\_runGetBatteryVoltage\_100ms() | ARCH\_SW\_PMP\_0014  ARCH\_SW\_PMP\_0020 |
| DSG\_PMP\_0020 | The motor resistance regarding the temperature shall be computed using a NVP look-up table. | PMP\_runGetMotorResistance() | ARCH\_SW\_PMP\_0383; ARCH\_SW\_PMP\_0384; ARCH\_SW\_PMP\_0385; |
| DSG\_PMP\_0021 | The calculated deficiency level shall made available to the rest of the application. | PMP\_runGetDeficiencyLevel() | ARCH\_SW\_PMP\_0320 |
| DSG\_PMP\_0022 | Power Overvoltage Autotest shall be executed every 10 ms. | PMP\_Autotest\_CheckPowerSupplyOV() | ARCH\_SW\_PMP\_1000; ARCH\_SW\_PMP\_1001; ARCH\_SW\_PMP\_1005; ARCH\_SW\_PMP\_1007; ARCH\_SW\_PMP\_1008 |
| DSG\_PMP\_0023 | In case an undervoltage was detected, the Power Overvoltage Autotest shall return KU8\_ATM\_TEST\_NOT\_DECIDED. | PMP\_Autotest\_CheckPowerSupplyOV() | ARCH\_SW\_PMP\_1000; ARCH\_SW\_PMP\_1001; ARCH\_SW\_PMP\_1005; ARCH\_SW\_PMP\_1007; ARCH\_SW\_PMP\_1008 |
| DSG\_PMP\_0024 | In case an overvoltage was previously detected and the measured voltage is below the deskilling threshold, the Power Overvoltage Autotest shall return KU8\_ATM\_TEST\_OK. | PMP\_Autotest\_CheckPowerSupplyOV() | ARCH\_SW\_PMP\_1000; ARCH\_SW\_PMP\_1001; ARCH\_SW\_PMP\_1005; ARCH\_SW\_PMP\_1007; ARCH\_SW\_PMP\_1008 |
| DSG\_PMP\_0025 | In case an overvoltage was previously detected and the measured voltage is over the deskilling threshold, the Power Overvoltage Autotest shall return KU8\_ATM\_TEST\_NOK. | PMP\_Autotest\_CheckPowerSupplyOV() | ARCH\_SW\_PMP\_1000; ARCH\_SW\_PMP\_1001; ARCH\_SW\_PMP\_1005; ARCH\_SW\_PMP\_1007; ARCH\_SW\_PMP\_1008 |
| DSG\_PMP\_0026 | In case an no overvoltage was previously detected and the measured voltage is below the qualification threshold, the Power Overvoltage Autotest shall return KU8\_ATM\_TEST\_OK. | PMP\_Autotest\_CheckPowerSupplyOV() | ARCH\_SW\_PMP\_1000; ARCH\_SW\_PMP\_1001; ARCH\_SW\_PMP\_1005; ARCH\_SW\_PMP\_1007; ARCH\_SW\_PMP\_1008 |
| DSG\_PMP\_0027 | In case an no overvoltage was previously detected and the measured voltage is over the qualification threshold, the Power Overvoltage Autotest shall return KU8\_ATM\_TEST\_NOK. | PMP\_Autotest\_CheckPowerSupplyOV() | ARCH\_SW\_PMP\_1000; ARCH\_SW\_PMP\_1001; ARCH\_SW\_PMP\_1005; ARCH\_SW\_PMP\_1007; ARCH\_SW\_PMP\_1008 |
| DSG\_PMP\_0028 | Power Undervoltage Autotest shall be executed every 10 ms. | PMP\_Autotest\_CheckPowerSupplyUV() | ARCH\_SW\_PMP\_1050; ARCH\_SW\_PMP\_1051; ARCH\_SW\_PMP\_1055; ARCH\_SW\_PMP\_1057; ARCH\_SW\_PMP\_1058; ARCH\_SW\_PMP\_1059; |
| DSG\_PMP\_0029 | In case an overvoltage was detected, the Power Undervoltage Autotest shall return KU8\_ATM\_TEST\_NOT\_DECIDED. | PMP\_Autotest\_CheckPowerSupplyUV() | ARCH\_SW\_PMP\_1050; ARCH\_SW\_PMP\_1051; ARCH\_SW\_PMP\_1055; ARCH\_SW\_PMP\_1057; ARCH\_SW\_PMP\_1058; ARCH\_SW\_PMP\_1059; |
| DSG\_PMP\_0030 | In case an undervoltage was previously detected and the measured voltage is over the deskilling threshold, the Power Undervoltage Autotest shall return KU8\_ATM\_TEST\_OK. | PMP\_Autotest\_CheckPowerSupplyUV() | ARCH\_SW\_PMP\_1050; ARCH\_SW\_PMP\_1051; ARCH\_SW\_PMP\_1055; ARCH\_SW\_PMP\_1057; ARCH\_SW\_PMP\_1058; ARCH\_SW\_PMP\_1059; |
| DSG\_PMP\_0031 | In case an undervoltage was previously detected and the measured voltage is under the deskilling threshold, the Power Undervoltage Autotest shall return KU8\_ATM\_TEST\_NOK. | PMP\_Autotest\_CheckPowerSupplyUV() | ARCH\_SW\_PMP\_1050; ARCH\_SW\_PMP\_1051; ARCH\_SW\_PMP\_1055; ARCH\_SW\_PMP\_1057; ARCH\_SW\_PMP\_1058; ARCH\_SW\_PMP\_1059; |
| DSG\_PMP\_0032 | In case an no undervoltage was previously detected and the measured voltage is over the qualification threshold, the Power Undervoltage Autotest shall return KU8\_ATM\_TEST\_OK. | PMP\_Autotest\_CheckPowerSupplyUV() | ARCH\_SW\_PMP\_1050; ARCH\_SW\_PMP\_1051; ARCH\_SW\_PMP\_1055; ARCH\_SW\_PMP\_1057; ARCH\_SW\_PMP\_1058; ARCH\_SW\_PMP\_1059; |
| DSG\_PMP\_0033 | In case an no undervoltage was previously detected and the measured voltage is under the qualification threshold, the Power Undervoltage Autotest shall return KU8\_ATM\_TEST\_NOK. | PMP\_Autotest\_CheckPowerSupplyUV() | ARCH\_SW\_PMP\_1050; ARCH\_SW\_PMP\_1051; ARCH\_SW\_PMP\_1055; ARCH\_SW\_PMP\_1057; ARCH\_SW\_PMP\_1058; ARCH\_SW\_PMP\_1059; |
| DSG\_PMP\_0034 | The sampled value of the unfiltered ECU temperature shall be taken from the ADC. | PMP\_runGetFilteredTemperature() | ARCH\_SW\_PMP\_9009 |
| DSG\_PMP\_0035 | Power Undervoltage Autotest During Activation shall be executed every 10 ms. | PMP\_Autotest\_CheckPowerSupplyUVDuringActivation () | 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 |
| DSG\_PMP\_0036 | In case an overvoltage was detected, the Power Undervoltage Autotest shall return KU8\_ATM\_TEST\_NOT\_DECIDED. | PMP\_Autotest\_CheckPowerSupplyUVDuringActivation() | 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 |
| DSG\_PMP\_0037 | In case an undervoltage was previously detected and the measured voltage is over the deskilling threshold, the Power Undervoltage During Activation Autotest shall return KU8\_ATM\_TEST\_OK. | PMP\_Autotest\_CheckPowerSupplyUVDuringActivation() | 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 |
| DSG\_PMP\_0038 | In case an undervoltage was previously detected and the measured voltage is under the deskilling threshold, the Power Undervoltage Autotest During Activation shall return KU8\_ATM\_TEST\_NOK. | PMP\_Autotest\_CheckPowerSupplyUVDuringActivation () | 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 |
| DSG\_PMP\_0039 | In case an no undervoltage was previously detected and the measured voltage is over the qualification threshold, the Power Undervoltage During Activation Autotest shall return KU8\_ATM\_TEST\_OK. | PMP\_Autotest\_CheckPowerSupplyUVDuringActivation () | 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 |
| DSG\_PMP\_0040 | In case an no undervoltage was previously detected and the measured voltage is under the qualification threshold, the Power Undervoltage Autotest During Activation shall return KU8\_ATM\_TEST\_NOK. | PMP\_Autotest\_CheckPowerSupplyUVDuringActivation () | 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; |
| DSG\_PMP\_0041 | Power Overvoltage Autotest shall be executed every 10 ms. | PMP\_Autotest\_CheckPowerSupplyOVDuringActivation () | ARCH\_SW\_PMP\_1070; ARCH\_SW\_PMP\_1071; ARCH\_SW\_PMP\_1075; ARCH\_SW\_PMP\_1077; ARCH\_SW\_PMP\_1078; ARCH\_SW\_PMP\_1079; ARCH\_SW\_PMP\_1090; |
| DSG\_PMP\_0042 | In case an undervoltage was detected, the Power Overvoltage Autotest shall return KU8\_ATM\_TEST\_NOT\_DECIDED. | PMP\_Autotest\_CheckPowerSupplyOVDuringActivation () | ARCH\_SW\_PMP\_1070; ARCH\_SW\_PMP\_1071; ARCH\_SW\_PMP\_1075; ARCH\_SW\_PMP\_1077; ARCH\_SW\_PMP\_1078; ARCH\_SW\_PMP\_1079; ARCH\_SW\_PMP\_1090; |
| DSG\_PMP\_0043 | In case an overvoltage was previously detected and the measured voltage is below the deskilling threshold, the Power Overvoltage Autotest shall return KU8\_ATM\_TEST\_OK. | PMP\_Autotest\_CheckPowerSupplyOVDuringActivation () | ARCH\_SW\_PMP\_1070; ARCH\_SW\_PMP\_1071; ARCH\_SW\_PMP\_1075; ARCH\_SW\_PMP\_1077; ARCH\_SW\_PMP\_1078; ARCH\_SW\_PMP\_1079; ARCH\_SW\_PMP\_1090; |
| DSG\_PMP\_0044 | In case an overvoltage was previously detected and the measured voltage is over the deskilling threshold, the Power Overvoltage Autotest shall return KU8\_ATM\_TEST\_NOK. | PMP\_Autotest\_CheckPowerSupplyOVDuringActivation () | ARCH\_SW\_PMP\_1070; ARCH\_SW\_PMP\_1071; ARCH\_SW\_PMP\_1075; ARCH\_SW\_PMP\_1077; ARCH\_SW\_PMP\_1078; ARCH\_SW\_PMP\_1079; ARCH\_SW\_PMP\_1090; |
| DSG\_PMP\_0045 | In case an no overvoltage was previously detected and the measured voltage is below the qualification threshold, the Power Overvoltage Autotest shall return KU8\_ATM\_TEST\_OK. | PMP\_Autotest\_CheckPowerSupplyOVDuringActivation () | ARCH\_SW\_PMP\_1070; ARCH\_SW\_PMP\_1071; ARCH\_SW\_PMP\_1075; ARCH\_SW\_PMP\_1077; ARCH\_SW\_PMP\_1078; ARCH\_SW\_PMP\_1079; ARCH\_SW\_PMP\_1090; |
| DSG\_PMP\_0046 | In case an no overvoltage was previously detected and the measured voltage is over the qualification threshold, the Power Overvoltage Autotest shall return KU8\_ATM\_TEST\_NOK. | PMP\_Autotest\_CheckPowerSupplyOVDuringActivation () | ARCH\_SW\_PMP\_1070; ARCH\_SW\_PMP\_1071; ARCH\_SW\_PMP\_1075; ARCH\_SW\_PMP\_1077; ARCH\_SW\_PMP\_1090; ARCH\_SW\_PMP\_1078; ARCH\_SW\_PMP\_1079; |
| DSG\_PMP\_0047 | In case the previous autotest result for ADC is KU8\_ATM\_TEST\_OK, the autotest shall return KU8\_ATM\_TEST\_OK. | PMP\_Autotest\_CheckTemperatureSensor() | ARCH\_SW\_PMP\_1081; ARCH\_SW\_PMP\_1082; |
| DSG\_PMP\_0048 | In case the previous autotest result for ADC is KU8\_ATM\_TEST\_NOK, the autotest shall return KU8\_ATM\_TEST\_NOK. | PMP\_Autotest\_CheckTemperatureSensor() | ARCH\_SW\_PMP\_1081; ARCH\_SW\_PMP\_1082; |
| DSG\_PMP\_0049 | In case an overvoltage was previously detected and the measured voltage is below the deskilling threshold, Autotest shall return KU8\_ATM\_TEST\_OK. | PMP\_Autotest\_CheckSystemOV () | ARCH\_SW\_PMP\_1102; ARCH\_SW\_PMP\_1101; ARCH\_SW\_PMP\_1110; |
| DSG\_PMP\_0050 | In case an overvoltage was previously detected and the measured voltage is over the deskilling threshold, the Autotest shall return KU8\_ATM\_TEST\_NOK. | PMP\_Autotest\_CheckSystemOV () | ARCH\_SW\_PMP\_1102; ARCH\_SW\_PMP\_1101; ARCH\_SW\_PMP\_1110; |
| DSG\_PMP\_0051 | In case an no overvoltage was previously detected and the measured voltage is below the qualification threshold, the Overvoltage Autotest shall return KU8\_ATM\_TEST\_OK. | PMP\_Autotest\_CheckSystemOV () | ARCH\_SW\_PMP\_1102; ARCH\_SW\_PMP\_1101; ARCH\_SW\_PMP\_1110; |
| DSG\_PMP\_0052 | In case an no overvoltage was previously detected and the measured voltage is over the qualification threshold, the Autotest shall return KU8\_ATM\_TEST\_NOK. | PMP\_Autotest\_CheckSystemOV () | ARCH\_SW\_PMP\_1102; ARCH\_SW\_PMP\_1101; ARCH\_SW\_PMP\_1110; |
| DSG\_PMP\_0053 | Power Overvoltage Autotest shall be executed every 100ms. | PMP\_Autotest\_CheckSystemOV () | ARCH\_SW\_PMP\_1110; |
| DSG\_PMP\_0054 | In case an overvoltage was detected, the Power Overvoltage Autotest shall return KU8\_ATM\_TEST\_NOT\_DECIDED. | PMP\_Autotest\_CheckSystemOV() | ARCH\_SW\_PMP\_1102; ARCH\_SW\_PMP\_1101; ARCH\_SW\_PMP\_1110; |
| DSG\_PMP\_0055 | Power Undervoltage Autotest During Activation shall be executed every 100 ms. | PMP\_Autotest\_CheckSystemUV() | ARCH\_SW\_PMP\_1100; ARCH\_SW\_PMP\_1111; ARCH\_SW\_PMP\_1112; |
| DSG\_PMP\_0056 | In case an overvoltage was detected, the Power Undervoltage Autotest shall return KU8\_ATM\_TEST\_NOT\_DECIDED. | PMP\_Autotest\_CheckSystemUV() | ARCH\_SW\_PMP\_1100; ARCH\_SW\_PMP\_1111; ARCH\_SW\_PMP\_1112; |
| DSG\_PMP\_0057 | In case an undervoltage was previously detected and the measured voltage is over the deskilling threshold, the Power Undervoltage During Activation Autotest shall return KU8\_ATM\_TEST\_OK. | PMP\_Autotest\_CheckSystemUV() | ARCH\_SW\_PMP\_1100; ARCH\_SW\_PMP\_1111; ARCH\_SW\_PMP\_1112; |
| DSG\_PMP\_0058 | In case an undervoltage was previously detected and the measured voltage is under the deskilling threshold, the Power Undervoltage Autotest During Activation shall return KU8\_ATM\_TEST\_NOK. | PMP\_Autotest\_CheckSystemUV() | ARCH\_SW\_PMP\_1100; ARCH\_SW\_PMP\_1111; ARCH\_SW\_PMP\_1112; |
| DSG\_PMP\_0059 | In case an no undervoltage was previously detected and the measured voltage is over the qualification threshold, the Power Undervoltage During Activation Autotest shall return KU8\_ATM\_TEST\_OK. | PMP\_Autotest\_CheckSystemUV() | ARCH\_SW\_PMP\_1100; ARCH\_SW\_PMP\_1111; ARCH\_SW\_PMP\_1112; |
| DSG\_PMP\_0060 | In case an no undervoltage was previously detected and the measured voltage is under the qualification threshold, the Power Undervoltage Autotest During Activation shall return KU8\_ATM\_TEST\_NOK. | PMP\_Autotest\_CheckSystemUV() | ARCH\_SW\_PMP\_1100; ARCH\_SW\_PMP\_1111; ARCH\_SW\_PMP\_1112; |
| DSG\_PMP\_0061 | Signal PN14\_SupBat\_Volt\_ST3 shall be converted in V in order to execute correct the autotest | PMP\_Autotest\_CheckSystemUV()  PMP\_Autotest\_CheckSystemOV() | ARCH\_SW\_PMP\_1120; |
| DSG\_PMP\_0062 | The board resistance shall be made available to the rest of the application. | PMP\_runGetBoardResistance() | ARCH\_SW\_PMP\_0382; ARCH\_SW\_PMP\_0381; |
| DSG\_PMP\_0063 | The HighSide switch Voltage shall be available to the rest of application | PMP\_runGetHS1Level() | ARCH\_SW\_PMP\_0203; ARCH\_SW\_PMP\_0204; |
| DSG\_PMP\_0064 | The value of the Negative Motor Voltage shall be available to the rest of application | PMP\_runGetMotorVn\_(Ext)\*() | ARCH\_SW\_PMP\_0021; ARCH\_SW\_PMP\_0022; |
| DSG\_PMP\_0065 | The value of the Positive Motor Voltage shall be available to the rest of application | PMP\_runGetMotorVp\_(Ext)\*() | ARCH\_SW\_PMP\_0023; ARCH\_SW\_PMP\_0024; |
| DSG\_PMP\_0066 | The value of the actual battery state shall be available to the rest of application | PMP\_runGetBatteryVoltageSurveyStatus | ARCH\_SW\_PMP\_0205; ARCH\_SW\_PMP\_0206; |

