|  |  |
| --- | --- |
| **PAL**  **Detailed Design Document** | |
| **Summary** | This is the Software Detailed Design Document for the PAL 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 \* 5

1.2. Purpose and Scope 5

1.3. Referenced documents 5

1.3.1. External documents 5

1.3.2. Internal Documents 5

1.4. Terminology and definitions 5

2.1. Overview 7

2.2. Traceability 7

3.1. Services 13

3.1.1. PAL\_Init 13

3.1.2. PAL\_runMainFunction 13

3.1.3. PAL\_runReadMotorCurrentInmA 14

3.1.4. PAL\_runReadMotorCurrentInA 15

3.1.5. PAL\_runSetPowerOrder 16

3.1.6. PAL\_runGetPWMOrder 17

3.1.7. PAL\_StartFreeWheelingState 17

3.2. Auto-Tests 17

3.2.1. PAL\_Autotest\_CheckHWSelfProtection 17

3.2.2. PAL\_Autotest\_CheckMotorCurrent 18

3.2.3. PAL\_Autotest\_CheckMotorThermalProctection 19

3.2.4. PAL\_Autotest\_CheckCommandConsistency 20

3.2.5. PAL\_Autotest\_CheckMosfetLowSC 21

3.2.6. PAL\_Autotest\_CheckMosfetHighSC 22

3.2.7. PAL\_Autotest\_CheckMosfetOCAT 23

3.2.8. PAL\_Autotest\_CheckHighSideSwRegulation 24

3.2.9. PAL\_Autotest\_CheckMotorConnection 25

3.2.10. PAL\_Autotest\_CheckMotorPowerOrder 26

3.2.11. PAL\_Autotest\_CheckMotorDisengagement 27

3.2.12. PAL\_Autotest\_CheckMotorSC 28

3.3. Internal local functions 29

3.3.1. pal\_bIsPowerStageInSelfProtection 29

3.3.2. pal\_u8MotorCurrent\_ManageMediumCurrentFailureDetection 30

3.3.3. pal\_u8MotorCurrent\_ManageHighCurrentFailureDetection 31

3.3.4. pal\_AT\_Init 32

3.3.5. pal\_Cfg\_Init 33

3.3.6. pal\_ComputeCurrentHalfBridgeP 34

3.3.7. pal\_ComputeCurrentHalfBridgeN 35

3.3.8. pal\_StopMotorRotation 36

3.3.9. pal\_DriveMotorInTensioning 36

3.3.10. pal\_DriveMotorInReleasing 37

3.3.11. pal\_DisablePowerStage 38

3.3.12. pal\_u8MotorOrder\_ManageMediumPWMFailureDetection 38

3.3.13. pal\_u8MotorOrder\_ManageHighPWMFailureDetection 39

3.3.14. pal\_HighSideSWRegFreeWheelingPhase 40

3.3.15. pal\_u8HighSideSWRegStep\_1 41

3.3.16. pal\_u8HighSideSWRegStep\_2 41

3.4. Types 42

3.4.1. PAL\_tHBridgeDefinition 42

3.4.2. PAL\_tHBridgeCurrentValues 42

3.4.3. PAL\_tBoostDriverDefinition 42

3.4.4. PAL\_tFullBridgeDefinition 42

3.4.5. PAL\_tPrelBoostTestData 43

3.4.6. PAL\_tVBoostTestData 43

3.4.7. PAL\_tAutoTestData 43

3.4.8. PAL\_tTopologyConfiguration 44

3.5. Variables 44

3.5.1. PAL\_kstHardWareTopology 44

3.5.2. PAL\_u8RequestedMode 44

3.5.3. PAL\_u8ModeState 44

3.5.4. PAL\_u8FreeWheelingTimer 44

3.5.5. PAL\_Cfg\_InternalData 45

3.5.6. PAL\_Cfg\_AT\_InternalData 45

3.6. Constants 45

3.6.1. PAL\_CFG\_TOPO\_HBRIDGE\_NB 45

3.6.2. PAL\_CFG\_KU8\_HB\_P\_ID 45

3.6.3. PAL\_CFG\_KU8\_HB\_N\_ID 45

3.6.4. PAL\_CFG\_TOPO\_SHUNT\_NB 45

3.6.5. PAL\_CFG\_TOPO\_FULL\_BRIDGE\_NB 46

3.6.6. PAL\_CFG\_FREEWHEELING\_TIME 46

3.6.7. PAL\_CFG\_PERIODIC\_FCT\_CALL 46

3.6.8. PAL\_CFG\_HIGH\_LIMIT\_LLD\_CMD 46

3.6.9. PAL\_CFG\_DUTY\_CYCLE\_ADAPT\_DEN 46

3.6.10. PAL\_CFG\_DUTY\_CYCLE\_ADAPT\_ROUND 47

3.6.11. PAL\_CFG\_DUTY\_CYCLE\_ADAPT\_A\_NUM 47

3.6.12. PAL\_CFG\_DUTY\_CYCLE\_ADAPT\_B\_NUM 47

3.6.13. PAL\_CFG\_SLOPE\_CTRL\_MAX\_VALUE 47

3.6.14. PAL\_CFG\_SLOPE\_DEFAULT\_VALUE 47

3.6.15. PAL\_CFG\_OFFSET\_CTRL\_MAX\_VALUE 47

3.6.16. PAL\_CFG\_OFFSET\_CTRL\_MIN\_VALUE 48

3.6.17. PAL\_CFG\_OFFSET\_DEFAULT\_VALUE 48

3.6.18. KU8\_NB\_INTERP\_PTS\_HB\_CALIB 48

3.6.19. PAL\_KU8\_INIT\_MODE\_STATE 48

3.6.20. PAL\_KU8\_DIAGNOSTIC\_MODE\_STATE 48

3.6.21. PAL\_KU8\_IDLE\_MODE\_STATE 49

3.6.22. PAL\_KU8\_ACTIVATION\_MODE\_STATE 49

3.6.23. PAL\_KU8\_OFF\_MODE\_STATE 49

3.6.24. PAL\_KU16\_MAX\_ADC\_VALUE 49

3.6.25. PAL\_KU16\_MAX\_ANALOG\_CH\_VALUE 49

3.6.26. PAL\_KU16\_DEFAULT\_HW\_SELF\_PROT\_THRS 49

3.6.27. PAL\_KU32\_CONVERT\_A\_TO\_MA 50

3.6.28. PAL\_KU8\_PERIODICITY\_10\_MS 50

3.6.29. PAL\_KU8\_CONVERT\_CENTIAMP\_TO\_MILLIAMP 50

3.6.30. PAL\_S32\_CONVERT\_A\_TO\_MA 50

3.6.31. PAL\_KU8\_OCAT\_PREPARE\_STEP\_1 50

3.6.32. PAL\_KU8\_OCAT\_EXECUTE\_STEP\_1\_PREPARE\_STEP\_2 51

3.6.33. PAL\_KU8\_OCAT\_EXECUTE\_STEP\_2\_PREPARE\_STEP\_3 51

3.6.34. PAL\_KU8\_OCAT\_EXECUTE\_STEP\_3\_PREPARE\_STEP\_4 51

3.6.35. PAL\_KU8\_OCAT\_EXECUTE\_STEP\_4 51

3.6.36. PAL\_KU8\_EIGHTY 51

3.6.37. PAL\_KU8\_DIVIDE\_BY\_64\_RIGHT\_SHIFT 51

3.6.38. PAL\_KU8\_CONVERT\_IN\_PERCENT 52

3.6.39. PAL\_KU32\_CONVERT\_MS\_TO\_100\_MS 52

3.6.40. PAL\_KU32\_SIXTEEN 52

3.6.41. PAL\_KU8\_MOTOR\_SHORT\_CIRCUIT\_RESITANCE\_MIN\_IDX 52

3.6.42. PAL\_KU8\_CONVERT\_DUTY\_CYCLE\_TO\_PWM\_ORDER\_FACTOR 52

3.6.43. PAL\_KU8\_HIGH\_SIDE\_SW\_REG\_INIT 53

3.6.44. PAL\_KU8\_HIGH\_SIDE\_SW\_REG\_ACTIVATION\_STEP\_1 53

3.6.45. PAL\_KU8\_HIGH\_SIDE\_SW\_REG\_ALGO\_STEP\_2 53

3.6.46. PAL\_KU8\_HIGH\_SIDE\_SW\_REG\_DEAD\_TIME 53

3.6.47. PAL\_KU8\_HIGH\_SIDE\_SW\_REG\_FREE\_WHEELING\_PRESCALER 53

# Tables of figures

[Figure 1: PAL\_Init diagram 12](#_Toc127434325)

[Figure 2: PAL\_runMainFunction diagram 13](#_Toc127434326)

[Figure 3: PAL\_runReadMotorCurrentInmA diagram 14](#_Toc127434327)

[Figure 4: PAL\_runReadMotorCurrentInA diagram 14](#_Toc127434328)

[Figure 5: PAL\_runSetPowerOrder diagram 15](#_Toc127434329)

[Figure 6: PAL\_Autotest\_CheckHWSelfProtection diagram 17](#_Toc127434330)

[Figure 7: PAL\_Autotest\_CheckMotorCurrent diagram 18](#_Toc127434331)

[Figure 8: PAL\_Autotest\_CheckMotorThermalProctection diagram 19](#_Toc127434332)

[Figure 9 PAL\_Autotest\_CheckCommandConsistency 20](#_Toc127434333)

[Figure 10 PAL\_Autotest\_CheckMosfetLowSC 21](#_Toc127434334)

[Figure 11PAL\_Autotest\_CheckMosfetHighSC 22](#_Toc127434335)

[Figure 12 PAL\_Autotest\_CheckMosfetOCAT 23](#_Toc127434336)

[Figure 13 PAL\_Autotest\_CheckHighSideSwRegulation 24](#_Toc127434337)

[Figure 14 PAL\_Autotest\_CheckMotorConnection 25](#_Toc127434338)

[Figure 15 PAL\_Autotest\_CheckMotorPowerOrder 26](#_Toc127434339)

[Figure 16 PAL\_Autotest\_CheckMotorDisengagement 27](#_Toc127434340)

[Figure 17 PAL\_Autotest\_CheckMotorSC 28](#_Toc127434341)

[Figure 18: pal\_bIsPowerStageInSelfProtection diagram 29](#_Toc127434342)

[Figure 19: pal\_u8MotorCurrent\_ManageMediumCurrentFailureDetection diagram 30](#_Toc127434343)

[Figure 20: pal\_u8MotorCurrent\_ManageHighCurrentFailureDetection diagram 31](#_Toc127434344)

[Figure 21: pal\_AT\_Init diagram 32](#_Toc127434345)

[Figure 22: pal\_Cfg\_Init diagram 33](#_Toc127434346)

[Figure 23: pal\_ComputeCurrentHalfBridgeP diagram 34](#_Toc127434347)

[Figure 24: pal\_ComputeCurrentHalfBridgeN diagram 35](#_Toc127434348)

[Figure 25: pal\_StopMotorRotation diagram 35](#_Toc127434349)

[Figure 26: pal\_DriveMotorInTensioning diagram 36](#_Toc127434350)

[Figure 27: pal\_DriveMotorInReleasing diagram 37](#_Toc127434351)

[Figure 28: pal\_DisablePowerStage diagram 37](#_Toc127434352)

[Figure 29 pal\_u8MotorOrder\_ManageMediumPWMFailureDetection 38](#_Toc127434353)

[Figure 30 pal\_u8MotorOrder\_ManageHighPWMFailureDetection 39](#_Toc127434354)

# General Information

## Revision history \*

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Author(s)** | **Description/comment** |
| 1.1.2.1 | 07.01.2022 | Septimiu-Darex C. Vintila | Initial revision. |
| 1.1.2.2 | 07.01.2022 | Septimiu-Darex C. Vintila | Services added. |
| 1.1.2.3 | 11.02.2022 | Septimiu-Darex C. Vintila | Corrections after review. |
| 1.1.2.4 | 29.03.2022 | Obada Mirela | Update traceabilty |
| 1.1.2.5 | 29.03.2022 | Obada Mirela | Update document |
| 1.1.2.6 | 11.04.2022 | Obada Mirela | Fix findings from review |
| 1.1.2.7 | 06.09.2022 | Madalina Serban | Added new functions and traceability |
| 1.1.2.8 | 08.09.2022 | Madalina Serban | Fixed review findings |
| 1.1.2.9 | 22.11.2022 | Mirela Obada | Add new req for tracebilty from arch |
| 1.1.2.10 | 10.02.2023 | Mirela Obada | Update document for 5.0 |
| 1.1.2.11 | 16.02.2023 | Mirela Obada | Update all diagrams and traceabilty for 5.0 |
| 1.1.2.12 | 16.02.2023 | Mirela Obada | Update 1 req for trasabilty |
| 1.1.2.13 | 17.02.2023 | Mirela Obada | Update 2 req for HSSwitch autotest |

*\* 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 PAL component

## Referenced documents

### External documents

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

### Internal Documents

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

## Terminology and definitions

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

|  |  |
| --- | --- |
| **Terminology** | **Meaning** |
| AAU | Atomic architectural unit |
| SW | software |
| PAL | Power Abstraction Layer |

# SW atomic architectural unit design

## Overview

The aim of the PAL module is to handle the interaction with the motor. The module controls the motor actions and also its self diagnostics.

