Document Title	Glossary
Document Category	Technical report (TR)
Document UID	55
Document Status	published
Part of AUTOSAR Standard	Foundation
Part of Standard Release	R26-11

Document Change History

Date	Release	Changed by	Description
2026-11-26	R26-11	AUTOSAR Release Management	• Replaced initialism LT by DLT (Log and Trace Protocol) • Added acronyms/initialisms: CM (Communication Module), DCA (Data Class Adapter), DMA (Direct Memory Access), DP (Data Point), HPC (High performance computer or high performance computing machine), VDP (Vehicle Data Protocol) • Added terms: Direct Memory Access, PNC Request Bit Vector, ShortName-path • Clarifications of further terms and acronyms
2025-11-27	R25-11	AUTOSAR Release Management	• Refactored abbreviation table • Changed acronyms: ASIL (Automotive Safety Integrity Level), ASAM (Association for Standardization of Automation and Measuring Systems), COM (Communication), DET (Default Error Tracer) • Added acronyms/initialisms: COMM (COMMunication Manager), HMI (Human Machine Interface) • Added definitions: Minimum Send Interval, Middleware, Software Cluster (CLassic Platform) • Removed acronym: TTCAN (Time Triggered CAN)

△

2024-11-27	R24-11	AUTOSAR Release Management	• Added terms for Trusted Platform, Adaptive Platform Machine Configuration, Inter-Integrated Circuit, Automotive API (Gateway) • Changed definition for Adaptive Application • Extended abbreviation list
2023-11-23	R23-11	AUTOSAR Release Management	• Improved definition of Processed Manifest • Extended abbreviation list
2022-11-24	R22-11	AUTOSAR Release Management	• Added terms for Can XL, Integrity Check Value, Firewall • Improved definitions of Foundation, Software Cluster (Adaptive Platform), Gateway, Use Case
2021-11-25	R21-11	AUTOSAR Release Management	• Added terms for Intermediate PNC coordinator, PN shutdown message, Top-level PNC coordinator, PNC leaf node • Added terms for Logging, Tracing, Profiling • Improved definition of System term
2020-11-30	R20-11	AUTOSAR Release Management	Added new terms: • E2E Protection Alive Counter • E2E Protection Sequence Counter • Vehicle State Manager • Health Indicator • System Health Monitor • Wake-up and sleep on dataline • Foundation • Intrusion Detection System • Onboard Security Event

▽

△

2019-11-28	R19-11	AUTOSAR Release Management	• Removed FlexRay specific terms • Added new terms: – Secure channel – Abstract Platform – Raw Data Stream – Signal Service Translation • Changed Document Status from Final to published
2019-03-29	1.5.1	AUTOSAR Release Management	• Editorial changes
2018-10-31	1.5.0	AUTOSAR Release Management	• Extended abbreviations • Added terms: – AUTOSAR Run-Time Interface – Bus Mirroring – Cluster – Executable Entity Cluster – Execution Order Constraint – Execution Time – LIN Bus Idle – Log and Trace – Logical Execution Time – Mappable Element – Security Event – Synchronization Points – Timed Communication • Changed OSEK references • Incorporated concepts as draft: – AUTOSAR Run-Time Interface – MCAL Multicore Distribution – Transport Layer Security

▽

△

2018-03-29	1.4.0	AUTOSAR Release Management	• Added terms: – Access Control Policy – Access Control Decision – MetaDataItem – Policy Decision Point (PDP) – Policy Enforcement Point (PEP) – Identity and Access Management (IAM) • Removed terms: – FlexRay Global Time – Meta-Model – MetaDataLength – Model – Multiple Configuration Sets Shipping – Template – Variation Definition Time • Changed terms: – AUTOSAR Definition – AUTOSAR Meta-model – AUTOSAR Model – AUTOSAR Service – AUTOSAR XML description – Link Time Configuration – Manifest – PDU Meta-Data
2017-12-08	1.3.0	AUTOSAR Release Management	• No content changes
2017-10-27	1.2.0	AUTOSAR Release Management	• No content changes

▽

2017-03-31	1.1.0	AUTOSAR Release Management	Added terms: • Adaptability • Adaptive Application • Adaptive Platform Foundation • Adaptive Platform Services • ASIL Decomposition • Audit • AUTOSAR Adaptive Platform • AUTOSAR Runtime for Adaptive Applications • Cascaded Switch • Cascading Failure • Classic Platform • Common Cause Failure • Dependent Failures • Diagnostic Coverage • Diversity • Ethernet Switch Port Groups • Executable • External Port • Failure Mode • Fault Reaction Time • Fault Tolerant Time Interval • Freedom from Interference • Functional Cluster • Functional Safety Concept • Functional Safety Requirement • Host ECU • Host Port • Hypervisor • Independence • Independent Failures • Internal Port • Link State Accumulation

			• Machine • Manifest • Master Switch • Microcontroller • Performance • Plausibility • Predictabiliy • Proven In Use Argument • Recovery • Safe State • Safety Case • Safety Goal • Safety Measure • Safety Mechanism • Service Discovery • Service Instance • Service Interface • Service Oriented Communication • Service Proxy • Service Skeleton • Slave Switch • Software package • Software Unit • Systematic Fault • Uplink Port • Virtualization Removed terms: • Accreditation Body • Accreditation • Attestation • Conformance Declaration • Conformance Test Agency (CTA) • First party

△

			• Implementation Conformance Statement • Interrupt Logic • Partial Model • Surveillance • Third party Changed terms: • Automotive Safety Integrity Levels • Availability • Acceptance Test Suite • Electronic Control Unit • Error • Fail-safe • Fail-silent • Failure Rate • Failure • Fault Tolerance • Fault • FlexRay Bus • FlexRay Cycle • FlexRay L-PDU-Identifier • FlexRay L-SDU-Identifier • FlexRay Matrix • FlexRay Slot Multiplexing • Graceful Degradation • Fail-degraded • Implementation Conformance Class 1 (ICC1) • Implementation Conformance Class 2 (ICC2) • Implementation Conformance Class 3 (ICC3) • Link Time Configuration • Partitioning • Protocol Control Information

▽

△

			△ • Protocol Data Unit • Post-build Time Configuration • Pre-Compile Time Configuration • Probability of Failure • Redundancy • Risk • Safety • Service Data Unit
2016-11-30	1.0.0	AUTOSAR Release Management	• Migration of document to standard • Following terms added: – AUTOSAR Blueprint – Bypassing – Hook – OS Event – Post-build Hooking – Pre-build Hooking – Rapid Prototyping (RP) – Rapid Prototyping Memory Interface – Rapid Prototyping Tool – Reentrancy – Standardized AUTOSAR Blueprint – Standardized Blueprint – Following terms changed: – Asset – Asynchronous Function – AUTOSAR Application Interface – Availability – ECU Abstraction Layer – Feature – Function – Microcontroller Abstraction Layer (MCAL)

