|  |  |
| --- | --- |
| **MMG**  **SW Detailed Design (DD)** | |
| **Summary** | This is the Software Detailed Design Document for the MMG component of *DAIMLER MMA* Project. |

|  |  |  |
| --- | --- | --- |
| **Author** | **Review** | **Approval** |
| SW Dev: Septimiu-Darex C. Vintila | 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 |
|  |  |  |
| **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 \* 6

1.2. Purpose and Scope 6

1.3. Referenced documents 7

1.3.1. External documents 7

1.3.2. Internal Documents 7

1.3.3. Terminology and definitions 7

2.1. Overview 8

2.2. Traceability 8

3.1. Services 18

3.1.1. MMG\_Init 18

3.1.2. MMG\_runUpdateModeStatus 20

3.1.3. MMG\_runManageModeDelays 21

3.1.4. MMG\_runCheckModeStatus 22

3.1.5. mmg\_UpdateModeStatus\_CriticalAutotestsNok 23

3.1.6. mmg\_UpdateModeStatus\_PowerVoltageUnstable 24

3.1.7. mmg\_UpdateModeStatus\_ExecutedCycleDelay 25

3.1.8. mmg\_UpdateModeStatus\_ExecutedCycle 26

3.1.9. mmg\_UpdateModeStatus\_ExecutedTensioningCycle 27

3.1.10. mmg\_UpdateModeStatus\_BusOff 28

3.1.11. mmg\_UpdateModeStatus\_NoHallEffectSensor 29

3.1.12. mmg\_UpdateModeStatus\_DefaultValues 30

3.1.13. mmg\_UpdateModeStatus\_PowerSupply\_AllCycles 31

3.1.14. mmg\_UpdateModeStatus\_EcuDefective\_AllCycles 31

3.1.15. mmg\_UpdateModeStatus\_EOLComfort\_BSR 32

3.1.16. mmg\_UpdateModeStatus\_ EOL\_LowForce\_TensioningCycles 32

3.1.17. mmg\_UpdateModeStatus\_Temperature\_EOL\_KL30\_TensioningCycles 34

3.1.18. mmg\_ManageModeDelays\_ExecutedCycle 35

3.1.19. mmg\_CheckIfCycleStepIsvalid 36

3.1.20. mmg\_CheckCycleEnable 37

3.1.21. mmg\_PreSafe\_CopyStartData 37

3.1.22. mmg\_PreSafe\_CopyStopData 38

3.1.23. mmg\_PreSafe\_CopyStateData 39

3.1.24. mmg\_PreSafe\_HandleStateData 40

3.1.25. mmg\_PreSafe\_HandleStartData 41

3.1.26. mmg\_PreSafe\_HandleStopData 41

3.1.27. mmg\_PreSafe\_Init 42

3.1.28. MMG\_runPreSafeRecorder 43

3.1.29. mmg\_PreSafe\_CheckEOLLimit 44

3.1.30. mmg\_AvailabilityDataStatus 44

3.1.31. mmg\_UpdateModeStatus\_SignalReleaseInhibition 45

3.1.32. MMG\_runUpdateNVMBlocks 46

3.1.33. mmg\_CheckCodingOfAPI 46

3.1.34. mmg\_CheckCodingRBTMFL 47

3.1.35. mmg\_CheckCodingRBTMFR 48

3.1.36. mmg\_CheckIgnitionON 48

3.2. Types 49

3.2.1. MMG\_stModeDelayType 49

3.2.2. MMG\_stModeConfigurationType 49

3.2.3. MMG\_KU8\_IDX 49

3.2.4. mmg\_stPreSafeRecorderStartData 50

3.2.5. mmg\_stPreSafeRecorderStopData 50

3.2.6. mmg\_stPreSafeRecorderStateData 50

3.2.7. mmg\_ stPreSafeRecorderManagement 50

3.3. Variables 51

3.3.1. mmg\_b8MMGHAsBeenInitialized 51

3.3.2. MMG\_u32ModesStatus 51

3.3.3. MMG\_u32ModesStatusComplement 51

3.3.4. mmg\_u16DelayToClearModeAfterCycle 51

3.3.5. MMG\_u16MaxDelayAfterCycleActivation 51

3.3.6. MMG\_astDelayAfterCycleThrsandModes 52

3.3.7. mmg\_u8PrevHESModeStatus 52

3.3.8. mmg\_stPreSafeManagement 52

3.3.9. mmg\_stPreSafeStartData 52

3.3.10. mmg\_stPreSafeStopData 52

3.3.11. mmg\_stPreSafeStateData 53

3.3.12. mmg\_stPreSafeInputData 53

3.3.13. mmg\_u32DeficiencyLevel 53

3.3.14. mmg\_u16LocalVoltage 53

3.3.15. mmg\_TimeStamp 53

3.3.16. mmg\_TimeUserData 53

3.3.17. mmg\_u8Return 54

3.3.18. u8oldAvailabilityDataCategory 54

3.3.19. MMG\_u16EnableCondition\_Status 54

3.4. Macros 54

3.4.1. MMG\_KU8\_CYCLE\_MODE\_DELAY\_INDEX 54

3.4.2. MMG\_KU8\_CYCLE\_MODE\_DELAY\_UV\_INDEX 54

3.4.3. MMG\_KU8\_NUMBER\_OF\_CYCLES\_MODES\_DELAYS 55

3.4.4. MMG\_U32\_INITIAL\_VALUES\_FOR\_ALL\_MODES 55

3.4.5. MMG\_KU8\_CFG\_LAST\_PRE\_TENSIONING\_CYCLE 55

3.4.6. MMG\_KU8\_CFG\_BSR1\_TENSIONING\_CYCLE 55

3.4.7. MMG\_KU8\_CFG\_HAPTIC\_WARNING\_10\_CYCLE 55

3.4.8. MMG\_KU8\_CFG\_HAPTIC\_WARNING\_41\_CYCLE 56

3.4.9. MMG\_KU8\_CFG\_PRODUCTION\_0\_CYCLE 56

3.4.10. MMG\_KU8\_CFG\_PRODUCTION\_1\_CYCLE 56

3.4.11. MMG\_KU16\_TIMEOUT\_INIT\_VALUE 56

3.4.12. M\_SET\_MODE\_MASK\_OFF 56

3.4.13. M\_SET\_MODE\_MASK\_ON 56

3.4.14. MMG\_KU8\_NB\_INHIB\_AEC\_ALL\_CYCLES 57

3.4.15. MMG\_KU8\_IDX\_POWERSUPPLY\_OV 57

3.4.16. MMG\_KU8\_IDX\_POWERSUPPLY\_UV 57

3.4.17. MMG\_KU8\_NB\_OF\_INHIB\_AEC\_FOR\_TENS\_CYCLES 57

3.4.18. MMG\_KU8\_IDX\_TEMPERATURE\_HIGH 57

3.4.19. MMG\_KU8\_IDX\_EOL\_LOW\_AND\_HIGH 58

3.4.20. MMG\_KU8\_IDX\_KL30\_UV\_TENS 58

3.4.21. MMG\_KU8\_IDX\_KL30\_OV\_TENS 58

3.4.22. MMG\_KU8\_PRESF\_TRIGGER\_OFF 58

3.4.23. MMG\_KU8\_PRESF\_TRIGGER\_ON 58

3.4.24. MMG\_KU8\_State\_DTC\_Is\_Not\_Set 59

3.4.25. MMG\_KU8\_State\_DTC\_Is\_Set 59

3.4.26. MMG\_KU8\_Tensioning\_Successfull 59

3.4.27. MMG\_KU8\_Tensioning\_NotStarted 59

3.4.28. MMG\_KU8\_Tensioning\_Abort 59

3.4.29. MMG\_KU8\_Tensioning\_NotDefined 59

3.4.30. MMG\_KU8\_SHIFT\_BELT\_HAND\_OVER 60

3.4.31. MMG\_KU8\_SHIFT\_IMPACTX 60

3.4.32. MMG\_KU8\_SHIFT\_ROLLOVERTYPE1 60

3.4.33. MMG\_KU8\_SHIFT\_ROLLOVERTYPE2 60

3.4.34. MMG\_KU8\_SHIFT\_PRESAFE\_DISABLED 60

3.4.35. MMG\_KU8\_SHIFT\_PRESAFE\_SUPP 61

3.4.36. MMG\_KU8\_SHIFT\_PRESAFE\_ABORT 61

3.4.37. MMG\_KU8\_SHIFT\_KL15OFF 61

3.4.38. MMG\_KU8\_SHIFT\_OV\_OR\_UV 61

3.4.39. MMG\_KU8\_SHIFT\_PRESAFE\_COUNTER 61

3.4.40. MMG\_KU8\_SHIFT\_SYSTEM\_FAILURE 61

3.4.41. MMG\_KU8\_VOLTAGE\_DIV 62

3.4.42. MMG\_KU16\_VOLTAGE\_MAX\_VAL 62

3.4.43. MMG\_KU32\_DEF\_LVL\_LOW\_THRS 62

3.4.44. MMG\_KU32\_DEF\_LVL\_MID\_THRS 62

3.4.45. MMG\_KU32\_DEF\_LVL\_HIGH\_THRS 62

3.4.46. MMG\_KU32\_DEF\_LVL\_HOT\_THRS 63

3.4.47. MMG\_KU16\_SEC\_TO\_MILI 63

3.4.48. MMG\_U16\_CONVERT\_TEMP\_TO\_RAW 63

3.4.49. KU8\_MMG\_ECU\_SIDE\_FRONT\_LEFT 63

3.4.50. KU8\_MMG\_ECU\_SIDE\_FRONT\_RIGHT 63

3.4.51. KU8\_MMG\_ECU\_LOCATION 63

3.4.52. KU8\_MMG\_PRESAFE\_RECORDER\_1\_ID 64

3.4.53. KU8\_MMG\_PRESAFE\_RECORDER\_2\_ID 64

3.4.54. MMG\_KU32\_AEC\_MASK\_GREEN\_STATUS 64

3.4.55. MMG\_KU32\_AEC\_MASK\_RED\_STATUS 64

3.4.56. MMG\_STATUS\_IS\_GREEN 65

3.4.57. MMG\_STATUS\_IS\_RED 65

3.4.58. MMG\_STATUS\_IS\_DEFAULT 65

3.4.59. MMG\_KU8\_IDX\_IMPL\_BUCKLE\_TENS 65

3.4.60. MMG\_KU8\_IDX\_TIMOEUT\_BUCKLEK\_TENS 65

3.4.61. MMG\_KU8\_CFG\_BSR3\_TENSIONING\_CYCLE 66

3.4.62. KU32\_MMG\_VEH\_EQP\_CYCLE\_ENABLE 66

3.4.63. KU32\_MMG\_AEC\_GROUP\_MASK 66

3.4.64. BOOL\_MMG\_CLEAR\_ENABLE\_CONDITION 66

3.4.65. BOOL\_MMG\_SET\_ENABLE\_CONDITION 67

3.4.66. MMG\_SET\_ENABLE\_CONDITION\_MASK\_OFF(mask) 67

3.4.67. MMG\_SET\_ENABLE\_CONDITION\_MASK\_ON(mask) 67

# Tables of figures

[Figure 1: MMG\_Init diagram 20](#_Toc157416770)

[Figure 2: MMG\_runUpdateModeStatus diagram 21](#_Toc157416771)

[Figure 3: MMG\_runManageModeDelays diagram 22](#_Toc157416772)

[Figure 4: MMG\_runCheckModeStatus diagram 23](#_Toc157416773)

[Figure 5: mmg\_UpdateModeStatus\_CriticalAutotestsNok diagram 24](#_Toc157416774)

[Figure 6: mmg\_UpdateModeStatus\_PowerVoltageUnstable diagram 25](#_Toc157416775)

[Figure 7: mmg\_UpdateModeStatus\_ExecutedCycleDelay diagram 26](#_Toc157416776)

[Figure 8: mmg\_UpdateModeStatus\_ExecutedCycle diagram 27](#_Toc157416777)

[Figure 9: mmg\_UpdateModeStatus\_ExecutedTensioningCycle diagram 28](#_Toc157416778)

[Figure 10: mmg\_UpdateModeStatus\_BusOff diagram 29](#_Toc157416779)

[Figure 11 mmg\_UpdateModeStatus\_ NoHallEffectSensor f diagram 30](#_Toc157416780)

[Figure 12 mmg\_UpdateModeStatus\_DefaultValues diagram 31](#_Toc157416781)

[Figure 13 mmg\_UpdateModeStatus\_PowerSupply\_AllCycles: 31](#_Toc157416782)

[Figure 14 mmg\_UpdateModeStatus\_EcuDefective\_AllCycles 32](#_Toc157416783)

[Figure 15 mmg\_UpdateModeStatus\_EOLComfort\_BSR 32](#_Toc157416784)

[Figure 16: mmg\_UpdateModeStatus\_ EOL\_LowForce\_TensioningCycles 34](#_Toc157416785)

[Figure 17: mmg\_UpdateModeStatus\_Temperature\_EOL\_KL30\_TensioningCycles 35](#_Toc157416786)

[Figure 18: mmg\_ManageModeDelays\_ExecutedCycle diagram 36](#_Toc157416787)

[Figure 19: mmg\_CheckIfCycleStepIsvalid diagram 37](#_Toc157416788)

[Figure 20: mmg\_CheckCycleEnable diagram 37](#_Toc157416789)

[Figure 21: mmg\_PreSafe\_CopyStartData 38](#_Toc157416790)

[Figure 22: mmg\_PreSafe\_CopyStopData 39](#_Toc157416791)

[Figure 23: mmg\_PreSafe\_CopyStateData 40](#_Toc157416792)

[Figure 24: Part1 mmg\_PreSafe\_HandleStateData 41](#_Toc157416793)

[Figure 25: Part2 mmg\_PreSafe\_HandleStateData 41](#_Toc157416794)

[Figure 26: Part1 mmg\_PreSafe\_HandleStartData 41](#_Toc157416795)

[Figure 27: Part2 mmg\_PreSafe\_HandleStartData 41](#_Toc157416796)

[Figure 28: Part1 mmg\_PreSafe\_HandleStopData 42](#_Toc157416797)

[Figure 29: Part1 mmg\_PreSafe\_HandleStopData 42](#_Toc157416798)

[Figure 30: mmg\_PreSafe\_Init 43](#_Toc157416799)

[Figure 31: Part1 MMG\_runPreSafeRecorder 43](#_Toc157416800)

[Figure 32: Part2 MMG\_runPreSafeRecorder 43](#_Toc157416801)

[Figure 33: Part3 MMG\_runPreSafeRecorder 44](#_Toc157416802)

[Figure 34: mmg\_PreSafe\_CheckEOLLimit 44](#_Toc157416803)

[Figure 35: mmg\_AvailabilityDataStatus 45](#_Toc157416804)

[Figure 36mmg\_UpdateModeStatus\_SignalReleaseInhibition 46](#_Toc157416805)

[Figure 37MMG\_runUpdateNVMBlocks 46](#_Toc157416806)

[Figure 38 Check coding of API 47](#_Toc157416807)

[Figure 39 Check coding of RBTMFL 48](#_Toc157416808)

[Figure 40 Check coding of RBTMFR 48](#_Toc157416809)

[Figure 41 Check Ignition status 49](#_Toc157416810)