The module is split in several files: PAL\_Autotests.c, PAL\_config.c, PAL\_CurrentMeasure.c, PAL\_main.c, PAL\_PowerLayerDriving.c

## Traceability

|  |  |  |  |
| --- | --- | --- | --- |
| Requirements | Criteria | Linked Runnable | Source |
| DSG\_PAL\_0001 | The default value for the component Mode State shall be PAL\_KU8\_DIAGNOSTIC\_MODE\_STATE. | PAL\_Init() | ARCH\_SW\_PAL\_0010 |
| DSG\_PAL\_0002 | The default value for the component Requested Mode shall be PAL\_KU8\_INIT\_MODE\_STATE. | PAL\_Init() | ARCH\_SW\_PAL\_0010 |
| DSG\_PAL\_0003 | The default value for the counter Free Wheeling Timer shall be PAL\_CFG\_FREEWHEELING\_TIME. | PAL\_Init() | ARCH\_SW\_PAL\_0010 |
| DSG\_PAL\_0004 | The configuration part of the component shall be initialized. | PAL\_Init() | ARCH\_SW\_PAL\_0010 |
| DSG\_PAL\_0005 | The component shall enable the H-Bridges related PWMs at start-up. | PAL\_Init() | ARCH\_SW\_PAL\_0010 |
| DSG\_PAL\_0006 | The running mode state of the component shall be evaluated every 2ms. | PAL\_runMainFunction() | ARCH\_SW\_PAL\_0450 |
| DSG\_PAL\_0007 | In case the running mode state is PAL\_KU8\_DIAGNOSTIC\_MODE\_STATE and the critical autotests are passed, the running mode state shall be set to PAL\_KU8\_IDLE\_MODE\_STATE. | PAL\_runMainFunction() | ARCH\_SW\_PAL\_0450; ARCH\_SW\_PAL\_0457 |
| DSG\_PAL\_0008 | In case the running mode state is PAL\_KU8\_IDLE\_MODE\_STATE and a motor activation order has been received since the last function call, the running mode state shall be set to PAL\_KU8\_ACTIVATION\_MODE\_STATE. | PAL\_runMainFunction() | ARCH\_SW\_PAL\_0450 ARCH\_SW\_PAL\_0457 |
| DSG\_PAL\_0009 | In case the running mode state is PAL\_KU8\_ACTIVATION\_MODE\_STATE and no motor activation order has been received since PAL\_CFG\_FREEWHEELING\_TIME, the running mode state shall be set to PAL\_KU8\_IDLE\_MODE\_STATE. | PAL\_runMainFunction() | ARCH\_SW\_PAL\_0450 ARCH\_SW\_PAL\_0457 |
| DSG\_PAL\_0010 | In case of Shut down request or in case of critical auto test have failed during startup phase, the running mode state shall be set to PAL\_KU8\_OFF\_MODE\_STATE. | PAL\_runMainFunction() | ARCH\_SW\_PAL\_0450; ARCH\_SW\_PAL\_0456; ARCH\_SW\_PAL\_0457 |
| DSG\_PAL\_0011 | The motor current shall be calculated based on the received motor power order. | PAL\_runReadMotorCurrentInmA() PAL\_runReadMotorCurrentInA() | ARCH\_SW\_PAL\_0080; ARCH\_SW\_PAL\_0100 |
| DSG\_PAL\_0012 | The motor current result shall be available in Amperes. | PAL\_runReadMotorCurrentInA() | ARCH\_SW\_PAL\_0100 |
| DSG\_PAL\_0013 | The motor current result shall be available in milliamperes. | PAL\_runReadMotorCurrentInmA() | ARCH\_SW\_PAL\_0080 |
| DSG\_PAL\_0014 | In case the motor is driven in tensioning direction, the motor current shall be calculated using Half Bridge P current. | PAL\_runReadMotorCurrentInmA() PAL\_runReadMotorCurrentInA() | ARCH\_SW\_PAL\_0080; ARCH\_SW\_PAL\_0100; ARCH\_SW\_PAL\_0085; ARCH\_SW\_PAL\_0105 |
| DSG\_PAL\_0015 | In case the motor is driven in tensioning direction, the motor current shall be calculated using Half Bridge N current. | PAL\_runReadMotorCurrentInmA() PAL\_runReadMotorCurrentInA() | ARCH\_SW\_PAL\_0080; ARCH\_SW\_PAL\_0100; ARCH\_SW\_PAL\_0086; ARCH\_SW\_PAL\_0106 |
| DSG\_PAL\_0016 | In case of a Null power order, the motor current shall be the sum of both Isense. | PAL\_runReadMotorCurrentInmA() PAL\_runReadMotorCurrentInA() | ARCH\_SW\_PAL\_0080; ARCH\_SW\_PAL\_0100; ARCH\_SW\_PAL\_0085; ARCH\_SW\_PAL\_0086; ARCH\_SW\_PAL\_0105; ARCH\_SW\_PAL\_0106 |
| DSG\_PAL\_0017 | The control of the motor power shall take place only in the modes allowed to control the power stage. | PAL\_runSetPowerOrder() | ARCH\_SW\_PAL\_0020; ARCH\_SW\_PAL\_0025; ARCH\_SW\_PAL\_0026; ARCH\_SW\_PAL\_0027; ARCH\_SW\_PAL\_0028; ARCH\_SW\_PAL\_0029 |
| DSG\_PAL\_0018 | In case the absolute value of motor command is 0, the power layer shall be disabled. | PAL\_runSetPowerOrder() | ARCH\_SW\_PAL\_0020; ARCH\_SW\_PAL\_0025; ARCH\_SW\_PAL\_0026; ARCH\_SW\_PAL\_0027; ARCH\_SW\_PAL\_0028; ARCH\_SW\_PAL\_0029 |
| DSG\_PAL\_0019 | In case the value of motor command has a negative value, the motor shall be driven in releasing direction. | PAL\_runSetPowerOrder() | ARCH\_SW\_PAL\_0020; ARCH\_SW\_PAL\_0025; ARCH\_SW\_PAL\_0026; ARCH\_SW\_PAL\_0027; ARCH\_SW\_PAL\_0028; ARCH\_SW\_PAL\_0029 |
| DSG\_PAL\_0020 | In case the value of motor command has a positive value, the motor shall be driven in tensioning direction. | PAL\_runSetPowerOrder() | ARCH\_SW\_PAL\_0020; ARCH\_SW\_PAL\_0025; ARCH\_SW\_PAL\_0026; ARCH\_SW\_PAL\_0027; ARCH\_SW\_PAL\_0028; ARCH\_SW\_PAL\_0029 |
| DSG\_PAL\_0021 | The Check HW Self Protection Autotest shall be executed every 10 ms. | PAL\_Autotest\_CheckHWSelfProtection() | ARCH\_SW\_PAL\_0140 |
| DSG\_PAL\_0022 | If both the currents measured on the H-bridge of the Vp motor pin and the H-bridge of the Vn motor pin are below the current sense threshold, the Power Overvoltage Autotest shall return KU8\_ATM\_TEST\_OK. | PAL\_Autotest\_CheckHWSelfProtection() | ARCH\_SW\_PAL\_0140; ARCH\_SW\_PAL\_0141; ARCH\_SW\_PAL\_0145; ARCH\_SW\_PAL\_0146 ARCH\_SW\_PAL\_0147 |
| DSG\_PAL\_0023 | If the current measured on the H-bridge of the Vp motor pin is higher than the current sense threshold, the Power Overvoltage Autotest shall return KU8\_ATM\_TEST\_NOK. | PAL\_Autotest\_CheckHWSelfProtection() | ARCH\_SW\_PAL\_0140; ARCH\_SW\_PAL\_0141; ARCH\_SW\_PAL\_0145; ARCH\_SW\_PAL\_0147 |
| DSG\_PAL\_0024 | If the current measured on the H-bridge of the Vn motor pin is higher than the current sense threshold, the Power Overvoltage Autotest shall return KU8\_ATM\_TEST\_NOK. | PAL\_Autotest\_CheckHWSelfProtection() | ARCH\_SW\_PAL\_0140; ARCH\_SW\_PAL\_0141; ARCH\_SW\_PAL\_0145; ARCH\_SW\_PAL\_0146 |
| DSG\_PAL\_0025 | The Check Motor Current Autotest shall be executed every 10 ms. | PAL\_AutotestCheckMotorCurrent() | ARCH\_SW\_PAL\_0340 |
| DSG\_PAL\_0026 | In case a PRO cycle is running, the Check Motor Current Autotest shall return KU8\_ATM\_TEST\_NOT\_DECIDED. | PAL\_AutotestCheckMotorCurrent() | ARCH\_SW\_PAL\_0340 ARCH\_SW\_PAL\_0342 |
| DSG\_PAL\_0027 | In case the executed cycle is a tensioning cycle and no medium current failure or high current failure was detected, the Check Motor Current Autotest shall return KU8\_ATM\_TEST\_OK. | PAL\_AutotestCheckMotorCurrent() | ARCH\_SW\_PAL\_0340 ARCH\_SW\_PAL\_0342 ARCH\_SW\_PAL\_0345 |
| DSG\_PAL\_0028 | In case the executed cycle is a tensioning cycle and a medium current failure was detected, the Check Motor Current Autotest shall return KU8\_ATM\_TEST\_NOK. | PAL\_AutotestCheckMotorCurrent() | ARCH\_SW\_PAL\_0340 ARCH\_SW\_PAL\_0342 ARCH\_SW\_PAL\_0345 |
| DSG\_PAL\_0029 | In case the executed cycle is a tensioning cycle and a high current failure was detected, the Check Motor Current Autotest shall return KU8\_ATM\_TEST\_NOK. | PAL\_AutotestCheckMotorCurrent() | ARCH\_SW\_PAL\_0340 ARCH\_SW\_PAL\_0342 ARCH\_SW\_PAL\_0345 |
| DSG\_PAL\_0030 | In case the executed cycle is a tensioning cycle and one of medium current failure detection or high current failure detection could not be executed, the Check Motor Current Autotest shall return KU8\_ATM\_TEST\_OK. | PAL\_AutotestCheckMotorCurrent() | ARCH\_SW\_PAL\_0340 ARCH\_SW\_PAL\_0342 ARCH\_SW\_PAL\_0345 |
| DSG\_PAL\_0031 | In case the executed cycle is a non-tensioning cycle and the time configured has not expired since the last KU8\_ATM\_TEST\_NOK result, the Check Motor Current Autotest shall return KU8\_ATM\_TEST\_NOK. | PAL\_AutotestCheckMotorCurrent() | ARCH\_SW\_PAL\_0340 ARCH\_SW\_PAL\_0341 ARCH\_SW\_PAL\_0342 ARCH\_SW\_PAL\_0346 |
| DSG\_PAL\_0032 | In case the executed cycle is a non-tensioning cycle and the time configured has expired since the last KU8\_ATM\_TEST\_NOK result, the Check Motor Current Autotest shall return KU8\_ATM\_TEST\_OK. | PAL\_AutotestCheckMotorCurrent() | ARCH\_SW\_PAL\_0340 ARCH\_SW\_PAL\_0341 ARCH\_SW\_PAL\_0342 ARCH\_SW\_PAL\_0346 |
| DSG\_PAL\_0033 | The Check Motor Thermal Protection Autotest shall be executed every 100 ms. | PAL\_Autotest\_CHeckMotorThermalProtection() | ARCH\_SW\_PAL\_0360 |
| DSG\_PAL\_0034 | In case the previous Check Motor Thermal Protection Autotest result was KU8\_ATM\_TEST\_NOK and the deficiency level is over the deskill threshold, the Check Motor Thermal Protection Autotest shall return KU8\_ATM\_TEST\_NOK. | PAL\_Autotest\_CHeckMotorThermalProtection() | ARCH\_SW\_PAL\_0360; ARCH\_SW\_PAL\_0361; ARCH\_SW\_PAL\_0365; ARCH\_SW\_PAL\_0366; ARCH\_SW\_PAL\_0367 |
| DSG\_PAL\_0035 | In case the previous Check Motor Thermal Protection Autotest result was KU8\_ATM\_TEST\_NOK and the deficiency level is under the deskill threshold, the Check Motor Thermal Protection Autotest shall return KU8\_ATM\_TEST\_OK. | PAL\_Autotest\_CHeckMotorThermalProtection() | ARCH\_SW\_PAL\_0360; ARCH\_SW\_PAL\_0361; ARCH\_SW\_PAL\_0365;ARCH\_SW\_PAL\_0366; ARCH\_SW\_PAL\_0367 |
| DSG\_PAL\_0036 | In case the previous Check Motor Thermal Protection Autotest result was KU8\_ATM\_TEST\_OK and the deficiency level is over the qualification threshold, the Check Motor Thermal Protection Autotest shall return KU8\_ATM\_TEST\_NOK. | PAL\_Autotest\_CHeckMotorThermalProtection() | ARCH\_SW\_PAL\_0360; ARCH\_SW\_PAL\_0361; ARCH\_SW\_PAL\_0365; ARCH\_SW\_PAL\_0366; ARCH\_SW\_PAL\_0367 |
| DSG\_PAL\_0037 | In case the previous Check Motor Thermal Protection Autotest result was KU8\_ATM\_TEST\_OK and the deficiency level is under the qualification threshold, the Check Motor Thermal Protection Autotest shall return KU8\_ATM\_TEST\_OK. | PAL\_Autotest\_CHeckMotorThermalProtection() | ARCH\_SW\_PAL\_0360; ARCH\_SW\_PAL\_0361; ARCH\_SW\_PAL\_0365; ARCH\_SW\_PAL\_0366; ARCH\_SW\_PAL\_0367 |
| DSG\_PAL\_0038 | PAL component will initialize the module according to the project configuration | Requirement covered by PAL\_Cfg\_Init() function | ARCH\_SW\_PAL\_0090,  ARCH\_SW\_PAL\_0095 |
| DSG\_PAL\_0039 | PAL component will initialize the autotest internal data | Requirement covered by PAL\_AT\_Init() function | ARCH\_SW\_PAL\_0110 |
| DSG\_PAL\_0040 | PAL component shall disable the HW when the motor is not activated | Requirement covered by PAL\_DisablePowerStage() function | ARCH\_SW\_PAL\_0070; ARCH\_SW\_PAL\_0075; ARCH\_SW\_PAL\_0076 |
| DSG\_PAL\_0041 | In case the power stage enable signals for the 2 half-bridge are consistent PAL\_Autotest\_CheckCommandConsistency autotest shall return KU8\_ATM\_TEST\_OK | Requirement covered by PAL\_Autotest\_CheckCommandConsistency() function | ARCH\_SW\_PAL\_0369; |
| DSG\_PAL\_0042 | In case the previous autotests for ADC, High Side SW, Driver command are not successfully executed, the autotest shall return KU8\_ATM\_TEST\_NOT\_DECIDED. | Requirement covered by PAL\_Autotest\_CheckMosfetLowSC() function | ARCH\_SW\_PAL\_0370;ARCH\_SW\_PAL\_0391;ARCH\_SW\_PAL\_0390; |
| DSG\_PAL\_0043 | In case the voltage on either “Motor +” or “Motor – “pin drops bellow a certain threshold, the autotest shall return KU8\_ATM\_TEST\_NOK | Requirement covered by PAL\_Autotest\_CheckMosfetLowSC() function | ARCH\_SW\_PAL\_0370;ARCH\_SW\_PAL\_0391;ARCH\_SW\_PAL\_0390; |
| DSG\_PAL\_0044 | In case the previous autotests for ADC, High Side SW, Driver command return KU8\_ATM\_TEST\_OK and the voltage on both pins is within the threshold, the autotest shall return KU8\_ATM\_TEST\_OK | Requirement covered by PAL\_Autotest\_CheckMosfetLowSC() function | ARCH\_SW\_PAL\_0370;ARCH\_SW\_PAL\_0391;ARCH\_SW\_PAL\_0390; |
| DSG\_PAL\_0045 | In case the previous autotests for ADC, High Side SW, Driver command are not successfully executed, the autotest shall return KU8\_ATM\_TEST\_NOT\_DECIDED. | Requirement covered by PAL\_Autotest\_CheckMosfetHighSC() function | ARCH\_SW\_PAL\_0372; ARCH\_SW\_PAL\_0395; ARCH\_SW\_PAL\_0396; |
| DSG\_PAL\_0046 | In case the voltage on either “Motor +” or “Motor – “pin drops bellow a certain threshold, the autotest shall return KU8\_ATM\_TEST\_NOK | Requirement covered by PAL\_Autotest\_CheckMosfetHighSC() function | ARCH\_SW\_PAL\_0372; ARCH\_SW\_PAL\_0395; ARCH\_SW\_PAL\_0396; |
| DSG\_PAL\_0047 | In case the previous autotests for ADC, High Side SW, Driver command return KU8\_ATM\_TEST\_OK and the voltage on both pins is within the threshold, the autotest shall return KU8\_ATM\_TEST\_OK | Requirement covered by PAL\_Autotest\_CheckMosfetHighSC() function | ARCH\_SW\_PAL\_0372; ARCH\_SW\_PAL\_0395; ARCH\_SW\_PAL\_0396; |
| DSG\_PAL\_0048 | In case the previous autotests for ADC, Driver Self protection, Motor Disconnection, Mosfet High Short Circuit, Mosfet Low Short Circuit, Driver command are not successfully executed, the autotest shall return KU8\_ATM\_TEST\_NOT\_DECIDED. | Requirement covered by PAL\_Autotest\_CheckMosfetOCAT() function | ARCH\_SW\_PAL\_0374; ARCH\_SW\_PAL\_0375; ARCH\_SW\_PAL\_0373; |
| DSG\_PAL\_0049 | In case the computed voltage on either “Motor +” or “Motor – “pin passes over a certain threshold, the autotest shall return KU8\_ATM\_TEST\_NOK | Requirement covered by PAL\_Autotest\_CheckMosfetOCAT() function | ARCH\_SW\_PAL\_0374; ARCH\_SW\_PAL\_0375; ARCH\_SW\_PAL\_0373; |
| DSG\_PAL\_0050 | In case the previous autotests for ADC, Driver Self protection, Motor Disconnection, Mosfet High Short Circuit, Mosfet Low Short Circuit, Driver command return KU8\_ATM\_TEST\_OK and the computed voltage from both pins are within a certain threshold, the autotest shall return KU8\_ATM\_TEST\_OK | Requirement covered by PAL\_Autotest\_CheckMosfetOCAT() function | ARCH\_SW\_PAL\_0374; ARCH\_SW\_PAL\_0375; ARCH\_SW\_PAL\_0373; |
| DSG\_PAL\_0051 | In case the previous autotests for ADC is not successfully executed, the autotest shall return KU8\_ATM\_TEST\_NOT\_DECIDED. | Requirement covered by PAL\_Autotest\_CheckHighSideSwRegulation() function | ARCH\_SW\_PAL\_0368; |
| DSG\_PAL\_0052 | If High Side SW voltage is inside the functional range, return OK | Requirement covered by PAL\_Autotest\_CheckHighSideSwRegulation() function | ARCH\_SW\_PAL\_0368; |
| DSG\_PAL\_0053 | If High Side SW voltage is not inside the functional range and maximum number of tries is reached, return NOK | Requirement covered by PAL\_Autotest\_CheckHighSideSwRegulation() function | ARCH\_SW\_PAL\_0368; |
| DSG\_PAL\_0054 | In case the previous autotests for ADC, Mosfet High Short Circuit, Mosfet Low Short Circuit, High Side SW regulation are not successfully executed, the autotest shall return KU8\_ATM\_TEST\_NOT\_DECIDED. | Requirement covered by PAL\_Autotest\_CheckMotorConnection() function | ARCH\_SW\_PAL\_0376; ARCH\_SW\_PAL\_0377; ARCH\_SW\_PAL\_0410; |
| DSG\_PAL\_0055 | In case the computed voltage difference between "Motor +" and "Motor -" goes over a certain threshold, the autotest shall return KU8\_ATM\_TEST\_NOK | Requirement covered by PAL\_Autotest\_CheckMotorConnection() function | ARCH\_SW\_PAL\_0376; ARCH\_SW\_PAL\_0377; ARCH\_SW\_PAL\_0410; |
| DSG\_PAL\_0056 | In case the battery is not stable during test execuction, the autotest shall return KU8\_ATM\_TEST\_NOT\_DECIDED. | Requirement covered by PAL\_Autotest\_CheckMotorConnection() function | ARCH\_SW\_PAL\_0376; ARCH\_SW\_PAL\_0377; ARCH\_SW\_PAL\_0410; |
| DSG\_PAL\_0057 | In case the previous autotests for ADC, Mosfet High Short Circuit, Mosfet Low Short Circuit, High Side SW regulation return KU8\_ATM\_TEST\_OK, the computed voltage difference between "Motor +" and "Motor -" are within a certain threshold and the battery is stable during the test, the autotest shall return KU8\_ATM\_TEST\_OK | Requirement covered by PAL\_Autotest\_CheckMotorConnection() function | ARCH\_SW\_PAL\_0376; ARCH\_SW\_PAL\_0377; ARCH\_SW\_PAL\_0410; |
| DSG\_PAL\_0058 | In case the previous autotests for ADC is not successfully executed and any PRO cycle is running, the autotest shall return KU8\_ATM\_TEST\_NOT\_DECIDED. | Requirement covered by PAL\_Autotest\_CheckMotorPowerOrder () function | ARCH\_SW\_PAL\_0381; ARCH\_SW\_PAL\_0382; ARCH\_SW\_PAL\_0390; ARCH\_SW\_PAL\_0391; ARCH\_SW\_PAL\_0392; ARCH\_SW\_PAL\_0393; |
| DSG\_PAL\_0059 | In case pal\_u8MotorOrder\_ManageMediumPWMFailureDetection or pal\_u8MotorOrder\_ManageHighPWMFailureDetection return KU8\_ATM\_TEST\_NOK, the autotest shall return KU8\_ATM\_TEST\_NOK. | Requirement covered by PAL\_Autotest\_CheckMotorPowerOrder () function | ARCH\_SW\_PAL\_0381; ARCH\_SW\_PAL\_0382; ARCH\_SW\_PAL\_0390; ARCH\_SW\_PAL\_0391; ARCH\_SW\_PAL\_0392; |
| DSG\_PAL\_0060 | In case pal\_u8MotorOrder\_ManageMediumPWMFailureDetection or pal\_u8MotorOrder\_ManageHighPWMFailureDetection return KU8\_ATM\_TEST\_OK, the autotest shall return KU8\_ATM\_TEST\_OK. | Requirement covered by PAL\_Autotest\_CheckMotorPowerOrder () function | ARCH\_SW\_PAL\_0381; ARCH\_SW\_PAL\_0382; ARCH\_SW\_PAL\_0390; ARCH\_SW\_PAL\_0391; ARCH\_SW\_PAL\_0392; |
| DSG\_PAL\_0061 | During any non tensioning cycles if there are any AECs qualified and the counter for motor order error is bellow a certain value, , the autotest shall return KU8\_ATM\_TEST\_NOK. | Requirement covered by PAL\_Autotest\_CheckMotorPowerOrder () function | ARCH\_SW\_PAL\_0381; ARCH\_SW\_PAL\_0382; ARCH\_SW\_PAL\_0390; ARCH\_SW\_PAL\_0391; ARCH\_SW\_PAL\_0392; |
| DSG\_PAL\_0070 | During any non tensioning cycles if there are no AECs qualified and al the counter for motor order error are set to zero, the autotest shall return KU8\_ATM\_TEST\_OK. | Requirement covered by PAL\_Autotest\_CheckMotorPowerOrder () function | ARCH\_SW\_PAL\_0381; ARCH\_SW\_PAL\_0382; ARCH\_SW\_PAL\_0390; ARCH\_SW\_PAL\_0391; ARCH\_SW\_PAL\_0392; |
| DSG\_PAL\_0063 | In case the previous autotests for ADC, Driver Self Protection are not successfully executed and any other cycles besides KU8\_BELT\_FUNCTION\_29, KU8\_BELT\_FUNCTION\_30, KU8\_BELT\_FUNCTION\_31 are running, the autotest shall return KU8\_ATM\_TEST\_NOT\_DECIDED. | Requirement covered by PAL\_Autotest\_CheckMotorDisengagement() function | ARCH\_SW\_PAL\_0378; ARCH\_SW\_PAL\_0379; ARCH\_SW\_PAL\_0380; ARCH\_SW\_PAL\_0400; ARCH\_SW\_PAL\_0401; ARCH\_SW\_PAL\_0402; ARCH\_SW\_PAL\_0403; |
| DSG\_PAL\_0062 | In case the motor current is above the threshold, and the counter for maximum failed occurrences reaches the maximum number of failures, the autotest shall return KU8\_ATM\_TEST\_NOK and the motor blocked counter is reset to zero. | Requirement covered by PAL\_Autotest\_CheckMotorDisengagement() function | ARCH\_SW\_PAL\_0378; ARCH\_SW\_PAL\_0379; ARCH\_SW\_PAL\_0380; ARCH\_SW\_PAL\_0400; ARCH\_SW\_PAL\_0401; ARCH\_SW\_PAL\_0402; ARCH\_SW\_PAL\_0403; |
| DSG\_PAL\_0069 | In case the motor currebt is bellow the threshold and the couter for maximum passed occurences is reached, autotest shall return KU8\_ATM\_TEST\_OK and the counter is reset to zero. | Requirement covered by PAL\_Autotest\_CheckMotorDisengagement() function | ARCH\_SW\_PAL\_0378; ARCH\_SW\_PAL\_0379; ARCH\_SW\_PAL\_0380; ARCH\_SW\_PAL\_0400; ARCH\_SW\_PAL\_0401; ARCH\_SW\_PAL\_0402; ARCH\_SW\_PAL\_0403; |
| DSG\_PAL\_0064 | In case MOSFET Short Circuit OR ADC OR HW Self protection auto-tests results are failed or no releasing cycle is executed, the autotest shall return KU8\_ATM\_TEST\_NOT\_DECIDED. | Requirement covered by PAL\_Autotest\_CheckMotorSC() function | ARCH\_SW\_PAL\_0383; ARCH\_SW\_PAL\_0384; ARCH\_SW\_PAL\_0385; ARCH\_SW\_PAL\_0386; ARCH\_SW\_PAL\_0387; ARCH\_SW\_PAL\_0388;ARCH\_SW\_PAL\_0389; |
| DSG\_PAL\_0065 | In case the measured Motor current is higher than the calculated current threshold and if the filter counter has reach the failed threshold, the autotest shall return KU8\_ATM\_TEST\_NOK and the counter is reset to zero. | Requirement covered by PAL\_Autotest\_CheckMotorSC() function | ARCH\_SW\_PAL\_0383; ARCH\_SW\_PAL\_0384; ARCH\_SW\_PAL\_0385; ARCH\_SW\_PAL\_0386; ARCH\_SW\_PAL\_0387; ARCH\_SW\_PAL\_0388;ARCH\_SW\_PAL\_0389; |
| DSG\_PAL\_0066 | In case the measured Motor current is below the calculated current threshold and the filter ccounter has not reached the passed threshold, the autotest shall return KU8\_ATM\_TEST\_NOK and the counter is incremented to 1. | Requirement covered by PAL\_Autotest\_CheckMotorSC() function | ARCH\_SW\_PAL\_0383; ARCH\_SW\_PAL\_0384; ARCH\_SW\_PAL\_0385; ARCH\_SW\_PAL\_0386; ARCH\_SW\_PAL\_0387; ARCH\_SW\_PAL\_0388;ARCH\_SW\_PAL\_0389; |
| DSG\_PAL\_0067 | In case the measured Motor current is below the calculated current threshold and the filter ccounter has reached the passed threshold, the autotest shall return KU8\_ATM\_TEST\_OK and the counter is reset to 0. | Requirement covered by PAL\_Autotest\_CheckMotorSC() function | ARCH\_SW\_PAL\_0383; ARCH\_SW\_PAL\_0384; ARCH\_SW\_PAL\_0385; ARCH\_SW\_PAL\_0386; ARCH\_SW\_PAL\_0387; ARCH\_SW\_PAL\_0388;ARCH\_SW\_PAL\_0389; |
| DSG\_PAL\_0068 | In case the absolute physical value of motor current is bellow the specified threshold, autotest shall return KU8\_ATM\_TEST\_OK. | Requirement covered by PAL\_Autotest\_CheckMotorSC() function | ARCH\_SW\_PAL\_0383; ARCH\_SW\_PAL\_0384; ARCH\_SW\_PAL\_0385; ARCH\_SW\_PAL\_0386; ARCH\_SW\_PAL\_0387; ARCH\_SW\_PAL\_0388;ARCH\_SW\_PAL\_0389 |
| DSG\_PAL\_0069 | The value of the power stage shall be set and available to the rest of application. | PAL\_StartFreeWheelingState() | ARCH\_SW\_PAL\_0077; ARCH\_SW\_PAL\_0078; ARCH\_SW\_PAL\_0079; |