# FEATURES

## Services

### PMP\_Init

|  |  |  |
| --- | --- | --- |
| Object | | |
| The function is called to initialize all the variables needed by the PMP component. | | |
| **Prototype** | | |
| void PMP\_Init(void) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Return value** | | |
| Type | Description | |
| NA | void | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| \* | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

Diagram

Description automatically generated

Figure : PMP\_Init diagram

### PMP\_runComputeDeficiencyLevel

|  |  |  |
| --- | --- | --- |
| Object | | |
| The function shall compute the estimation of the energy stored in the motor, also called deficiency level. | | |
| **Prototype** | | |
| void PMP\_runComputeDeficiencyLevel(void) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Return value** | | |
| Type | Description | |
| NA | void | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| \* | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

Diagram

Description automatically generated

Figure : PMP\_runComputeDeficiencyLevel diagram

### PMP\_runGetFilteredTemperature

|  |  |  |
| --- | --- | --- |
| Object | | |
| The function is called to provide the FILTERED and CORRECTED value of the ECU temperature in LSB | | |
| **Prototype** | | |
| void PMP\_runGetFilteredTemperature\_deg(s16InternalTemperature\_deg\_Type \* ps16InternalTemperature\_deg) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu16FilteredTemperature | u16InternalTemperatureType | Pointer used to store the requested result |
| **Return value** | | |
| Type | Description | |
| NA | void | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| Called by BFE | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

Diagram

Description automatically generated

Figure : PMP\_runGetFilteredTemperature diagram

### PMP\_runGetFilteredTemperature\_deg

|  |  |  |
| --- | --- | --- |
| Object | | |
| The function is used to provide the FILTERED and CORRECTED value of the ECU temperature in degrees. | | |
| **Prototype** | | |
| void PMP\_runGetFilteredTemperature\_deg(s16InternalTemperature\_deg\_Type \* ps16InternalTemperature\_deg) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| ps16InternalTemperature\_deg | s16InternalTemperature\_deg\_Type | Pointer used to store the requested result |
| **Return value** | | |
| Type | Description | |
| NA | void | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| Called by BFE | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

Diagram

Description automatically generated

Figure : PMP\_runGetFilteredTemperature\_deg diagram

### PMP\_runSurveyBatteryVoltage

|  |  |  |
| --- | --- | --- |
| Object | | |
| The function shall be used to monitor the battery voltage. | | |
| **Prototype** | | |
| void PMP\_runSurveyBatteryVoltage ( void ) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Return value** | | |
| Type | Description | |
| NA | void | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| Called by AdcIf | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