# General Information

## Revision history \*

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Author(s)** | **Description/comment** |
| 1.1.4.1 | 04.02.2022 | Septimiu-Darex C. Vintila | Initial revision. |
| 1.1.4.2 | 04.02.2022 | Septimiu-Darex C. Vintila | Design fixes after review. |
| 1.1.4.3 | 08.02.2022 | Septimiu-Darex C. Vintila | Traceability added. |
| 1.1.4.4 | 15.02.2022 | Septimiu-Darex C. Vintila | Requirements fixed. |
| 1.1.4.5 | 10.03.2022 | Andreea Negrea | Add check for validity of cycles. |
| 1.1.4.6 | 17.03.2022 | Andreea Negrea | Add fix after review. |
| 1.1,4.7 | 29.03.2022 | Mirela Obada | Update traceabilty. |
| 1.1.4.8 | 29.03.2022 | Mirela Obada | Update document. |
| 1.1.4.9 | 11.04.2022 | Mirela Obada | Fix findings from review. |
| 1.1.4.10 | 14.06.2022 | Mirela Obada | Add traceability and update of figures. |
| 1.1.4.11 | 17.06.2022 | Mirela Obada | Update after review. |
| 1.1.4.12 | 20.06.2022 | Mirela Obada | Update functions template. |
| 1.1.4.13 | 21.06.2022 | Mirela Obada | Update tabel of content. |
| 1.1.4.14 | 21.06.2022 | Mirela Obada | Update version of document. |
| 1.1.4.15 | 22.06.2022 | Mirela Obada | Update traceability. |
| 1.1.4.16 | 22.06.2022 | Mirela Obada | Update revision history. |
| 1.1.4.17 | 22.06.2022 | Mirela Obada | Modify requirement 0042 because it was doubled. |
| 1.1.4.18 | 31.08.2022 | Stefan Dominte | Update for 3.0 release. |
| 1.1.4.19 | 15.11.2022 | Stefan Dominte | Update for 4.0 release. |
| 1.1.4.20 | 22.11.2022 | Stefan Dominte | Update after review. |
| 1.1.4.21 | 19.01.2023 | Andreea Negrea | Update with 5.0 belt functionality |
| 1.1.4.22 | 09.02.2023 | Stefan Dominte | Update with 5.0 presafe recorder |
| 1.1.4.23 | 13.02.2023 | Tudor Gligor | Update with 5.0 AvailabilityDataStatus |
| 1.1.4.24 | 15.02.2023 | Stefan Dominte | Update after review |
| 1.1.4.25 | 20.02.2023 | Stefan Dominte | Traceabilty update |
| 1.1.4.26 | 21.02.2023 | Stefan Dominte | Traceabilty re-update |
| 1.1.4.27 | 02.03.2023 | Andreea Negrea | Update with new context for EOL Low Force |
| 1.1.4.28 | 06.03.2023 | Stefan Dominte | Update for R5.1z |
| 1.1.4.29 | 07.03.2023 | Stefan Dominte | Update with inhibiton context for pyro activation |
| 1.1.4.30 | 08.03.2023 | Stefan Dominte | Update after review |
| 1.1.4.31 | 05.05.2023 | Andreea Negrea | Update for 6.0 |
| 1.1.4.32 | 20/06/2023 | Madalina Serban | Update for 6.1 |
| 1.1.4.33 | 21/06/2023 | Madalina Serban | Updated traceability |
| 1.1.4.34 | 22/06/2023 | Madalina Serban | Updated traceability |
| 1.1.4.35 | 22/06/2023 | Madalina Serban | Updated internal document version |
| 1.1.4.36 | 30/08/2023 | Mirela Obada | Update for R7.0 release |
| 1.1.4.37 | 06/10/2023 | Tudor Gligor | Update mmg\_AvailabilityDataStatus |
| 1.1.4.38 | 13/10/2023 | Madalina Serban | Updated diagrams |
| 1.1.4.39 | 19/01/2024 | Mirela Obada | Update for 8.1 |
| 1.1.4.40 | 29/01/2024 | Mirela Obada | Update after SRM |

*\* 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 MMG component

## Referenced documents

### External documents

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

### Internal Documents

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

### 1.3.3. 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 |
| MMG | Mode ManaGement |

# SW atomic architectural unit design

## Overview

Purpose of the MMG component is to:

* Gather data from other components and calculate the overall status of the system.
* Communicate to other components the status of the system
* Determine at Init if available tensioning cycles are valid or not (different from 0xFF)
* Gather data from other components in order to populate PreSafe Recorder with specific information
* Gather data from other components to populate the RoELite structure with Availability Data

**MMG**

Scheduler

**PAL**

**PMP**

MMG\_Init()

MMG\_runUpdateModeStatus()

MMG\_runAutotestCall10ms()MMG\_runAutotestCall100ms()

MMG\_runManageModeDelays()

MMG\_runMainFunction()

PAL\_Autotest\_CheckHWSelfProtection()

PAL\_Autotest\_CheckMotorCurrent()

PAL\_Autotest\_CheckMotorThermalProtection()

PMP\_Autotest\_CheckPowerSupplyOV()

PMP\_Autotest\_CheckPowerSupplyUV()

DEM

DEM\_SetEventStatus()

DEM\_GetEventStatus()

**BFE**

**PRE**

**SFR**

MMG\_runCheckModeStatus()

BFE\_ReadExecutedCycle()

MMG\_runGetTestResult()

MMG\_runGetTestResult()