# FEATURES

## Services

### PAL\_Init

|  |  |  |
| --- | --- | --- |
| **Object :** | | |
| This function shall initialize the PAL SW unit. | | |
| **Input Parameters :** | | |
| Name | Type | Description |
| none |  |  |
| **Output Parameters f** | | |
| Name | Type | Description |
| none |  |  |
| **Dynamic Aspect :** | | |
| Will be called at system startup | | |
| Who : | Scheduler | |
| **Return value :** | | |
| Type : | Description | |
| none | - | |
| **Constraints :** | | |
| none | | |
| **Static aspect :** | | |
|  | | |

Diagram

Description automatically generated

Figure : PAL\_Init diagram

### PAL\_runMainFunction

|  |  |  |
| --- | --- | --- |
| **Object :** | | |
| Cyclic runnable in charge of the internal state machine. Handles freewheeling timer and disabling power stage. | | |
| **Input Parameters :** | | |
| Name | Type | Description |
| none |  |  |
| **Output Parameters :** | | |
| Name | Type | Description |
| none |  |  |
| **Dynamic Aspect :** | | |
| Will be called at 2 ms | | |
| Who : | Scheduler | |
| **Return value :** | | |
| Type : | Description | |
| none |  | |
| **Constraints :** | | |
| none | | |
| **Static aspect :** | | |
|  | | |

Diagram

Description automatically generated

Figure : PAL\_runMainFunction diagram

### PAL\_runReadMotorCurrentInmA

|  |  |  |
| --- | --- | --- |
| **Object :** | | |
| Function that provides the current in Milli Ampere | | |
| **Input Parameters :** | | |
| Name | Type | Description |
| none |  |  |
| **Output Parameters :** | | |
| Name | Type | Description |
| ps32MotorCurrentInmA | s32MotorCurrentInmAType | Pointer used to store the requested result |
| **Dynamic Aspect :** | | |
| Called on request | | |
| Who : | BFE | |
| **Return value :** | | |
| Type : | Description | |
| none |  | |
| **Constraints :** | | |
| none | | |
| **Static aspect :** | | |
|  | | |

Diagram

Description automatically generated

Figure : PAL\_runReadMotorCurrentInmA diagram

### PAL\_runReadMotorCurrentInA

|  |  |  |
| --- | --- | --- |
| **Object :** | | |
| Function that provides the current in Ampere | | |
| **Input Parameters :** | | |
| Name | Type | Description |
| none |  |  |
| **Output Parameters :** | | |
| Name | Type | Description |
| ps8MotorCurrentInA | s8MotorCurrentInAType | Pointer used to store the requested result |
| **Dynamic Aspect :** | | |
| Called on request | | |
| Who : | BFE, PMP | |
| **Return value :** | | |
| Type : | Description | |
| none |  | |
| **Constraints :** | | |
| none | | |
| **Static aspect :** | | |
|  | | |