Diagram, schematic

Description automatically generated

Figure : PMP\_runSurveyBatteryVoltage diagram

### PMP\_runGetBatteryVoltage

|  |  |  |
| --- | --- | --- |
| Object | | |
| The function shall provide the Battery Voltage | | |
| Prototype | | |
| void PMP\_runGetBatteryVoltage (u16VoltageType \* pu16Voltage) | | |
| Input parameters | | |
| Name | Type | Description |
| NA | NA | NA |
| Output parameters | | |
| Name | Type | Description |
| pu16Voltage | u16VoltageType | Pointer used to store the requested result |
| Return value | | |
| Type | Description | |
| NA | void | |
| Dynamic aspect | | |
| Who(callers) | Description | |
| Called by BFE | \* | |
| Static aspect | | |
| NA | | |
| Constrains | | |
| NA | | |

Diagram, schematic

Description automatically generated

Figure : PMP\_runGetBatteryVoltage diagram

### PMP\_runGetBatteryVoltage\_2ms

|  |  |  |
| --- | --- | --- |
| Object | | |
| The function shall provide the Battery Voltage filtered for 2ms periodic use | | |
| **Prototype** | | |
| void PMP\_runGetBatteryVoltage\_2ms (u16VoltageType \* pu16Voltage) | | |
| Input parameters | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu16Voltage | u16VoltageType | Pointer used to store the requested result |
| **Return value** | | |
| Type | Description | |
| NA | void | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| Called by BFE | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

Diagram, schematic

Description automatically generated

Figure : PMP\_runGetBatteryVoltage\_2ms diagram

### PMP\_runGetBatteryVoltage\_10ms

|  |  |  |
| --- | --- | --- |
| Object | | |
| The function shall provide the Battery Voltage filtered for 10ms periodic use | | |
| **Prototype** | | |
| void PMP\_runGetBatteryVoltage\_10ms (u16VoltageType \* pu16Voltage) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu16Voltage | u16VoltageType | Pointer used to store the requested result |
| **Return value** | | |
| Type | Description | |
| NA | void | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| Called by BFE | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

Diagram, schematic

Description automatically generated

Figure : PMP\_runGetBatteryVoltage\_10ms diagram

### PMP\_runGetDeficiencyLevel

|  |  |  |
| --- | --- | --- |
| Object | | |
| The function shall provide the Deficiency Level | | |
| **Prototype** | | |
| void PMP\_runGetDeficiencyLevel(u32DeficiencyLevelType \* pu32DeficiencyLevel) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu32DeficiencyLevel | u32DeficiencyLevelType | Pointer used to store the requested result |
| **Return value** | | |
| Type | Description | |
| NA | void | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| Called by PAL | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

Diagram

Description automatically generated

Figure : PMP\_runGetDeficiencyLevel diagram

### PMP\_runGetMotorResistance

|  |  |  |
| --- | --- | --- |
| Object | | |
| This function returns an estimated motor resistance from the ECU measured temperature, using linear interpolation. | | |
| **Prototype** | | |
| void PMP\_runGetMotorResistance(u16InternalTemperatureType u16ECUInternalTemperature, u16Resistance\_mOhmType \* pu16Resistance\_mOhm) | | |
| **Input parameters** | | |
| Name | Type | Description |
| u16ECUInternalTemperature | u16InternalTemperatureType | Temperature to be used in computation |
| **Output parameters** | | |
| Name | Type | Description |
| u16ResistanceinOhmType | pu16ResistanceinOhm | Pointer used to store the requested result |
| **Return value** | | |
| Type | Description | |
| NA | void | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| Called by BFE | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

A picture containing chart

Description automatically generated

Figure 10: PMP\_runGetMotorResistance diagram

### pmp\_GetCorrectedTemperature

|  |  |  |
| --- | --- | --- |
| Object | | |
| The function is called to provide the CORRECTED Temperature of the ECU. | | |
| **Prototype** | | |
| **void** **pmp\_GetCorrectedTemperature**(u16InternalTemperatureType \* **const** pu16Temperature) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu16Temperature | u16InternalTemperatureType | Constant pointer used to store the requested result |
| **Return value** | | |
| Type | Description | |
| NA | void | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| Called by PMP\_runGetFilteredTemperature() | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

Diagram

Description automatically generated

Figure pmp\_GetCorrectedTemperature

### pmp\_runGetBatteryVoltage

|  |  |  |
| --- | --- | --- |
| Object | | |
| The function shall provide the Battery Voltage filtered for 100ms periodic use | | |
| **Prototype** | | |
| void pmp\_runGetBatteryVoltage (u16VoltageType \* pu16Voltage) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu16Voltage | u16VoltageType | Pointer used to store the requested result |
| **Return value** | | |
| Type | Description | |
| NA | void | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| \* | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

Diagram, schematic

Description automatically generated

Figure PMP\_runGetBatteryVoltage

### pmp\_runGetBatteryVoltage\_20ms

|  |  |  |
| --- | --- | --- |
| Object | | |
| The function shall provide the Battery Voltage filtered for 20ms periodic use | | |
| **Prototype** | | |
| void pmp\_runGetBatteryVoltage (u16VoltageType \* pu16Voltage) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu16Voltage | u16VoltageType | Pointer used to store the requested result |
| **Return value** | | |
| Type | Description | |
| NA | void | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| \* | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

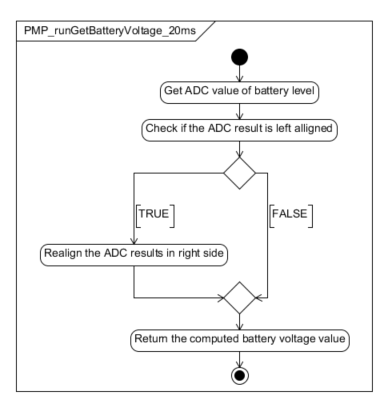


Figure PMP\_runGetBatteryVoltage\_20ms

### pmp\_runGetBatteryVoltage\_100ms

|  |  |  |
| --- | --- | --- |
| Object | | |
| The function shall provide the Battery Voltage filtered for 100ms periodic use | | |
| **Prototype** | | |
| void pmp\_runGetBatteryVoltage\_100 (u16VoltageType \* pu16Voltage) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu16Voltage | u16VoltageType | Pointer used to store the requested result |
| **Return value** | | |
| Type | Description | |
| NA | void | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| \* | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

Diagram, schematic

Description automatically generated

Figure PMP\_runGetBatteryVoltage\_100ms

### pmp\_runGetBatteryVoltage\_10ms

|  |  |  |
| --- | --- | --- |
| Object | | |
| The function shall provide the Battery Voltage filtered for 10ms periodic use | | |
| **Prototype** | | |
| void pmp\_runGetBatteryVoltage (u16VoltageType \* pu16Voltage) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu16Voltage | u16VoltageType | Pointer used to store the requested result |
| **Return value** | | |
| Type | Description | |
| NA | void | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| \* | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

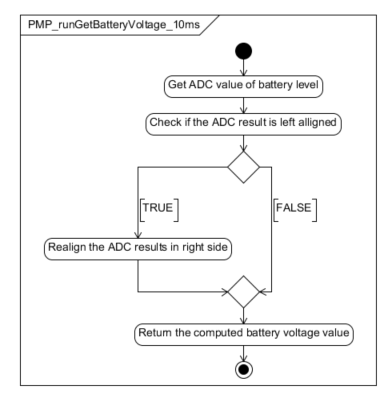


Figure PMP\_runGetBatteryVoltage\_10ms

### pmp\_runGetBatteryVoltage\_2ms

|  |  |  |
| --- | --- | --- |
| Object | | |
| The function shall provide the Battery Voltage filtered for 2ms periodic use | | |
| **Prototype** | | |
| void pmp\_runGetBatteryVoltage (u16VoltageType \* pu16Voltage) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu16Voltage | u16VoltageType | Pointer used to store the requested result |
| **Return value** | | |
| Type | Description | |
| NA | void | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| \* | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

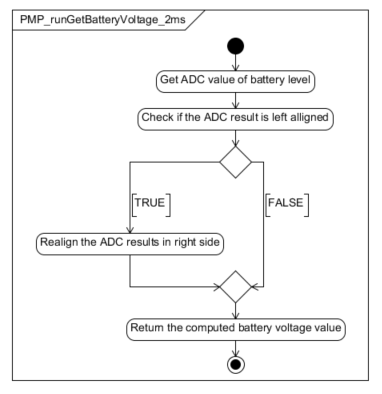


Figure PMP\_runGetBatteryVoltage\_2ms

### PMP\_runGetBoardResistance

|  |  |  |
| --- | --- | --- |
| Object | | |
| The function is called to provide the ECU Resistance. | | |
| **Prototype** | | |
| EXPORTED void PMP\_runGetBoardResistance (u16InternalTemperatureType u16ECUInternalTemperature, u16Resistance\_mOhmType \*pu16Resistance\_mOhm) | | |
| **Input parameters** | | |
| Name | Type | Description |
| u16ECUInternalTemperature | u16InternalTemperatureType | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu16Resistance\_mOhm | u16Resistance\_mOhmType\* | NA |
| **Return value** | | |
| Type | Description | |
| NA | - | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| \* | \* | |
| **Static aspect** | | |
|  | | |
| **Constrains** | | |
|  | | |

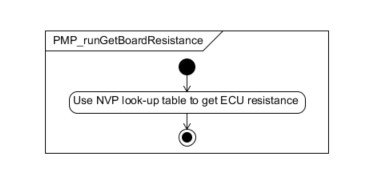


Figure PMP\_runGetBoardResistance

### PMP\_runGetMotorVn

|  |  |  |
| --- | --- | --- |
| Object | | |
| The function shall be used to get the V- voltage value from Half bridge power stage. | | |
| **Prototype** | | |
| void PMP\_runGetMotorVn (u16VoltageType \*pu16Voltage) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu16Voltage | u16VoltageType\* | NA |
| **Return value** | | |
| Type | Description | |
| NA | - | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| \* | \* | |
| **Static aspect** | | |
|  | | |
| **Constrains** | | |
|  | | |
|  |  |  |