▽

△

2015-07-31	4.2.2	AUTOSAR Release Management	• Following terms changed: – ECU Abstraction Layer – Standardized AUTOSAR Interface – Hook – OS Event – Post-build Hooking – Pre-build Hooking • Following terms removed: – Software Module
2014-10-31	4.2.1	AUTOSAR Release Management	• Following terms changed: – Data Variant Coding – OS-Application – Post-build time configuration – Standardized AUTOSAR Interface
2014-03-31	4.1.3	AUTOSAR Release Management	• Extended Abbreviations • Following terms changed: – Software Component (SW-C)
2013-03-15	4.1.1	AUTOSAR Administration	• Extended Abbreviations • Following terms added: – Application Interface – Asynchronous Functions – AUTOSAR Application Interface – Dynamic PDU – Life Cycle – MetaDataLength – PDU Meta-Data – Pretended Networking – Synchronous Functions
2011-12-22	4.0.3	AUTOSAR Administration	• Extended Abbreviations • Following terms added: – Callback – Callout – ECU

▽

△

2009-12-18	4.0.1	AUTOSAR Administration	• Following terms added: – AUTOSAR Partial Model – Bus Wake-Up – Empty Function
2009-02-02	3.1.4	AUTOSAR Administration	• Following terms added: – Automotive Safety Integrity Levels (ASIL) – Bit Position – Category 1 Interrupt – Category 2 Interrupt – Code Generator – Coordinate – E2E Profile – Error Detection Rate – Failure Rate – ICC1 (Implementation Conformance Class 1) – ICC2 (Implementation Conformance Class 2) – ICC3 (Implementation Conformance Class 3) – Interrupt Frames – Interrupt Handler – Interrupt Logic – Meta-Model – Mode – Model – Network Interface (NWI) – NM Coordination Cluster – NM Coordinator – Rate Conversion – Residual Error Rate – SAE J1939 – Safety Protocol ▽

▽

			– Software Component Interface (SW-CI) – Synchronize – Variability – Variant – Variation Binding – Variation Binding Time – Variation Definition Time – Variation Point – Formal adaptations • Legal disclaimer revised
2008-08-13	3.1.1	AUTOSAR Administration	• Legal disclaimer revised
2007-12-21	3.0.1	AUTOSAR Administration	• Following terms added: – Debugging – Implementation Conformance Statement – Document meta information extended – Small layout adaptations made
2007-01-24	2.1.15	AUTOSAR Administration	• Following terms added: – FlexRay – Vendor ID – Callback – Interrupt frames – Interrupt vector table – Accreditation – Accreditation Body – Conformance Test Agency – Assessment – Surveillance – Attestation – (Conformance) Declaration – First party and – Third party

△

			△ – Safety – ECU Configuration – ECU Configuration Description • Legal disclaimer revised
2006-05-16	2.0	AUTOSAR Administration	• removed and added some terms • rework of several descriptions • formal changes
2006-05-16	1.0	AUTOSAR Administration	• Initial release

Disclaimer

This work (specification and/or software implementation) and the material contained in it, as released by AUTOSAR, is for the purpose of information only. AUTOSAR and the companies that have contributed to it shall not be liable for any use of the work.

The material contained in this work is protected by copyright and other types of intellectual property rights. The commercial exploitation of the material contained in this work requires a license to such intellectual property rights.

This work may be utilized or reproduced without any modification, in any form or by any means, for informational purposes only. For any other purpose, no part of the work may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The work has been developed for automotive applications only. It has neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Table of Contents

1 Introduction 27

2 Acronyms, Abbreviations (and Initialisms) 28
2.1 Explanation . . . 28

3 How to read this document 35
3.1 <Term> Template . . . 35

4 Definitions 36
4.1 Abstract Platform . . . 36
4.2 Acceptance Test Suite . . . 36
4.3 Access Control Decision . . . 36
4.4 Access Control Policy . . . 37
4.5 Adaptability . . . 37
4.6 Adaptive Application . . . 37
4.7 Adaptive Platform Foundation . . . 38
4.8 Adaptive Platform Services . . . 38
4.9 Adaptive Platform Machine Configuration . . . 38
4.10 Application Programming Interface . . . 38
4.11 Application Software Component . . . 39
4.12 Architecture . . . 39
4.13 Artifact . . . 39
4.14 ASIL Decomposition . . . 40
4.15 Asserted Property . . . 40
4.16 Assessment . . . 40
4.17 Asset . . . 40
4.18 Asynchronous Communication . . . 41
4.19 Asynchronous Function . . . 41
4.20 Atomic Software Component . . . 41
4.21 Audit . . . 42
4.22 Authenticity . . . 42
4.23 Automotive API . . . 42
4.24 Automotive API Gateway . . . 42
4.25 Automotive Safety Integrity Levels . . . 43
4.26 AUTOSAR Adaptive Platform . . . 43
4.27 AUTOSAR Application Interface . . . 43
4.28 AUTOSAR Authoring Tool . . . 43
4.29 AUTOSAR Blueprint . . . 44
4.30 AUTOSAR Converter Tool . . . 44
4.31 AUTOSAR Definition . . . 44
4.32 AUTOSAR Interface . . . 44
4.33 AUTOSAR Meta-model . . . 45

4.34 AUTOSAR Model . . . 45
4.35 AUTOSAR Partial Model . . . 45
4.36 AUTOSAR Processor Tool . . . 46
4.37 AUTOSAR Runtime for Adaptive Applications . . . 46
4.38 AUTOSAR Run-Time Interface (ARTI) . . . 46
4.39 AUTOSAR Service . . . 46
4.40 AUTOSAR Software . . . 47
4.41 AUTOSAR Tool . . . 47
4.42 AUTOSAR XML description . . . 47
4.43 AUTOSAR XML Schema . . . 48
4.44 Availability . . . 48
4.45 Basic Software . . . 49
4.46 Basic Software Module . . . 50
4.47 Bit Position . . . 50
4.48 Blueprint . . . 50
4.49 Bulk Data . . . 50
4.50 Bus Mirroring . . . 51
4.51 Bus Wake-Up . . . 51
4.52 Bypassing . . . 51
4.53 Calibration . . . 52
4.54 Call Point . . . 52
4.55 Callback . . . 52
4.56 Callout . . . 53
4.57 CAN XL . . . 53
4.58 Cascaded Switch . . . 54
4.59 Cascading Failure . . . 54
4.60 Category 1 Interrupt . . . 54
4.61 Category 2 Interrupt . . . 55
4.62 Causality of Transmission . . . 55
4.63 Classic Platform . . . 55
4.64 Client . . . 56
4.65 Client-Server Communication . . . 56
4.66 Client-Server Interface . . . 56
4.67 Cluster Signal . . . 57
4.68 Code Generator . . . 57
4.69 Code Variant Coding . . . 57
4.70 Common Cause Failure . . . 57
4.71 Communication Attribute . . . 58
4.72 Complex Driver . . . 58
4.73 Composition . . . 58
4.74 Compositionality . . . 59
4.75 Conditioned Signal . . . 59
4.76 Confidentiality . . . 59