Diagram, schematic

Description automatically generated

Figure : PAL\_runReadMotorCurrentInA diagram

### PAL\_runSetPowerOrder

|  |  |  |
| --- | --- | --- |
| **Object :** | | |
| External service called by upper layer in order to drive the motor in case of belt function activation | | |
| **Input Parameters :** | | |
| Name | Type | Description |
| s16MotorPowerOrder | s16MotorPowerOrderType | Requested motor power order |
| u16BoostDutyCycle | u16BoostPWMOrderType | Requested Duty Cycle |
| **Output Parameters :** | | |
| Name | Type | Description |
| ps8MotorCurrentInA | s8MotorCurrentInAType | Pointer used to store the requested result |
| **Dynamic Aspect :** | | |
| Called on request | | |
| Who : | BFE | |
| **Return value :** | | |
| Type : | Description | |
| none |  | |
| **Constraints :** | | |
| none | | |
| **Static aspect :** | | |
|  | | |

Diagram, schematic

Description automatically generated

Figure : PAL\_runSetPowerOrder diagram

### PAL\_runGetPWMOrder

|  |  |  |
| --- | --- | --- |
| Object | | |
| This function shall provide the current applied power order. | | |
| **Prototype** | | |
| EXPORTED void PAL\_runGetPWMOrder (s16MotorPowerOrderType \*ps16MotorPowerOrder) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| ps16MotorPowerOrder | s16MotorPowerOrderType\* | Applied motor power order |
| **Return value** | | |
| Type | Description | |
| NA | - | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| \* | \* | |
| **Static aspect** | | |
|  | | |
| **Constrains** | | |
|  | | |
|  |  |  |

### PAL\_StartFreeWheelingState

|  |  |  |
| --- | --- | --- |
| Object | | |
| This function shall set the power H-bridge in free-wheeling state. | | |
| **Prototype** | | |
| EXPORTED void PAL\_StartFreeWheelingState (void) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Return value** | | |
| Type | Description | |
| NA | - | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| \* | \* | |
| **Static aspect** | | |
|  | | |

## Auto-Tests

### PAL\_Autotest\_CheckHWSelfProtection

|  |  |  |
| --- | --- | --- |
| **Object :** | | |
| This function has to test the Hardware self-protection on Half Bridges | | |
| **Input Parameters :** | | |
| Name | Type | Description |
| none |  |  |
| **Output Parameters :** | | |
| Name | Type | Description |
| pu8TestResult | u8TestResultType | Pointer used to store the auto test result |
| **Dynamic Aspect :** | | |
| Called on request | | |
| Who : | MMG | |
| **Return value :** | | |
| Type : | Description | |
| none |  | |
| **Constraints :** | | |
| none | | |
| **Static aspect :** | | |
|  | | |

Diagram

Description automatically generated

Figure : PAL\_Autotest\_CheckHWSelfProtection diagram

### PAL\_Autotest\_CheckMotorCurrent

|  |  |  |
| --- | --- | --- |
| **Object :** | | |
| This function has to test that a high motor current will not be applied for a too long time (scope is to protect motor and HW components) | | |
| **Input Parameters :** | | |
| Name | Type | Description |
| none |  |  |
| **Output Parameters :** | | |
| Name | Type | Description |
| pu8TestResult | u8TestResultType | Pointer used to store the auto test result |
| **Dynamic Aspect :** | | |
| Called on request | | |
| Who : | MMG | |
| **Return value :** | | |
| Type : | Description | |
| none |  | |
| **Constraints :** | | |
| none | | |
| **Static aspect :** | | |
|  | | |

Diagram

Description automatically generated

Figure : PAL\_Autotest\_CheckMotorCurrent diagram

### PAL\_Autotest\_CheckMotorThermalProctection

|  |  |  |
| --- | --- | --- |
| **Object :** | | |
| This function has to test that the motor is not too hot (high temperature will damage the motor and the electronic) | | |
| **Input Parameters :** | | |
| Name | Type | Description |
| none |  |  |
| **Output Parameters :** | | |
| Name | Type | Description |
| pu8TestResult | u8TestResultType | Pointer used to store the auto test result |
| **Dynamic Aspect :** | | |
| Called on request | | |
| Who : | MMG | |
| **Return value :** | | |
| Type : | Description | |
| none |  | |
| **Constraints :** | | |
| none | | |
| **Static aspect :** | | |
|  | | |

Diagram

Description automatically generated

Figure : PAL\_Autotest\_CheckMotorThermalProctection diagram

### PAL\_Autotest\_CheckCommandConsistency

|  |  |  |
| --- | --- | --- |
| Object | | |
| This function shall indicate if the power stage activation state is consistent with the order. | | |
| **Prototype** | | |
| void PAL\_Autotest\_CheckCommandConsistency (u8TestResultType \*pu8TestResult) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu8TestResult | u8TestResultType\* | Status of the test |
| **Return value** | | |
| Type | Description | |
| NA | - | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| \* | \* | |
| **Static aspect** | | |
|  | | |
| **Constrains** | | |
|  | | |
|  |  |  |
|  |  |  |

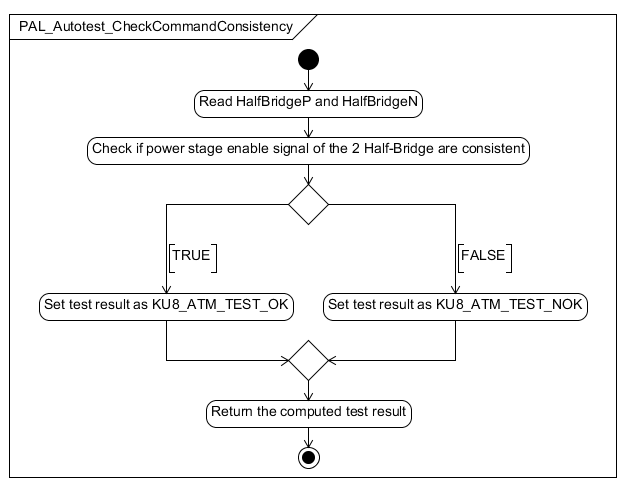


Figure PAL\_Autotest\_CheckCommandConsistency

### PAL\_Autotest\_CheckMosfetLowSC

|  |  |  |
| --- | --- | --- |
| Object | | |
| This function shall test if the low MOFSET of the power bridge are in short circuit. | | |
| **Prototype** | | |
| void PAL\_Autotest\_CheckMosfetLowSC (u8TestResultType \*pu8TestResult) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu8TestResult | u8TestResultType\* | Status of the test |
| **Return value** | | |
| Type | Description | |
| NA | - | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| \* | \* | |
| **Static aspect** | | |
|  | | |
| **Constrains** | | |
|  | | |
|  |  |  |
|  |  |  |

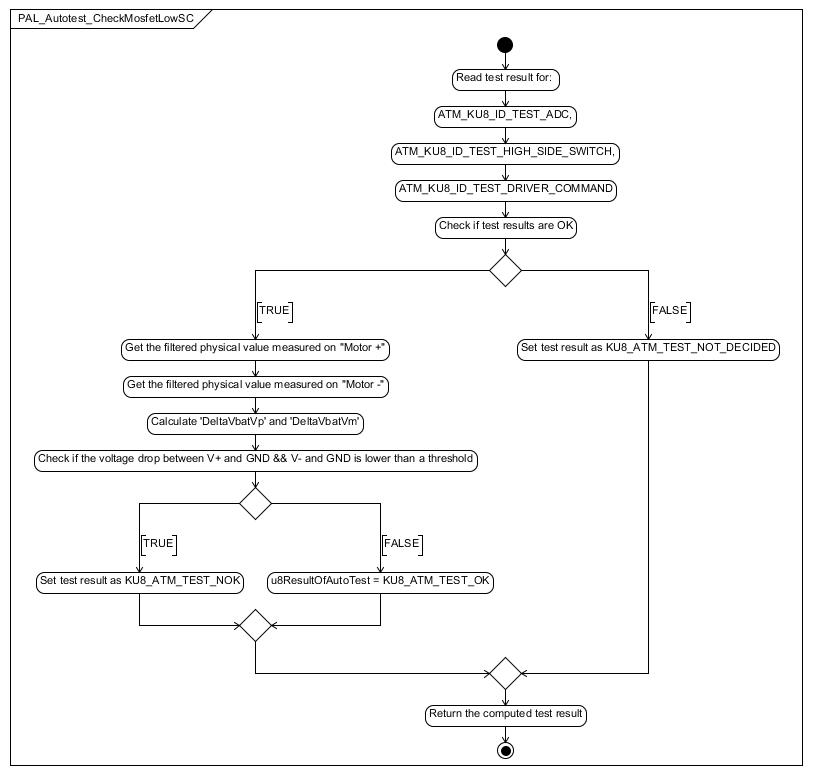


Figure PAL\_Autotest\_CheckMosfetLowSC

### PAL\_Autotest\_CheckMosfetHighSC

|  |  |  |
| --- | --- | --- |
| Object | | |
| This function shall test if the high MOFSET of the power bridge are in short circuit. | | |
| **Prototype** | | |
| void PAL\_Autotest\_CheckMosfetHighSC (u8TestResultType \*pu8TestResult) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu8TestResult | u8TestResultType\* | Status of the test |
| **Return value** | | |
| Type | Description | |
| NA | - | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| \* | \* | |
| **Static aspect** | | |
|  | | |
| **Constrains** | | |
|  | | |
|  |  |  |
|  |  |  |

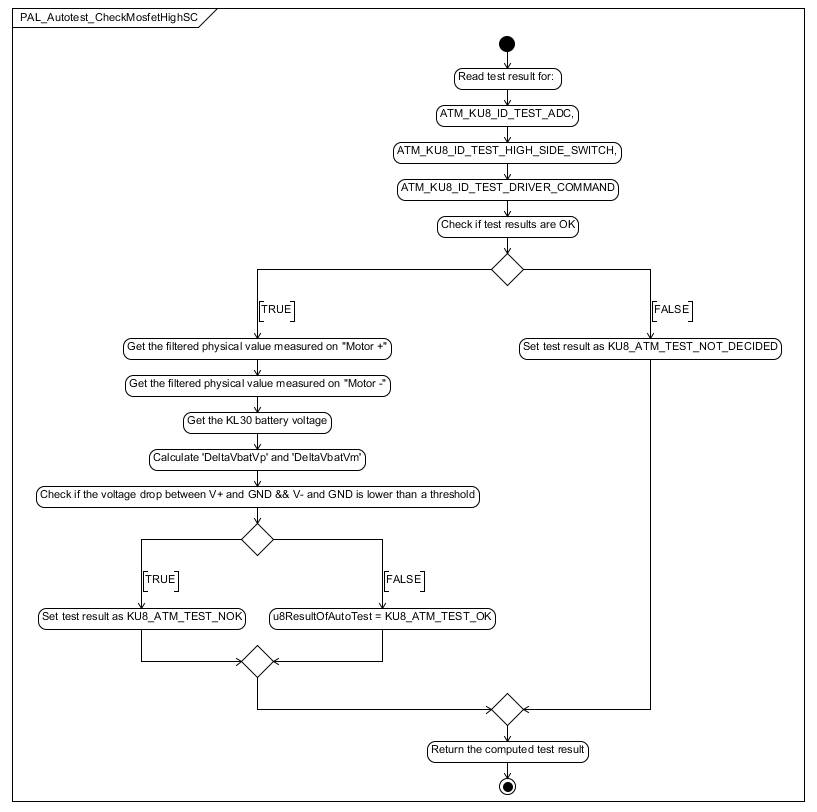


Figure PAL\_Autotest\_CheckMosfetHighSC

### PAL\_Autotest\_CheckMosfetOCAT

|  |  |  |
| --- | --- | --- |
| Object | | |
| This function shall test if MOFSET of the power bridge stay in open circuit when ordered | | |
| **Prototype** | | |
| void PAL\_Autotest\_CheckMosfetOCAT (u8TestResultType \*pu8TestResult) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu8TestResult | u8TestResultType\* | Status of the test |
| **Return value** | | |
| Type | Description | |
| NA | - | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| \* | \* | |
| **Static aspect** | | |
|  | | |
| **Constrains** | | |
|  | | |
|  |  |  |
|  |  |  |

Diagram

Description automatically generated

Figure PAL\_Autotest\_CheckMosfetOCAT

### PAL\_Autotest\_CheckHighSideSwRegulation

|  |  |  |
| --- | --- | --- |
| Object | | |
| This function shall checkif high switch regulation works well. | | |
| **Prototype** | | |
| void PAL\_Autotest\_CheckHighSideSwRegulation (u8TestResultType \*pu8TestResult) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu8TestResult | u8TestResultType\* | Status of the test |
| **Return value** | | |
| Type | Description | |
| NA | - | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| \* | \* | |
| **Static aspect** | | |
|  | | |
| **Constrains** | | |
|  | | |
|  |  |  |
|  |  |  |

Diagram

Description automatically generated

Figure PAL\_Autotest\_CheckHighSideSwRegulation

### PAL\_Autotest\_CheckMotorConnection

|  |  |  |
| --- | --- | --- |
| Object | | |
| This function shall test if the motor is well connected | | |
| **Prototype** | | |
| void PAL\_Autotest\_CheckMotorConnection (u8TestResultType \*pu8TestResult) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu8TestResult | u8TestResultType\* | Status of the test |
| **Return value** | | |
| Type | Description | |
| NA | - | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| \* | \* | |
| **Static aspect** | | |
|  | | |
| **Constrains** | | |
|  | | |
|  |  |  |
|  |  |  |