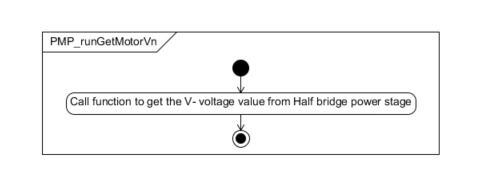


Figure PMP\_runGetMotorVn

### PMP\_runGetMotorVn\_20ms

|  |  |  |
| --- | --- | --- |
| Object | | |
| The function shall be used to get the V- voltage value from Half bridge power stage filtered for 20ms periodic use. | | |
| **Prototype** | | |
| void PMP\_runGetMotorVn\_20ms (u16VoltageType \*pu16Voltage) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu16Voltage | u16VoltageType\* | NA |
| **Return value** | | |
| Type | Description | |
| NA | - | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| \* | \* | |
| **Static aspect** | | |
|  | | |
| **Constrains** | | |
|  | | |
|  |  |  |

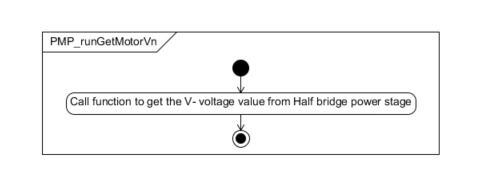


Figure PMP\_runGetMotorVn\_20ms

### PMP\_runGetMotorVn\_100ms

|  |  |  |
| --- | --- | --- |
| Object | | |
| The function shall be used to get the V- voltage value from Half bridge power stage filtered for 100ms periodic use. | | |
| **Prototype** | | |
| void PMP\_runGetMotorVn\_100ms (u16VoltageType \*pu16Voltage) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu16Voltage | u16VoltageType\* | NA |
| **Return value** | | |
| Type | Description | |
| NA | - | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| \* | \* | |
| **Static aspect** | | |
|  | | |
| **Constrains** | | |
|  | | |
|  |  |  |

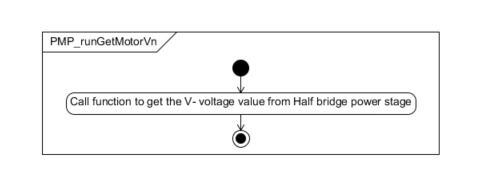


Figure PMP\_runGetMotorVn\_100ms

### PMP\_runGetMotorVp

|  |  |  |
| --- | --- | --- |
| Object | | |
| The function shall be used to get the V+ voltage value from Half bridge power stage. | | |
| **Prototype** | | |
| EXPORTED void PMP\_runGetMotorVp (u16VoltageType \*pu16Voltage) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu16Voltage | u16VoltageType\* | NA |
| **Return value** | | |
| Type | Description | |
| NA | - | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| \* | \* | |
| **Static aspect** | | |
|  | | |
| **Constrains** | | |
|  | | |
|  |  |  |

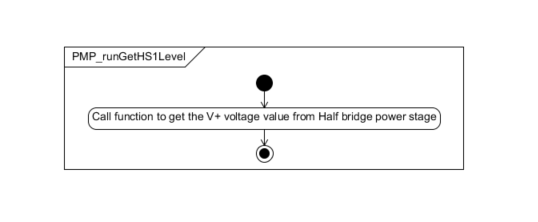


Figure PMP\_runGetMotorVp

### PMP\_runGetMotorVp\_20ms

|  |  |  |
| --- | --- | --- |
| Object | | |
| The function shall be used to get the V+ voltage value from Half bridge power stage filtered for 20ms periodic use. | | |
| **Prototype** | | |
| EXPORTED void PMP\_runGetMotorVp\_20ms (u16VoltageType \*pu16Voltage) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu16Voltage | u16VoltageType\* | NA |
| **Return value** | | |
| Type | Description | |
| NA | - | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| \* | \* | |
| **Static aspect** | | |
|  | | |
| **Constrains** | | |
|  | | |
|  |  |  |

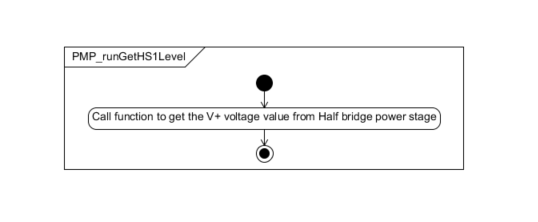


Figure PMP\_runGetMotorVp\_20ms

### PMP\_runGetMotorVp\_100ms

|  |  |  |
| --- | --- | --- |
| Object | | |
| The function shall be used to get the V+ voltage value from Half bridge power stage filtered for 100ms periodic use. | | |
| **Prototype** | | |
| EXPORTED void PMP\_runGetMotorVp\_100ms (u16VoltageType \*pu16Voltage) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu16Voltage | u16VoltageType\* | NA |
| **Return value** | | |
| Type | Description | |
| NA | - | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| \* | \* | |
| **Static aspect** | | |
|  | | |
| **Constrains** | | |
|  | | |
|  |  |  |

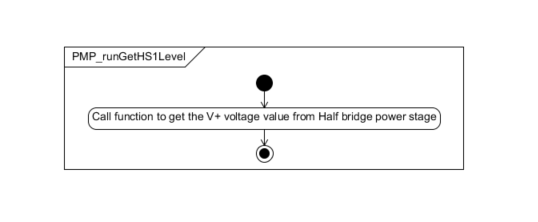


Figure PMP\_runGetMotorVp\_100ms

### PMP\_runGetHS1Level

|  |  |  |
| --- | --- | --- |
| Object | | |
| The function shall be used to compute the value of the HighSide Switch Voltage level. | | |
| **Prototype** | | |
| EXPORTED void PMP\_runGetHS1Level (u16VoltageType \*pu16Voltage) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu16Voltage | u16VoltageType\* | NA |
| **Return value** | | |
| Type | Description | |
| NA | - | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| \* | \* | |
| **Static aspect** | | |
|  | | |
| **Constrains** | | |
|  | | |
|  |  |  |

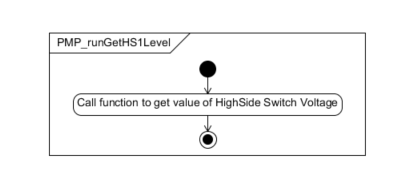


Figure PMP\_runGetHS1Level

### PMP\_runGetBatteryVoltageSurveyStatus

|  |  |  |
| --- | --- | --- |
| Object | | |
| The function shall be used to return the value of the battery voltage survey status. | | |
| **Prototype** | | |
| void PMP\_runGetBatteryVoltageSurveyStatus (u8BatteryVoltageSurveyStatusType \*pu8BatteryVoltageSurveyStatus) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu8BatteryVoltageSurveyStatus | u8BatteryVoltageSurveyStatusType\* | NA |
| **Return value** | | |
| Type | Description | |
| NA | - | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| \* | \* | |
| **Static aspect** | | |
|  | | |
| **Constrains** | | |
|  | | |
|  |  |  |

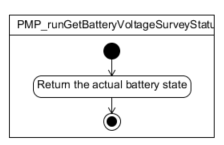


Figure PMP\_runGetBatteryVoltageSurveyStatus

## Auto-Tests

### PMP\_Autotest\_CheckTemperatureSensor

|  |  |  |
| --- | --- | --- |
| Object | | |
| This auto-test aims at detecting an SBC temperature error. The temperature provided by the SBC module is compared to a low and high threshold. If the temperature value is out of range, this auto-test returns "failed" result, otherwise it returns "passed" result. | | |
| **Prototype** | | |
| void PMP\_Autotest\_CheckTemperatureSensor (u8TestResultType \*pu8TestResult) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu8TestResult | u8TestResultType\* | NA |
| **Return value** | | |
| Type | Description | |
| NA | - | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| \* | \* | |
| **Static aspect** | | |
|  | | |
| **Constrains** | | |
|  | | |
|  |  |  |
|  |  |  |

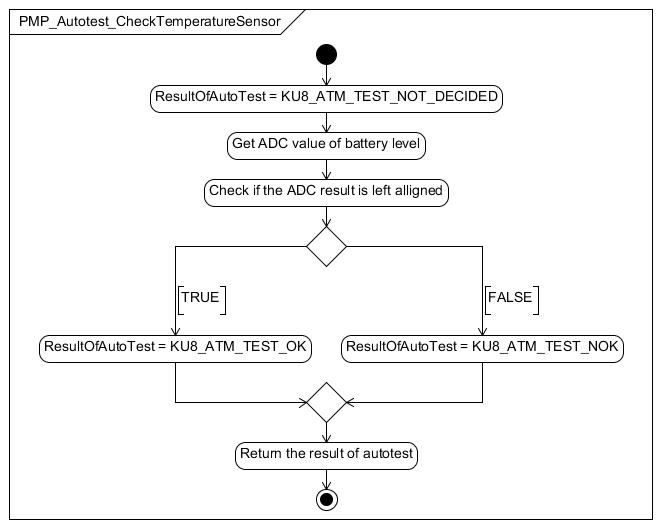


Figure PMP\_Autotest\_CheckTemperatureSensor

### PMP\_Autotest\_CheckPowerSupplyOV

|  |  |  |
| --- | --- | --- |
| Object | | |
| This is the test function for overvoltage on the Power stage battery voltage. | | |
| **Prototype** | | |
| void PMP\_Autotest\_CheckPowerSupplyOV(u8TestResultType \* pu8TestResult) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu8TestResult | u8TestResultType | Pointer used to store the auto test result |
| **Return value** | | |
| Type | Description | |
| NA | void | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| Called by ATM | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