4.77 Configuration . . . 60
4.78 Confirmation . . . 60
4.79 Connector . . . 60
4.80 Control Flow . . . 60
4.81 Coordinate . . . 61
4.82 Data . . . 61
4.83 Data Element . . . 61
4.84 Data Flow . . . 61
4.85 Data Variant Coding . . . 62
4.86 Deadline . . . 62
4.87 Debugging . . . 62
4.88 Dependability . . . 63
4.89 Dependent Failure . . . 63
4.90 Diagnostic Coverage . . . 63
4.91 Diagnostic Event . . . 63
4.92 Direct Memory Access . . . 64
4.93 Diversity . . . 64
4.94 Dynamic PDU . . . 64
4.95 Dynamic Routing . . . 65
4.96 E2E Profile . . . 65
4.97 E2E Protection Alive Counter . . . 65
4.98 E2E Protection Sequence Counter . . . 66
4.99 ECU Abstraction Layer . . . 66
4.100 ECU Configuration . . . 66
4.101 ECU Configuration Description . . . 67
4.102 ECU HW . . . 67
4.103 ECU Instance . . . 67
4.104 Electrical Signal . . . 68
4.105 Electronic Control Unit (ECU) . . . 68
4.106 Empty Function . . . 68
4.107 Entry Point . . . 68
4.108 Error . . . 69
4.109 Error Detection Rate . . . 69
4.110 Ethernet Switch Port Groups . . . 69
4.111 Event . . . 70
4.112 Event Message (SOME/IP) . . . 70
4.113 Executable . . . 70
4.114 Executable Entity Cluster . . . 70
4.115 Execution Order Constraint . . . 71
4.116 Execution Time . . . 72
4.117 Exit Point . . . 72
4.118 External Port . . . 72
4.119 Fail-operational . . . 72

4.120 Fail-safe . . . 73
4.121 Fail-silent . . . 73
4.122 Failure Mode . . . 73
4.123 Failure . . . 74
4.124 Failure Rate . . . 74
4.125 Fault . . . 74
4.126 Fault Detection . . . 75
4.127 Fault Reaction . . . 75
4.128 Fault Reaction Time . . . 75
4.129 Fault Tolerance . . . 76
4.130 Fault Tolerant Time Interval . . . 76
4.131 Feature . . . 76
4.132 Firewall . . . 76
4.133 Flag . . . 77
4.134 FlexRay . . . 77
4.135 Foundation . . . 77
4.136 Frame . . . 78
4.137 Frame PDU . . . 78
4.138 Freedom from Interference . . . 78
4.139 Freshness . . . 79
4.140 Function . . . 79
4.141 Functional Cluster . . . 79
4.142 Functional Network . . . 80
4.143 Functional Safety Concept . . . 80
4.144 Functional Safety Requirement . . . 80
4.145 Functional Unit . . . 80
4.146 Functionality . . . 81
4.147 Gateway . . . 81
4.148 Gateway ECU . . . 81
4.149 Graceful Degradation . . . 81
4.150 Hardware Abstraction Layer . . . 82
4.151 Hardware Connection . . . 82
4.152 Hardware Element . . . 82
4.153 Hardware Interrupt . . . 83
4.154 Hardware Port . . . 83
4.155 Health Indicator . . . 83
4.156 Hook . . . 84
4.157 Host ECU . . . 84
4.158 Host Port . . . 85
4.159 Hypervisor . . . 85
4.160 Identity and Access Management (IAM) . . . 85
4.161 Identity Information . . . 85
4.162 Implementation Conformance Class 1 (ICC1) . . . 86

4.163 Implementation Conformance Class 2 (ICC2) . . . 86
4.164 Implementation Conformance Class 3 (ICC3) . . . 88
4.165 Independence . . . 88
4.166 Independent Failures . . . 88
4.167 Indication . . . 88
4.168 Integration . . . 89
4.169 Integration Code . . . 89
4.170 Integrity . . . 89
4.171 Integrity Check Value . . . 90
4.172 Inter-Integrated Circuit I2C . . . 90
4.173 Intermediate PNC Coordinator . . . 90
4.174 Internal Port . . . 91
4.175 Interrupt Frame . . . 91
4.176 Interrupt Handler . . . 91
4.177 Interrupt Service Routine . . . 91
4.178 Interrupt Vector Table . . . 92
4.179 Interrupt . . . 92
4.180 Intrusion Detection System . . . 92
4.181 Invalid Flag . . . 93
4.182 Invalid Value of Signal . . . 93
4.183 I-PDU . . . 93
4.184 Life Cycle . . . 94
4.185 LIN Bus Idle . . . 94
4.186 Link State Accumulation . . . 94
4.187 Link Time Configuration . . . 94
4.188 Logical Execution Time (LET) . . . 95
4.189 Log and Trace . . . 96
4.190 Logging . . . 96
4.191 Machine . . . 96
4.192 Manifest . . . 97
4.193 Mappable Element . . . 97
4.194 Mapping . . . 97
4.195 Master Switch . . . 98
4.196 MCAL Signal . . . 98
4.197 Meta-data . . . 98
4.198 MetaDataItem . . . 98
4.199 Microcontroller . . . 99
4.200 Microcontroller Abstraction Layer . . . 99
4.201 Middleware . . . 100
4.202 Minimum Send Interval . . . 100
4.203 Mistake . . . 101
4.204 Mode . . . 101
4.205 Multimedia Stream . . . 101

4.206 Multiplexed PDU . . . 102
4.207 Network . . . 102
4.208 Network Interface . . . 102
4.209 NM Coordination Cluster . . . 103
4.210 NM Coordinator . . . 103
4.211 Non-repudiation . . . 103
4.212 Notification . . . 104
4.213 Onboard Security Event . . . 105
4.214 OS Application . . . 105
4.215 OS Event . . . 105
4.216 Partitioning . . . 106
4.217 Protocol Control Information . . . 106
4.218 Protocol Data Unit (PDU) . . . 106
4.219 PDU Meta-Data . . . 107
4.220 PDU Timeout . . . 107
4.221 Performance . . . 108
4.222 Peripheral Hardware . . . 108
4.223 Personalization . . . 108
4.224 Plausibility . . . 109
4.225 PNC Leaf Node . . . 109
4.226 PNC Request Bit Vector . . . 109
4.227 PN shutdown message . . . 110
4.228 Policy Decision Point (PDP) . . . 110
4.229 Policy Enforcement Point (PEP) . . . 110
4.230 Port . . . 110
4.231 Port Interface . . . 111
4.232 Post-build Time Configuration . . . 111
4.233 Post-build Hooking . . . 111
4.234 Pre-build Hooking . . . 112
4.235 Pre-Compile Time Configuration . . . 112
4.236 Predictabiliy . . . 112
4.237 Pretended Networking . . . 112
4.238 Private Interface . . . 113
4.239 Probability of Failure . . . 113
4.240 Procedure Call . . . 113
4.241 Process . . . 114
4.242 Processed Manifest . . . 115
4.243 Profiling . . . 115
4.244 Proven In Use Argument . . . 115
4.245 Provide Port . . . 116
4.246 Rapid Prototyping . . . 116
4.247 Rapid Prototyping Memory Interface . . . 116
4.248 Rapid Prototyping Tool . . . 116