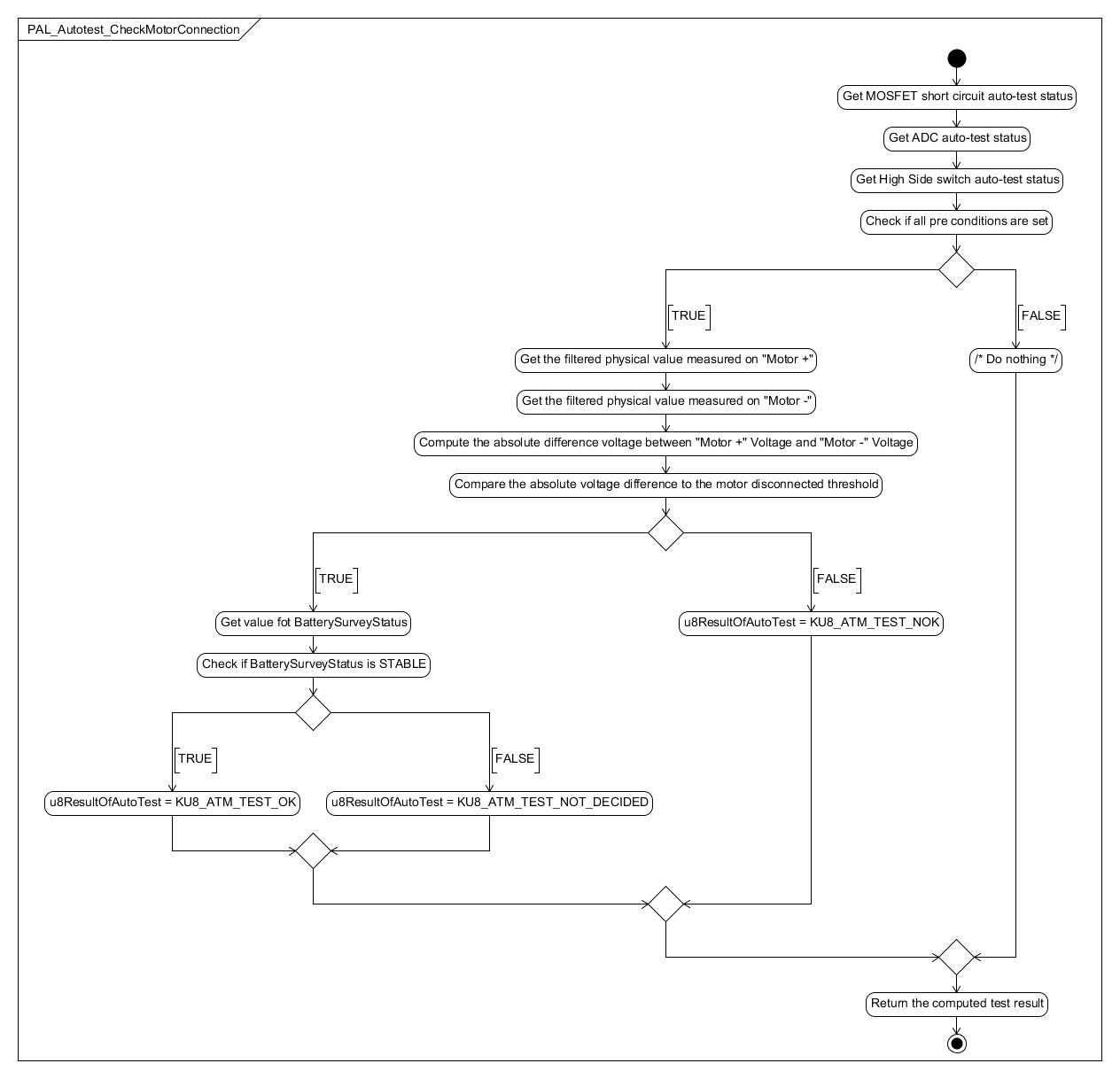


Figure PAL\_Autotest\_CheckMotorConnection

### PAL\_Autotest\_CheckMotorPowerOrder

|  |  |  |
| --- | --- | --- |
| Object | | |
| This function shall check if the PWM order is greater than 2 different thresholds during a specified time | | |
| **Prototype** | | |
| void PAL\_Autotest\_CheckMotorPowerOrder (u8TestResultType \*pu8TestResult) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu8TestResult | u8TestResultType\* | Status of the test |
| **Return value** | | |
| Type | Description | |
| NA | - | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| \* | \* | |
| **Static aspect** | | |
|  | | |
| **Constrains** | | |
|  | | |

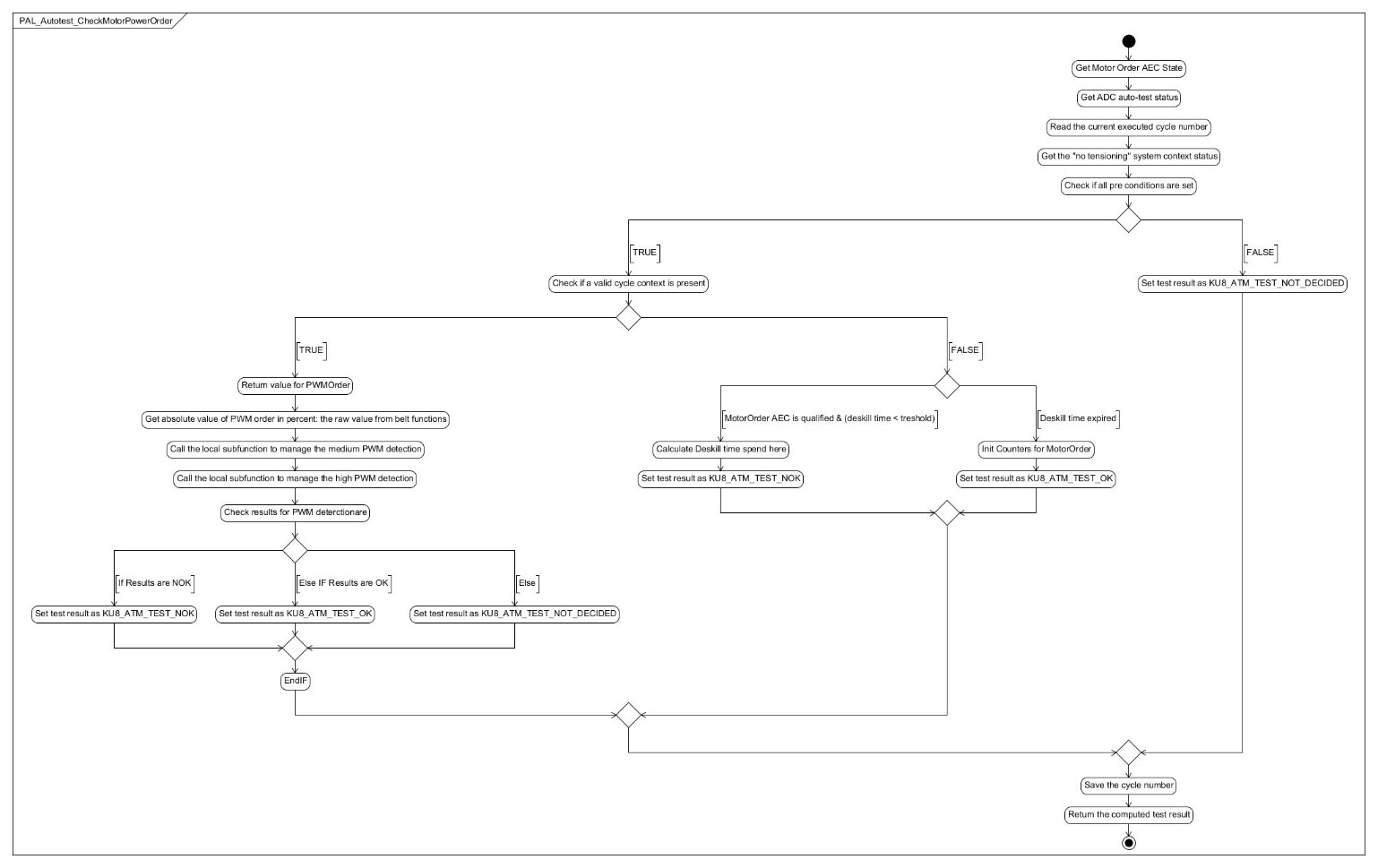


Figure PAL\_Autotest\_CheckMotorPowerOrder

### PAL\_Autotest\_CheckMotorDisengagement

|  |  |  |
| --- | --- | --- |
| Object | | |
| This function shall check whether the motor is blocked while performing a releasing mode | | |
| **Prototype** | | |
| void PAL\_Autotest\_CheckMotorDisengagement (u8TestResultType \*pu8TestResult) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu8TestResult | u8TestResultType\* | Status of the test |
| **Return value** | | |
| Type | Description | |
| NA | - | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| \* | \* | |
| **Static aspect** | | |
|  | | |
| **Constrains** | | |
|  | | |
|  |  |  |
|  |  |  |

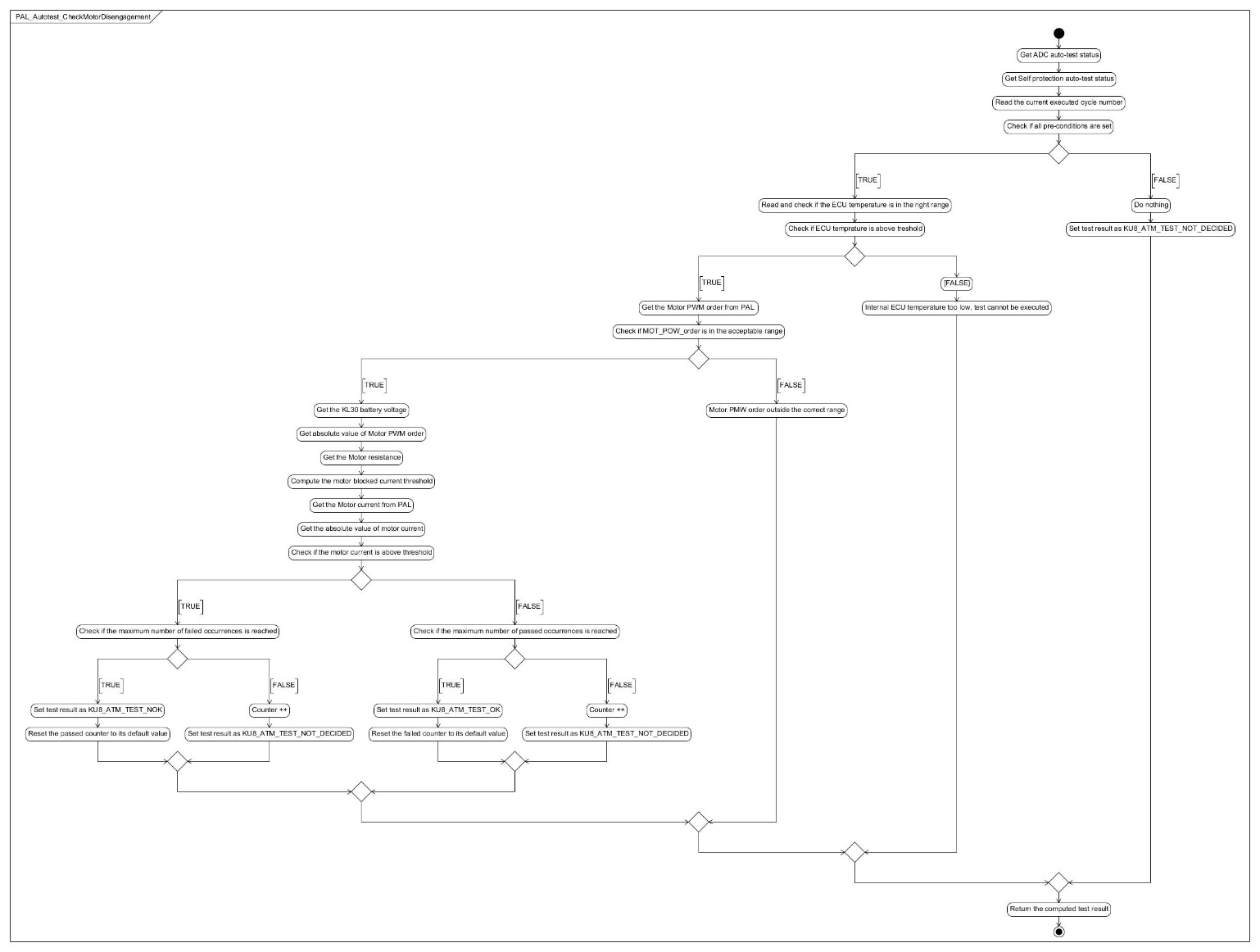


Figure PAL\_Autotest\_CheckMotorDisengagement

### PAL\_Autotest\_CheckMotorSC

|  |  |  |
| --- | --- | --- |
| Object | | |
| This function shall check if the motor is in short circuit during a start-up release cycle | | |
| **Prototype** | | |
| void PAL\_Autotest\_CheckMotorSC (u8TestResultType \*pu8TestResult) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| pu8TestResult | u8TestResultType\* | Status of the test |
| **Return value** | | |
| Type | Description | |
| NA | - | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| \* | \* | |
| **Static aspect** | | |
|  | | |
| **Constrains** | | |
|  | | |
|  |  |  |
|  |  |  |

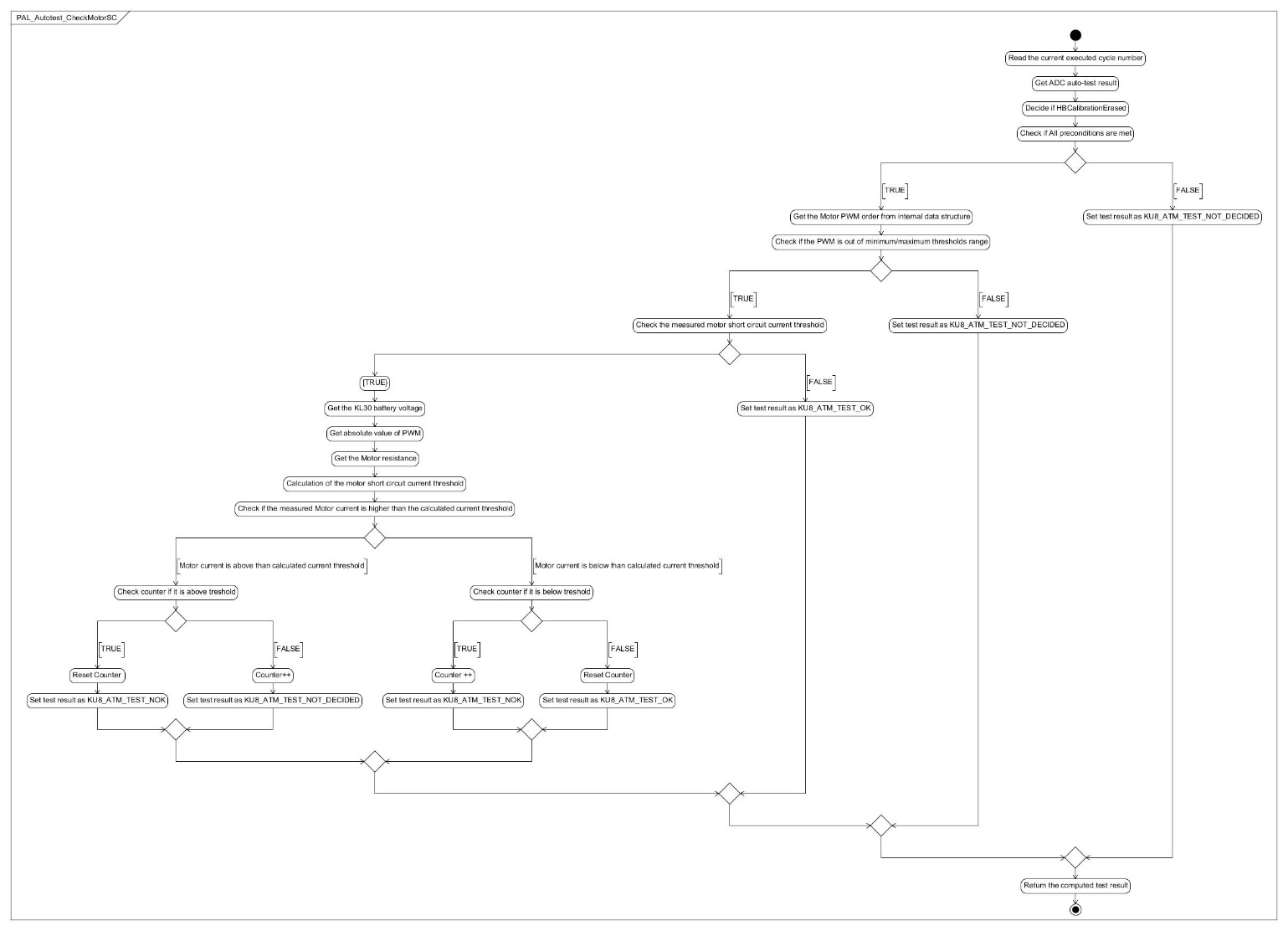


Figure PAL\_Autotest\_CheckMotorSC

## Internal local functions

### pal\_bIsPowerStageInSelfProtection

|  |  |  |
| --- | --- | --- |
| **Object :** | | |
| This function aims at detect the HW self-protection | | |
| **Input Parameters :** | | |
| Name | Type | Description |
| none |  |  |
| **Output Parameters :** | | |
| Name | Type | Description |
| none |  |  |
| **Dynamic Aspect :** | | |
| Called on request | | |
| Who : | PAL\_Autotest\_CheckHWSelfProtection() | |
| **Return value :** | | |
| Type : | Description | |
| boolean | The result value on uint8 with the ATM return value format  B\_TRUE : Self-protection detected  B\_FALSE: No self-protection detected | |
| **Constraints :** | | |
| none | | |
| **Static aspect :** | | |
|  | | |