## Traceability

|  |  |  |  |
| --- | --- | --- | --- |
| Requirements | Criteria | Linked Runnable | Source |
| DSG\_MMG\_0001 | The “Critical auto tests NOK” mode shall be updated every 10 ms based on the available information. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0004 |
| DSG\_MMG\_0002 | The “Power voltage unstable” mode shall be updated every 10 ms based on the available information. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0004 |
| DSG\_MMG\_0003 | The “Executed cycle with delay” modes shall be updated every 10 ms based on the available information. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0004 |
| DSG\_MMG\_0004 | The “Executed cycle” mode shall be updated every 10 ms based on the available information. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0004 |
| DSG\_MMG\_0005 | The “Executed tensioning cycle” mode shall be updated every 10 ms based on the available information. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0004 |
| DSG\_MMG\_0007 | The “No hall effect sensor” mode shall be updated every 10 ms based on the available information. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0004 |
| DSG\_MMG\_0008 | The “Inhibition of all cycles” mode shall be updated every 10 ms based on the available information. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0004 |
| DSG\_MMG\_0009 | The “Inhibition of all cycles by ecu defect” mode shall be updated every 10 ms based on the available information. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0004 |
| DSG\_MMG\_0010 | The “EOL counter inhibition” mode shall be updated every 10 ms based on the available information. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0004 |
| DSG\_MMG\_0011 | The “Inhibition of tensioning” mode shall be updated every 10 ms based on the available information. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0004 |
| DSG\_MMG\_0012 | The delays for the executed cycles shall be updated every 10 ms. | MMG\_runManageModeDelays | ARCH\_SW\_MMG\_0007 |
| DSG\_MMG\_0013 | The status for all modes shall be available on request. | MMG\_runCheckModeStatus | ARCH\_SW\_MMG\_0006 |
| DSG\_MMG\_0014 | At INIT all cycles shall be checked to have at least one step different from 0xFF. | MMG\_Init() | ARCH\_SW\_MMG\_0008; |
| DSG\_MMG\_0015 | At INIT first steps shall be checked to have first byte different from 0xFF. | MMG\_Init() | ARCH\_SW\_MMG\_0008; |
| DSG\_MMG\_0016 | If MMG has been initialized local variable MMG\_b8MMGHAsBeenInitialized is B\_TRUE) the contexts will be periodically updated. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0004; |
| DSG\_MMG\_0017 | **mmg\_UpdateModeStatus\_CriticalAutotestsNok** local function will read the startup Autotest status for setting the MMG\_KU32\_MASK\_CRITICAL\_AT\_NOT context. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0250 |
| DSG\_MMG\_0018 | **mmg\_UpdateModeStatus\_PowerVoltageUnstable** local function will read the Battery voltage survey status for setting the MMG\_KU32\_MASK\_POWER\_VOLTAGE\_UNSTABLE context. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0251 |
| DSG\_MMG\_0019 | **mmg\_UpdateModeStatus\_ExecutedCycleDelay** local function will read the executed cycle delay status for setting the MMG\_KU32\_MASK\_NO\_BELTFUNCTIONS\_DELAY context. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0252 |
| DSG\_MMG\_0020 | **mmg\_UpdateModeStatus\_ExecutedCycle** local function will read the current execute cycle and pyro activation status for setting the MMG\_KU32\_MASK\_NO\_BELTFUNCTIONS context. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0253; ARCH\_SW\_MMG\_0254 |
| DSG\_MMG\_0021 | **mmg\_UpdateModeStatus\_NoHallEffectSensor** local function will read the Hall Efect Sensor AEC status for setting the MMG\_KU32\_MASK\_NO\_HALL\_EFFECT\_SENSOR context. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0140; ARCH\_SW\_MMG\_0205 |
| DSG\_MMG\_0022 | **mmg\_UpdateModeStatus\_PowerSupply\_AllCycles** local function will compute the NVP\_stVehicleEquipmentData. Undervoltage and NVP\_stVehicleEquipmentData.Overvoltage variables in order to set the MMG\_KU32\_MASK\_AEC\_ECU\_DEFECT\_ALL\_CYCLES context. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0150; ARCH\_SW\_MMG\_0151 |
| DSG\_MMG\_0023 | **mmg\_UpdateModeStatus\_EcuDefective\_AllCycles** local function will read the AEC group status and check if Group mask hardware AEC is set in order to compute MMG\_KU32\_MASK\_AEC\_ECU\_DEFECT\_ALL\_CYCLES context. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0152; ARCH\_SW\_MMG\_0205 |
| DSG\_MMG\_0024 | **mmg\_UpdateModeStatus\_EcuDefective\_AllCycles** local function will read the AEC group status and check if Group SW/HW Self protection status AEC is set in order to compute MMG\_KU32\_MASK\_AEC\_ECU\_DEFECT\_ALL\_CYCLES context. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0153 |
| DSG\_MMG\_0025 | **mmg\_UpdateModeStatus\_Temperature\_EOL\_KL30\_TensioningCycles** local function will read the AEC group status and check if Temperature high AEC is set in order to compute MMG\_KU32\_MASK\_AEC\_INHIB\_TENSIONING\_CYCLES context. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0133; ARCH\_SW\_MMG\_0205 |
| DSG\_MMG\_0026 | **mmg\_UpdateModeStatus\_Temperature\_EOL\_KL30\_TensioningCycles** local function will read the AEC group status and check if EOL low and high AEC is set in order to compute MMG\_KU32\_MASK\_AEC\_INHIB\_TENSIONING\_CYCLES context. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0205 |
| DSG\_MMG\_0114 | **mmg\_UpdateModeStatus\_EOL\_Low\_Force** function will read the AEC group status and check if EOL low and high AEC is set in order to compute MMG\_KU32\_MASK\_EOL\_LOW\_FORCE | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0138; |
| DSG\_MMG\_0027 | **mmg\_UpdateModeStatus\_Temperature\_EOL\_KL30\_TensioningCycles** local function will compute NVP\_stVehicleEquipmentData.UndervoltageAbort variable in order to set MMG\_KU32\_MASK\_AEC\_INHIB\_TENSIONING\_CYCLES context. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0135 |
| DSG\_MMG\_0028 | **mmg\_UpdateModeStatus\_Temperature\_EOL\_KL30\_TensioningCycles** local function will compute the NVP\_stVehicleEquipmentData.OvervoltageAbort in order to set the MMG\_KU32\_MASK\_AEC\_INHIB\_TENSIONING\_CYCLES context. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0157 |
| DSG\_MMG\_0029 | **mmg\_UpdateModeStatus\_Temperature\_EOL\_KL30\_TensioningCycles** local function will read the AEC group status and check if implausible data for buckle signal AEC is set in order to compute MMG\_KU32\_MASK\_AEC\_INHIB\_TENSIONING\_CYCLES context. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0158 |
| DSG\_MMG\_0030 | **mmg\_UpdateModeStatus\_Temperature\_EOL\_KL30\_TensioningCycles** local function will read the AEC group status and check if timeout for buckle signal AEC is set in order to compute MMG\_KU32\_MASK\_AEC\_INHIB\_TENSIONING\_CYCLES context. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0159 |
| DSG\_MMG\_0031 | **mmg\_UpdateModeStatus\_Temperature\_EOL\_KL30\_TensioningCycles** local function will read the AEC group status and check if control steering configuration AEC is set in order to compute MMG\_KU32\_MASK\_AEC\_INHIB\_TENSIONING\_CYCLES context. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0160 |
| DSG\_MMG\_0032 | **mmg\_UpdateModeStatus\_EOLComfort\_BSR** local function will read the AEC group status and check if EOL comfort AEC is set in order to compute MMG\_KU32\_MASK\_EOL\_COUNTER\_INHIBITION context. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0162; ARCH\_SW\_MMG\_0205 |
| DSG\_MMG\_0033 | **mmg\_UpdateModeStatus\_CriticalAutotestsNok** local function will read the last critical autotest ID as an input. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0200 |
| DSG\_MMG\_0034 | **mmg\_UpdateModeStatus\_CriticalAutotestsNok** local function will read the status of the last executed autotest as an input. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0201 |
| DSG\_MMG\_0035 | **mmg\_UpdateModeStatus\_PowerVoltageUnstable** local function will read the battery voltage status. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0202 |
| DSG\_MMG\_0036 | **mmg\_UpdateModeStatus\_ExecutedCycle** local function will read the executed cycle as an input. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0203 |
| DSG\_MMG\_0037 | **mmg\_UpdateModeStatus\_ExecutedTensioningCycle** local function will read the executed cycle as an input. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0203 |
| DSG\_MMG\_0038 | **mmg\_UpdateModeStatus\_ExecutedCycle** local function will read the pyro activation as an input. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0204 |
| DSG\_MMG\_0039 | **mmg\_UpdateModeStatus\_NoHallEffectSensor** local function will notify BMM about restoring the puls counting. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0206 |
| DSG\_MMG\_0040 | **mmg\_UpdateModeStatus\_Temperature\_EOL\_KL30\_TensioningCycles local function** will be called in order to check if the battery voltage is outside Normal and Extended Operating range. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0207; ARCH\_SW\_MMG\_0208 |
| DSG\_MMG\_0041 | Local function **mmg\_AvailabilityDataStatus** will send the AvailabilityDataCategory towards DiagFunction only if it changed from last sta-te | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0209 |
| DSG\_MMG\_0113 | **mmg\_UpdateModeStatus\_SignalReleaseInhibition** local function will compute signal release inhibition in order to compute MMG\_KU32\_MASK\_SIGNAL\_RELEASE\_INHIBITION context. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0301 |
| DSG\_MMG\_0122 | **Rte\_Read\_prrCustomerSpecific\_b8SignalRelInhibition shall be called to** check the value of release inhibition for costumer signals | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0302 |
| DSG\_MMG\_0042 | **mmg\_CheckCycleEnable** local function will write the validity of cycle 0-19. | MMG\_Init | ARCH\_SW\_MMG\_0300 |
| DSG\_MMG\_0043 | **mmg\_ManageModeDelays\_ExecutedCycle** local function will read the reset cause as an input | MMG\_runManageModeDelays | ARCH\_SW\_MMG\_0400 |
| DSG\_MMG\_0044 | **mmg\_ManageModeDelays\_ExecutedCycle** local function will read the executed cycle as an input. | MMG\_runManageModeDelays | ARCH\_SW\_MMG\_0401 |
| DSG\_MMG\_0045 | **mmg\_ManageModeDelays\_ExecutedCycle** local function will read the pyro activation status as an input. | MMG\_runManageModeDelays | ARCH\_SW\_MMG\_0402 |
| DSG\_MMG\_0053 | **mmg\_PreSafe\_HandleStateData** local function will read the AEC group status in order to set Abortion of a Presafe cycle bit. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0403 |
| DSG\_MMG\_0054 | **mmg\_PreSafe\_HandleStateData** local function will read the AEC group status in order to set Under and Over voltage bit. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0403 |
| DSG\_MMG\_0055 | **mmg\_PreSafe\_HandleStateData** local function will read the AEC group status in order to set Overtemperature bit. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0403 |
| DSG\_MMG\_0056 | **mmg\_PreSafe\_HandleStateData** local function will read the AEC group status in order to set System failure bit. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0403 |
| DSG\_MMG\_0057 | Local variable **mmm\_stPreSafeInputData** will hold the input signals received from CIL module. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0404 |
| DSG\_MMG\_0058 | Local variable **mmg\_u32DeficiencyLevel** will hold the deficiency value received from PMP module. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0405 |
| DSG\_MMG\_0059 | Local variable **mmg\_u16LocalVoltage** will hold the local voltage value received from PMP module. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0406 |
| DSG\_MMG\_0060 | Local variable **mmg\_TimeUserData** will hold the system time stamp value received from StbM module. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0407 |
| DSG\_MMG\_0061 | **mmg\_PreSafe\_CheckEOLLimit**local function will read the EOL counters Autotests status in order to set the Presafe counter bit. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0408 |
| DSG\_MMG\_0062 | Local variable **mmg\_stPreSafeStartData** will hold the value of the Start Block and will be saved in RAM when a requested profile is inhibited or started successful | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0009 |
| DSG\_MMG\_0063 | Local variable **mmg\_stPreSafeStopData** will hold the value of the Stop Block and will be saved in RAM when a requested profile is ended successful, aborted or inhibited. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0010 |
| DSG\_MMG\_0064 | Local variable **mmg\_stPreSafeStateData** will hold the value of the State Block and will be saved in RAM when a start or stop block is saved. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0011 |
| DSG\_MMG\_0065 | Local variable **mmg\_stPreSafeManagement.** **PreSafeBufferID** will hold the ID of the Recorder that will be saved in RAM and will be updated every time a recorder is saved. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0012 |
| DSG\_MMG\_0066 | Local variables **mmg\_stPreSafeManagement.PreSafeState** and **mmg\_stPreSafeManagement.** **PreSafePreviousState** will hold the tensioning state and the previous tensioning state of a executed profile. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0013; ARCH\_SW\_MMG\_0014 |
| DSG\_MMG\_0067 | If **mmg\_stPreSafeManagement.PreSafeState** and **mmg\_stPreSafeManagement.** **PreSafePreviousState** are not equal than a new profile is executed and a Presafe Recorder needs to be saved. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0013; ARCH\_SW\_MMG\_0014 |
| DSG\_MMG\_0068 | Local variables **mmg\_stPreSafeStartData.Start1BuckleSwitch** and **mmg\_stPreSafeStopData.Stop1BuckleSwitch** will hold the buckle switch state received from CIL module. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0016 |
| DSG\_MMG\_0069 | Local variables **mmg\_stPreSafeStartData.** **Start1BeltHandOver** and **mmg\_stPreSafeStopData.Stop1 BeltHandOver** will hold the belt hand over state received from CIL module. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0017 |
| DSG\_MMG\_0070 | Local variables **mmg\_stPreSafeStartData.Start2** **PreSafeLvl** and **mmg\_stPreSafeStopData.Stop2** **PreSafeLvl** will hold the Presafe level received from CIL module. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0018 |
| DSG\_MMG\_0071 | Local variables **mmg\_stPreSafeStartData.Start2** **ImpactX** and **mmg\_stPreSafeStopData.Stop2** **ImpactX** will be set with the value 0x01 if ImpactX signal received from CIL module is 0x01 or will be set with the value 0x00 if ImpactX signal received from CIL module is 0x00. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0019 |
| DSG\_MMG\_0072 | Local variables **mmg\_stPreSafeStartData.Start2** **RollOverType1** and **mmg\_stPreSafeStopData.Stop2** **RollOverType1** will be set with the value 0x01 if RollOverType1 signal received from CIL module is 0x01 or will be set with the value 0x00 if RollOverType1 signal received from CIL module is 0x00. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0020 |
| DSG\_MMG\_0073 | Local variables **mmg\_stPreSafeStartData.Start2** **RollOverType2** and **mmg\_stPreSafeStopData.Stop2** **RollOverType2** will be set with the value 0x01 if RollOverType2 signal received from CIL module is 0x01 or will be set with the value 0x00 if RollOverType2 signal received from CIL module is 0x00. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0021 |
| DSG\_MMG\_0074 | Local variables **mmg\_stPreSafeStartData.** **StartVoltage** and **mmg\_stPreSafeStopData.StopVoltage** will hold the value of the computed KL15 voltage. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0022 |
| DSG\_MMG\_0075 | Local variables **mmg\_stPreSafeStartData.** **StartTemperature2** and **mmg\_stPreSafeStopData.** **StopTemperature2** will always be set with the value 0x00. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0023 |
| DSG\_MMG\_0076 | Local variables **mmg\_stPreSafeStartData.** **StartTemperature1** and **mmg\_stPreSafeStopData.** **StopTemperature1** will be set with the value 0x00 if deficiency value read from PMP module is lower than 1200. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0023 |
| DSG\_MMG\_0077 | Local variables **mmg\_stPreSafeStartData.** **StartTemperature1** and **mmg\_stPreSafeStopData.** **StopTemperature1** will be set with the value 0x01 if deficiency value read from PMP module is equal or higher than 1200 and lower than 32000. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0023 |
| DSG\_MMG\_0078 | Local variables **mmg\_stPreSafeStartData.** **StartTemperature1** and **mmg\_stPreSafeStopData.** **StopTemperature1** will be set with the value 0x02 if deficiency value read from PMP module is equal or higher than 32000 and lower than 150000. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0023 |
| DSG\_MMG\_0079 | Local variables **mmg\_stPreSafeStartData.** **StartTemperature1** and **mmg\_stPreSafeStopData.** **StopTemperature1** will be set with the value 0x03 if deficiency value read from PMP module is equal or higher than 150000 and lower than 160000. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0023 |
| DSG\_MMG\_0080 | Local variables **mmg\_stPreSafeStartData.** **StartTemperature1** and **mmg\_stPreSafeStopData.** **StopTemperature1** will be set with the value 0x04 if deficiency value read from PMP module is equal or higher than 160000. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0023 |
| DSG\_MMG\_0081 | Local variables **mmg\_stPreSafeStartData.** **StartTemperature1** and **mmg\_stPreSafeStopData.** **StopTemperature1** will be set with the value 0x0FF if deficiency value read from PMP module is invalid. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0023 |
| DSG\_MMG\_0082 | Local variables **mmg\_stPreSafeStartData.** **Start SystemTime1** and **mmg\_stPreSafeStopData.** **StopSystemTime1** will hold the first byte of the system time read from the SbtM module. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0024 |
| DSG\_MMG\_0083 | Local variables **mmg\_stPreSafeStartData.** **Start SystemTime2** and **mmg\_stPreSafeStopData.** **StopSystemTime2** will hold the second byte of the system time read from the SbtM module. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0024 |
| DSG\_MMG\_0084 | Local variables **mmg\_stPreSafeStartData.** **Start SystemTime3** and **mmg\_stPreSafeStopData.** **StopSystemTime3** will hold the third byte of the system time read from the SbtM module. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0024 |
| DSG\_MMG\_0085 | Local variables **mmg\_stPreSafeStartData.** **Start SystemTime4** and **mmg\_stPreSafeStopData.** **StopSystemTime4** will hold the fourth byte of the system time read from the SbtM module. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0024 |
| DSG\_MMG\_0086 | Local variables **mmg\_stPreSafeStartData.** **Start SystemTime5** and **mmg\_stPreSafeStopData.** **StopSystemTime5** will always be set with 0x00 value. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0024 |
| DSG\_MMG\_0087 | Local variables **mmg\_stPreSafeStartData, mmg\_stPreSafeStopData and mmg\_stPreSafeStateData** will not set reserved bits and leave them with 0x00 value. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0025 |
| DSG\_MMG\_0088 | Local variable **mmg\_stPreSafeStateData.State1TensioningState** will be set the value 0x00 if tensioning state received from CIL module is equal to 0x00. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0026 |
| DSG\_MMG\_0089 | Local variable **mmg\_stPreSafeStateData.State1TensioningState** will be set the value 0x00 if tensioning state received from CIL module is equal to 0x01. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0026 |
| DSG\_MMG\_0090 | Local variable **mmg\_stPreSafeStateData.State1TensioningState** will be set the value 0x01 if tensioning state received from CIL module is equal to 0x02. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0026 |
| DSG\_MMG\_0091 | Local variable **mmg\_stPreSafeStateData.State1TensioningState** will be set the value 0x02 if tensioning state received from CIL module is equal to 0x03. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0026 |
| DSG\_MMG\_0092 | Local variable **mmg\_stPreSafeStateData.State1TensioningState** will be set the value 0x01 if presafe function is disabled or if the cycle requested is disabled. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0027 |
| DSG\_MMG\_0093 | Local variable **mmg\_stPreSafeStateData.State1TensioningState** will be set the value 0x00 if presafe function is not disabled and if the cycle requested is not disabled. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0027 |
| DSG\_MMG\_0094 | Local variable **mmg\_stPreSafeStateData.** **State1PreSafeSupp** will be set the value 0x00 if tensioning suppression type received from CIL module is equal to 0x00. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0028 |
| DSG\_MMG\_0095 | Local variable **mmg\_stPreSafeStateData.** **State1PreSafeSupp** will be set the value 0x00 if tensioning suppression type received from CIL module is equal to 0x01 and ECU location is right. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0028 |
| DSG\_MMG\_0096 | Local variable **mmg\_stPreSafeStateData.** **State1PreSafeSupp** will be set the value 0x01 if tensioning suppression type received from CIL module is equal to 0x01 and ECU location is left. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0028 |
| DSG\_MMG\_0097 | Local variable **mmg\_stPreSafeStateData.** **State1PreSafeSupp** will be set the value 0x01 if tensioning suppression type received from CIL module is equal to 0x02 and ECU location is right. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0028 |
| DSG\_MMG\_0098 | Local variable **mmg\_stPreSafeStateData.** **State1PreSafeSupp** will be set the value 0x00 if tensioning suppression type received from CIL module is equal to 0x02 and ECU location is left. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0028 |
| DSG\_MMG\_0099 | Local variable **mmg\_stPreSafeStateData.** **State1PreSafeSupp** will be set the value 0x01 if tensioning suppression type received from CIL module is equal to 0x03. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0028 |
| DSG\_MMG\_0100 | Local variable **mmg\_stPreSafeStateData.State1PreSafeAbort** will be set the value 0x00 if no Timeout or Implausible data AEC is set. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0029 |
| DSG\_MMG\_0101 | Local variable **mmg\_stPreSafeStateData.State1PreSafeAbort** will be set the value 0x01 if any Timeout or Implausible data AEC is set. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0029 |
| DSG\_MMG\_0102 | Local variable **mmg\_stPreSafeStateData.** **State1KL15** will be set the value 0x01 if Ignition signal received from CIL module is different than 0x04. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0030 |
| DSG\_MMG\_0103 | Local variable **mmg\_stPreSafeStateData.** **State1KL15** will be set the value 0x00 if Ignition signal received from CIL module is equal to 0x04. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0030 |
| DSG\_MMG\_0104 | Local variable **mmg\_stPreSafeStateData.** **State1UvOROV** will be set the value 0x00 if Undevoltage and Overvoltage AECs are not set. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0031 |
| DSG\_MMG\_0105 | Local variable **mmg\_stPreSafeStateData.** **State1UvOROV** will be set the value 0x01 if Undevoltage or Overvoltage AECs are set. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0031 |
| DSG\_MMG\_0106 | Local variable **mmg\_stPreSafeStateData.** **State2OverTemp**will be set the value 0x00 if Self protection AEC is not set. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0032 |
| DSG\_MMG\_0107 | Local variable **mmg\_stPreSafeStateData.** **State2OverTemp** will be set the value 0x01 if Self protection AEC is set. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0032 |
| DSG\_MMG\_0108 | Local variable **mmg\_stPreSafeStateData.** **State2SystemFailure** will be set the value 0x00 if Hardware AEC is not set. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0034 |
| DSG\_MMG\_0109 | Local variable **mmg\_stPreSafeStateData.** **State2SystemFailure** will be set the value 0x01 if Hardware AEC is set. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0034 |
| DSG\_MMG\_0110 | Local function **mmg\_PreSafe\_CheckEOLLimit** will set the **mmg\_stPreSafeStateData.State2PreSafeCounter** local variable with the value 0x00 if the EOL counter mapped on the executed cycle has not reached the maximum value. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0033 |
| DSG\_MMG\_0111 | Local function **mmg\_PreSafe\_CheckEOLLimit** will set the **mmg\_stPreSafeStateData.State2PreSafeCounter** local variable with the value 0x01 if the EOL counter mapped on the executed cycle has reached the maximum value. | MMG\_runPreSafeRecorder | ARCH\_SW\_MMG\_0033 |
| DSG\_MMG\_0112 | **mmg\_UpdateModeStatus\_AllCycles local function** will be called in order to check if pyro activation is on and then all Tensioning cycles will be inhibitted. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0154 |
| DSG\_MMG\_0115 | **NvM\_GetErrorStatus** shall be called in order to get the error status for each block:   * ExecutionCountersBlock * CANInputBlockStatus * AECsBlockStatus * ServicesRecorder1 * ServicesRecorder2 | MMG\_runUpdateNVMBlocks() | ARCH\_SW\_MMG\_0303 |
| DSG\_MMG\_0116 | **NvM\_WriteBlock** shall be called in order to write updated blocks | MMG\_runUpdateNVMBlocks() | ARCH\_SW\_MMG\_0304 |
| DSG\_MMG\_0117 | **Rte\_IsUpdated\_prrExecutionCountersBlockStatus\_b8NVMBlockStatus** shall be called in order to check if execution counters Block has been updated | MMG\_runUpdateNVMBlocks() | ARCH\_SW\_MMG\_0305 |
| DSG\_MMG\_0118 | **Rte\_IsUpdated\_prrCANInputBlockStatus\_b8NVMBlockStatus** shall be called in order to check if CAN Input signals Block has been updated | MMG\_runUpdateNVMBlocks() | ARCH\_SW\_MMG\_0306 |
| DSG\_MMG\_0119 | **Rte\_IsUpdated\_prrAECsBlockStatus\_b8NVMBlockStatus** shall be called in order to check if AECs Block has been updated | MMG\_runUpdateNVMBlocks() | ARCH\_SW\_MMG\_0307 |
| DSG\_MMG\_0120 | **Rte\_IrvRead\_MMG\_runUpdateNVMBlocks\_b8PreSafeRecorder1Update** shall be called in order to check if PreSafeRecorder1 Block has been updated | MMG\_runUpdateNVMBlocks() | ARCH\_SW\_MMG\_0308 |
| DSG\_MMG\_0121 | **Rte\_IrvRead\_MMG\_runUpdateNVMBlocks\_b8PreSafeRecorder2Update** shall be called in order to check if PreSafeRecorder2 Block has been updated | MMG\_runUpdateNVMBlocks() | ARCH\_SW\_MMG\_0309 |
| DSG\_MMG\_0122 | If step 0 from a cycle is 0xFF then the return status of validity check shall be ‘B\_FALSE’. | mmg\_CheckIfCycleStepIsvalid() | ARCH\_SW\_MMG\_0310 |
| DSG\_MMG\_0123 | If cycle number is not in range then the return status of validity check shall be ‘B\_FALSE’. | mmg\_CheckIfCycleStepIsvalid() | ARCH\_SW\_MMG\_0311 |
| DSG\_MMG\_0124 | If step number is not in range then the return status of validity check shall be ‘B\_FALSE’. | mmg\_CheckIfCycleStepIsvalid() | ARCH\_SW\_MMG\_0313 |
| DSG\_MMG\_0125 | For each cycle, if Week or Year are 0xFF, then cycle validity is false. | mmg\_CheckCycleEnable() | ARCH\_SW\_MMG\_0312; |
| DSG\_MMG\_0126 | For each step if step order is different from:   * 0x00: PWM order ; * 0x01: Current order ; * 0x02: Velocity Control order type; * 0x03: Velocity Control + backup PWM order type; * 0x04: Voltage order type; * 0x05: Velocity Control + backup Current order type;   0x06: Velocity Control + backup Voltage order  Then step is invalid. | mmg\_CheckIfCycleStepIsvalid() | ARCH\_SW\_MMG\_0315; |
| DSG\_MMG\_0127 | If step duration is over 655350 ms then the return status of validity check shall be ‘B\_FALSE’. | mmg\_CheckIfCycleStepIsvalid() | ARCH\_SW\_MMG\_0314; |
| DSG\_MMG\_0128 | **mmg\_CheckIgnitionON** local function will compute signal release inhibition in order to compute MMG\_KU32\_MASK\_ENABLE\_IGNITION\_ONcontext. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0320; |
| DSG\_MMG\_0129 | **mmg\_CheckCodingOfAPI** local function will compute signal release inhibition in order to compute MMG\_KU32\_MASK\_ENABLE\_CODING\_API context. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0321; |
| DSG\_MMG\_0130 | **mmg\_CheckCodingRBTMFL** local function will compute signal release inhibition in order to compute MMG\_KU32\_MASK\_ENABLE\_CODING\_RBTMFL context. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0322; |
| DSG\_MMG\_0131 | **mmg\_CheckCodingRBTMFR** local function will compute signal release inhibition in order to compute MMG\_KU32\_MASK\_ENABLE\_CODING\_RBTMFR context. | MMG\_runUpdateModeStatus | ARCH\_SW\_MMG\_0323; |