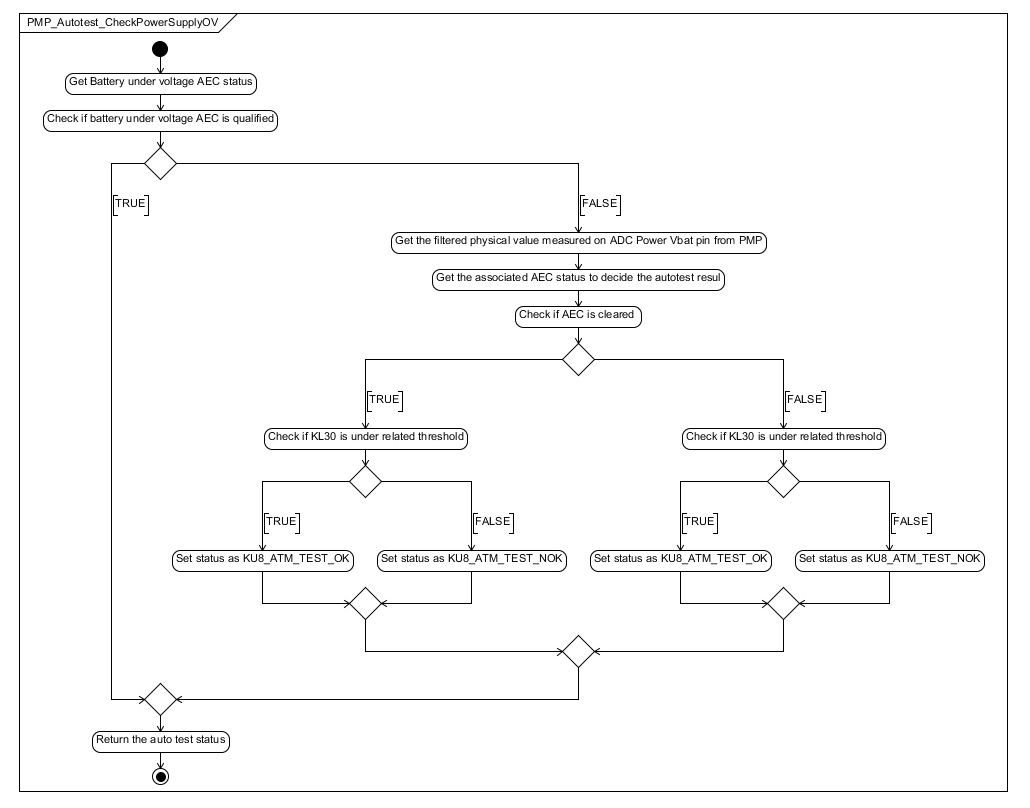


Figure PMP\_Autotest\_CheckPowerSupplyOV

### PMP\_Autotest\_CheckPowerSupplyUV

|  |  |  |
| --- | --- | --- |
| Object | | |
| The function shall check if an undervoltage is detected on the power stage of the battery voltage. | | |
| **Prototype** | | |
| void PMP\_Autotest\_CheckPowerSupplyUV (u8TestResultType \*pu8TestResult) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu8TestResult | u8TestResultType\* | NA |
| **Return value** | | |
| Type | Description | |
| NA | - | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| \* | \* | |
| **Static aspect** | | |
|  | | |
| **Constrains** | | |
|  | | |

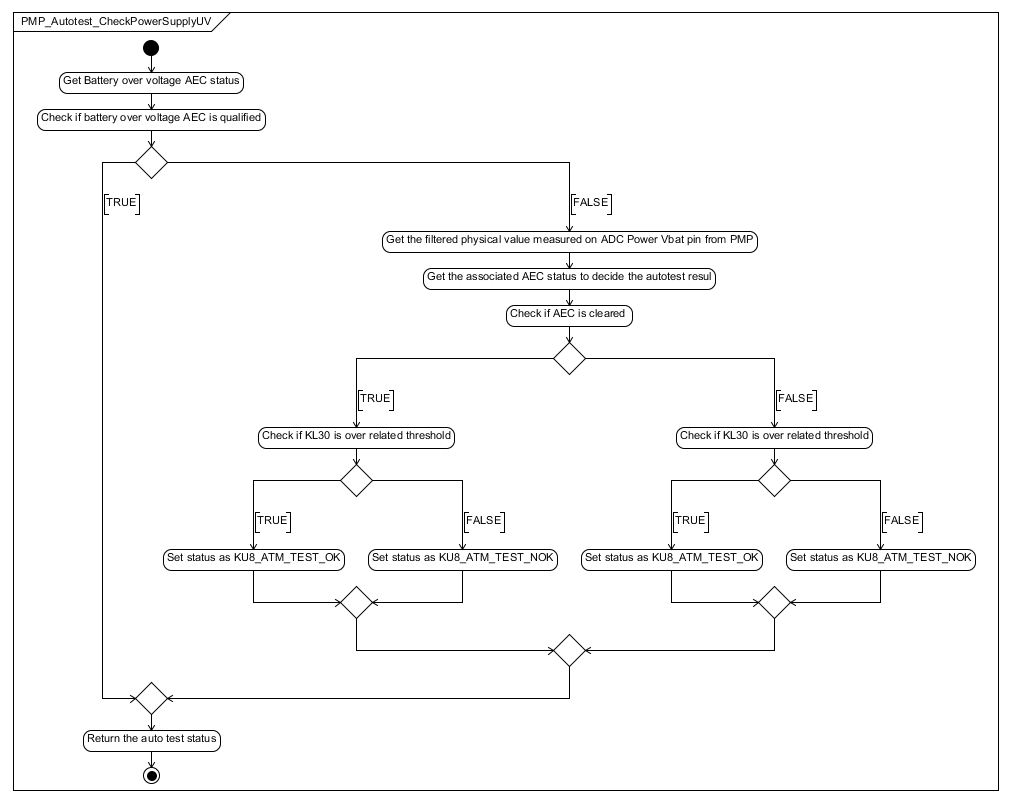


Figure PMP\_Autotest\_CheckPowerSupplyUV

### PMP\_Autotest\_CheckPowerSupplyUVDuringActivation

|  |  |  |
| --- | --- | --- |
| Object | | |
| The function shall check if an undervoltage is detected on the power stage of the battery voltage during a tensioning profile execution. | | |
| **Prototype** | | |
| void PMP\_Autotest\_CheckPowerSupplyUVDuringActivation (u8TestResultType \*pu8TestResult) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu8TestResult | u8TestResultType\* | NA |
| **Return value** | | |
| Type | Description | |
| NA | - | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| \* | \* | |
| **Static aspect** | | |
|  | | |
| **Constrains** | | |
|  | | |

A diagram of a flowchart

Description automatically generated

Figure PMP\_Autotest\_CheckPowerSupplyUVDuringActivation

### PMP\_Autotest\_CheckPowerSupplyOVDuringActivation

|  |  |  |
| --- | --- | --- |
| Object | | |
| This is the test function for overvoltage on the Power stage battery voltage during tensioning | | |
| **Prototype** | | |
| **void** **PMP\_Autotest\_CheckPowerSupplyOVDuringActivation**(u8TestResultType \* pu8TestResult) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu8TestResult | u8TestResultType | Pointer used to store the auto test result |
| **Return value** | | |
| Type | Description | |
| NA | void | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| Called by ATM | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

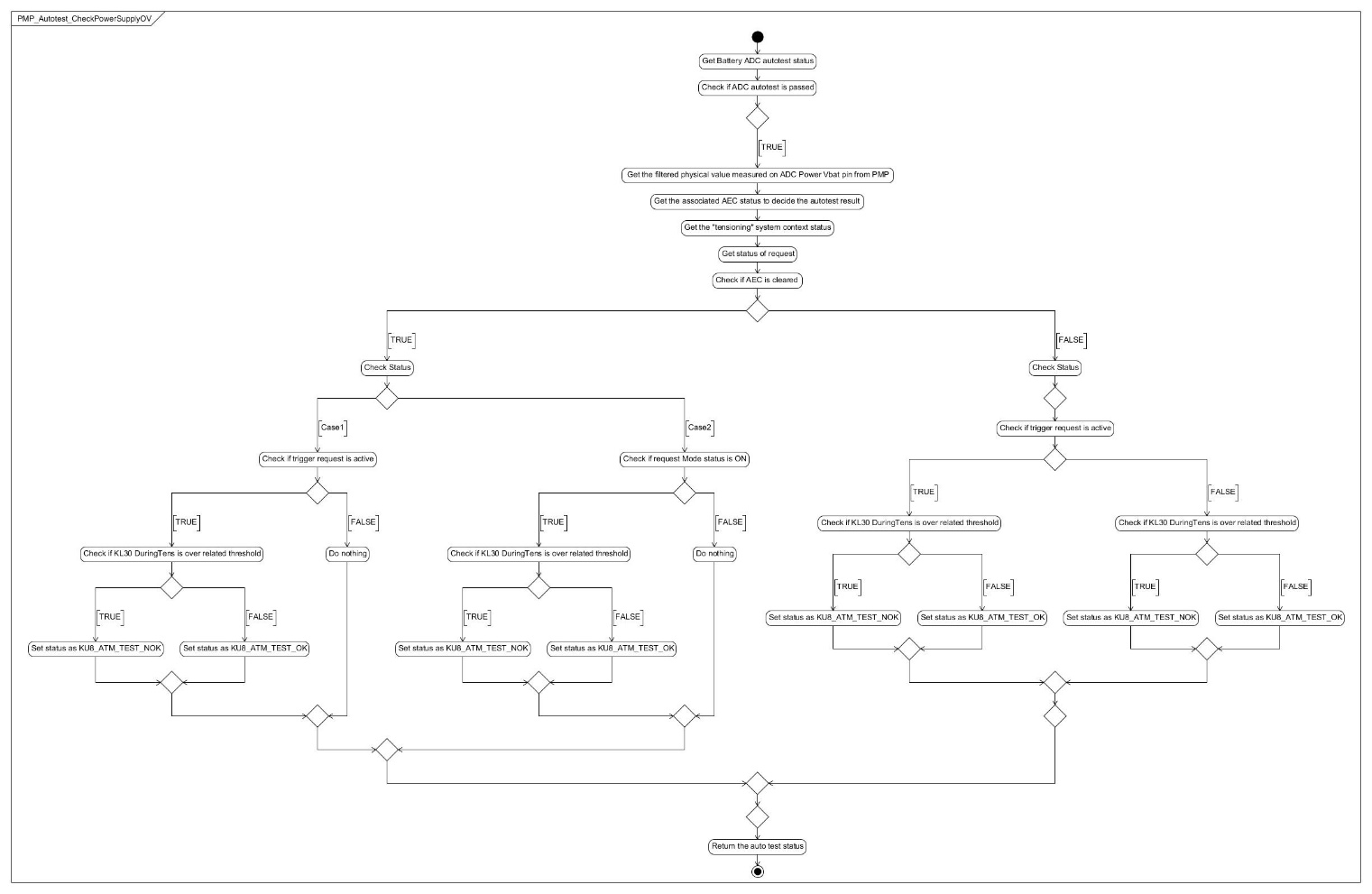


Figure PMP\_Autotest\_CheckPowerSupplyOVDuringActivation

### PMP\_Autotest\_CheckSystemUV

|  |  |  |
| --- | --- | --- |
| Object | | |
| This is the test function for undervoltage in the system. | | |
| **Prototype** | | |
| void PMP\_Autotest\_CheckSystemUV(u8TestResultType \* pu8TestResult) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu8TestResult | u8TestResultType | Pointer used to store the auto test result |
| **Return value** | | |
| Type | Description | |
| NA | void | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| Called by ATM | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