Diagram

Description automatically generated

Figure : pal\_bIsPowerStageInSelfProtection diagram

### pal\_u8MotorCurrent\_ManageMediumCurrentFailureDetection

|  |  |  |
| --- | --- | --- |
| **Object :** | | |
| This function aims at manage a time counter and a reset counter to detect if the motor has been over a "medium current threshold" for more than the associated "medium current time threshold", since the last time the motor current was continuously under or equal to the "medium current threshold" for a "medium current reset time threshold". | | |
| **Input Parameters :** | | |
| Name | Type | Description |
| cu32PhysicalMotorCurrent\_mA | const uint32 | Physical Motor Current in mA |
| **Output Parameters :** | | |
| Name | Type | Description |
| none |  |  |
| **Dynamic Aspect :** | | |
| Called on request | | |
| Who : | PAL\_Autotest\_CheckMotorCurrent() | |
| **Return value :** | | |
| Type : | Description | |
| uint8 | The result value on uint8 with the ATM return value format  ATM\_U8\_NOK: failure detected  ATM\_U8\_NOT\_DECIDED: no failure detected | |
| **Constraints :** | | |
| none | | |
| **Static aspect :** | | |
|  | | |

Diagram

Description automatically generated

Figure : pal\_u8MotorCurrent\_ManageMediumCurrentFailureDetection diagram

### pal\_u8MotorCurrent\_ManageHighCurrentFailureDetection

|  |  |  |
| --- | --- | --- |
| **Object :** | | |
| This function aims at manage a time counter and a reset counter to detect if the motor has been over a "high current threshold" for more than the associated "high current time threshold", since the last time the motor current was continuously under or equal to the "high current threshold" for a "high current reset time threshold". | | |
| **Input Parameters :** | | |
| Name | Type | Description |
| cu32PhysicalMotorCurrent\_mA | const uint32 | Physical Motor Current in mA |
| **Output Parameters :** | | |
| Name | Type | Description |
| none |  |  |
| **Dynamic Aspect :** | | |
| Called on request | | |
| Who : | PAL\_Autotest\_CheckMotorCurrent() | |
| **Return value :** | | |
| Type : | Description | |
| uint8 | The result value on uint8 with the ATM return value format  ATM\_U8\_NOK: failure detected  ATM\_U8\_NOT\_DECIDED: no failure detected | |
| **Constraints :** | | |
| none | | |
| **Static aspect :** | | |
|  | | |

Diagram

Description automatically generated

Figure : pal\_u8MotorCurrent\_ManageHighCurrentFailureDetection diagram

### pal\_AT\_Init

|  |  |  |
| --- | --- | --- |
| **Object :** | | |
| This function initializes all static data used during PAL auto tests. | | |
| **Input Parameters :** | | |
| Name | Type | Description |
| none |  |  |
| **Output Parameters :** | | |
| Name | Type | Description |
| none |  |  |
| **Dynamic Aspect :** | | |
| Called on request | | |
| Who : | pal\_Cfg\_Init() | |
| **Return value :** | | |
| Type : | Description | |
| none |  | |
| **Constraints :** | | |
| none | | |
| **Static aspect :** | | |
|  | | |

A picture containing box and whisker chart

Description automatically generated

Figure : pal\_AT\_Init diagram

### pal\_Cfg\_Init

|  |  |  |
| --- | --- | --- |
| **Object :** | | |
| Initialization function for the configuration | | |
| **Input Parameters :** | | |
| Name | Type | Description |
| none |  |  |
| **Output Parameters :** | | |
| Name | Type | Description |
| none |  |  |
| **Dynamic Aspect :** | | |
| Called on request | | |
| Who : | PAL\_Init() | |
| **Return value :** | | |
| Type : | Description | |
| none |  | |
| **Constraints :** | | |
| none | | |
| **Static aspect :** | | |
|  | | |

Diagram, schematic

Description automatically generated

Figure : pal\_Cfg\_Init diagram

### pal\_ComputeCurrentHalfBridgeP

|  |  |  |
| --- | --- | --- |
| **Object :** | | |
| Function that provides the current in Milli Ampere for Half Bridge P | | |
| **Input Parameters :** | | |
| Name | Type | Description |
| none |  |  |
| **Output Parameters :** | | |
| Name | Type | Description |
| ps32MeasuredCurrent | sint32 \* const | Measured current in mA |
| **Dynamic Aspect :** | | |
| Called on request | | |
| Who : | PAL\_runReadMotorCurrentInmA() | |
| **Return value :** | | |
| Type : | Description | |
| none |  | |
| **Constraints :** | | |
| none | | |
| **Static aspect :** | | |
|  | | |

Diagram

Description automatically generated

Figure : pal\_ComputeCurrentHalfBridgeP diagram

### pal\_ComputeCurrentHalfBridgeN

|  |  |  |
| --- | --- | --- |
| **Object :** | | |
| Function that provides the current in Milli Ampere for HN | | |
| **Input Parameters :** | | |
| Name | Type | Description |
| none |  |  |
| **Output Parameters :** | | |
| Name | Type | Description |
| ps32MeasuredCurrent | sint32 \* const | Measured current in mA |
| **Dynamic Aspect :** | | |
| Called on request | | |
| Who : | PAL\_runReadMotorCurrentInmA() | |
| **Return value :** | | |
| Type : | Description | |
| none |  | |
| **Constraints :** | | |
| none | | |
| **Static aspect :** | | |
|  | | |

Diagram

Description automatically generated

Figure : pal\_ComputeCurrentHalfBridgeN diagram

### pal\_StopMotorRotation

|  |  |  |
| --- | --- | --- |
| **Object :** | | |
| Function used to stop the motor rotation | | |
| **Input Parameters :** | | |
| Name | Type | Description |
| none |  |  |
| **Output Parameters :** | | |
| Name | Type | Description |
| none |  |  |
| **Dynamic Aspect :** | | |
| Called on request | | |
| Who : | PAL\_runSetPowerOrder() | |
| **Return value :** | | |
| Type : | Description | |
| none |  | |
| **Constraints :** | | |
| none | | |
| **Static aspect :** | | |
|  | | |

A picture containing box and whisker chart

Description automatically generated

Figure : pal\_StopMotorRotation diagram

### pal\_DriveMotorInTensioning

|  |  |  |
| --- | --- | --- |
| **Object :** | | |
| Function used to drive the motor in tensioning direction | | |
| **Input Parameters :** | | |
| Name | Type | Description |
| cu16AbsoluteMotorCommand | const uint16 | Motor command |
| cu16BoosterMotorCommand | const uint16 | Motor booster command |
| **Output Parameters :** | | |
| Name | Type | Description |
| none |  |  |
| **Dynamic Aspect :** | | |
| Called on request | | |
| Who : | PAL\_runSetPowerOrder() | |
| **Return value :** | | |
| Type : | Description | |
| none |  | |
| **Constraints :** | | |
| none | | |
| **Static aspect :** | | |
|  | | |

Diagram

Description automatically generated

Figure : pal\_DriveMotorInTensioning diagram

### pal\_DriveMotorInReleasing

|  |  |  |
| --- | --- | --- |
| **Object :** | | |
| Function used to drive the motor in releasing direction | | |
| **Input Parameters :** | | |
| Name | Type | Description |
| cu16AbsoluteMotorCommand | const uint16 | Motor command |
| **Output Parameters :** | | |
| Name | Type | Description |
| none |  |  |
| **Dynamic Aspect :** | | |
| Called on request | | |
| Who : | PAL\_runSetPowerOrder() | |
| **Return value :** | | |
| Type : | Description | |
| none |  | |
| **Constraints :** | | |
| none | | |
| **Static aspect :** | | |
|  | | |

Diagram

Description automatically generated

Figure : pal\_DriveMotorInReleasing diagram

### pal\_DisablePowerStage

|  |  |  |
| --- | --- | --- |
| **Object :** | | |
| External service that will disable the HW components that drive the motor | | |
| **Input Parameters :** | | |
| Name | Type | Description |
| none |  |  |
| **Output Parameters :** | | |
| Name | Type | Description |
| none |  |  |
| **Dynamic Aspect :** | | |
| Called on request | | |
| Who : | PAL\_runMainFunction | |
| **Return value :** | | |
| Type : | Description | |
| none |  | |
| **Constraints :** | | |
| none | | |
| **Static aspect :** | | |
|  | | |

Diagram, schematic

Description automatically generated

Figure : pal\_DisablePowerStage diagram

### pal\_u8MotorOrder\_ManageMediumPWMFailureDetection

|  |  |  |
| --- | --- | --- |
| Object | | |
| This function aims at manage a time counter and a reset counter to detect if the PWM order has been over a "medium current threshold" for more than the associated "medium current time threshold", since the last time the PWM order was continuously under or equal to the "medium current threshold" for a "medium current reset time threshold". | | |
| **Prototype** | | |
| LOCAL uint8 pal\_u8MotorOrder\_ManageMediumPWMFailureDetection (const uint8 ku8AbsoluteValue\_MotorPWMOrder\_Percent) | | |
| **Input parameters** | | |
| Name | Type | Description |
| ku8AbsoluteValue\_MotorPWMOrder\_Percent | const uint8 | The motor pwm value |
| **Output parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Return value** | | |
| Type | Description | |
| uint8 | The result value | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| PAL\_Autotest\_CheckMotorPowerOrder | \* | |
| **Static aspect** | | |
|  | | |
| **Constrains** | | |
|  | | |
|  |  |  |

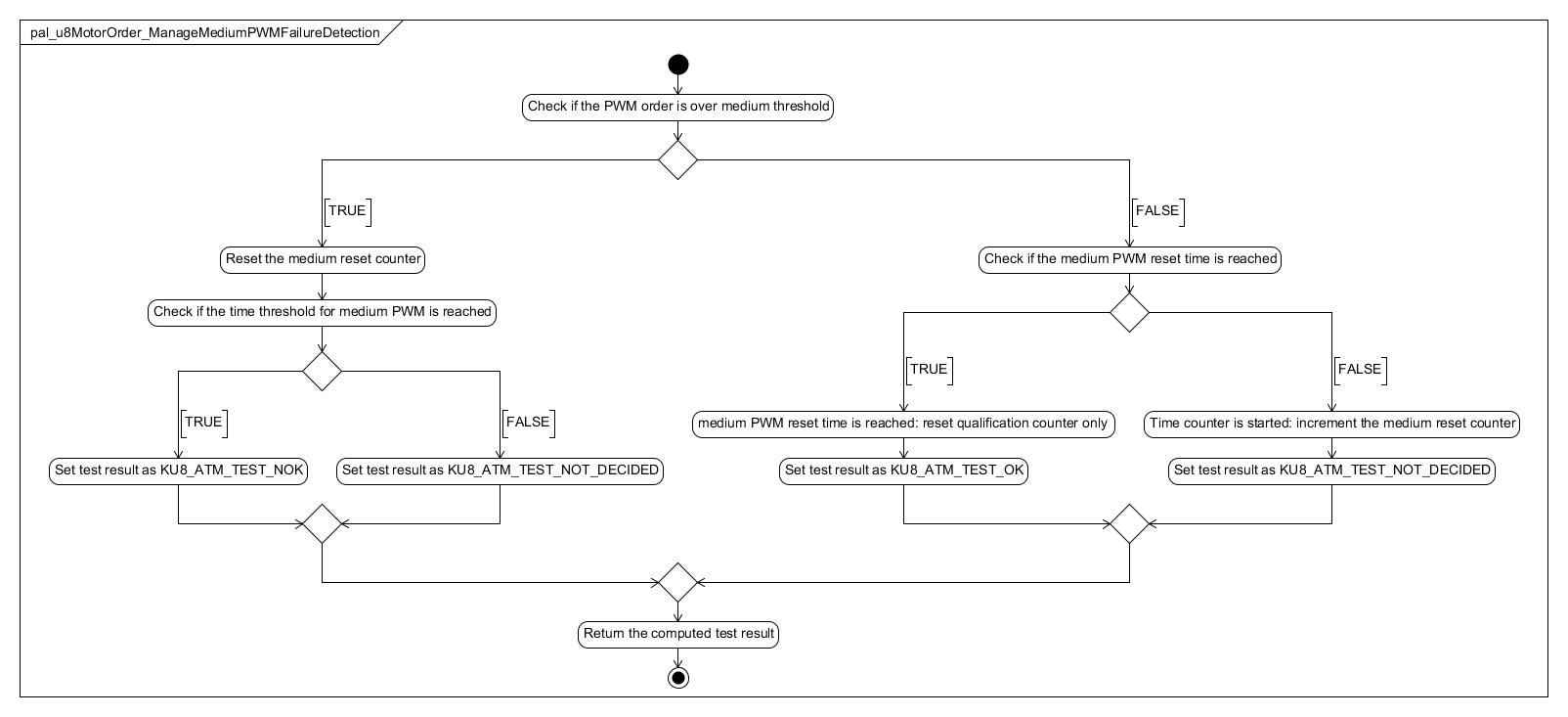


Figure pal\_u8MotorOrder\_ManageMediumPWMFailureDetection

### pal\_u8MotorOrder\_ManageHighPWMFailureDetection

|  |  |  |
| --- | --- | --- |
| Object | | |
| This function aims at manage a time counter and a reset counter to detect if the PWM order has been over a "high current threshold" for more than the associated "high current time threshold", since the last time the PWM order was continuously under or equal to the "high current threshold" for a "high current reset time threshold". | | |
| **Prototype** | | |
| LOCAL uint8 pal\_u8MotorOrder\_ManageHighPWMFailureDetection (const uint8 ku8AbsoluteValue\_MotorPWMOrder\_Percent) | | |
| **Input parameters** | | |
| Name | Type | Description |
| ku8AbsoluteValue\_MotorPWMOrder\_Percent | const uint8 | Motor Pwm value |
| **Output parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Return value** | | |
| Type | Description | |
| uint8 | The result value on uint8 with the ATM return value | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| PAL\_Autotest\_CheckMotorPowerOrder | \* | |
| **Static aspect** | | |
|  | | |
| **Constrains** | | |
|  | | |
|  |  |  |