# FEATURES

## Services

### MMG\_Init

|  |  |  |
| --- | --- | --- |
| Object | | |
| Function is in charge of initialization driver’s variables and parameters | | |
| **Prototype** | | |
| FUNC(void, MMG\_AC\_ModeManagement\_CODE) MMG\_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 | |
| Will be called at system startup | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

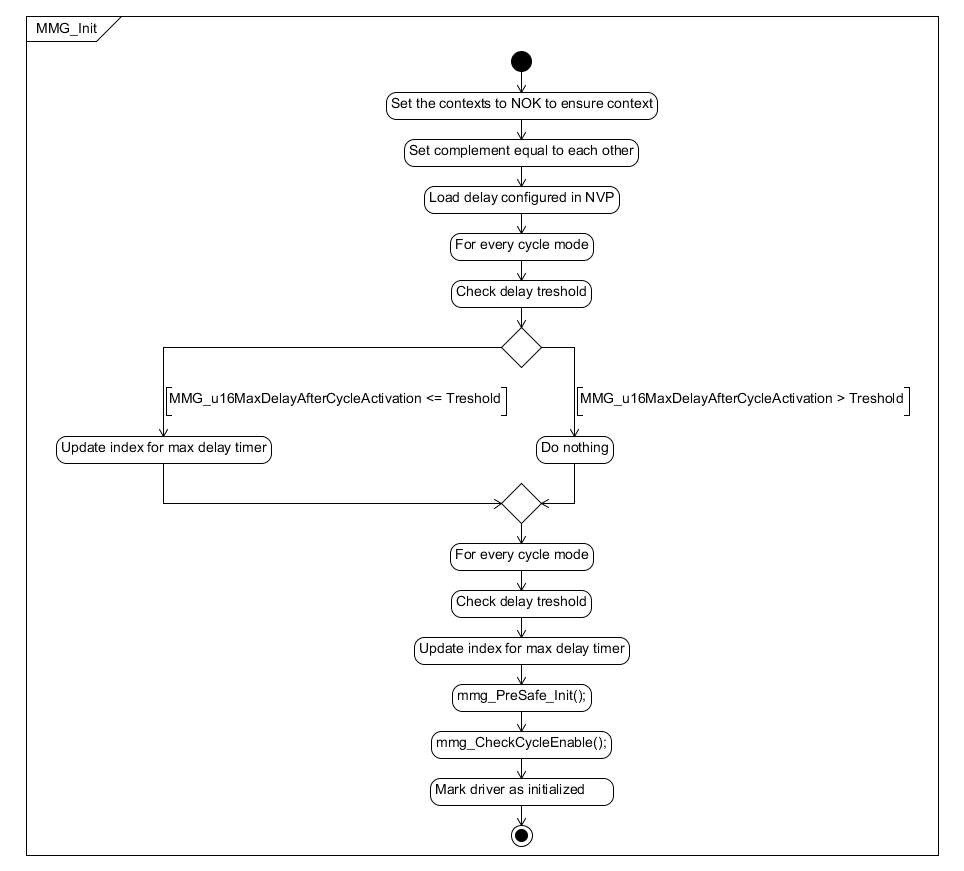


Figure : MMG\_Init diagram

### MMG\_runUpdateModeStatus

|  |  |  |
| --- | --- | --- |
| Object | | |
| This is the MMG main function that refreshes the status of every modes each time it is called. | | |
| **Prototype** | | |
| FUNC(void, MMG\_AC\_ModeManagement\_CODE) MMG\_runUpdateModeStatus(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 | | |

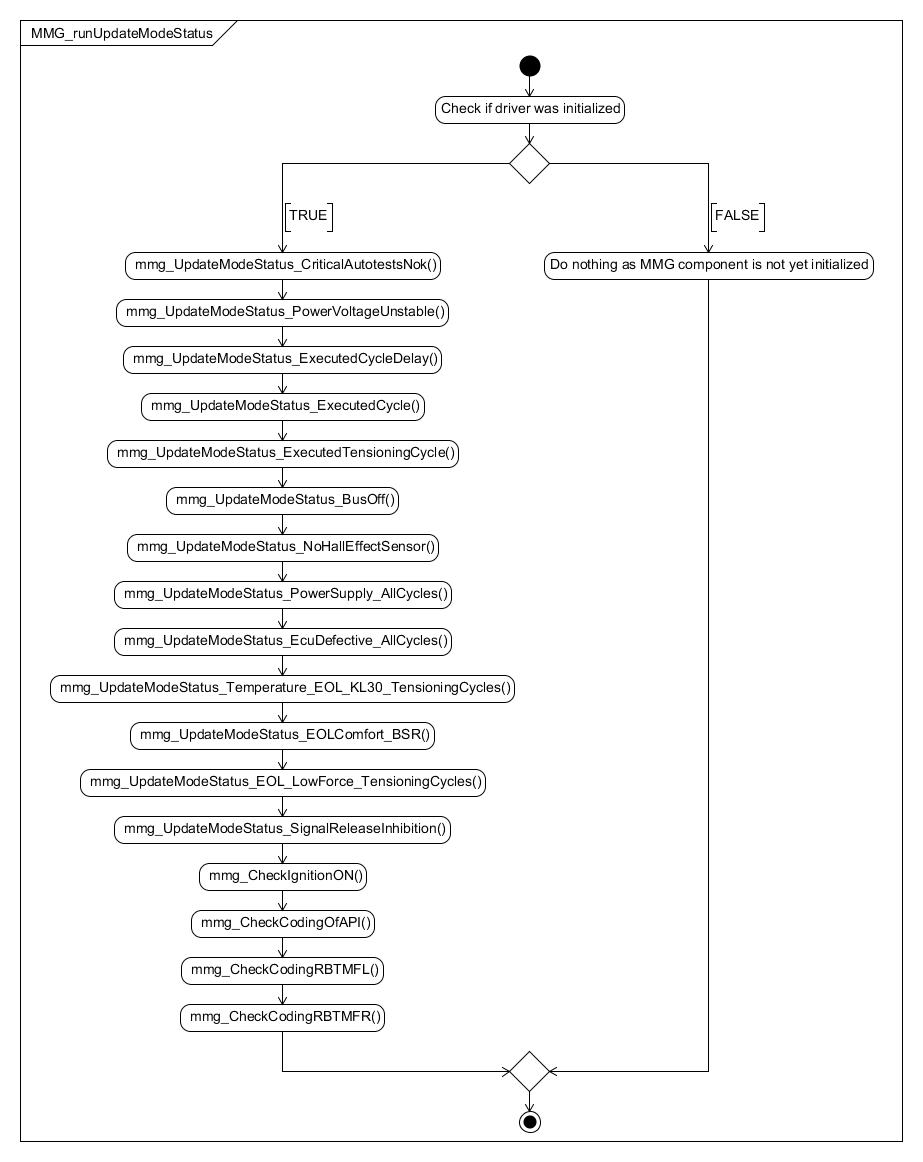


Figure : MMG\_runUpdateModeStatus diagram

### MMG\_runManageModeDelays

|  |  |  |
| --- | --- | --- |
| Object | | |
| This is the MMG function that handles all the timers needed by the MMG modes. | | |
| **Prototype** | | |
| FUNC(void, MMG\_AC\_ModeManagement\_CODE) MMG\_runManageModeDelays(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 : MMG\_runManageModeDelays diagram

### MMG\_runCheckModeStatus

|  |  |  |
| --- | --- | --- |
| Object | | |
| This is the mode checking service. It indicates whether all the required modes are activated. | | |
| **Prototype** | | |
| FUNC(void, MMG\_AC\_ModeManagement\_CODE) MMG\_runCheckModeStatus(u32ModeMaskType u32ModesToCheck, P2VAR(u8ModeStatusType, AUTOMATIC, RTE\_MMG\_AC\_MODEMANAGEMENT\_APPL\_VAR) bModeStatus) | | |
| **Input parameters** | | |
| Name | Type | Description |
| u32ModesToCheck | u32ModeMaskType | Mask of the selected modes |
| **Output parameters** | | |
| Name | Type | Description |
| \* bModeStatus | u8ModeStatusType | Status of the required selected modes |
| **Return value** | | |
| Type | Description | |
| NA | void | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| Called by PAL, PRE, SFR, CIL | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

Diagram, schematic

Description automatically generated

Figure : MMG\_runCheckModeStatus diagram

### mmg\_UpdateModeStatus\_CriticalAutotestsNok

|  |  |  |
| --- | --- | --- |
| Object | | |
| This is the executing mode function in charge of manages the “critical auto tests nok” mode. | | |
| **Prototype** | | |
| void mmg\_UpdateModeStatus\_CriticalAutotestsNok(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 | |
| MMG\_runUpdateModeStatus | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

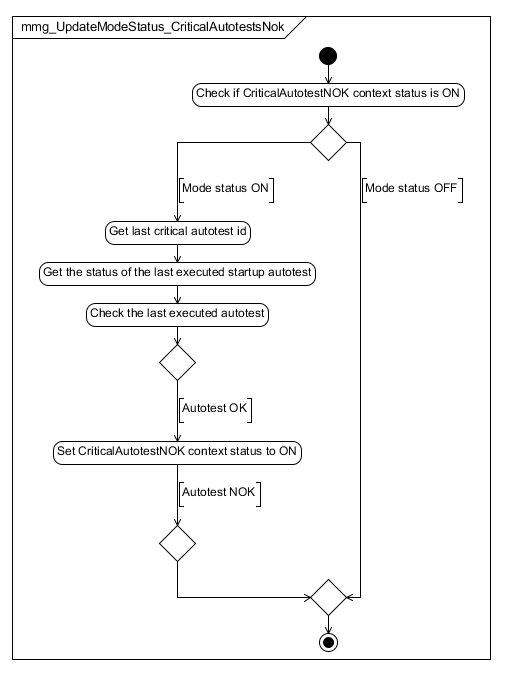


Figure : mmg\_UpdateModeStatus\_CriticalAutotestsNok diagram

### mmg\_UpdateModeStatus\_PowerVoltageUnstable

|  |  |  |
| --- | --- | --- |
| Object | | |
| This is the executing mode function in charge of manages the “power voltage unstable” mode. | | |
| **Prototype** | | |
| void mmg\_UpdateModeStatus\_PowerVoltageUnstable(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 | |
| MMG\_runUpdateModeStatus | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

Diagram

Description automatically generated

Figure : mmg\_UpdateModeStatus\_PowerVoltageUnstable diagram

### mmg\_UpdateModeStatus\_ExecutedCycleDelay

|  |  |  |
| --- | --- | --- |
| Object | | |
| This is the executing mode function in charge of manages the “executed cycle with delay” modes. | | |
| **Prototype** | | |
| void mmg\_UpdateModeStatus\_ExecutedCycleDelay(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 | |
| MMG\_runUpdateModeStatus | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

Diagram

Description automatically generated

Figure : mmg\_UpdateModeStatus\_ExecutedCycleDelay diagram

### mmg\_UpdateModeStatus\_ExecutedCycle

|  |  |  |
| --- | --- | --- |
| Object | | |
| This is the executing mode function in charge of manages the “executed cycle” mode. | | |
| **Prototype** | | |
| void mmg\_UpdateModeStatus\_ExecutedCycle(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 | |
| MMG\_runUpdateModeStatus | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

Diagram

Description automatically generated

Figure : mmg\_UpdateModeStatus\_ExecutedCycle diagram

### mmg\_UpdateModeStatus\_ExecutedTensioningCycle

|  |  |  |
| --- | --- | --- |
| Object | | |
| This is the executing mode function in charge of manages the “executed tensioning cycle” mode. | | |
| **Prototype** | | |
| void mmg\_UpdateModeStatus\_ExecutedTensioningCycle(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 | |
| MMG\_runUpdateModeStatus | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

Diagram, schematic

Description automatically generated

Figure : mmg\_UpdateModeStatus\_ExecutedTensioningCycle diagram

### mmg\_UpdateModeStatus\_BusOff

|  |  |  |
| --- | --- | --- |
| Object | | |
| This is the executing mode function in charge of manages the “bus off” mode. | | |
| **Prototype** | | |
| void mmg\_UpdateModeStatus\_BusOff(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 | |
| MMG\_runUpdateModeStatus | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

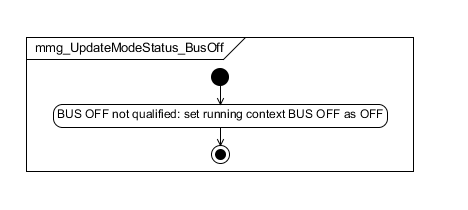


Figure : mmg\_UpdateModeStatus\_BusOff diagram

### mmg\_UpdateModeStatus\_NoHallEffectSensor

|  |  |  |
| --- | --- | --- |
| Object | | |
| This is the executing mode function in charge of manages the “no hall effect sensor” mode. | | |
| **Prototype** | | |
| void mmg\_UpdateModeStatus\_NoHallEffectSensor(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 | |
| MMG\_runUpdateModeStatus | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