4.249 Rate Conversion . . . 117
4.250 Raw Data Stream . . . 117
4.251 Recovery . . . 117
4.252 Redundancy . . . 118
4.253 Reentrancy . . . 118
4.254 Reliability . . . 119
4.255 Relocatability . . . 119
4.256 Require Port . . . 119
4.257 Required Property . . . 120
4.258 Residual Error Rate . . . 120
4.259 Resource . . . 120
4.260 Resource-Management . . . 121
4.261 Response Time . . . 121
4.262 Risk . . . 121
4.263 Robustness . . . 121
4.264 RTE Event . . . 122
4.265 Runnable Entity . . . 122
4.266 SAE J1939 . . . 122
4.267 Safe State . . . 123
4.268 Safety . . . 123
4.269 Safety Analysis . . . 123
4.270 Safety Case . . . 124
4.271 Safety Goal . . . 124
4.272 Safety Measure . . . 124
4.273 Safety Mechanism . . . 125
4.274 Safety Protocol . . . 125
4.275 Sample Application . . . 125
4.276 Scalability . . . 125
4.277 Scheduler . . . 126
4.278 Service Data Unit . . . 126
4.279 Security . . . 126
4.280 Secure Channel . . . 126
4.281 Security Event . . . 127
4.282 Sender-Receiver Communication . . . 127
4.283 Sender-Receiver Interface . . . 127
4.284 Sensor-Actuator SW-Component . . . 128
4.285 Server . . . 128
4.286 Service . . . 128
4.287 Service Discovery . . . 129
4.288 Service Instance . . . 129
4.289 Service Interface . . . 129
4.290 Service-Oriented Communication . . . 129
4.291 Service Port . . . 130

4.292 Service Proxy . . . 130
4.293 Service Skeleton . . . 130
4.294 Services Layer . . . 131
4.295 ShortName-path . . . 131
4.296 ShortName-path (relative) . . . 132
4.297 ShortName-path (absolute) . . . 132
4.298 Signal Service Translation . . . 132
4.299 Slave Switch . . . 132
4.300 Software Cluster (Adaptive Platform) . . . 133
4.301 Software Cluster (Classic Platform) . . . 133
4.302 Software Component . . . 133
4.303 Software Component Interface . . . 134
4.304 Software Configuration . . . 134
4.305 Software Interrupt . . . 135
4.306 Software Package . . . 135
4.307 Software Platform . . . 135
4.308 Software Signal . . . 135
4.309 Software Unit . . . 136
4.310 Special Periphery Access . . . 136
4.311 Standard Periphery Access . . . 136
4.312 Standard Software . . . 136
4.313 Standardized AUTOSAR Blueprint . . . 137
4.314 Standardized AUTOSAR Interface . . . 137
4.315 Standardized Blueprint . . . 137
4.316 Standardized Interface . . . 138
4.317 Static Configuration . . . 138
4.318 Synchronization Points . . . 138
4.319 Synchronization . . . 139
4.320 Synchronous Communication . . . 139
4.321 Synchronous Function . . . 139
4.322 System . . . 140
4.323 System Constraint . . . 140
4.324 System Health Monitor . . . 140
4.325 System Signal . . . 141
4.326 Systematic Fault . . . 141
4.327 Task . . . 141
4.328 Technical Signal . . . 141
4.329 Timed Communication . . . 142
4.330 Timeout . . . 142
4.331 Top-level PNC Coordinator . . . 143
4.332 Tracing . . . 143
4.333 Trusted Platform . . . 144
4.334 Uplink Port . . . 144

4.335 Use Case . . . 144
4.336 Validation . . . 144
4.337 Variability . . . 145
4.338 Variant . . . 145
4.339 Variant Coding . . . 145
4.340 Variation Binding . . . 146
4.341 Variation Binding Time . . . 146
4.342 Variation Point . . . 146
4.343 Vehicle State Manager . . . 147
4.344 Vehicle Variant . . . 147
4.345 Vendor ID . . . 147
4.346 Verification . . . 148
4.347 VFB View . . . 148
4.348 Virtual Functional Bus . . . 148
4.349 Virtual Integration . . . 149
4.350 Virtualization . . . 149
4.351 Wake-up and Sleep on Dataline . . . 149
4.352 Worst Case Execution Time . . . 150
4.353 Worst Case Response Time . . . 150

A Index of AUTOSAR entities 151
A.1 AUTOSAR Document categories . . . 151
A.2 AUTOSAR Documents . . . 151
A.3 BSW modules . . . 161
A.4 Functional Clusters . . . 164
A.5 M2 Constraint Imposition Times . . . 165
A.6 Modeling keywords . . . 167

References

[1] Specification of Default Error Tracer
AUTOSAR_CP_SWS_DefaultErrorTracer

[2] Specification of RTE Software
AUTOSAR_CP_SWS_RTE

[3] ISO/IEC 9646-1:1994 - Information technology - Open Systems Interconnection - Conformance testing methodology and framework
https://www.iso.org

[4] ISO 17356-3: Road vehicles – Open interface for embedded automotive applications – Part 3: OSEK/VDX Operating System (OS)

[5] ITEA Project 00009 EAST-EEA Embedded Electronic Architecture - Glossary Version 6.1

[6] IEEE 1471-2000: IEEE Recommended Practice for Architectural Description for Software- Intensive Systems

[7] ISO 26262-1:2018 Part 1: Vocabulary
https://www.iso.org

[8] IEEE 1517-1999: IEEE Standard for Information Technology – Software Life Cycle Processes – Reuse Processes

[9] Explanation of Automotive API
AUTOSAR_AP_EXP_AutomotiveAPI

[10] Standardization Template
AUTOSAR_FO_TPS_StandardizationTemplate

[11] Virtual Functional Bus
AUTOSAR_CP_TR_VFB

[12] Unified Modeling Language: Superstructure, Version 2.0, OMG Available Specification, ptc/05-07-04
http://www.omg.org/cgi-bin/apps/doc?formal/05-07-04

[13] Specification of AUTOSAR Run-Time Interface
AUTOSAR_CP_SWS_ARTI

[14] CiA 610-1 version 1.0.0 (DSP) - CAN XL specifications and test plans - Part 1: Data link layer and physical coding sub-layer requirements
http://www.can-cia.org

[15] CiA 611-1 version 1.0.0 (DSP) - CAN XL higher layer functions - Part 1: Definition of service data unit types
http://www.can-cia.org

[16] Lehrbuch Grundlagen der Informatik

[17] ISO/IEC 61511 Part 1 Information technology – Software life cycle process, First Edition

[18] ISO 7498 – Information processing systems – Open Systems Interconnection – Basic Reference Model
https://www.iso.org
ISO/IEC 7498-1:1994

[19] Binding Specification Version 1.4.1

[20] ISO/IEC 2382 Part 1 Information technology – Vocabulary – Fundamental Terms, Third Edition

[21] Software Component Template
AUTOSAR_CP_TPS_SoftwareComponentTemplate

[22] Specification of Operating System
AUTOSAR_CP_SWS_OS

[23] Specification of Diagnostic Event Manager
AUTOSAR_CP_SWS_DiagnosticEventManager

[24] Specification of DMA Driver
AUTOSAR_CP_SWS_DMADriver

[25] ISO 26262-6:2018 Part 6: Product development at the software level
https://www.iso.org

[26] Layered Software Architecture
AUTOSAR_CP_EXP_LayeredSoftwareArchitecture

[27] Specification of ECU Configuration
AUTOSAR_CP_TPS_ECUConfiguration

[28] Specification of Timing Extensions for Classic Platform
AUTOSAR_CP_TPS_TimingExtensions