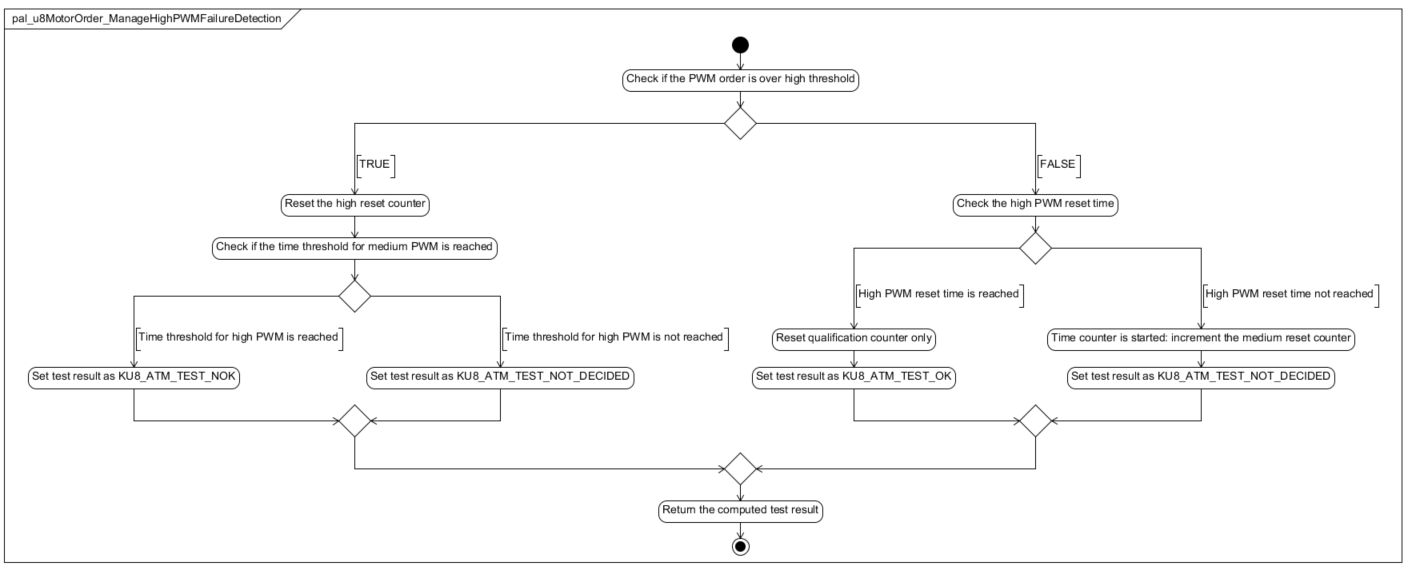


Figure pal\_u8MotorOrder\_ManageHighPWMFailureDetection

### pal\_HighSideSWRegFreeWheelingPhase

|  |  |  |
| --- | --- | --- |
| Object | | |
| This function is called in case of a failed occurrence of HS SW auto test | | |
| **Prototype** | | |
| LOCAL void pal\_HighSideSWRegFreeWheelingPhase (void) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Return value** | | |
| Type | Description | |
| NA | - | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| PAL\_Autotest\_CheckHighSideSwRegulation | \* | |
| **Static aspect** | | |
|  | | |
| **Constrains** | | |
|  | | |
|  |  |  |
|  |  |  |

### pal\_u8HighSideSWRegStep\_1

|  |  |  |
| --- | --- | --- |
| Object | | |
| This function has to manage the step 1 for HS SW auto test | | |
| **Prototype** | | |
| LOCAL uint8 pal\_u8HighSideSWRegStep\_1 (void) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Return value** | | |
| Type | Description | |
| uint8 | The test result | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| PAL\_Autotest\_CheckHighSideSwRegulation | \* | |
| **Static aspect** | | |
|  | | |
| **Constrains** | | |
|  | | |
|  |  |  |
|  |  |  |

### pal\_u8HighSideSWRegStep\_2

|  |  |  |
| --- | --- | --- |
| Object | | |
| This function has to manage the step 2 for HS SW auto test | | |
| **Prototype** | | |
| LOCAL uint8 pal\_u8HighSideSWRegStep\_2 (void) | | |
| **Input parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Output parameters** | | |
| Name | Type | Description |
| NA | NA | NA |
| **Return value** | | |
| Type | Description | |
| uint8 | The test result | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| PAL\_Autotest\_CheckHighSideSwRegulation | \* | |
| **Static aspect** | | |
|  | | |
| **Constrains** | | |
|  | | |
|  |  |  |
|  |  |  |

## Types

### PAL\_tHBridgeDefinition

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Field description** |
| u16LinarSlopeDutyCycle | uint16 | Linear Slope Duty Cycle |
| s16LinarOffsetDutyCycle | sint16 | Linear Offset Duty Cycle |
| u16IdleDutyCycleLevel | uint16 | Idle Duty Cycle Level |
| u16ReadEnableDioId | uint16 | Read Enable Dio Id |
| u16DriveEnableDioId | uint16 | Drive Enable Dio Id |
| u8PwmChannelId | uint8 | Pwm Channel Id |

### PAL\_tHBridgeCurrentValues

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Field description** |
| u32SlopeToApplyNumerator | uint32 | Slope To Apply Numerator |
| s32OffsetToApplyNumerator | sint32 | Offset To Apply Numerator |
| s16RawDutyCycleCommand | sint16 | Raw Duty Cycle Command |
| s16AdaptedDutyCycleCommand | sint16 | Adapted Duty Cycle Command |
| u8EnableCommand | uint8 | Enable Command |
| bIsPwrStgInSelfProtection | boolean | Driver Is PwrStg In Self Protection |

### PAL\_tBoostDriverDefinition

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Field description** |
| u16IdleDutyCycleLevel | uint16 | Idle Duty Cycle Level |
| u8PwmChannelId | uint8 | Pwm Channel Id |
| u8PwmCPChannelId | uint8 | Pwm CP Channel Id |
| u8DioChannelHighId | uint16 | Dio Channel High Id |
| u8DioChannelLowId | uint16 | Dio Channel Low Id |

### PAL\_tFullBridgeDefinition

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Field description** |
| u16LinarSlopeNumDutyCycle | uint16 | Linear Slope Num Duty Cycle |
| u16LinarSlopeDenDutyCycle | uint16 | Linear Slope Den Duty Cycle |
| u16LinarOffsetNumDutyCycle | uint16 | Linear Offset Num Duty Cycle |
| u16LinarOffsetDenDutyCycle | uint16 | Linear Offset Den Duty Cycle |
| u16IdleDutyCycleLevel | uint16 | Idle Duty Cycle Level |
| u16ReadEnableDioId | uint16 | Read Enable Dio Id |
| u16DriveEnableDioId | uint16 | Drive Enable Dio Id |
| u8DirectionDioId | uint8 | Direction Dio Id |
| u8PwmChannelId | uint8 | Pwm Channel Id |

### PAL\_tPrelBoostTestData

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Field description** |
| u16PreBoostDeadTimeCounter | uint16 | Pre Boost Dead Time Counter |
| u8PreliminaryBoostTestState | uint8 | Preliminary Boost Test State |
| u8PreBoostStep1State | uint8 | Pre Boost Step 1 State |
| u8PreBoostStep2State | uint8 | Pre Boost Step 2 State |
| u8PreBoostStep3State | uint8 | Pre Boost Step 3 State |
| u8PreBoostStep4State | uint8 | Pre Boost Step 4 State |
| u8PreBoostNbTries | uint8 | Pre Boost Number of Tries |
| u8PreBoostStep1ChargeCounter | uint8 | Pre Boost Step 1 Charge Counter |
| u8PreBoostStep1DischargeCounter | uint8 | Pre Boost Step 1 Discharge Counter |

### PAL\_tVBoostTestData

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Field description** |
| u16BoostStep1VPMotorStart | uint16 | Boost Step 1 VP Motor Start |
| u16DeadTimeCounter | uint16 | Dead Time Counter |
| u8BoostTestState | uint8 | Boost Test State |
| u8BoostStep1State | uint8 | Boost Step 1 State |
| u8BoostStep2State | uint8 | Boost Step 2 State |
| u8BoostStep1StateMachineCounter | uint8 | Boost Step 1 State Machine Counter |
| u8BoostStep2StateMachineCounter | uint8 | Boost Step 2 State Machine Counter |
| u8BoostNbrOfTryCounter | uint8 | Boost Number Of Try Counter |
| u8BoostBatteryConditions | uint8 | Boost Battery Conditions |
| u8LostBatteryConditionsCounter | uint8 | Lost Battery Conditions Counter |

### PAL\_tAutoTestData

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Field description** |
| stPreliminaryBoostData | PAL\_tPrelBoostTestData | Preliminary Boost Data |
| stVboostData | PAL\_tVBoostTestData | Vboost Data |
| u32MotorOrderMediumThrsTimeCounter | uint32 | Motor Order Medium Thrs Time Counter |
| u32MotorCurrentMediumThrsTimeCounter | uint32 | Motor Current Medium Thrs Time Counter |
| u16HighSideSw\_DeadTimeCounter | uint16 | High Side Sw Dead Time Counter |
| u16MotorOrderMediumThrsTimeResetCounter | uint16 | Motor Order Medium Thrs Time Reset Counter |
| u16MotorCurrentMediumThrsTimeResetCounter | uint16 | Motor Current Medium Thrs Time Reset Counter |
| u16MotorOrderHighThrsTimeCounter | uint16 | Motor Order High Thrs Time Counter |
| u16MotorCurrentHighThrsTimeCounter | uint16 | Motor Current High Thrs Time Counter |
| u16MotorOrderHighThrsTimeResetCounter | uint16 | Motor Order High Thrs Time Reset Counter |
| u16MotorCurrentHighThrsTimeResetCounter | uint16 | Motor Current High Thrs Time Reset Counter |
| u16MotorOrderErrorResetTimeCounter | uint16 | Motor Order Error Reset Time Counter |
| u16MotorCurrentErrorResetTimeCounter | uint16 | Motor Current Error Reset Time Counter |
| u8PowerOrderExecutedCycleMemory | uint8 | Power Order Executed Cycle Memory |
| u8PowerCurrentExecutedCycleMemory | uint8 | Power Current Executed Cycle Memory |
| u8MosfetOC\_State | uint8 | Mosfet OC State |
| u8MosfetOC\_NbTries | uint8 | Mosfet OC Number of Tries |
| u8MotorBlockedFailedCntr | uint8 | Motor Blocked Failed Counter |
| u8MotorBlockedPassedCntr | uint8 | Motor Blocked Passed Counter |
| u8MotorSCFailedCntr | uint8 | Motor SC Failed Counter |
| u8MotorSCPassedCntr | uint8 | Motor SC Passed Counter |
| u8HighSideSw\_TestState | uint8 | High Side Sw Test State |
| u8HighSideSw\_NbTries | uint8 | High Side Sw Number of Tries |
| u8HighSideSw\_FreeWheelingTimer | uint8 | High Side Sw Free Wheeling Timer |

### PAL\_tTopologyConfiguration

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Field description** |
| stHBridgeDef | PAL\_tHBridgeDefinition | H-Bridge Definition |
| stBoosterDef | PAL\_tBoostDriverDefinition | Booster Definition |
| stFullBridgeDef | PAL\_tFullBridgeDefinition | Full Bridge Definition |

## Variables

### PAL\_kstHardWareTopology

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| PAL\_tTopologyConfiguration | N.A. | |
| **Description** | | |
| Hardware topology used by ECU | | |
| **Definition** | | |
| const PAL\_tTopologyConfiguration | | |

### PAL\_u8RequestedMode

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| uint8 | N.A. | |
| **Description** | | |
| Global data used to set the state machine status for the PAL component | | |
| **Definition** | | |
| static uint8 PAL\_u8RequestedMode | | |

### PAL\_u8ModeState

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| uint8 | N.A. | |
| **Description** | | |
| Global data used to know the state machine status for the PAL component | | |
| **Definition** | | |
| uint8 PAL\_u8ModeState | | |

### PAL\_u8FreeWheelingTimer

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| uint8 | N.A. | |
| **Description** | | |
| Timer used to set the state machine status for the PAL component | | |
| **Definition** | | |
| uint8 PAL\_u8FreeWheelingTimer | | |

### PAL\_Cfg\_InternalData

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| PAL\_tHBridgeCurrentValues | N.A. | |
| **Description** | | |
| This variable is used to store static data | | |
| **Definition** | | |
| PAL\_tHBridgeCurrentValues PAL\_Cfg\_InternalData | | |

### PAL\_Cfg\_AT\_InternalData

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| PAL\_tAutoTestData | N.A. | |
| **Description** | | |
| This variable is used to store autotests data | | |
| **Definition** | | |
| PAL\_tAutoTestData PAL\_Cfg\_AT\_InternalData | | |

## Constants

### PAL\_CFG\_TOPO\_HBRIDGE\_NB

|  |  |
| --- | --- |
| Name | Value |
| H-Bridge number | 2 |
| **Definition** | |
| #define PAL\_CFG\_TOPO\_HBRIDGE\_NB | |
| **Description** | |
| Number of the H-Bridges | |

### PAL\_CFG\_KU8\_HB\_P\_ID

|  |  |
| --- | --- |
| Name | Value |
| H-Bridge P Id | 0 |
| **Definition** | |
| #define PAL\_CFG\_KU8\_HB\_P\_ID | |
| **Description** | |
| ID for the first H-Bridge (P) | |

### PAL\_CFG\_KU8\_HB\_N\_ID

|  |  |
| --- | --- |
| Name | Value |
| H-Bridge N Id | 1 |
| **Definition** | |
| #define PAL\_CFG\_KU8\_HB\_N\_ID | |
| **Description** | |
| ID for the second H-Bridge (N) | |

### PAL\_CFG\_TOPO\_SHUNT\_NB

|  |  |
| --- | --- |
| Name | Value |
| Shunt number | 0 |
| **Definition** | |
| #define PAL\_CFG\_TOPO\_SHUNT\_NB | |
| **Description** | |
| Macro to enable/disable the booster function | |

### PAL\_CFG\_TOPO\_FULL\_BRIDGE\_NB

|  |  |
| --- | --- |
| Name | Value |
| Full bridge number | 0 |
| **Definition** | |
| #define PAL\_CFG\_TOPO\_FULL\_BRIDGE\_NB | |
| **Description** | |
| Macro to enable/disable the booster function | |

### PAL\_CFG\_FREEWHEELING\_TIME

|  |  |
| --- | --- |
| Name | Value |
| Freewheeling time | 100 |
| **Definition** | |
| #define PAL\_CFG\_FREEWHEELING\_TIME | |
| **Description** | |
| Configuration time for the freewheeling functionality | |

### PAL\_CFG\_PERIODIC\_FCT\_CALL

|  |  |
| --- | --- |
| Name | Value |
| Period function call | 2 |
| **Definition** | |
| #define PAL\_CFG\_PERIODIC\_FCT\_CALL | |
| **Description** | |
| Periodicity of the main function (in ms) | |