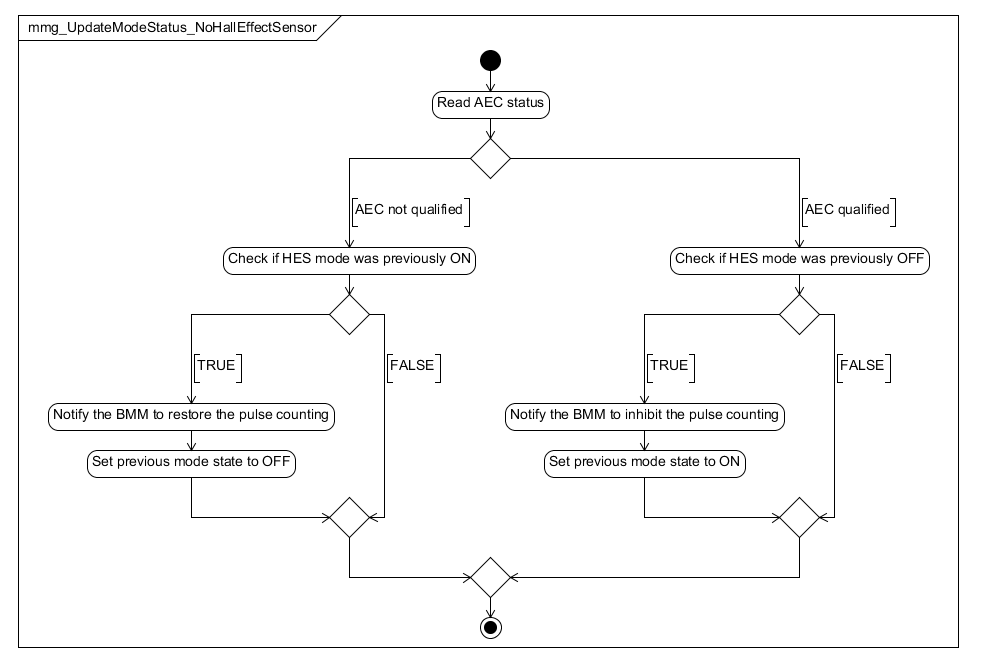


Figure mmg\_UpdateModeStatus\_ NoHallEffectSensor f diagram

### mmg\_UpdateModeStatus\_DefaultValues

|  |  |  |
| --- | --- | --- |
| Object | | |
| This is the executing mode function in charge of manages the “default values” mode. | | |
| **Prototype** | | |
| void mmg\_UpdateModeStatus\_DefaultValues(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 | |
| MMG\_runUpdateModeStatus | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

Diagram, schematic

Description automatically generated

Figure mmg\_UpdateModeStatus\_DefaultValues diagram

### mmg\_UpdateModeStatus\_PowerSupply\_AllCycles

|  |  |  |
| --- | --- | --- |
| Object | | |
| This is the executing mode function in charge of manages the inhibition of all cycles by Power supply | | |
| **Prototype** | | |
| void mmg\_UpdateModeStatus\_PowerSupply\_AllCycles(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 | |
| MMG\_runUpdateModeStatus | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |



Figure mmg\_UpdateModeStatus\_PowerSupply\_AllCycles:

### mmg\_UpdateModeStatus\_EcuDefective\_AllCycles

|  |  |  |
| --- | --- | --- |
| Object | | |
| This is the executing mode function in charge of manages the inhibition of all cycles by ECU defective | | |
| **Prototype** | | |
| void mmg\_UpdateModeStatus\_EcuDefective\_AllCycles(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 | |
| MMG\_runUpdateModeStatus | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |



Figure mmg\_UpdateModeStatus\_EcuDefective\_AllCycles

### mmg\_UpdateModeStatus\_EOLComfort\_BSR

|  |  |  |
| --- | --- | --- |
| Object | | |
| This is the executing mode function in charge of manages the inhibition of BSR by EOL Comfort | | |
| **Prototype** | | |
| void mmg\_UpdateModeStatus\_EOLComfort\_BSR(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 | |
| MMG\_runUpdateModeStatus | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |



Figure mmg\_UpdateModeStatus\_EOLComfort\_BSR

### mmg\_UpdateModeStatus\_ EOL\_LowForce\_TensioningCycles

|  |  |  |
| --- | --- | --- |
| Object | | |
| This is the executing mode function in charge of manages the inhibition of tensioning cycles by : LOW Force Counter | | |
| **Prototype** | | |
| void mmg\_UpdateModeStatus EOL\_LowForce\_TensioningCycles (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 | |
| MMG\_runUpdateModeStatus | \* | |
| **Static aspect** | | |
| NA | | |

Diagram

Description automatically generated

Figure : mmg\_UpdateModeStatus\_ EOL\_LowForce\_TensioningCycles

### mmg\_UpdateModeStatus\_Temperature\_EOL\_KL30\_TensioningCycles

|  |  |  |
| --- | --- | --- |
| Object | | |
| This is the executing mode function in charge of manages the inhibition of tensioning cycles by : Temperature high ,KL30 OV and UV. | | |
| **Prototype** | | |
| void mmg\_UpdateModeStatus\_Temperature\_EOL\_KL30\_TensioningCycles(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 | |
| MMG\_runUpdateModeStatus | \* | |
| **Static aspect** | | |
| NA | | |



Figure : mmg\_UpdateModeStatus\_Temperature\_EOL\_KL30\_TensioningCycles

### mmg\_ManageModeDelays\_ExecutedCycle

|  |  |  |
| --- | --- | --- |
| Object | | |
| This is the function in charge of managing the timers for the Executed Cycle modes. | | |
| **Prototype** | | |
| void mmg\_ManageModeDelays\_ExecutedCycle(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 | |
| MMG\_runManageModeDelays | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

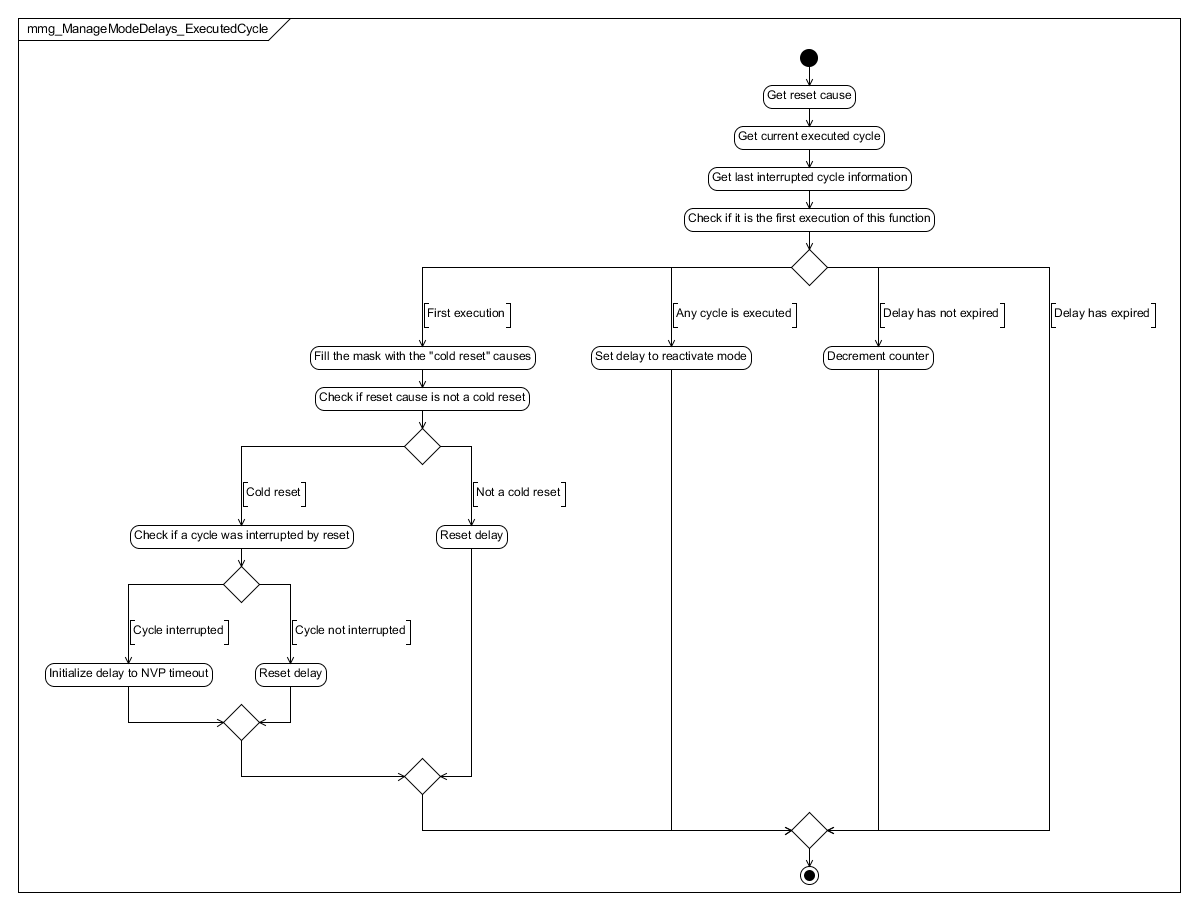


Figure : mmg\_ManageModeDelays\_ExecutedCycle diagram

### mmg\_CheckIfCycleStepIsvalid

|  |  |  |
| --- | --- | --- |
| Object | | |
| This function is in charge of verifying if cycle has valid steps. | | |
| **Prototype** | | |
| LOCAL boolean mmg\_CheckIfCycleStepIsvalid(const uint16 u16Step0) | | |
| **Input parameters** | | |
| Name | Type | Description |
| u16Step0 | uint16 | Step0 from cycle that will be verified |
| **Output parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Return value** | | |
| Type | Description | |
| boolean b8Ret | Function return if a cycle is valid or not. | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| mmg\_CheckCycleEnable | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |



Figure : mmg\_CheckIfCycleStepIsvalid diagram

### mmg\_CheckCycleEnable

|  |  |  |
| --- | --- | --- |
| Object | | |
| This is the executing mode function in charge of manages the “bus off” mode. | | |
| **Prototype** | | |
| LOCAL void mmg\_CheckCycleEnable(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 | |
| MMG\_Init | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |



Figure : mmg\_CheckCycleEnable diagram

### mmg\_PreSafe\_CopyStartData

|  |  |  |
| --- | --- | --- |
| Object | | |
| Function is in charge of copying start data to RAM structure | | |
| **Prototype** | | |
| LOCAL void mmg\_PreSafe\_CopyStartData(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 | |
| mmg\_PreSafe\_HandleStartData | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

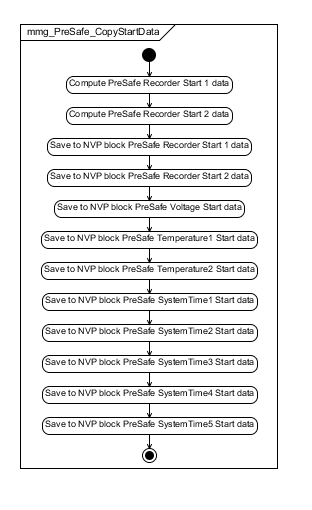


Figure : mmg\_PreSafe\_CopyStartData

### mmg\_PreSafe\_CopyStopData

|  |  |  |
| --- | --- | --- |
| Object | | |
| Function is in charge of copying stop data to RAM structure | | |
| **Prototype** | | |
| LOCAL void mmg\_PreSafe\_CopyStopData(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 | |
| mmg\_PreSafe\_HandleStopData | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

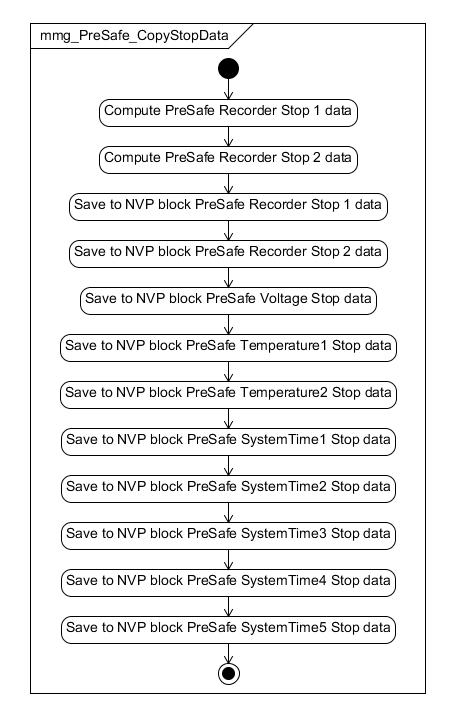


Figure : mmg\_PreSafe\_CopyStopData

### mmg\_PreSafe\_CopyStateData

|  |  |  |
| --- | --- | --- |
| Object | | |
| Function is in charge of copying state data to RAM structure | | |
| **Prototype** | | |
| LOCAL void mmg\_PreSafe\_CopyStateData(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 | |
| mmg\_PreSafe\_HandleStateData | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

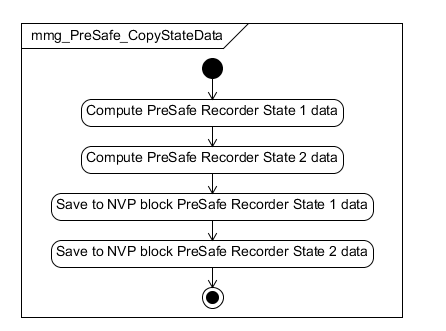


Figure : mmg\_PreSafe\_CopyStateData

### mmg\_PreSafe\_HandleStateData

|  |  |  |
| --- | --- | --- |
| Object | | |
| Function is in charge of handling input data and complete the raw state data to local structure | | |
| **Prototype** | | |
| LOCAL void mmg\_PreSafe\_HandleStateData(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 | |
| MMG\_runPreSafeRecorder | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

Note: Because of the size of the function, the diagram was splitted in two pictures.



Figure : Part1 mmg\_PreSafe\_HandleStateData



Figure : Part2 mmg\_PreSafe\_HandleStateData

### mmg\_PreSafe\_HandleStartData

|  |  |  |
| --- | --- | --- |
| Object | | |
| Function is in charge of handling input data and complete the raw start data to local structure | | |
| **Prototype** | | |
| LOCAL void mmg\_PreSafe\_HandleStartData(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 | |
| MMG\_runPreSafeRecorder | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

Note: Because of the size of the function, the diagram was splitted in two pictures.



Figure : Part1 mmg\_PreSafe\_HandleStartData



Figure : Part2 mmg\_PreSafe\_HandleStartData

### mmg\_PreSafe\_HandleStopData

|  |  |  |
| --- | --- | --- |
| Object | | |
| Function is in charge of handling input data and complete the raw stop data to local structure | | |
| **Prototype** | | |
| LOCAL void mmg\_PreSafe\_HandleStopData(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 | |
| MMG\_runPreSafeRecorder | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

Note: Because of the size of the function, the diagram was splitted in two pictures.