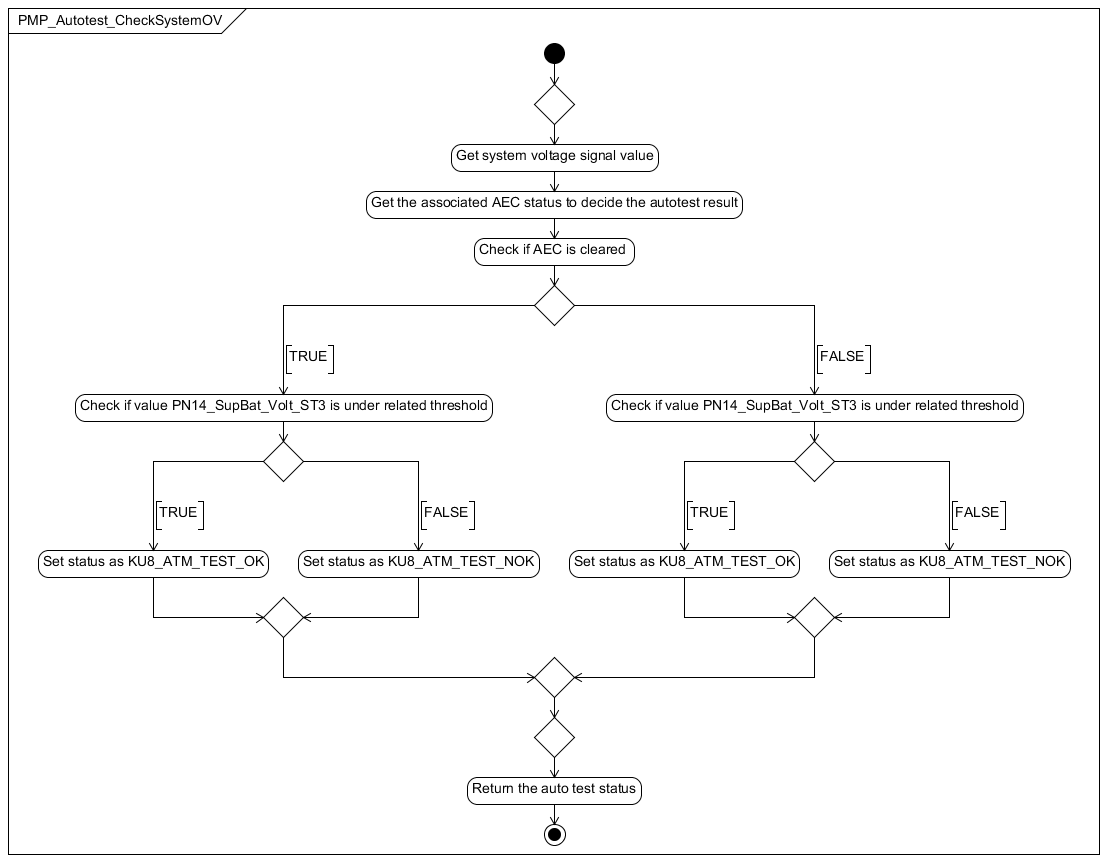


Figure PMP\_Autotest\_CheckSystemUV

### PMP\_Autotest\_CheckSystemOV

|  |  |  |
| --- | --- | --- |
| Object | | |
| This is the test function for overvoltage in the system. | | |
| **Prototype** | | |
| void PMP\_Autotest\_CheckSystemOV(u8TestResultType \* pu8TestResult) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu8TestResult | u8TestResultType | Pointer used to store the auto test result |
| **Return value** | | |
| Type | Description | |
| NA | void | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| Called by ATM | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

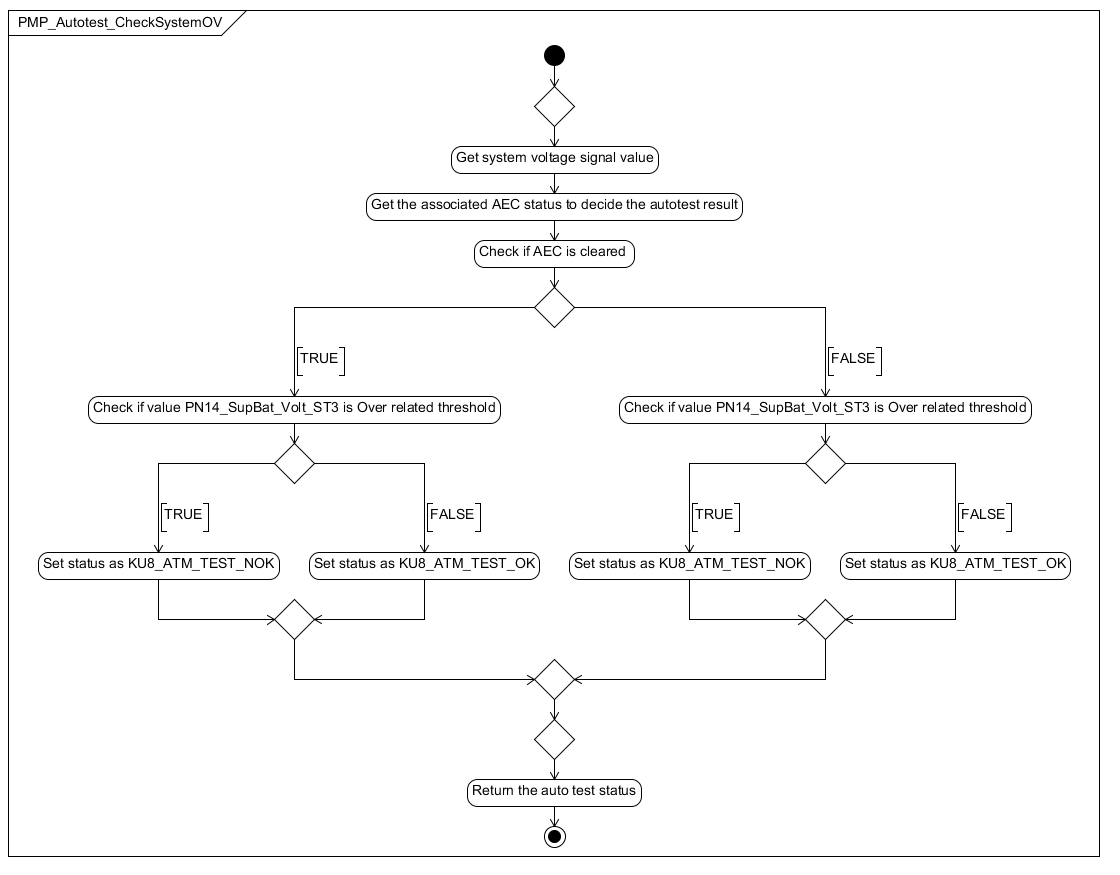


Figure PMP\_Autotest\_CheckSystemOV

## Types

### PMP\_stBatterySurveyParamType

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Field description** |
| u16OldRawBatteryVoltage | uint16 | Previous battery voltage |
| u16MinBatteryVoltageADCFormat | uint16 | Minimum threshold for battery voltage |
| u16MaxBatteryVoltageADCFormat | uint16 | Maximum threshold for battery voltage |
| u16MinDiffVoltageToValidateBatterySurvey | uint16 | Voltage difference used in validation |
| u8CounterToValidateBatterySurvey | uint8 | Internal counter |
| u8BatteryVoltageSurveyState | u8BatteryVoltageSurveyStatusType | Battery Survey State |

## Variables

### PMP\_stBatterySurveyParam

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| PMP\_stBatterySurveyParamType | N.A. | |
| **Description** | | |
| Variable represents the structure containing all variables used by the battery survey mechanism | | |
| **Definition** | | |
| static PMP\_stBatterySurveyParamType PMP\_stBatterySurveyParam | | |

### PMP\_u16OldADCTemperature\_LSB

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| u16InternalTemperatureType | N.A. | |
| **Description** | | |
| Variable is used to store old temperature read from ADC | | |
| **Definition** | | |
| static u16InternalTemperatureType PMP\_u16OldADCTemperature\_LSB | | |

### PMP\_u16OldCorrectedTemperature\_LSB

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| u16InternalTemperatureType | N.A. | |
| **Description** | | |
| Variable is used to store old corrected temperature | | |
| **Definition** | | |
| static u16InternalTemperatureType PMP\_u16OldCorrectedTemperature\_LSB | | |

## Macros

### PMP\_KU8\_MOTOR\_RESISTANCE\_LINEAR\_INTERPOLATION\_NB\_POINTS

|  |  |
| --- | --- |
| Name | Value |
| Motor resistance linear interpolation points number | 7 |
| **Definition** | |
| #define PMP\_KU8\_MOTOR\_RESISTANCE\_LINEAR\_INTERPOLATION\_NB\_POINTS | |
| **Description** | |
| Number of points used in motor resistance linear interpolation | |

### PMP\_RIGHT\_ALIGNED\_ADC

|  |  |
| --- | --- |
| Name | Value |
| ADC alignment to the right status | 0 |
| **Definition** | |
| #define PMP\_RIGHT\_ALIGNED\_ADC | |
| **Description** | |
| Defines if the ADC result is right aligned | |

### PMP\_LEFT\_ALIGNED\_ADC

|  |  |
| --- | --- |
| Name | Value |
| ADC alignment to the left status | 1 |
| **Definition** | |
| #define PMP\_LEFT\_ALIGNED\_ADC | |
| **Description** | |
| Defines if the ADC result is left aligned | |

### PMP\_ALIGNMENT\_ADC\_CHANNELS

|  |  |
| --- | --- |
| Name | Value |
| ADC Channels alignment | PMP\_RIGHT\_ALIGNED\_ADC |
| **Definition** | |
| #define PMP\_ALIGNMENT\_ADC\_CHANNELS | |
| **Description** | |
| ADC Channels alignment in the project | |