[29] ISO 15765-2 – Road vehicles – Diagnostics on Controller Area Networks (CAN) – Part2: Network layer services

[30] ISO 14229-1 – Unified diagnostic services (UDS) – Part 1: Specification and requirements (Release 2006-12)
https://www.iso.org

[31] AUTOSAR Feature Model Exchange Format
AUTOSAR_FO_TPS_FeatureModelExchangeFormat

[32] ISO 17458-1:2013, Road vehicles - FlexRay communication system
https://www.iso.org

[33] Specification of FlexRay Driver
AUTOSAR_CP_SWS_FlexRayDriver

[34] Specification of FlexRay Interface

AUTOSAR_CP_SWS_FlexRayInterface

[35] System Template
AUTOSAR_CP_TPS_SystemTemplate

[36] ISO 17356-1: Road vehicles – Open interface for embedded automotive applications – Part 1: General structure and terms, definitions and abbreviated terms

[37] Specification of ECU Resource Template
AUTOSAR_CP_TPS_ECUResourceTemplate

[38] Translation/Adaptation from VDI Lexikon Informatik und Kommunikationstechnik

[39] Specification of Health Monitoring
AUTOSAR_FO_ASWS_HealthMonitoring

[40] ISO/IEC 2382 Part 20 Information technology – Vocabulary – System Development, First Edition

[41] NIST: Secure Hash Standard (SHS)
http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf

[42] IEEE 1588-2019: IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems

[43] UM10204 - I2C-bus specification and user manual
https://www.nxp.com/docs/en/user-guide/UM10204.pdf

[44] ISO 17356-4: Road vehicles – Open interface for embedded automotive applications – Part 4: OSEK/VDX Communication (COM)

[45] ISO 17987:2016 (all parts), Road vehicles – Local Interconnect Network (LIN)
https://www.iso.org

[46] Specification of Communication Management
AUTOSAR_AP_SWS_CommunicationManagement

[47] Specification of Manifest
AUTOSAR_AP_TPS_ManifestSpecification

[48] DIN 40041 Ausgabe:1990-12 Zuverlässigkeit; Begriffe

[49] IEEE Standard for a High-Performance Serial Bus
http://www.1394ta.org/Technology/Specifications/specifications.htm

[50] Specification of Network Management Interface
AUTOSAR_CP_SWS_NetworkManagementInterface

[51] ISO/IEC 27000:2018 Information technology - Security techniques - Information security management systems - Overview and vocabulary
https://www.iso.org

[52] ISO/IEC 2382 Part 15 Information technology – Vocabulary – Programming Languages, First Edition

[53] ISO/IEC 2382 Part 1 Information technology – Vocabulary – Security, Second Edition

[54] Generic Structure Template
AUTOSAR_FO_TPS_GenericStructureTemplate

[55] Specification of Update and Configuration Management
AUTOSAR_AP_SWS_UpdateAndConfigurationManagement

[56] Specification of Software Cluster Connection module
AUTOSAR_CP_SWS_SoftwareClusterConnection

[57] XML Specification of Application Interfaces
AUTOSAR_CP_MOD_AISpecification

[58] IEEE Standard for System, Software, and Hardware Verification and Validation

[59] General Specification of Basic Software Modules
AUTOSAR_CP_SWS_BSWGeneral

[60] Specification of Adaptive Platform Core
AUTOSAR_AP_SWS_Core

[61] SW-C and System Modeling Guide
AUTOSAR_CP_TR_SWCModelingGuide

1 Introduction

This document constitutes the comprehensive glossary of AUTOSAR. It contains definitions of all major abbreviations, acronyms and terms used within AUTOSAR. Please note that the document is not intended to be exhaustive. It is important to consult the glossaries of the various working group documents, as these contain additional specific terms within their respective domains.

2 Acronyms, Abbreviations (and Initialisms)

2.1 Explanation

An *Abbreviation* is a shortened form of a word or phrase, like "Temp." for "Temperature", whereas an *Acronym* is a specific type of abbreviation formed from the initial letters of a phrase and pronounced as a single word like "AUTOSAR", which stands for "AUTomotive Open System Architecture". An *Initialism* is similar to an Acronym but is pronounced letter by letter, like "ECU" for "Electronic Control Unit", and there are hybrid forms of an Initialism and an Acronym, like "VLAN" for "Virtual LAN. However, in both colloquial and technical contexts, the terms Abbreviation, Acronym and Initialism are used interchangeably, and that is also the case in this document.

Acronym/ Abbreviation	Description
A	
AA	Adaptive Application
ABC	Abstract Base Class
ABI	Application Binary Interface
ADC	Analog Digital Converter
AES	Advanced Encryption Standard
AMM	Application Mode Management
AP	AUTOSAR Adaptive Platform
API	Application Programming Interface
ARA	AUTOSAR Runtime for Adaptive Applications
ARP	Address Resolution Protocol
ARTI	AUTOSAR Run-Time Interface
ARXML	AUTOSAR XML
ASAM	Association for Standardization of Automation and Measuring Systems
ASD	Abstract System Description
ASIL	Automotive Safety Integrity Levels
ASW	Application SoftWare
ATP	AUTOSAR Template Profile
ATS	Acceptance Test Suite
AUTOSAR	AUTomotive Open System Architecture
B	
BFx	Bitfield functions for fixed point
BSW	Basic Software
BIST	Built-In Self Tests
BSWM	BSW Mode manager

△

Acronym/ Abbreviation	Description
BSWMD	Basic Software Module Description
C	
CAN	Controller Area Network
CC	Communication Controller
CCF	Common Cause Failure
CD	Complex Driver
CP	AUTOSAR Classic Platform
COM	Communication
COMM	COMmunication Manager
CM	Communication Module
CRC	Cyclic Redundancy Check
D	
DAC	Digital to Analog Converter
DCA	Data Class Adapter
DDS	Data Distribution Service
DEM	Diagnostic Event Manager
DES	Data Encryption Standard
DET	Default Error Tracer, supports development and runtime error reporting, see [1, Specification of Default Error Tracer]
DEXT	Diagnostic EXTract
DHCP	Dynamic Host Configuration Protocol
DIO	Digital Input/Output
DLC	Data Length Code
DLT	Log and Trace Protocol
DM	Diagnostic Manager
DMA	Direct Memory Access
DoIP	Diagnostics over Internet Protocol
DoS	Denial of Service
DP	Data Point
DSL	Domain Specific Languages
DTC	Diagnostic Trouble Code
DTD	Document Type Definition
E	
E2E	End-to-End
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm

▽

△

Acronym/ Abbreviation	Description
ECIES	Elliptic Curve Integrated Encryption Scheme
ECU	Electronic Control Unit
EDDSA	Edwards-Curve Digital Signature Algorithm
EEC	Executable Entity Cluster
EEPROM	Electrically Erasable Programmable Read-Only Memory
EM	Execution Management
EOC	Execution Order Constraint
EOCEERG	Execution Order Constraint Executable Entity Reference Groupe
F	
FC	Functional Cluster
FIFO	First In First Out
FIBEX	Field Bus Exchange Format
FO	AUTOSAR Foundation
FOTA	Firmware-Over-The-Air
FPU	Floating Point Unit
FQDN	Fully-Qualified Domain Name
FW	FireWire (by Apple, aka IEEE 1394)
G	
GCM	Galios/Counter Modester
GENIVI	GENeva In-Vehicle Infotainment
GPT	General Purpose Timer
GSM	Global System for Mobile Communication
H	
HMAC	Hash-based Message Authentication Code
HMI	Human Machine Interface
HPC	High performance computer or high performance computing machine
HTMSS	Hardware Test Management Startup and Shutdown
I	
I2C	Inter-Integrated Circuit
I-PDU	Interaction Layer Protocol Data Unit, see: I-PDU
ICC	Implementation Conformance Class, see: Implementation Conformance Class 1 ff.
ICMP	Internet Control Message Protocol
ICOM	Intelligent COMmunication controller
ICU	Input Capture Unit
ICV	Integrity Check Value
ID	IDentifier

▽

△

Acronym/ Abbreviation	Description
IDL	Interface Description Language
IEC	International Electrotechnical Commission
IFI	Interpolation Floating point
IFx	Interpolation Fixed point
IO	Input/Output
IP	Internet Protocol
ISO	International Standardization Organization
ISR	Interrupt Service Routine
IVC	In-Vehicle Communication
J	
JSON	JavaScript Object Notation
L	
LAN	Local Area Network
LBAP	Language Binding for the Adaptive Platform AP
L-PDU	Link Layer Protocol Data Units
L-SDU	Link Layer Service Data Units
LET	Logical Execution Time
LIFO	Last In First Out
LIN	Local Interconnected Network
LSB	Least Significant Bit
M	
MAC	Media Access Control
MAC	Message Authentication Code
mC	Microcontroller
MC	Measurement and Calibration
MCAL	Microcontroller Abstraction Layer
MCBD	Multicore BSW Distribution
MCU	MicroController Unit
MD	Message Digest
MDIO	Management Data Input/Output
ME	Mappable Element
MFI	Mathematical Floating point
MFx	Mathematical Fixed point
MIPS	Million Instructions Per Second
MMD	MDIO Manageable Device
MMU	Memory Management Unit
MMI	Man Machine Interface

▽

△

Acronym/ Abbreviation	Description
MOST	Media Oriented Systems Transport
mP	MicroProcessor
MPU	Memory Protection Unit
MSB	Most Significant Bit
MSTP	Microcontroller Specific Test Package
MTU	Maximum Transmission Unit
N	
NM	Network Management
N-PDU	Protocol Data Unit of the Network layer (transport protocol)
N-SDU	Service Data Unit of the Network layer (transport protocol)
NvM	Non-Volatile Memory
NVRAM	Non-Volatile Random Access Memory
O	
OBD	On-Board Diagnostic
OEM	Original Equipment Manufacturer
OIL	ISO 17356-6 (OSEK/VDX Implementation Language)
OS	Operating System
OSEK	Open Systems and their interfaces for Automotive Electronics[1]
P	
PCI	Protocol Control Information
PDEP	Profile of Data Exchange Point
PDU	Protocol Data Unit
PHM	Platform Health Management
PKCS	Public Key Cryptography Standards
PNC	Partial Network Cluster
POSIX	Portable Operating System Interface
PS	Product Supplier
PSK	Least Significant Bit
PTP	Precision Time Protocol
PWM	Pulse Width Modulation
R	
RAM	Random Access Memory
RDG	Runnable Dependency Graph
REST	Representational State Transfer

▽

[1] The original German phrase for this acronym is: **O**ffene **S**ysteme und deren Schnittstellen für die **E**lektronik im **K**raftfahrzeug

△

Acronym/ Abbreviation	Description
RfC	Request for Change
ROM	Read-Only Memory
RP	Rapid Prototyping
RPC	Remote Procedure Call
RSA	Cryptographic approach according to Rivest, Shamir, and Adleman
RTE	Run-Time Environment, provides the infrastructure services that enable communication to occur between AUTOSAR SWCs as well as acting as the means by which AUTOSAR SWCs access basic software modules including the OS and communication service. This includes the RTE's responsibility for the realization of the scheduling of the SWCs. See [2, Specification of RTE Software]
S	
SAE	Society of Automotive Engineers, see: SAE J1939
SD	Service Discovery
SDG	Special Data Group
SDU	Service Data Unit
SEooC	Safety Element out of Context
SDV	Software-Defined Vehicle
SHA	Secure Hash Algorithm
SHM	System Health Monitoring
SHWA	Safe Hardware Acceleration
SIL	Safety Integrity Level
SL-LET	System Level Logical Execution Time
SM	State Management
SOA	Service-Oriented Architecture
SOC	Service-Oriented Communication
SOME/IP	Scalable Service-Oriented Middleware over IP
SOVD	Service-Oriented Vehicle Diagnostics
SP	Synchronization Points
SPEM	Software and Systems Process Engineering Meta-model
SPI	Serial Peripheral Interface
SST	Signal Service Translation
SWC/SW-C	Software Component
SWCT/SWC-T	SWC-Template
SYST/SYS-T	System Template
T	
TC	Timed Communication
TC	Test Case

▽

△

Acronym/ Abbreviation	Description
TCP	Transmission Control Protocol
TD	Timing Description
TIMEX	TIMing EXTensions
TLS	Transport Layer Security
TLV	Tag Length Value
TP	Transport Protocol
TSN	Time Sensitive Network
TSY	Time Synchronization
TTL	Time To Live
TTP	Time Triggered Protocol
U	
UDP	User (Universal) Datagram Protocol
UDS	Unified Diagnostic Services
UDPNM	UDP Network Management
UML	Unified Modeling Language
USB	Universal Serial Bus
UTF	Universal coded character set Transformation Format
UUID	Universally Unique Identifier
V	
V2X	Vehicle-To-Everything
VDP	Vehicle Data Protocol
VFB	Virtual Functional Bus
VISS	Vehicle Information Service Specification
VLAN	Virtual Local Area Network
VMM	Vehicle Mode Management
VSA	Variable Size Array
VSM	Vehicle State Manager
W	
WCET	Worst Case Execution Time
WCRT	Worst Case Response Time
X	
XCP	Universal Calibration Protocol

Table 2.1: Acronyms and Abbreviations

3 How to read this document

The title of the sub-chapters is identical to the term to be defined.