Figure : Part1 mmg\_PreSafe\_HandleStopData



Figure : Part1 mmg\_PreSafe\_HandleStopData

### mmg\_PreSafe\_Init

|  |  |  |
| --- | --- | --- |
| Object | | |
| Function is in charge of initializing the PreSafe structures | | |
| **Prototype** | | |
| FUNC(void, MMG\_AC\_ModeManagement\_CODE) mmg\_PreSafe\_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 | |
| MMG\_Init | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

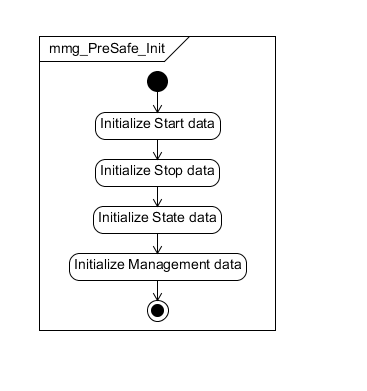


Figure : mmg\_PreSafe\_Init

### MMG\_runPreSafeRecorder

|  |  |  |
| --- | --- | --- |
| Object | | |
| Function is in charge handling the mode management of Presafe recorder | | |
| **Prototype** | | |
| FUNC(void, MMG\_AC\_ModeManagement\_CODE) MMG\_runPreSafeRecorder(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 | |
| Every 10 ms | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

Note: Because of the size of the function, the diagram was splitted in three pictures.



Figure : Part1 MMG\_runPreSafeRecorder



Figure : Part2 MMG\_runPreSafeRecorder



Figure : Part3 MMG\_runPreSafeRecorder

### mmg\_PreSafe\_CheckEOLLimit

|  |  |  |
| --- | --- | --- |
| Object | | |
| Function is in charge of checking EOL counter inhibition status | | |
| **Prototype** | | |
| LOCAL void mmg\_PreSafe\_CheckEOLLimit(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 | | |



Figure : mmg\_PreSafe\_CheckEOLLimit

### mmg\_AvailabilityDataStatus

|  |  |  |
| --- | --- | --- |
| Object | | |
| Function is in charge to handle the AvailabilityDataStatus and route it to DiagFunction | | |
| **Prototype** | | |
| void mmg\_AvailabilityDataStatus(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 | |
| MMG\_runUpdateModeStatus | \* | |
| **Static aspect** | | |
| NA | | |
| **Constrains** | | |
| NA | | |

A diagram of a diagram

Description automatically generated with medium confidence

Figure : mmg\_AvailabilityDataStatus

### mmg\_UpdateModeStatus\_SignalReleaseInhibition

|  |  |  |  |
| --- | --- | --- | --- |
| Object | | | |
| This is the executing mode function in charge of inhibitiing release profile by customer signals. | | | |
| **Prototype** | | | |
| void mmg\_UpdateModeStatus\_SignalReleaseInhibition (void) | | | |
| **Parameters** | | | |
| void | | | |
| **Exceptions** | | | |
| None | | | |
| **Precondition** | | | |
| None | | | |
| **Postcondition** | | | |
| None | | | |
| **Input parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Output parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Return value** | | | |
| Type | Description | | |
| NA | void | | |
| **Dynamic aspect** | | | |
| Who(callers) | Description | | |
| \* | \* | | |
| **Static aspect** | | | |
| \* | | | |
| **Constrains** | | | |
|  | | | |



Figure mmg\_UpdateModeStatus\_SignalReleaseInhibition

### MMG\_runUpdateNVMBlocks

|  |  |  |  |
| --- | --- | --- | --- |
| Object | | | |
| The function checks if a specific NVM block needs written and send a request write for the specific block | | | |
| **Prototype** | | | |
| FUNC(void, MMG\_AC\_ModeManagement\_CODE) MMG\_runUpdateNVMBlocks(void) | | | |
| **Parameters** | | | |
| void | | | |
| **Exceptions** | | | |
| None | | | |
| **Precondition** | | | |
| None | | | |
| **Postcondition** | | | |
| None | | | |
| **Input parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Output parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Return value** | | | |
| Type | Description | | |
| NA | void | | |
| **Dynamic aspect** | | | |
| Who(callers) | Description | | |
| \* | \* | | |
| **Static aspect** | | | |
| \* | | | |
| **Constrains** | | | |
|  | | | |



Figure MMG\_runUpdateNVMBlocks

### mmg\_CheckCodingOfAPI

|  |  |  |  |
| --- | --- | --- | --- |
| Object | | | |
| The function is called to mange CodingOfAPI enable condition context. | | | |
| **Prototype** | | | |
| void mmg\_CheckCodingOfAPI (void) | | | |
| **Parameters** | | | |
| void | | | |
| **Exceptions** | | | |
| None | | | |
| **Precondition** | | | |
| None | | | |
| **Postcondition** | | | |
| None | | | |
| **Input parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Output parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Return value** | | | |
| Type | Description | | |
| NA | void | | |
| **Dynamic aspect** | | | |
| Who(callers) | Description | | |
| \* | \* | | |
| **Static aspect** | | | |
| \* | | | |
| **Constrains** | | | |
|  | | | |



Figure Check coding of API

### mmg\_CheckCodingRBTMFL

|  |  |  |  |
| --- | --- | --- | --- |
| Object | | | |
| The function is called to mange CodingRBTMFL enable condition context. | | | |
| **Prototype** | | | |
| void mmg\_CheckCodingRBTMFL (void) | | | |
| **Parameters** | | | |
| void | | | |
| **Exceptions** | | | |
| None | | | |
| **Precondition** | | | |
| None | | | |
| **Postcondition** | | | |
| None | | | |
| **Input parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Output parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Return value** | | | |
| Type | Description | | |
| NA | void | | |
| **Dynamic aspect** | | | |
| Who(callers) | Description | | |
| \* | \* | | |
| **Static aspect** | | | |
| \* | | | |
| **Constrains** | | | |
|  | | | |



Figure Check coding of RBTMFL

### mmg\_CheckCodingRBTMFR

|  |  |  |  |
| --- | --- | --- | --- |
| Object | | | |
| The function is called to mange CodingRBTMFR enable condition context. | | | |
| **Prototype** | | | |
| void mmg\_CheckCodingRBTMFR (void) | | | |
| **Parameters** | | | |
| void | | | |
| **Exceptions** | | | |
| None | | | |
| **Precondition** | | | |
| None | | | |
| **Postcondition** | | | |
| None | | | |
| **Input parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Output parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Return value** | | | |
| Type | Description | | |
| NA | void | | |
| **Dynamic aspect** | | | |
| Who(callers) | Description | | |
| \* | \* | | |
| **Static aspect** | | | |
| \* | | | |
| **Constrains** | | | |
|  | | | |



Figure Check coding of RBTMFR

### mmg\_CheckIgnitionON

|  |  |  |  |
| --- | --- | --- | --- |
| Object | | | |
| The function is called to mange IgnitionON enable condition context. | | | |
| **Prototype** | | | |
| void mmg\_CheckIgnitionON (void) | | | |
| **Parameters** | | | |
| void | | | |
| **Exceptions** | | | |
| None | | | |
| **Precondition** | | | |
| None | | | |
| **Postcondition** | | | |
| None | | | |
| **Input parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Output parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Return value** | | | |
| Type | Description | | |
| NA | void | | |
| **Dynamic aspect** | | | |
| Who(callers) | Description | | |
| \* | \* | | |
| **Static aspect** | | | |
| \* | | | |
| **Constrains** | | | |
|  | | | |



Figure Check Ignition status

## Types

### MMG\_stModeDelayType

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Field description** |
| u16DelayThrs | uint16 | Holds the value of delay to wait after cycle activation |
| u32AssociatedMode | uint32 | Holds the mask of the associated MMG mode |

### MMG\_stModeConfigurationType

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Field description** |
| pfModeCallback | tSystemContextCallbackFct | Pointer to function callback |

### MMG\_KU8\_IDX

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Field description** |
| MMG\_KU8\_IDX\_HWSELFPROT | enumeration type (0) | Index for HW Self Protection Autotest |
| MMG\_KU8\_IDX\_MOTORCURRENT | enumeration type (1) | Index for Motor Current Autotest |
| MMG\_KU8\_IDX\_MOTORTEMP | enumeration type (2) | Index for Motor Thermal Protection Autotest |
| MMG\_KU8\_IDX\_UNDERVOLTAGE | enumeration type (3) | Index for Undervoltage Autotest |
| MMG\_KU8\_IDX\_OVERVOLTAGE | enumeration type (4) | Index for Overvoltage Autotest |
| MMG\_KU8\_IDX\_UNDERVOLTAGE\_TENS | enumeration type (5) | Index for Undervoltage During Activation Autotest |
| MMG\_KU8\_IDX\_OVERVOLTAGE\_TENS | enumeration type (6) | Index for Overvoltage During Activation Autotest |
| MMG\_KU8\_NR\_OF\_AUTOTESTS | enumeration type (7) | Total number of autotests used |

### mmg\_stPreSafeRecorderStartData

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Field description** |
| Start1BuckleSwitch | uint8 : 2 | To store the Start1 Buckle Switch |
| Start1BeltHandOver | uint8 : 2 | To store the Start1 Belt Hand Over |
| Start2PreSafeLvl | uint8 : 3 | To store the Start2 PreSafe Lvl |
| Start2ImpactX | uint8 : 1 | To store the Start2 ImpactX |
| Start2RollOverType1 | uint8 : 1 | To store the Start2 RollOverType1 |
| Start2RollOverType2 | uint8 : 1 | To store the Start2 RollOverType2 |
| StartVoltage | uint8 : 8 | To store the Start Voltage |
| StartTemperature1 | uint8 : 8 | To store the Start Temperature1 |
| StartTemperature2 | uint8 : 8 | To store the Start Temperature2 |
| StartSystemTime1 | uint8 : 8 | To store the Start SystemTime1 |
| StartSystemTime2 | uint8 : 8 | To store the Start SystemTime2 |
| StartSystemTime3 | uint8 : 8 | To store the Start SystemTime3 |
| StartSystemTime4 | uint8 : 8 | To store the Start SystemTime4 |
| StartSystemTime5 | uint8 : 8 | To store the Start SystemTime5 |

### mmg\_stPreSafeRecorderStopData

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Field description** |
| Stop1BuckleSwitch | uint8 : 2 | To store the Stop1 Buckle Switch |
| Stop1BeltHandOver | uint8 : 2 | To store the Stop1 Belt Hand Over |
| Stop2PreSafeLvl | uint8 : 3 | To store the Stop2 PreSafe Lvl |
| Stop2ImpactX | uint8 : 1 | To store the Stop2 ImpactX |
| Stop2RollOverType1 | uint8 : 1 | To store the Stop2 RollOverType1 |
| Stop2RollOverType2 | uint8 : 1 | To store the Stop2 RollOverType2 |
| StopVoltage | uint8 : 8 | To store the Stop Voltage |
| StopTemperature1 | uint8 : 8 | To store the Stop Temperature1 |
| StopTemperature2 | uint8 : 8 | To store the Stop Temperature2 |
| StopSystemTime1 | uint8 : 8 | To store the Stop SystemTime1 |
| StopSystemTime2 | uint8 : 8 | To store the Stop SystemTime2 |
| StopSystemTime3 | uint8 : 8 | To store the Stop SystemTime3 |
| StopSystemTime4 | uint8 : 8 | To store the Stop SystemTime4 |
| StopSystemTime5 | uint8 : 8 | To store the Stop SystemTime5 |

### mmg\_stPreSafeRecorderStateData

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Field description** |
| State1TensioningState | uint8 : 2 | To store the State1 Tensioning State |
| State1PreSafeDisabled | uint8 : 1 | To store the State1 PreSafe disabled |
| State1PreSafeSupp | uint8 : 1 | To store the State1 PreSafe suppresion |
| State1PreSafeAbort | uint8 : 1 | To store the State1 PreSafe abortion |
| State1KL15 | uint8 : 1 | To store the State1 KL15 |
| State1UvOROV | uint8 : 1 | To store the State1 Overvoltage or Undervoltahe |
| State2OverTemp | uint8 : 1 | To store the State2 Overtemperature |
| State2PreSafeCounter | uint8 : 1 | To store the State2 EOL PreSafe Counter |
| State2SystemFailure | uint8 : 1 | To store the State2 System/Component failure |

### mmg\_ stPreSafeRecorderManagement

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Field description** |
| PreSafeState | uint8 | To store the PreSafe state |
| PreSafePreviousState | uint8 | To store the PreSafe previous state |
| PreSafeWriteTrigger | uint8 | To store the PreSafe write trigger |
| PreSafeBufferID | uint8 | To store the PreSafe ID to be written |
| StartedProfile | uint8 | To store the started profile |
| ProfileInterrupedStatus | uint8 | To store the status of a new profile interrupted |
| NewProfileExecuted | uint8 | To store the new profile interrupted |

## Variables

### mmg\_b8MMGHAsBeenInitialized

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| Boolean | 0x55 | |
| **Description** | | |
| Variables is used to detect if MMG component has been initialized | | |
| **Definition** | | |
| static boolean MMG\_b8MMGHAsBeenInitialized | | |

### MMG\_u32ModesStatus

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| uint32 | N.A. | |
| **Description** | | |
| Variable is used to set a running context | | |
| **Definition** | | |
| uint32 MMG\_u32ModesStatus | | |

### MMG\_u32ModesStatusComplement

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| uint32 | N.A. | |
| **Description** | | |
| Variable is used for processing information when set a running context | | |
| **Definition** | | |
| uint32 MMG\_u32ModesStatusComplement | | |

### mmg\_u16DelayToClearModeAfterCycle

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| uint16 | 0xFFFF | |
| **Description** | | |
| Variable represents the counter used to manage timeout for "Executed cycle" mode | | |
| **Definition** | | |
| static uint16 MMG\_u16DelayToClearModeAfterCycle | | |

### MMG\_u16MaxDelayAfterCycleActivation

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| uint16 | 0x00 | |
| **Description** | | |
| Variable which will be loaded, during MMG\_Init, with the maximal value of delays configured in NVP | | |
| **Definition** | | |
| uint16 MMG\_u16MaxDelayAfterCycleActivation | | |

### MMG\_astDelayAfterCycleThrsandModes

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| \* stModeDelayType | N.A. | |
| **Description** | | |
| Vector variable that represents the table defining delays associated to cycle execution MMG modes | | |
| **Definition** | | |
| stModeDelayType MMG\_astDelayAfterCycleThrsandModes[KU8\_NUMBER\_OF\_MMG\_CYCLES\_MODES\_DELAYS] | | |

### mmg\_u8PrevHESModeStatus

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| uint8 | 0x85 | |
| **Description** | | |
| Variable that stores the previous status of Hall Effect Sensor mode | | |
| **Definition** | | |
| static uint8 MMG\_u8PrevHESModeStatus | | |

### mmg\_stPreSafeManagement

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| mmg\_stPreSafeRecorderManagement | { KU8\_FALSE, KU8\_NO\_CYCLE, KU8\_NO\_CYCLE, KU8\_TENSIONING\_NONE, KU8\_TENSIONING\_NONE, MMG\_KU8\_PRESF\_TRIGGER\_OFF, NVP\_BLOCK\_ID\_PRE\_SAFE\_RECORDER\_ID\_RamBlockData.NVP\_u8PreSafeRecorderID } | |
| **Description** | | |
| Variable that stores the PreSafe Recorder management data | | |
| **Definition** | | |
| static mmg\_stPreSafeRecorderManagement mmg\_stPreSafeManagement | | |

### mmg\_stPreSafeStartData

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| mmg\_stPreSafeRecorderStartData | {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00} | |
| **Description** | | |
| Variable that stores the PreSafe Recorder start block data | | |
| **Definition** | | |
| static mmg\_stPreSafeRecorderStartData mmg\_stPreSafeStartData | | |