### PAL\_CFG\_HIGH\_LIMIT\_LLD\_CMD

|  |  |
| --- | --- |
| Name | Value |
| High limit LLD Command | 0x8000 |
| **Definition** | |
| #define PAL\_CFG\_HIGH\_LIMIT\_LLD\_CMD | |
| **Description** | |
| Output limits for duty cycle values sent to PWM driver | |

### PAL\_CFG\_DUTY\_CYCLE\_ADAPT\_DEN

|  |  |
| --- | --- |
| Name | Value |
| Configured Duty Cycle DEN adapt | 625 |
| **Definition** | |
| #define PAL\_CFG\_DUTY\_CYCLE\_ADAPT\_DEN | |
| **Description** | |
| Constant value used to adapt duty cycle to HW constraints | |

### PAL\_CFG\_DUTY\_CYCLE\_ADAPT\_ROUND

|  |  |
| --- | --- |
| Name | Value |
| Configured Duty Cycle round adapt | 312 |
| **Definition** | |
| #define PAL\_CFG\_DUTY\_CYCLE\_ADAPT\_ROUND | |
| **Description** | |
| Constant value used to adapt duty cycle to HW constraints | |

### PAL\_CFG\_DUTY\_CYCLE\_ADAPT\_A\_NUM

|  |  |
| --- | --- |
| Name | Value |
| Configured Duty Cycle A Number adapt | 32 |
| **Definition** | |
| #define PAL\_CFG\_DUTY\_CYCLE\_ADAPT\_A\_NUM | |
| **Description** | |
| Constant value used to adapt duty cycle to HW constraints | |

### PAL\_CFG\_DUTY\_CYCLE\_ADAPT\_B\_NUM

|  |  |
| --- | --- |
| Name | Value |
| Configured Duty Cycle B Number adapt | 2048 |
| **Definition** | |
| #define PAL\_CFG\_DUTY\_CYCLE\_ADAPT\_B\_NUM | |
| **Description** | |
| Constant value used to adapt duty cycle to HW constraints | |

### PAL\_CFG\_SLOPE\_CTRL\_MAX\_VALUE

|  |  |
| --- | --- |
| Name | Value |
| Configured Slope control maximum value | 200 |
| **Definition** | |
| #define PAL\_CFG\_SLOPE\_CTRL\_MAX\_VALUE | |
| **Description** | |
| Constant value used to adapt duty cycle to HW constraints | |

### PAL\_CFG\_SLOPE\_DEFAULT\_VALUE

|  |  |
| --- | --- |
| Name | Value |
| Configured Slope default value | 100 |
| **Definition** | |
| #define PAL\_CFG\_SLOPE\_DEFAULT\_VALUE | |
| **Description** | |
| Constant value used to adapt duty cycle to HW constraints | |

### PAL\_CFG\_OFFSET\_CTRL\_MAX\_VALUE

|  |  |
| --- | --- |
| Name | Value |
| Configured Control Offset maximum value | 10000 |
| **Definition** | |
| #define PAL\_CFG\_OFFSET\_CTRL\_MAX\_VALUE | |
| **Description** | |
| Constant value used to adapt duty cycle to HW constraints | |

### PAL\_CFG\_OFFSET\_CTRL\_MIN\_VALUE

|  |  |
| --- | --- |
| Name | Value |
| Configured Control Offset minimum value | -10000 |
| **Definition** | |
| #define PAL\_CFG\_OFFSET\_CTRL\_MIN\_VALUE | |
| **Description** | |
| Constant value used to adapt duty cycle to HW constraints | |

### PAL\_CFG\_OFFSET\_DEFAULT\_VALUE

|  |  |
| --- | --- |
| Name | Value |
| Configured Offset default value | 0 |
| **Definition** | |
| #define PAL\_CFG\_OFFSET\_DEFAULT\_VALUE | |
| **Description** | |
| Constant value used to adapt duty cycle to HW constraints | |

### KU8\_NB\_INTERP\_PTS\_HB\_CALIB

|  |  |
| --- | --- |
| Name | Value |
| Interp. Points Half-Bridge calibration | 5 |
| **Definition** | |
| #define KU8\_NB\_INTERP\_PTS\_HB\_CALIB | |
| **Description** | |
| Constant value used in H-Bridge calibration | |

### PAL\_KU8\_INIT\_MODE\_STATE

|  |  |
| --- | --- |
| Name | Value |
| Init Mode State | 0x00 |
| **Definition** | |
| #define PAL\_KU8\_INIT\_MODE\_STATE | |
| **Description** | |
| Mask used to store the state machine status for PAL component | |

### PAL\_KU8\_DIAGNOSTIC\_MODE\_STATE

|  |  |
| --- | --- |
| Name | Value |
| Diagnostic Mode State | 0x01 |
| **Definition** | |
| #define PAL\_KU8\_DIAGNOSTIC\_MODE\_STATE | |
| **Description** | |
| Mask used to store the state machine status for PAL component | |

### PAL\_KU8\_IDLE\_MODE\_STATE

|  |  |
| --- | --- |
| Name | Value |
| Idle Mode State | 0x02 |
| **Definition** | |
| #define PAL\_KU8\_IDLE\_MODE\_STATE | |
| **Description** | |
| Mask used to store the state machine status for PAL component | |

### PAL\_KU8\_ACTIVATION\_MODE\_STATE

|  |  |
| --- | --- |
| Name | Value |
| Activation Mode State | 0x04 |
| **Definition** | |
| #define PAL\_KU8\_ACTIVATION\_MODE\_STATE | |
| **Description** | |
| Mask used to store the state machine status for PAL component | |

### PAL\_KU8\_OFF\_MODE\_STATE

|  |  |
| --- | --- |
| Name | Value |
| Off Mode State | 0x08 |
| **Definition** | |
| #define PAL\_KU8\_OFF\_MODE\_STATE | |
| **Description** | |
| Mask used to store the state machine status for PAL component | |

### PAL\_KU16\_MAX\_ADC\_VALUE

|  |  |
| --- | --- |
| Name | Value |
| Maximum ADC value | 1024 |
| **Definition** | |
| #define PAL\_KU16\_MAX\_ADC\_VALUE | |
| **Description** | |
| Maximum threshold for ADC value | |

### PAL\_KU16\_MAX\_ANALOG\_CH\_VALUE

|  |  |
| --- | --- |
| Name | Value |
| Maximum Analog Channel Value | 5000 |
| **Definition** | |
| #define PAL\_KU16\_MAX\_ANALOG\_CH\_VALUE | |
| **Description** | |
| Maximum voltage (mV) that can be measured directly by the ADC | |

### PAL\_KU16\_DEFAULT\_HW\_SELF\_PROT\_THRS

|  |  |
| --- | --- |
| Name | Value |
| Default HW Self Protection Threshold | 819 |
| **Definition** | |
| #define PAL\_KU16\_DEFAULT\_HW\_SELF\_PROT\_THRS | |
| **Description** | |
| Default value (in absolute number) for the threshold used in HW Self Protection AT | |

### PAL\_KU32\_CONVERT\_A\_TO\_MA

|  |  |
| --- | --- |
| Name | Value |
| Convert Ampere to milli Ampere | 1000 |
| **Definition** | |
| #define PAL\_KU32\_CONVERT\_A\_TO\_MA | |
| **Description** | |
| Macro to convert Ampere in milli Ampere | |

### PAL\_KU8\_PERIODICITY\_10\_MS

|  |  |
| --- | --- |
| Name | Value |
| 10 ms Periodicity | 10 |
| **Definition** | |
| #define PAL\_KU8\_PERIODICITY\_10\_MS | |
| **Description** | |
| Periodicity of the Motor Current AT | |

### PAL\_KU8\_CONVERT\_CENTIAMP\_TO\_MILLIAMP

|  |  |
| --- | --- |
| Name | Value |
| Convert centi Ampere to milli Ampere | 10 |
| **Definition** | |
| #define PAL\_KU8\_CONVERT\_CENTIAMP\_TO\_MILLIAMP | |
| **Description** | |
| Macro to convert centi Ampere in milli Ampere | |

### PAL\_S32\_CONVERT\_A\_TO\_MA

|  |  |
| --- | --- |
| Name | Value |
| Convert Ampere to milli Ampere | 1000 |
| **Definition** | |
| #define PAL\_KU32\_CONVERT\_A\_TO\_MA | |
| **Description** | |
| Macro to convert Ampere in milli Ampere | |

### PAL\_KU8\_OCAT\_PREPARE\_STEP\_1

|  |  |
| --- | --- |
| Name | Value |
| Used for MOSFET Open Circuit state machine | 1 |
| **Definition** | |
| #define PAL\_KU8\_OCAT\_PREPARE\_STEP\_1 | |
| **Description** | |
| Macro used for MOSFET Open Circuit state machine | |

### PAL\_KU8\_OCAT\_EXECUTE\_STEP\_1\_PREPARE\_STEP\_2

|  |  |
| --- | --- |
| Name | Value |
| Used for MOSFET Open Circuit state machine | 2 |
| **Definition** | |
| #define PAL\_KU8\_OCAT\_EXECUTE\_STEP\_1\_PREPARE\_STEP\_2 | |
| **Description** | |
| Macro used for MOSFET Open Circuit state machine | |

### PAL\_KU8\_OCAT\_EXECUTE\_STEP\_2\_PREPARE\_STEP\_3

|  |  |
| --- | --- |
| Name | Value |
| Used for MOSFET Open Circuit state machine | 3 |
| **Definition** | |
| #define PAL\_KU8\_OCAT\_EXECUTE\_STEP\_2\_PREPARE\_STEP\_3 | |
| **Description** | |
| Macro used for MOSFET Open Circuit state machine | |

### PAL\_KU8\_OCAT\_EXECUTE\_STEP\_3\_PREPARE\_STEP\_4

|  |  |
| --- | --- |
| Name | Value |
| Used for MOSFET Open Circuit state machine | 4 |
| **Definition** | |
| #define PAL\_KU8\_OCAT\_EXECUTE\_STEP\_3\_PREPARE\_STEP\_4 | |
| **Description** | |
| Macro used for MOSFET Open Circuit state machine | |

### PAL\_KU8\_OCAT\_EXECUTE\_STEP\_4

|  |  |
| --- | --- |
| Name | Value |
| Used for MOSFET Open Circuit state machine | 5 |
| **Definition** | |
| #define PAL\_KU8\_OCAT\_EXECUTE\_STEP\_4 | |
| **Description** | |
| Macro used for MOSFET Open Circuit state machine | |

### PAL\_KU8\_EIGHTY

|  |  |
| --- | --- |
| Name | Value |
| Motor disengagement auto-test parameter | 80 |
| **Definition** | |
| #define PAL\_KU8\_EIGHTY | |
| **Description** | |
| Macro used for Motor disengagement auto-test | |

### PAL\_KU8\_DIVIDE\_BY\_64\_RIGHT\_SHIFT

|  |  |
| --- | --- |
| Name | Value |
| Motor disengagement auto-test parameter | 6 |
| **Definition** | |
| #define PAL\_KU8\_DIVIDE\_BY\_64\_RIGHT\_SHIFT | |
| **Description** | |
| Macro used for Motor disengagement auto-test | |

### PAL\_KU8\_CONVERT\_IN\_PERCENT

|  |  |
| --- | --- |
| Name | Value |
| Motor disengagement auto-test parameter | 100 |
| **Definition** | |
| #define PAL\_KU8\_DIVIDE\_BY\_64\_RIGHT\_SHIFT | |
| **Description** | |
| Macro used for Motor disengagement auto-test | |

### PAL\_KU32\_CONVERT\_MS\_TO\_100\_MS

|  |  |
| --- | --- |
| Name | Value |
| 100ms periodicity | 100 |
| **Definition** | |
| #define PAL\_KU32\_CONVERT\_MS\_TO\_100\_MS | |
| **Description** | |
| Macro used for Motor disengagement auto-test | |

### PAL\_KU32\_SIXTEEN

|  |  |
| --- | --- |
| Name | Value |
| Motor Short Circuit autotest parameter | 16 |
| **Definition** | |
| #define PAL\_KU32\_SIXTEEN | |
| **Description** | |
| Macro used for Motor Short Circuit autotest | |

### PAL\_KU8\_MOTOR\_SHORT\_CIRCUIT\_RESITANCE\_MIN\_IDX

|  |  |
| --- | --- |
| Name | Value |
| Select the minimum resistance from the related NVM parameters | 0 |
| **Definition** | |
| #define PAL\_KU8\_MOTOR\_SHORT\_CIRCUIT\_RESITANCE\_MIN\_IDX | |
| **Description** | |
| Macro used to select the minimum resistance from the related NVM parameters | |

### PAL\_KU8\_CONVERT\_DUTY\_CYCLE\_TO\_PWM\_ORDER\_FACTOR

|  |  |
| --- | --- |
| Name | Value |
| used to convert a duty cycle into PWM order | 64 |
| **Definition** | |
| #define PAL\_KU8\_CONVERT\_DUTY\_CYCLE\_TO\_PWM\_ORDER\_FACTOR | |
| **Description** | |
| Macro used to convert a duty cycle into PWM order | |

### PAL\_KU8\_HIGH\_SIDE\_SW\_REG\_INIT

|  |  |
| --- | --- |
| Name | Value |
| Used for High Side Switch state machine | 0 |
| **Definition** | |
| #define PAL\_KU8\_HIGH\_SIDE\_SW\_REG\_INIT | |
| **Description** | |
| Macro Used for High Side Switch state machine | |

### PAL\_KU8\_HIGH\_SIDE\_SW\_REG\_ACTIVATION\_STEP\_1

|  |  |
| --- | --- |
| Name | Value |
| Used for High Side Switch state machine | 1 |
| **Definition** | |
| #define PAL\_KU8\_HIGH\_SIDE\_SW\_REG\_ACTIVATION\_STEP\_1 | |
| **Description** | |
| Macro Used for High Side Switch state machine | |

### PAL\_KU8\_HIGH\_SIDE\_SW\_REG\_ALGO\_STEP\_2

|  |  |
| --- | --- |
| Name | Value |
| Used for High Side Switch state machine | 2 |
| **Definition** | |
| #define PAL\_KU8\_HIGH\_SIDE\_SW\_REG\_ALGO\_STEP\_2 | |
| **Description** | |
| Macro Used for High Side Switch state machine | |

### PAL\_KU8\_HIGH\_SIDE\_SW\_REG\_DEAD\_TIME

|  |  |
| --- | --- |
| Name | Value |
| Used for High Side Switch state machine | 3 |
| **Definition** | |
| #define PAL\_KU8\_HIGH\_SIDE\_SW\_REG\_DEAD\_TIME | |
| **Description** | |
| Macro Used for High Side Switch state machine | |

### PAL\_KU8\_HIGH\_SIDE\_SW\_REG\_FREE\_WHEELING\_PRESCALER

|  |  |
| --- | --- |
| Name | Value |
| High Side Switch Free wheeling phase parameter | 20 |
| **Definition** | |
| #define PAL\_KU8\_HIGH\_SIDE\_SW\_REG\_FREE\_WHEELING\_PRESCALER | |
| **Description** | |
| Macro Used for High Side Switch Free wheeling | |

# EEPROM

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

Refer to this document for more details.

# Configuration

No special configuration for PAL software component.

# Compilation Options

No special configuration for PAL software component.