### PMP\_KU16\_ADC\_10BIT\_RANGE

|  |  |
| --- | --- |
| Name | Value |
| Range of 10-bit ADC | 1024 |
| **Definition** | |
| #define PMP\_KU16\_ADC\_10BIT\_RANGE | |
| **Description** | |
| Represents the maximum value a 10-bit ADC can have | |

### PMP\_KU16\_ADC\_8BIT\_RANGE

|  |  |
| --- | --- |
| Name | Value |
| Range of 8-bit ADC | 256 |
| **Definition** | |
| #define PMP\_KU16\_ADC\_8BIT\_RANGE | |
| **Description** | |
| Represents the maximum value a 8-bit ADC can have | |

### PMP\_KU16\_ADC\_FULL\_RANGE

|  |  |
| --- | --- |
| Name | Value |
| ADC full range | PMP\_KU16\_ADC\_10BIT\_RANGE |
| **Definition** | |
| #define PMP\_KU16\_ADC\_FULL\_RANGE | |
| **Description** | |
| Represents the full range of the ADC used in the project. | |

### PMP\_KU16\_ADC\_REFERENCE\_MV

|  |  |
| --- | --- |
| Name | Value |
| ADC reference in mV | 5000 |
| **Definition** | |
| #define PMP\_KU16\_ADC\_REFERENCE\_MV | |
| **Description** | |
| Represents the ADC voltage reference in mV | |

### PMP\_KU32\_HVI\_INTERNAL\_FACTOR

|  |  |
| --- | --- |
| Name | Value |
| HVI Internal Factor | 66 |
| **Definition** | |
| #define PMP\_KU32\_HVI\_INTERNAL\_FACTOR | |
| **Description** | |
| High Voltage Input factor internally applied by the MCU on high voltage analog inputs (mapped to KL30/KL30BG). | |

### PMP\_KU32\_KL30\_RAW\_MUL\_FACTOR

|  |  |
| --- | --- |
| Name | Value |
| KL30 Raw Multiplication Factor | PMP\_KU16\_ADC\_FULL\_RANGE |
| **Definition** | |
| #define PMP\_KU32\_KL30\_RAW\_MUL\_FACTOR | |
| **Description** | |
| KL30 & KL30bg conversion factors to be applied on voltage threshold parameter to convert from mV into 10 bits ADC raw format. | |

### PMP\_KU32\_KL30\_RAW\_DIV\_FACTOR

|  |  |
| --- | --- |
| Name | Value |
| KL30 Raw Division Factor | (PMP\_KU32\_HVI\_INTERNAL\_FACTOR \* PMP\_KU16\_ADC\_REFERENCE\_MV) / KU8\_TEN) |
| **Definition** | |
| #define PMP\_KU32\_KL30\_RAW\_DIV\_FACTOR | |
| **Description** | |
| KL30 & KL30bg conversion factors to be applied on voltage threshold parameter to convert from mV into 10 bits ADC raw format. | |

### PMP\_KU32\_KL30\_MUL\_FACTOR

|  |  |
| --- | --- |
| Name | Value |
| KL30 Multiplication Factor | (PMP\_KU32\_HVI\_INTERNAL\_FACTOR \* PMP\_KU16\_ADC\_REFERENCE\_MV) / KU8\_TEN |
| **Definition** | |
| #define PMP\_KU32\_KL30\_MUL\_FACTOR | |
| **Description** | |
| KL30 & KL30bg conversion factors to be applied on voltage threshold parameter to convert from mV into 10 bits ADC raw format. | |

### PMP\_KU32\_KL30\_DIV\_FACTOR

|  |  |
| --- | --- |
| Name | Value |
| KL30 Division Factor | PMP\_KU16\_ADC\_FULL\_RANGE |
| **Definition** | |
| #define PMP\_KU32\_KL30\_DIV\_FACTOR | |
| **Description** | |
| KL30 & KL30bg conversion factors to be applied on voltage threshold parameter to convert from mV into 10 bits ADC raw format. | |

### PMP\_KU32\_VN\_MUL\_FACTOR

|  |  |
| --- | --- |
| Name | Value |
| VN Multiplication Factor | 112 \* PMP\_KU16\_ADC\_REFERENCE\_MV |
| **Definition** | |
| #define PMP\_KU32\_VN\_MUL\_FACTOR | |
| **Description** | |
| Motor V Minus conversion factor | |

### PMP\_KU32\_VN\_DIV\_FACTOR

|  |  |
| --- | --- |
| Name | Value |
| VN Division Factor | 12 \* PMP\_KU16\_ADC\_FULL\_RANGE |
| **Definition** | |
| #define PMP\_KU32\_VN\_DIV\_FACTOR | |
| **Description** | |
| Motor V Minus conversion factor | |

### PMP\_KU32\_VP\_MUL\_FACTOR

|  |  |
| --- | --- |
| Name | Value |
| VP Multiplication Factor | 112 \* PMP\_KU16\_ADC\_REFERENCE\_MV |
| **Definition** | |
| #define PMP\_KU32\_VP\_MUL\_FACTOR | |
| **Description** | |
| Motor V Plus conversion factor | |

### PMP\_KU32\_VP\_DIV\_FACTOR

|  |  |
| --- | --- |
| Name | Value |
| VP Division Factor | 12 \* PMP\_KU16\_ADC\_FULL\_RANGE |
| **Definition** | |
| #define PMP\_KU32\_VP\_DIV\_FACTOR | |
| **Description** | |
| Motor V Plus conversion factor | |

### PMP\_KU32\_HS\_MUL\_FACTOR

|  |  |
| --- | --- |
| Name | Value |
| HS Multiplication Factor | 112 \* PMP\_KU16\_ADC\_REFERENCE\_MV |
| **Definition** | |
| #define PMP\_KU32\_HS\_MUL\_FACTOR | |
| **Description** | |
| HS1 conversion factor | |

### PMP\_KU32\_HS\_DIV\_FACTOR

|  |  |
| --- | --- |
| Name | Value |
| HS Division Factor | 12 \* PMP\_KU16\_ADC\_FULL\_RANGE |
| **Definition** | |
| #define PMP\_KU32\_HS\_DIV\_FACTOR | |
| **Description** | |
| HS1 conversion factor | |

### PMP\_KU32\_REFLOW\_MUL\_FACTOR

|  |  |
| --- | --- |
| Name | Value |
| REFLOW Multiplication Factor | PMP\_KU16\_ADC\_REFERENCE\_MV |
| **Definition** | |
| #define PMP\_KU32\_REFLOW\_MUL\_FACTOR | |
| **Description** | |
| Low reference conversion factor | |

### PMP\_KU32\_REFLOW\_DIV\_FACTOR

|  |  |
| --- | --- |
| Name | Value |
| REFLOW Division Factor | PMP\_KU16\_ADC\_FULL\_RANGE |
| **Definition** | |
| #define PMP\_KU32\_VP\_DIV\_FACTOR | |
| **Description** | |
| Low reference conversion factor | |

### PMP\_KU32\_REFMID\_MUL\_FACTOR

|  |  |
| --- | --- |
| Name | Value |
| REFMID Multiplication Factor | PMP\_KU16\_ADC\_REFERENCE\_MV |
| **Definition** | |
| #define PMP\_KU32\_REFMID\_MUL\_FACTOR | |
| **Description** | |
| Mid reference conversion factor | |

### PMP\_KU32\_REFMID\_DIV\_FACTOR

|  |  |
| --- | --- |
| Name | Value |
| REFMID Division Factor | PMP\_KU16\_ADC\_FULL\_RANGE |
| **Definition** | |
| #define PMP\_KU32\_REFMID\_DIV\_FACTOR | |
| **Description** | |
| Mid reference conversion factor | |

### PMP\_KU32\_REFHIGH\_MUL\_FACTOR

|  |  |
| --- | --- |
| Name | Value |
| REFHIGH Multiplication Factor | PMP\_KU16\_ADC\_REFERENCE\_MV |
| **Definition** | |
| #define PMP\_KU32\_REFHIGH\_MUL\_FACTOR | |
| **Description** | |
| High reference conversion factor | |

### PMP\_KU32\_REFHIGH\_DIV\_FACTOR

|  |  |
| --- | --- |
| Name | Value |
| REFHIGH Division Factor | PMP\_KU16\_ADC\_FULL\_RANGE |
| **Definition** | |
| #define PMP\_KU32\_REFHIGH\_DIV\_FACTOR | |
| **Description** | |
| High reference conversion factor | |

### PMP\_KU32\_REFCPUREG\_MUL\_FACTOR

|  |  |
| --- | --- |
| Name | Value |
| REFCPUREG Multiplication Factor | PMP\_KU16\_ADC\_REFERENCE\_MV |
| **Definition** | |
| #define PMP\_KU32\_REFCPUREG\_MUL\_FACTOR | |
| **Description** | |
| CPU Regulator Voltage conversion factor | |

### PMP\_KU32\_REFCPUREG\_DIV\_FACTOR

|  |  |
| --- | --- |
| Name | Value |
| REFCPUREG Division Factor | PMP\_KU16\_ADC\_FULL\_RANGE |
| **Definition** | |
| #define PMP\_KU32\_REFCPUREG\_DIV\_FACTOR | |
| **Description** | |
| CPU Regulator Voltage conversion factor | |

### PMP\_KU16\_ADC\_RANGE\_MV

|  |  |
| --- | --- |
| Name | Value |
| ADC Range (mV) | 5000 |
| **Definition** | |
| #define KU16\_ADC\_RANGE\_MV | |
| **Description** | |
| ADC input full scale in mV (ADC reference voltage) | |

### PMP\_KU16\_ADC\_OUT\_SCALE

|  |  |
| --- | --- |
| Name | Value |
| ADC Out Scale | 1024 |
| **Definition** | |
| #define KU16\_ADC\_OUT\_SCALE | |
| **Description** | |
| ADC full scale output value | |

### PMP\_KU8\_NB\_INTERP\_PTS\_TEMP\_CALIB

|  |  |
| --- | --- |
| Name | Value |
| Number of points in temperature calibration. | 7 |
| **Definition** | |
| #define PMP\_KU8\_NB\_INTERP\_PTS\_TEMP\_CALIB | |
| **Description** | |
| Number of points in temperature calibration. | |