### mmg\_stPreSafeStopData

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| mmg\_stPreSafeRecorderStopData | {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00} | |
| **Description** | | |
| Variable that stores the PreSafe Recorder stop block data | | |
| **Definition** | | |
| static mmg\_stPreSafeRecorderStopData mmg\_stPreSafeStopData | | |

### mmg\_stPreSafeStateData

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| mmg\_stPreSafeRecorderStateData | {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00} | |
| **Description** | | |
| Variable that stores the PreSafe Recorder state block data | | |
| **Definition** | | |
| static mmg\_stPreSafeRecorderStateData mmg\_stPreSafeStateData | | |

### mmg\_stPreSafeInputData

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| REC\_PreSafeRecInputType | {KU8\_TENSIONING\_NONE,I\_C02\_BCKL\_SW\_STAT\_SNA,I\_C04\_PRESF\_LVL\_V2\_NOLVL,KU8\_ZERO,I\_C03\_NO\_LT\_RT\_BOTH\_NO,KU8\_ZERO,KU8\_ZERO,I\_C03\_ISW\_STAT\_IGN\_LOCK, I\_C02\_BELTHDOVR\_STAT\_SNA} | |
| **Description** | | |
| Variable that stores the Pre Safe input data received from CIL module | | |
| **Definition** | | |
| static REC\_PreSafeRecInputType mmg\_stPreSafeInputData | | |

### mmg\_u32DeficiencyLevel

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| u32DeficiencyLevelType | 0x00 | |
| **Description** | | |
| Variable that stores the deficiency level | | |
| **Definition** | | |
| static u32DeficiencyLevelType mmg\_u32DeficiencyLevel | | |

### mmg\_u16LocalVoltage

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| u16VoltageType | 0x00 | |
| **Description** | | |
| Variable that stores the Local voltage received from PMP module | | |
| **Definition** | | |
| static u16VoltageType mmg\_u16LocalVoltage | | |

### mmg\_TimeStamp

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| StbM\_TimeStampType | {0x00, 0x00, 0x00, 0x00} | |
| **Description** | | |
| Variable that stores the time stampe data | | |
| **Definition** | | |
| static StbM\_TimeStampType mmg\_TimeStamp | | |

### mmg\_TimeUserData

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| StbM\_UserDataType | {0x00, 0x00, 0x00, 0x00} | |
| **Description** | | |
| Variable that stores the user time stampe data | | |
| **Definition** | | |
| static StbM\_UserDataType mmg\_TimeUserData | | |

### mmg\_u8Return

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| Std\_ReturnType | 0x01 | |
| **Description** | | |
| Variable that stores return value for time stampe RTE call | | |
| **Definition** | | |
| static Std\_ReturnType mmg\_u8Return | | |

### u8oldAvailabilityDataCategory

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| Uint8 | MMG\_STATUS\_IS\_DEFAULT (0x01) | |
| **Description** | | |
| Variable that stores initial value of MMG status for the Availability Data Calculations | | |
| **Definition** | | |
| static uint8 u8oldAvailabilityDataCategory | | |

### MMG\_u16EnableCondition\_Status

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| uint8 | KU8\_ZERO | |
| **Description** | | |
| This variable holds the status of the profile inhibition causes. | | |
| **Definition** | | |
| uint8 MMG\_u16EnableCondition\_Status = KU8\_ZERO | | |
| **Remarks** | | |
| None. | | |

## Macros

### MMG\_KU8\_CYCLE\_MODE\_DELAY\_INDEX

|  |  |
| --- | --- |
| Name | Value |
| Cycle Mode Delay Index | 0 |
| **Definition** | |
| #define MMG\_KU8\_CYCLE\_MODE\_DELAY\_INDEX | |
| **Description** | |
| indicates the index value corresponding to the MMG cycle modes which needs delay in table astDelayAfterCycleThrsandModes | |

### MMG\_KU8\_CYCLE\_MODE\_DELAY\_UV\_INDEX

|  |  |
| --- | --- |
| Name | Value |
| Cycle Mode Delay UV Index | 1 |
| **Definition** | |
| #define MMG\_KU8\_CYCLE\_MODE\_DELAY\_UV\_INDEX | |
| **Description** | |
| indicates the index value corresponding to the MMG cycle modes which needs delay in table astDelayAfterCycleThrsandModes | |

### MMG\_KU8\_NUMBER\_OF\_CYCLES\_MODES\_DELAYS

|  |  |
| --- | --- |
| Name | Value |
| Number of cycle mode delays | 2 |
| **Definition** | |
| #define MMG\_KU8\_NUMBER\_OF\_CYCLES\_MODES\_DELAYS | |
| **Description** | |
| indicates the index value corresponding to the MMG cycle modes which needs delay in table astDelayAfterCycleThrsandModes | |

### MMG\_U32\_INITIAL\_VALUES\_FOR\_ALL\_MODES

|  |  |
| --- | --- |
| Name | Value |
| Initial value for all modes | (MMG\_KU32\_MASK\_CRITICAL\_AT\_NOK |MMG\_KU32\_MASK\_AEC\_INHIB\_ALL\_CYCLES  | MMG\_KU32\_MASK\_AEC\_INHIB\_TENSIONING\_CYCLES) |
| **Definition** | |
| #define MMG\_U32\_INITIAL\_VALUES\_FOR\_ALL\_MODES | |
| **Description** | |
| This constant is used to initialize the modes | |

### MMG\_KU8\_CFG\_LAST\_PRE\_TENSIONING\_CYCLE

|  |  |
| --- | --- |
| Name | Value |
| Last Pre-Tensioning Cycle | KU8\_BELT\_FUNCTION\_17 |
| **Definition** | |
| #define MMG\_KU8\_CFG\_LAST\_PRE\_TENSIONING\_CYCLE | |
| **Description** | |
| Internal macro for a belt function | |

### MMG\_KU8\_CFG\_BSR1\_TENSIONING\_CYCLE

|  |  |
| --- | --- |
| Name | Value |
| Belt Slack Reduction cycle 1 | KU8\_BELT\_FUNCTION\_26 |
| **Definition** | |
| #define MMG\_KU8\_CFG\_BSR1\_TENSIONING\_CYCLE | |
| **Description** | |
| Internal macro for a belt function | |

### MMG\_KU8\_CFG\_HAPTIC\_WARNING\_10\_CYCLE

|  |  |
| --- | --- |
| Name | Value |
| Haptic Warning 10 cycle | KU8\_BELT\_FUNCTION\_18 |
| **Definition** | |
| #define MMG\_KU8\_CFG\_HAPTIC\_WARNING\_10\_CYCLE | |
| **Description** | |
| Internal macro for a belt function | |

### MMG\_KU8\_CFG\_HAPTIC\_WARNING\_41\_CYCLE

|  |  |
| --- | --- |
| Name | Value |
| Haptic Warning 41 cycle | KU8\_BELT\_FUNCTION\_19 |
| **Definition** | |
| #define MMG\_KU8\_CFG\_HAPTIC\_WARNING\_41\_CYCLE | |
| **Description** | |
| Internal macro for a belt function | |

### MMG\_KU8\_CFG\_PRODUCTION\_0\_CYCLE

|  |  |
| --- | --- |
| Name | Value |
| Production 0 cycle | KU8\_BELT\_FUNCTION\_33 |
| **Definition** | |
| #define MMG\_KU8\_CFG\_PRODUCTION\_0\_CYCLE | |
| **Description** | |
| Internal macro for a belt function | |

### MMG\_KU8\_CFG\_PRODUCTION\_1\_CYCLE

|  |  |
| --- | --- |
| Name | Value |
| Production 1 cycle | KU8\_BELT\_FUNCTION\_34 |
| **Definition** | |
| #define MMG\_KU8\_CFG\_PRODUCTION\_1\_CYCLE | |
| **Description** | |
| Internal macro for a belt function | |

### MMG\_KU16\_TIMEOUT\_INIT\_VALUE

|  |  |
| --- | --- |
| Name | Value |
| Timeout initial value | 0xFFFF |
| **Definition** | |
| #define MMG\_KU16\_TIMEOUT\_INIT\_VALUE | |
| **Description** | |
| Init value of delay for MMG\_KU32\_MASK\_BELTFUNCTIONS\_DELAY and MMG\_KU32\_MASK\_BELTFUNCTIONS\_DELAY\_UV modes | |

### M\_SET\_MODE\_MASK\_OFF

|  |  |
| --- | --- |
| Name | Value |
| Mode mask set off | mask |
| **Definition** | |
| #define M\_SET\_MODE\_MASK\_OFF(mask) \  U32\_BIT\_CLEAR (MMG\_u32ModesStatus, (mask)); \  U32\_BIT\_SET (MMG\_u32ModesStatusComplement, (mask)) | |
| **Description** | |
| NA | |

### M\_SET\_MODE\_MASK\_ON

|  |  |
| --- | --- |
| Name | Value |
| Mode mask set on | mask |
| **Definition** | |
| #define M\_SET\_MODE\_MASK\_ON(mask) \  U32\_BIT\_CLEAR (MMG\_u32ModesStatus, (mask)); \  U32\_BIT\_SET (MMG\_u32ModesStatusComplement, (mask)) | |
| **Description** | |
| NA | |

### MMG\_KU8\_NB\_INHIB\_AEC\_ALL\_CYCLES

|  |  |
| --- | --- |
| Name | Value |
| Number of Auto tests which are able to inhibit cycles | 0x02 |
| **Definition** | |
| #define MMG\_KU8\_NB\_INHIB\_AEC\_ALL\_CYCLES | |
| **Description** | |
| NA | |

### MMG\_KU8\_IDX\_POWERSUPPLY\_OV

|  |  |
| --- | --- |
| Name | Value |
| Index for OV autotest | 0x00 |
| **Definition** | |
| #define MMG\_KU8\_IDX\_POWERSUPPLY\_OV | |
| **Description** | |
| NA | |

### MMG\_KU8\_IDX\_POWERSUPPLY\_UV

|  |  |
| --- | --- |
| Name | Value |
| Index for UV autotest | 0x01 |
| **Definition** | |
| #define MMG\_KU8\_IDX\_POWERSUPPLY\_UV | |
| **Description** | |
| NA | |

### MMG\_KU8\_NB\_OF\_INHIB\_AEC\_FOR\_TENS\_CYCLES

|  |  |
| --- | --- |
| Name | Value |
| Number of Aecs which are able to inhibit and abort TENSIONING cycles | 0x04 |
| **Definition** | |
| #define MMG\_KU8\_NB\_OF\_INHIB\_AEC\_FOR\_TENS\_CYCLES | |
| **Description** | |
| NA | |

### MMG\_KU8\_IDX\_TEMPERATURE\_HIGH

|  |  |
| --- | --- |
| Name | Value |
| Index for Temperature high autotest | 0x00 |
| **Definition** | |
| #define MMG\_KU8\_IDX\_TEMPERATURE\_HIGH | |
| **Description** | |
| NA | |

### MMG\_KU8\_IDX\_EOL\_LOW\_AND\_HIGH

|  |  |
| --- | --- |
| Name | Value |
| Index for EOL low and high autotest | 0x01 |
| **Definition** | |
| #define MMG\_KU8\_IDX\_EOL\_LOW\_AND\_HIGH | |
| **Description** | |
| NA | |

### MMG\_KU8\_IDX\_KL30\_UV\_TENS

|  |  |
| --- | --- |
| Name | Value |
| Index for KL30 UV tensioning autotest | 0x02 |
| **Definition** | |
| #define MMG\_KU8\_IDX\_KL30\_UV\_TENS | |
| **Description** | |
| NA | |

### MMG\_KU8\_IDX\_KL30\_OV\_TENS

|  |  |
| --- | --- |
| Name | Value |
| Index for KL30 OV tensioning autotest | 0x03 |
| **Definition** | |
| #define MMG\_KU8\_IDX\_KL30\_OV\_TENS | |
| **Description** | |
| NA | |

### MMG\_KU8\_PRESF\_TRIGGER\_OFF

|  |  |
| --- | --- |
| Name | Value |
| PreSafe trigger state off | 85 |
| **Definition** | |
| #define MMG\_KU8\_PRESF\_TRIGGER\_OFF | |
| **Description** | |
| NA | |

### MMG\_KU8\_PRESF\_TRIGGER\_ON

|  |  |
| --- | --- |
| Name | Value |
| PreSafe trigger state on | 170 |
| **Definition** | |
| #define MMG\_KU8\_PRESF\_TRIGGER\_ON | |
| **Description** | |
| NA | |

### MMG\_KU8\_State\_DTC\_Is\_Not\_Set

|  |  |
| --- | --- |
| Name | Value |
| PreSafe state DTC is not set | 0x00 |
| **Definition** | |
| #define MMG\_KU8\_State\_DTC\_Is\_Not\_Set | |
| **Description** | |
| NA | |

### MMG\_KU8\_State\_DTC\_Is\_Set

|  |  |
| --- | --- |
| Name | Value |
| PreSafe state DTC is set | 0x01 |
| **Definition** | |
| #define MMG\_KU8\_State\_DTC\_Is\_Set | |
| **Description** | |
| NA | |

### MMG\_KU8\_Tensioning\_Successfull

|  |  |
| --- | --- |
| Name | Value |
| PreSafe tensioning is successfull | 0x00 |
| **Definition** | |
| #define MMG\_KU8\_Tensioning\_Successfull | |
| **Description** | |
| NA | |

### MMG\_KU8\_Tensioning\_NotStarted

|  |  |
| --- | --- |
| Name | Value |
| PreSafe tensioning is not started | 0x01 |
| **Definition** | |
| #define MMG\_KU8\_Tensioning\_NotStarted | |
| **Description** | |
| NA | |

### MMG\_KU8\_Tensioning\_Abort

|  |  |
| --- | --- |
| Name | Value |
| PreSafe tensioning is aborted | 0x02 |
| **Definition** | |
| #define MMG\_KU8\_Tensioning\_Abort | |
| **Description** | |
| NA | |

### MMG\_KU8\_Tensioning\_NotDefined

|  |  |
| --- | --- |
| Name | Value |
| PreSafe tensioning is not defined | 0x03 |
| **Definition** | |
| #define MMG\_KU8\_Tensioning\_NotDefined | |
| **Description** | |
| NA | |

### MMG\_KU8\_SHIFT\_BELT\_HAND\_OVER

|  |  |
| --- | --- |
| Name | Value |
| PreSafe shift index for belt hand over | 0x02 |
| **Definition** | |
| #define MMG\_KU8\_SHIFT\_BELT\_HAND\_OVER | |
| **Description** | |
| NA | |

### MMG\_KU8\_SHIFT\_IMPACTX

|  |  |
| --- | --- |
| Name | Value |
| PreSafe shift index for Impact X | 0x04 |
| **Definition** | |
| #define MMG\_KU8\_SHIFT\_IMPACTX | |
| **Description** | |
| NA | |

### MMG\_KU8\_SHIFT\_ROLLOVERTYPE1

|  |  |
| --- | --- |
| Name | Value |
| PreSafe shift index for RollOvert Type 1 | 0x06 |
| **Definition** | |
| #define MMG\_KU8\_SHIFT\_ROLLOVERTYPE1 | |
| **Description** | |
| NA | |