3.1 <Term> Template

Definition	<term to be defined>
Initiator	<`Functional Cluster` which responsible for the term>
Further Explanations	<further explanation of the definition>
Comment	<comment or hints>
Example	<example of the term>
Reference	<reference of definition>

4 Definitions

4.1 Abstract Platform

Definition	A `Software Platform` defined by AUTOSAR for the definition of functional level communications independent of Classic / Adaptive / Offboard deployments.
Initiator	System Design
Further Explanations	An Abstract Platform exists only in the methodological and modeling sense and is not a physical deployable platform. In the context of AUTOSAR Classic and AUTOSAR Adaptive functional level communications means VFB level communications.
Comment	–
Example	–
Reference	–

4.2 Acceptance Test Suite

Definition	A test case description used in the context of Acceptance Testing
Initiator	Acceptance Testing
Further Explanations	ISO 9646 distinguishes between Abstract Test Suites and `Executable` Test Suites. For AUTOSAR the earlier relates to the Acceptance Test Specifications, whereas the latter to the test implementations or Acceptance Test Suites.
Comment	–
Example	–
Reference	[3], Parts 1,2 and 4

4.3 Access Control Decision

Definition	The Access Control Decision is a Boolean value indicating if the requested operation is permitted or not. It is based on the identity of the caller and the `Access Control Policy`
Initiator	Security
Further Explanations	In the case of `Identity and Access Management`, the 'caller' is an `Adaptive Applications`.
Comment	–
Example	–
Reference	–

4.4 Access Control Policy

Definition	Access Control Policies are bound to the targets of calls (i.e. `Service` interfaces) and are used to express what `Identity Information` is necessary to access those interfaces.
Initiator	Security
Further Explanations	Policies can be provided through `Configuration` / modeling or by statically pre-programming them into the PDP.
Comment	–
Example	–
Reference	–

4.5 Adaptability

Definition	Adaptability is the ability of a system to adjust itself to changed circumstances in its environment in order to continue to provide the intended `Functionality`.
Initiator	Safety
Further Explanations	One should distinguish between changes in the environment of the system/vehicle ("run-time adaptability") and changes in the development environment where software `Architecture` (like AUTOSAR) is used ("design-time predictability").
Comment	–
Example	–
Reference	• Antonio Carlos Schneider Beck, Carlos Arthur Lang Lisboa, Luigi Carro (eds.), Adaptable Embedded Systems, Springer Science & Business Media, 27 Nov 2012 • Twan Basten, Roelof Hamberg, Frans Reckers, Jacques Verriet, Model-Based Design of Adaptive Embedded Systems, Springer Science & Business Media, 15 Mar 2013

4.6 Adaptive Application

Definition	Software that follows the Adaptive AUTOSAR specifications and therefore can be deployed onto an Adaptive Platform instance. It consists of its implementation, operational data (e.g. map data) and its `Meta-data` given by the Application Design Model. In order to be deployable on different Adaptive Platforms, it only uses ARA programming interfaces including POSIX profile PSE51 APIs.
Initiator	Execution Management
Further Explanations	`Adaptive Applications` are generally more coarse grain than SW-Cs of the `Classic Platform`. They use exclusively Adaptive Platform APIs, and may offer and use services. They are implemented by one or several executables, byte code or libraries with defined entry points and may comprise multiple parts (e.g. libraries, data files).
Comment	The goal of Adaptive Platform is to achieve portability of `Adaptive Applications` among different implementations of the Adaptive Platform at least on source-code level, potentially also on object-code level.
Example	–
Reference	–

4.7 Adaptive Platform Foundation

Definition	Part of an Adaptive Platform implementation, which provides standardized platform `Functionality` to Applications via software interfaces (APIs).
Initiator	Software and `Architecture`
Further Explanations	The Adaptive Platform Foundation includes of core system functionalities such as OS, Execution Manager, Communication Management and Persistency.
Comment	The goal of Adaptive Platform is to achieve portability of `Adaptive Applications` among different implementations of the Adaptive Platform at least on source-code level, potentially also on object-code level.
Example	–
Reference	–

4.8 Adaptive Platform Services

Definition	Standard platform services that is provided by an application which is part of AUTOSAR platform implementation.
Initiator	Communication
Further Explanations	–
Comment	–
Example	–
Reference	–

4.9 Adaptive Platform Machine Configuration

Definition	Configuration model in the AUTOSAR Adaptive Platform that defines the target `Configuration` content whose scope is local to a specific target `Machine`.
Initiator	WG-MT
Further Explanations	The Adaptive Platform Machine Configuration is uploaded with executable code to a target `Machine` and supports the `Integration` of `Executables` onto the machine.
Comment	–
Example	–
Reference	–

4.10 Application Programming Interface

Definition	An Application Programming Interface (API) is the prescribed method of a specific software part by which a programmer writing a program can make requests to that software part.
Initiator	WG-MT
Further Explanations	–
Comment	–

▽

△

Example	[4] (OSEK/VDX Operating System)
Reference	–

4.11 Application Software Component

Definition	An Application Software Component is a specific `Software Component` which realizes a defined `Functionality` on application level and runs on the AUTOSAR infrastructure. It communicates only through the AUTOSAR Runtime Environment (RTE).
Initiator	Software and `Architecture`
Further Explanations	Application Software Components are located "above" the AUTOSAR Runtime Environment.
Comment	–
Example	–
Reference	–

4.12 Architecture

Definition	The fundamental organization of a system embodied in its components, their static and dynamic relationships to each other, and to the environment, and the principles guiding its design and evolution.
Initiator	Software and `Architecture`
Further Explanations	–
Comment	"Static and dynamic" added to definition of [5] .
Example	–
Reference	[6], [5]

4.13 Artifact

Definition	This is a Work Product definition that provides a description and definition for tangible work product types. Artifacts may be composed of other artifacts. At a high level, an artifact is represented as a single conceptual file.
Initiator	Methodology and Templates
Further Explanations	–
Comment	–
Example	–
Reference	–

4.14 ASIL Decomposition

Definition	See ISO-26262 ([7]) ID 1.7
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	See [7], ID 1.7

4.15 Asserted Property

Definition	A property or quality of a design entity (e.g. SW component or system) is asserted, if the design entity guarantees that this property or quality is fulfilled.
Initiator	Software and `Architecture`
Further Explanations	A property or quality of a design unit can be asserted by the design unit itself or in combination with another design unit.
Comment	–
Example	If the worst case `Execution Time` of a `Task` (w.r.t. a certain CPU etc.) is asserted to be 3 ms, the execution time of this task will under any circumstances be less than or equal to 3 ms.
Reference	Compare `Required Property`

4.16 Assessment

Definition	See ISO-26262 ([7]), ID 1.4
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	See [7], ID 1.4

4.17 Asset

Definition	An item that has been designed for use in multiple contexts.
Initiator	Software and `Architecture`
Further Explanations	–
Comment	–

▽

△

Example	An asset can be design, specifications, source code, documentation, test suits, manual procedures, etc. From a security perspective anything that has a value to any of the stakeholders such as critical data (information, software) and critical functions, that could potentially be subject to attacks and possibly, but not necessarily, motivates countermeasures.
Reference	[8], [5]

4.18 Asynchronous Communication

Definition	Asynchronous communication does not block the sending software entity. The sending software entity continues its operation without getting a response from the communication partner(s).
Initiator	Communication
Further Explanations	There could be an acknowledgement by the communication system about the sending of the information. A later response to the sending software entity is possible.
Comment	–
Example	–
Reference	–

4.19 Asynchronous Function

Definition	A `Function` is called asynchronous if the described `Functionality` is not guaranteed to be completed the moment the `Function` returns to the caller.
Initiator	Software and `Architecture`
Further Explanations	–
Comment	–
Example	–
Reference	–

4.20 Atomic Software Component

Definition	Non-composed Software-Component.
Initiator	Software and `Architecture`
Further Explanations	An Atomic `Software Component` might access HW or not, therefore not all Atomic SW-Cs are relocatable.
Comment	–
Example	Application Software-Component, Complex Driver
Reference	–