### PMP\_KU8\_DEG\_RANGE

|  |  |
| --- | --- |
| Name | Value |
| DEG Range | 135 |
| **Definition** | |
| #define PMP\_KU8\_DEG\_RANGE | |
| **Description** | |
| DEG range | |

### PMP\_KU8\_DEG\_LOW\_RANGE

|  |  |
| --- | --- |
| Name | Value |
| DEG low range | 40 |
| **Definition** | |
| #define PMP\_KU8\_DEG\_LOW\_RANGE | |
| **Description** | |
| DEG low range | |

### PMP\_KU32\_TYP\_H\_SLOPE\_UV\_PER\_DEG

|  |  |
| --- | --- |
| Name | Value |
| TYP H Slope UV per DEG | (PMP\_KU32\_TYP\_SLOPE\_UV\_PER\_DEG+1u)/2u |
| **Definition** | |
| #define PMP\_KU32\_TYP\_H\_SLOPE\_UV\_PER\_DEG | |
| **Description** | |
| Half sensor's typical temperature slop (used to round division by PMP\_KU16\_ADC\_OUT\_SCALE) | |

### PMP\_KU16\_ADC\_OUT\_H\_SCALE

|  |  |
| --- | --- |
| Name | Value |
| ADC Out H Scale | (PMP\_KU16\_ADC\_OUT\_SCALE+1u)/2u |
| **Definition** | |
| #define PMP\_KU16\_ADC\_OUT\_H\_SCALE | |
| **Description** | |
| ADC half scale output value (used to round division by PMP\_KU16\_ADC\_OUT\_SCALE) | |

### PMP\_KU32\_ADC\_RANGE\_UV

|  |  |
| --- | --- |
| Name | Value |
| ADC Range UV | 1000 \* MP\_KU16\_ADC\_RANGE\_MV |
| **Definition** | |
| #define PMP\_KU32\_ADC\_RANGE\_UV | |
| **Description** | |
| ADC input full scale in uV | |

### PMP\_KU32\_ADC\_H\_RANGE\_UV

|  |  |
| --- | --- |
| Name | Value |
| ADC H Range UV | (PMP\_KU32\_ADC\_RANGE\_UV+1u)/2u |
| **Definition** | |
| #define PMP\_KU32\_ADC\_H\_RANGE\_UV | |
| **Description** | |
| ADC input half scale in uV (used to round division by PMP\_KU16\_ADC\_OUT\_SCALE) | |

### PMP\_KU32\_REF\_TEMP\_UV

|  |  |
| --- | --- |
| Name | Value |
| REF Temperature UV | 1000 \* PMP\_KU16\_REF\_TEMP\_MV |
| **Definition** | |
| #define PMP\_KU32\_REF\_TEMP\_UV | |
| **Description** | |
| Temperature sensor output in uV at its reference temperature | |

### PMP\_KU32\_ADC\_INPUT\_UV

|  |  |
| --- | --- |
| Name | Value |
| ADC Input UV | ((((output) \* PMP\_KU32\_ADC\_RANGE\_UV) + PMP\_KU16\_ADC\_OUT\_H\_SCALE) / PMP\_KU16\_ADC\_OUT\_SCALE) |
| **Definition** | |
| #define PMP\_KU32\_ADC\_INPUT\_UV(output) | |
| **Description** | |
| ADC input voltage in uV, for a given ADC output value | |

### PMP\_KU32\_ADC\_OUTPUT

|  |  |
| --- | --- |
| Name | Value |
| ADC Output | ((((input\_uv) \* PMP\_KU16\_ADC\_OUT\_SCALE) + PMP\_KU32\_ADC\_H\_RANGE\_UV) / PMP\_KU32\_ADC\_RANGE\_UV) |
| **Definition** | |
| #define PMP\_KU32\_ADC\_OUTPUT(input\_uv) | |
| **Description** | |
| ADC output value, for a given ADC input voltage in uV | |

### PMP\_KU32\_REF\_TEMP\_ADC

|  |  |
| --- | --- |
| Name | Value |
| Reference Temperature ADC | PMP\_KU32\_ADC\_OUTPUT(PMP\_KU32\_REF\_TEMP\_UV) |
| **Definition** | |
| #define PMP\_KU32\_REF\_TEMP\_ADC | |
| **Description** | |
| Temperature sensor output raw value at its reference temperature | |

### PMP\_KS16\_ADC\_TO\_DEG

|  |  |
| --- | --- |
| Name | Value |
| ADC to Degree | (((PMP\_KU8\_DEG\_RANGE \* temp\_adc) / PMP\_KU16\_ADC\_OUT\_SCALE)  - PMP\_KU8\_DEG\_LOW\_RANGE) |
| **Definition** | |
| #define PMP\_KS16\_ADC\_TO\_DEG(temp\_adc) | |
| **Description** | |
| Macro to convert a temperature from ADC output scale to degrees C, using a given slope parameter in micro volt per degree before the reference point | |

### PMP\_KU16\_DEG\_TO\_ADC

|  |  |
| --- | --- |
| Name | Value |
| Degree to ADC | PMP\_KU32\_ADC\_OUTPUT(PMP\_KU32\_REF\_TEMP\_UV - ((KU16\_REF\_TEMP\_DEG - temp\_deg) \* (uint32)slope)) |
| **Definition** | |
| #define PMP\_KU16\_DEG\_TO\_ADC(temp\_deg,slope) | |
| **Description** | |
| Macro to convert a temperature from degrees C to ADC output scale, using a given slope parameter in micro volt per degree before the reference point | |

### PMP\_KU16\_COMPUTE\_TEMP\_SLOP

|  |  |
| --- | --- |
| Name | Value |
| Compute Temperature Slop | (uint16)(((PMP\_KU32\_REF\_TEMP\_UV - KU32\_ADC\_INPUT\_UV(temp\_adc)) + ((((PMP\_KU16\_REF\_TEMP\_DEG-(temp\_deg))+1u)/2u)+1u)) / (PMP\_KU16\_REF\_TEMP\_DEG-(temp\_deg))) |
| **Definition** | |
| #define PMP\_KU16\_COMPUTE\_TEMP\_SLOP(temp\_deg,temp\_adc) | |
| **Description** | |
| Macro to compute a slop in uV/degree from the reference point using the real MCU temperature in degree and the ADC scaled temperature read | |

### PMP\_KU16\_FIX\_TEMP\_ADC

|  |  |
| --- | --- |
| Name | Value |
| Fix Temperature ADC | (uint16)(PMP\_KU32\_REF\_TEMP\_ADC - ((((PMP\_KU32\_REF\_TEMP\_ADC - temp\_adc) \* Typ\_slope) + Cal\_h\_slope) / Cal\_slope)) |
| **Definition** | |
| #define PMP\_KU16\_FIX\_TEMP\_ADC(temp\_adc,Cal\_slope,Cal\_h\_slope,Typ\_slope) | |
| **Description** | |
| Macro to fix a temperature in ADC output scale:  - consider that for (temp\_adc) temperature, the sensor observed the same slope as during the calibration (Cal\_slope)  - the new output temperature will respect the sensor typical slope (Typ\_slope) | |

### PMP\_KU32\_TYP\_SLOPE\_UV\_PER\_DEG

|  |  |
| --- | --- |
| Name | Value |
| TYP Slope UV per Degree | 5220 |
| **Definition** | |
| #define PMP\_KU32\_TYP\_SLOPE\_UV\_PER\_DEG | |
| **Description** | |
| Sensor's typical temperature slop in uV/degree C | |

### PMP\_KU16\_REF\_TEMP\_DEG

|  |  |
| --- | --- |
| Name | Value |
| Reference Temperature Degree | 150 |
| **Definition** | |
| #define PMP\_KU16\_REF\_TEMP\_DEG | |
| **Description** | |
| Temperature sensor's reference temperature in degree C | |

### PMP\_KU16\_REF\_TEMP\_MV

|  |  |
| --- | --- |
| Name | Value |
| Reference of temperature in mV | 2250 |
| **Definition** | |
| #define PMP\_KU16\_REF\_TEMP\_MV | |
| **Description** | |
| Temperature sensor output in mV at its reference temperature | |

### PMP\_KS16\_TEMP\_HIGH

|  |  |
| --- | --- |
| Name | Value |
| Temperature high | 35 |
| **Definition** | |
| #define PMP\_KS16\_TEMP\_HIGH | |
| **Description** | |
| High threshold for temperature | |

### PMP\_KU32\_TWO

|  |  |
| --- | --- |
| Name | Value |
| TWO | 2 |
| **Definition** | |
| #define PMP\_KU32\_TWO | |
| **Description** | |
| Internal macro for number 2 | |

### PMP\_KU32\_ONE\_HUNDRED

|  |  |
| --- | --- |
| Name | Value |
| ONE HUNDRED | 100 |
| **Definition** | |
| #define PMP\_KU32\_ONE\_HUNDRED | |
| **Description** | |
| Internal macro for number 100 | |

### PMP\_KU16\_ADC\_LEFT\_ALIGNED\_BIT\_SHIFT

|  |  |
| --- | --- |
| Name | Value |
| 6 bits Shift for left coded Raw values | 64 |
| **Definition** | |
| #define PMP\_KU16\_ADC\_LEFT\_ALIGNED\_BIT\_SHIFT | |
| **Description** | |
| 6 bits Shift for left coded Raw values | |

### PMP\_KU16\_FIVE\_HUNDRED

|  |  |
| --- | --- |
| Name | Value |
| PMP\_KU16\_FIVE\_HUNDRED | 500U |
| **Definition** | |
| #define PMP\_KU16\_FIVE\_HUNDRED | |
| **Description** | |
| Define used to convert system voltage variable | |

### PMP\_KU8\_NO\_REQUEST

|  |  |
| --- | --- |
| Name | Value |
| PMP\_KU8\_NO\_REQUEST | 255U |
| **Definition** | |
| #define PMP\_KU8\_NO\_REQUEST | |
| **Description** | |
| Define used to check if any request is active or not | |

# EEPROM

The EEPROM parameters are all specified in NVP\_param config .xls

Refer to this document for more details.

# Configuration

No special configuration for PMP software component.

# Compilation Options

No special configuration for PMP software component.