### MMG\_KU8\_SHIFT\_ROLLOVERTYPE2

|  |  |
| --- | --- |
| Name | Value |
| PreSafe shift index for RollOvert Type 2 | 0x07 |
| **Definition** | |
| #define MMG\_KU8\_SHIFT\_ROLLOVERTYPE2 | |
| **Description** | |
| NA | |

### MMG\_KU8\_SHIFT\_PRESAFE\_DISABLED

|  |  |
| --- | --- |
| Name | Value |
| PreSafe shift index for PreSafe disabled | 0x02 |
| **Definition** | |
| #define MMG\_KU8\_SHIFT\_PRESAFE\_DISABLED | |
| **Description** | |
| NA | |

### MMG\_KU8\_SHIFT\_PRESAFE\_SUPP

|  |  |
| --- | --- |
| Name | Value |
| PreSafe shift index for PreSafe suppresion | 0x03 |
| **Definition** | |
| #define MMG\_KU8\_SHIFT\_PRESAFE\_SUPP | |
| **Description** | |
| NA | |

### MMG\_KU8\_SHIFT\_PRESAFE\_ABORT

|  |  |
| --- | --- |
| Name | Value |
| PreSafe shift index for PreSafe abortion | 0x04 |
| **Definition** | |
| #define MMG\_KU8\_SHIFT\_PRESAFE\_ABORT | |
| **Description** | |
| NA | |

### MMG\_KU8\_SHIFT\_KL15OFF

|  |  |
| --- | --- |
| Name | Value |
| PreSafe shift index for PreSafe KL 15 0ff | 0x05 |
| **Definition** | |
| #define MMG\_KU8\_SHIFT\_KL15OFF | |
| **Description** | |
| NA | |

### MMG\_KU8\_SHIFT\_OV\_OR\_UV

|  |  |
| --- | --- |
| Name | Value |
| PreSafe shift index for PreSafe OV or UV | 0x06 |
| **Definition** | |
| #define MMG\_KU8\_SHIFT\_OV\_OR\_UV | |
| **Description** | |
| NA | |

### MMG\_KU8\_SHIFT\_PRESAFE\_COUNTER

|  |  |
| --- | --- |
| Name | Value |
| PreSafe shift index for PreSafe counter | 0x01 |
| **Definition** | |
| #define MMG\_KU8\_SHIFT\_PRESAFE\_COUNTER | |
| **Description** | |
| NA | |

### MMG\_KU8\_SHIFT\_SYSTEM\_FAILURE

|  |  |
| --- | --- |
| Name | Value |
| PreSafe shift index for PreSafe system failure | 0x02 |
| **Definition** | |
| #define MMG\_KU8\_SHIFT\_SYSTEM\_FAILURE | |
| **Description** | |
| NA | |

### MMG\_KU8\_VOLTAGE\_DIV

|  |  |
| --- | --- |
| Name | Value |
| PreSafe local voltage divisor | 100 |
| **Definition** | |
| #define MMG\_KU8\_VOLTAGE\_DIV | |
| **Description** | |
| NA | |

### MMG\_KU16\_VOLTAGE\_MAX\_VAL

|  |  |
| --- | --- |
| Name | Value |
| PreSafe local voltage max value | 254 |
| **Definition** | |
| #define MMG\_KU16\_VOLTAGE\_MAX\_VAL | |
| **Description** | |
| NA | |

### MMG\_KU32\_DEF\_LVL\_LOW\_THRS

|  |  |
| --- | --- |
| Name | Value |
| Deficiency level low threshold | 1200 |
| **Definition** | |
| #define MMG\_KU32\_DEF\_LVL\_LOW\_THRS | |
| **Description** | |
| NA | |

### MMG\_KU32\_DEF\_LVL\_MID\_THRS

|  |  |
| --- | --- |
| Name | Value |
| Deficiency level medium threshold | 32000 |
| **Definition** | |
| #define MMG\_KU32\_DEF\_LVL\_MID\_THRS | |
| **Description** | |
| NA | |

### MMG\_KU32\_DEF\_LVL\_HIGH\_THRS

|  |  |
| --- | --- |
| Name | Value |
| Deficiency level high threshold | 150000 |
| **Definition** | |
| #define MMG\_KU32\_DEF\_LVL\_HIGH\_THRS | |
| **Description** | |
| NA | |

### MMG\_KU32\_DEF\_LVL\_HOT\_THRS

|  |  |
| --- | --- |
| Name | Value |
| Deficiency level hot threshold | 160000 |
| **Definition** | |
| #define MMG\_KU32\_DEF\_LVL\_HOT\_THRS | |
| **Description** | |
| NA | |

### MMG\_KU16\_SEC\_TO\_MILI

|  |  |
| --- | --- |
| Name | Value |
| PreSafe seconds to miliseconds | 1000 |
| **Definition** | |
| #define MMG\_KU16\_SEC\_TO\_MILI | |
| **Description** | |
| NA | |

### MMG\_U16\_CONVERT\_TEMP\_TO\_RAW

|  |  |
| --- | --- |
| Name | Value |
| PreSafe convert raw value to physical value | (uint16)((uint32)((uint32)((sint32)((PMP\_TEMP)\*MMG\_KU16\_TEMPERATURE\_MUL10) + MMG\_KU16\_TEMPERATURE\_OFFSET) \* MMG\_KU16\_TEMPERATURE\_MUL1000) / MMG\_KU16\_TEMPERATURE\_FACTOR) |
| **Definition** | |
| #define MMG\_KU16\_SEC\_TO\_MILI | |
| **Description** | |
| NA | |

### KU8\_MMG\_ECU\_SIDE\_FRONT\_LEFT

|  |  |
| --- | --- |
| Name | Value |
| Ecu location side left | 0x9C |
| **Definition** | |
| #define KU8\_MMG\_ECU\_SIDE\_FRONT\_LEFT | |
| **Description** | |
| NA | |

### KU8\_MMG\_ECU\_SIDE\_FRONT\_RIGHT

|  |  |
| --- | --- |
| Name | Value |
| Ecu location side right | 0x99 |
| **Definition** | |
| #define KU8\_MMG\_ECU\_SIDE\_FRONT\_RIGHT | |
| **Description** | |
| NA | |

### KU8\_MMG\_ECU\_LOCATION

|  |  |
| --- | --- |
| Name | Value |
| NVP Ecu location | NVP\_BLOCK\_ID\_ECU\_LOCATION\_RamBlockData.NVP\_u8ECULocation |
| **Definition** | |
| #define KU8\_MMG\_ECU\_LOCATION | |
| **Description** | |
| NA | |

### KU8\_MMG\_PRESAFE\_RECORDER\_1\_ID

|  |  |
| --- | --- |
| Name | Value |
| Presafe recorder 1 ID value | 0x01 |
| **Definition** | |
| #define KU8\_MMG\_PRESAFE\_RECORDER\_1\_ID | |
| **Description** | |
| NA | |

### KU8\_MMG\_PRESAFE\_RECORDER\_2\_ID

|  |  |
| --- | --- |
| Name | Value |
| Presafe recorder 1 ID value | 0x02 |
| **Definition** | |
| #define KU8\_MMG\_PRESAFE\_RECORDER\_2\_ID | |
| **Description** | |
| NA | |

### MMG\_KU32\_AEC\_MASK\_GREEN\_STATUS

|  |  |
| --- | --- |
| Name | Value |
| Green AEC Group Status Mask | KU32\_AEC\_GROUP\_MASK\_OVER\_VO | KU32\_AEC\_GROUP\_MASK\_UNDER\_VO | KU32\_AEC\_GROUP\_MASK\_OVER\_VO\_TENS | KU32\_AEC\_GROUP\_MASK\_UNDER\_VO\_TENS | KU32\_AEC\_GROUP\_MASK\_NVM | KU32\_AEC\_GROUP\_MASK\_WARM\_RESET | KU32\_AEC\_GROUP\_MASK\_EOL\_COMFORT | KU32\_AEC\_GROUP\_MASK\_EOL\_MAX | KU32\_AEC\_GROUP\_MASK\_IMPLAUSIBLE\_DATA\_POWERTRAIN | KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_IGNITION | KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_BH\_R | KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_BH\_L | KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_POWERTRAIN | KU32\_AEC\_GROUP\_MASK\_SYSTEM\_OV | KU32\_AEC\_GROUP\_MASK\_SYSTEM\_UV |
| **Definition** | |
| #define MMG\_KU32\_AEC\_MASK\_GREEN\_STATUS | |
| **Description** | |
| NA | |

### MMG\_KU32\_AEC\_MASK\_RED\_STATUS

|  |  |
| --- | --- |
| Name | Value |
| Red AEC Group Status Mask | KU32\_AEC\_GROUP\_MASK\_HARDWARE | KU32\_AEC\_GROUP\_MASK\_SELF\_PROTECTION | KU32\_AEC\_GROUP\_MASK\_MOTOR | KU32\_AEC\_GROUP\_MASK\_EOL\_LOW\_HIGH | KU32\_AEC\_GROUP\_MASK\_STEERING\_CONFIG | KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_PRESAFE | KU32\_AEC\_GROUP\_MASK\_IMPLAUSIBLE\_DATA\_PRESAFE | KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_BUCKLE | KU32\_AEC\_GROUP\_MASK\_IMPLAUSIBLE\_DATA\_BUCKLE\_ORC | KU32\_AEC\_GROUP\_MASK\_IMPLAUSIBLE\_DATA\_IGNITION |
| **Definition** | |
| #define MMG\_KU32\_AEC\_MASK\_RED\_STATUS | |
| **Description** | |
| NA | |

### MMG\_STATUS\_IS\_GREEN

|  |  |
| --- | --- |
| Name | Value |
| Macro use as return for green status | 0x00 |
| **Definition** | |
| #define MMG\_STATUS\_IS\_GREEN | |
| **Description** | |
| NA | |

### MMG\_STATUS\_IS\_RED

|  |  |
| --- | --- |
| Name | Value |
| Macro use as return for red status | 0x02 |
| **Definition** | |
| #define MMG\_STATUS\_IS\_RED | |
| **Description** | |
| NA | |

### MMG\_STATUS\_IS\_DEFAULT

|  |  |
| --- | --- |
| Name | Value |
| Macro use as return for default status | 0x01 |
| **Definition** | |
| #define MMG\_STATUS\_IS\_DEFAULT | |
| **Description** | |
| NA | |

### MMG\_KU8\_IDX\_IMPL\_BUCKLE\_TENS

|  |  |
| --- | --- |
| Name | Value |
| Macro use for implausible buckle tensioning | 0x02 |
| **Definition** | |
| #define MMG\_KU8\_IDX\_IMPL\_BUCKLE\_TENS | |
| **Description** | |
| NA | |

### MMG\_KU8\_IDX\_TIMOEUT\_BUCKLEK\_TENS

|  |  |
| --- | --- |
| Name | Value |
| Macro use for timeout buckle tensioning | 0x03 |
| **Definition** | |
| #define MMG\_KU8\_IDX\_TIMOEUT\_BUCKLEK\_TENS | |
| **Description** | |
| NA | |

### MMG\_KU8\_CFG\_BSR3\_TENSIONING\_CYCLE

|  |  |
| --- | --- |
| Name | Value |
| Belt Slack Reduction cycle 3 | KU8\_BELT\_FUNCTION\_28 |
| **Definition** | |
| #define MMG\_KU8\_CFG\_BSR3\_TENSIONING\_CYCLE | |
| **Description** | |
| Internal macro for a belt function | |

### KU32\_MMG\_VEH\_EQP\_CYCLE\_ENABLE

|  |  |
| --- | --- |
| Name | Value |
| Macro used for checking if Profile is enable in vehicle equipment | ((KU8\_ONE) << ((bitPosition)-KU8\_ONE)) |
| **Definition** | |
| #define MMG\_KU8\_CFG\_BSR3\_TENSIONING\_CYCLE | |
| **Description** | |
| NA | |

### KU32\_MMG\_AEC\_GROUP\_MASK

|  |  |
| --- | --- |
| Name | Value |
| General group mask for AECs that inhibate Presafe cycles | (KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_PRESAFE | KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_BUCKLE | KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_IGNITION |KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_POWERTRAIN | KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_BH\_L | KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_BH\_R | KU32\_AEC\_GROUP\_MASK\_IMPLAUSIBLE\_DATA\_POWERTRAIN | KU32\_AEC\_GROUP\_MASK\_IMPLAUSIBLE\_DATA\_BUCKLE\_ORC | KU32\_AEC\_GROUP\_MASK\_IMPLAUSIBLE\_DATA\_IGNITION | KU32\_AEC\_GROUP\_MASK\_IMPLAUSIBLE\_DATA\_PRESAFE) |
| **Definition** | |
| #define KU32\_MMG\_AEC\_GROUP\_MASK (KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_PRESAFE | KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_BUCKLE | KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_IGNITION |KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_POWERTRAIN | KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_BH\_L | KU32\_AEC\_GROUP\_MASK\_TIMEOUT\_BH\_R | KU32\_AEC\_GROUP\_MASK\_IMPLAUSIBLE\_DATA\_POWERTRAIN | KU32\_AEC\_GROUP\_MASK\_IMPLAUSIBLE\_DATA\_BUCKLE\_ORC | KU32\_AEC\_GROUP\_MASK\_IMPLAUSIBLE\_DATA\_IGNITION | KU32\_AEC\_GROUP\_MASK\_IMPLAUSIBLE\_DATA\_PRESAFE) | |
| **Description** | |
| NA | |

### BOOL\_MMG\_CLEAR\_ENABLE\_CONDITION

|  |  |
| --- | --- |
| Name | Value |
| BOOL\_MMG\_CLEAR\_ENABLE\_CONDITION | (0u) |
| **Definition** | |
| #define BOOL\_MMG\_CLEAR\_ENABLE\_CONDITION (0u) | |
| **Description** | |
| Macro used to clear enable condition in DEM. | |

### BOOL\_MMG\_SET\_ENABLE\_CONDITION

|  |  |
| --- | --- |
| Name | Value |
| BOOL\_MMG\_SET\_ENABLE\_CONDITION | (1u) |
| **Definition** | |
| #define BOOL\_MMG\_SET\_ENABLE\_CONDITION (1u) | |
| **Description** | |
| Macro used to set enable condition in DEM. | |

### MMG\_SET\_ENABLE\_CONDITION\_MASK\_OFF(mask)

|  |  |
| --- | --- |
| Name | Value |
| MMG\_SET\_ENABLE\_CONDITION\_MASK\_OFF(mask) | (mask) U8\_BIT\_CLEAR(MMG\_u16EnableCondition\_Status, (mask)) |
| **Definition** | |
| #define MMG\_SET\_ENABLE\_CONDITION\_MASK\_OFF(mask) U8\_BIT\_CLEAR(MMG\_u16EnableCondition\_Status, (mask)) | |
| **Description** | |
| Macro used for bit clearing inhibition status. | |

### MMG\_SET\_ENABLE\_CONDITION\_MASK\_ON(mask)

|  |  |
| --- | --- |
| Name | Value |
| MMG\_SET\_ENABLE\_CONDITION\_MASK\_ON(mask) | (mask) U8\_BIT\_SET(MMG\_u16EnableCondition\_Status, (mask)) |
| **Definition** | |
| #define MMG\_SET\_ENABLE\_CONDITION\_MASK\_ON(mask) U8\_BIT\_SET(MMG\_u16EnableCondition\_Status, (mask)) | |
| **Description** | |
| Macro used for bit setting inhibition status. | |

# EEPROM

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

Refer to this document for more details.

# Configuration

No special configuration for MMG software component.

# Compilation Options

No special configuration for MMG software component.