4.21 Audit

Definition	See ISO-26262 ([7]), ID 1.5
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	See [7], ID 1.5

4.22 Authenticity

Definition	The property that data originated from its purported source.
Initiator	Security
Further Explanations	–
Comment	–
Example	–
Reference	NIST SP 800-38B under Authenticity

4.23 Automotive API

Definition	The Automotive API is an interface that allows data-centric communication with the vehicle. It defines how other systems can access selected vehicle data securely and independently of the in-vehicle representation using a standardized interface across vehicle types and manufacturers.
Initiator	Communication
Further Explanations	–
Comment	–
Example	–
Reference	See [9]

4.24 Automotive API Gateway

Definition	The Automotive API Gateway is the `Functional Cluster` that offers the `Automotive API` and can thus connect clients to vehicle internal data.
Initiator	Communication
Further Explanations	–
Comment	–
Example	–
Reference	See [9]

4.25 Automotive Safety Integrity Levels

Definition	See ISO-26262 ([7]), ID 1.7
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	See [7], ID 1.7

4.26 AUTOSAR Adaptive Platform

Definition	An adaptive computing platform standardized by AUTOSAR. In a narrow term, it refers to its specification. In a broad term, it may refer to an instance of Adaptive Platform implementation.
Initiator	Software and `Architecture`
Further Explanations	–
Comment	–
Example	–
Reference	–

4.27 AUTOSAR Application Interface

Definition	A set of `Blueprints` which are standardized by AUTOSAR and which can be used for creating `AUTOSAR Interfaces` of an `Adaptive Application`. `AUTOSAR Interfaces` that are derived from `Standardized AUTOSAR Blueprints` are `Standardized AUTOSAR Interfaces`.
Initiator	Application Interfaces
Further Explanations	–
Comment	–
Example	–
Reference	

4.28 AUTOSAR Authoring Tool

Definition	An AUTOSAR Tool used to create and modify `AUTOSAR XML descriptions`.
Initiator	Methodology and Templates
Further Explanations	–
Comment	–
Example	System Description Editor
Reference	–

4.29 AUTOSAR Blueprint

Definition	An AUTOSAR Blueprint is a `Blueprint` for an AUTOSAR element. It also includes that it is specified within the AUTOSAR project which attributes are mandatory to be specified for the blueprint of a specific class of AUTOSAR element types as well as how to derive an AUTOSAR object from that blueprint.
Initiator	Application Interfaces
Further Explanations	The AUTOSAR meta-model supports the pre-definition of model elements taken as the basis for further modeling. These pre-definitions are called blueprints. [TPS_STDT_00002]
Comment	–
Example	–
Reference	[10]

4.30 AUTOSAR Converter Tool

Definition	An AUTOSAR Tool used to create AUTOSAR XML files by converting information from other AUTOSAR XML files.
Initiator	Methodology and Templates
Further Explanations	–
Comment	–
Example	ECU Flattener
Reference	–

4.31 AUTOSAR Definition

Definition	This is the definition of parameters which can have values. One could say that the parameter values are instances of the definitions. But in the meta-model hierarchy of AUTOSAR, definitions are also instances of the meta-model and therefore considered as a description.
Initiator	Methodology and Templates
Further Explanations	–
Comment	–
Example	–
Reference	–

4.32 AUTOSAR Interface

Definition	The AUTOSAR Interface of a software component refers to the collection of all Ports (`Port`) of that component through which it interacts with other components.
Initiator	Software and `Architecture`
Further Explanations	–

▽

△

Comment	Note that an AUTOSAR Interface is different from a `Port Interface`. The latter characterizes one specific `Port` of a component.
Example	–
Reference	See AUTOSAR Specification of [11] Chapter 4 *"Communication on the VFB"*.

4.33 AUTOSAR Meta-model

Definition	The AUTOSAR meta-model is a UML2.0 model that defines the language for describing AUTOSAR systems and related artifacts.
Initiator	Methodology and Templates
Further Explanations	–
Comment	The `AUTOSAR XML Schema` is derived from the AUTOSAR meta-model.
Example	–
Reference	[12]

4.34 AUTOSAR Model

Definition	This is an instance of the `AUTOSAR Meta-model`. The AUTOSAR Model represents aspects suitable to the intended use according to the AUTOSAR methodology.
Initiator	Methodology and Templates
Further Explanations	–
Comment	–
Example	–
Reference	–

4.35 AUTOSAR Partial Model

Definition	In AUTOSAR, the possible `Partitioning` of models is marked in the meta-model by <<atp Splitable>>. One partial model is represented in an `AUTOSAR XML description` by one file. The partial model does not need to fulfill all semantic constraints applicable to an AUTOSAR model.
Initiator	Methodology and Templates
Further Explanations	–
Comment	–
Example	–
Reference	–

4.36 AUTOSAR Processor Tool

Definition	An AUTOSAR Tool used to create non-AUTOSAR files by processing information from AUTOSAR XML files.
Initiator	Methodology and Templates
Further Explanations	–
Comment	–
Example	RTE Generator
Reference	–

4.37 AUTOSAR Runtime for Adaptive Applications

Definition	A set of standard application interfaces provided by `Functional Clusters`, which belong to either `Adaptive Platform Foundation` or `Adaptive Platform Services`.
Initiator	General
Further Explanations	–
Comment	–
Example	–
Reference	–

4.38 AUTOSAR Run-Time Interface (ARTI)

Definition	The AUTOSAR Run-Time Interface shall define an interface between build tools and `Debugging`/`Tracing` tools.
Initiator	Software and `Architecture`
Further Explanations	The interface shall ease and speed up the `Debugging`, `Tracing` and `Verification` of `System` behavior as well as round-trip engineering.
Comment	
Example	–
Reference	AUTOSAR Specification of ARTI ([13])

4.39 AUTOSAR Service

Definition	The term `Service` is used in the layered software architecture to denote the highest layer of the `AUTOSAR Software` architecture that interacts with the application. The term service is used in the meaning defined by the service-oriented architecture. This meaning has the strongest relation to the usage of the term service on the `AUTOSAR Adaptive Platform`. The term service is also used to describe the handling of diagnostic services, e.g. UDS service ReadDataByIdentifier, for the communication between a diagnostic tester and a diagnostic stack on an (AUTOSAR) `Electronic Control Unit` (ECU).
Initiator	Software and `Architecture`

▽

△

Further Explanations	–
Comment	–
Example	–
Reference	–

4.40 AUTOSAR Software

Definition	Application Software or Basic/Platform Software developed according to the AUTOSAR Standard.
Initiator	–
Further Explanations	–
Comment	–
Example	–
Reference	–

4.41 AUTOSAR Tool

Definition	This is a software tool which supports one or more tasks defined as AUTOSAR tasks in the methodology. Depending on the supported tasks, an AUTOSAR tool can act as an `AUTOSAR Authoring Tool`, an `AUTOSAR Converter Tool`, an `AUTOSAR Processor Tool` or as a combination of those.
Initiator	Methodology and Templates
Further Explanations	–
Comment	–
Example	–
Reference	–

4.42 AUTOSAR XML description

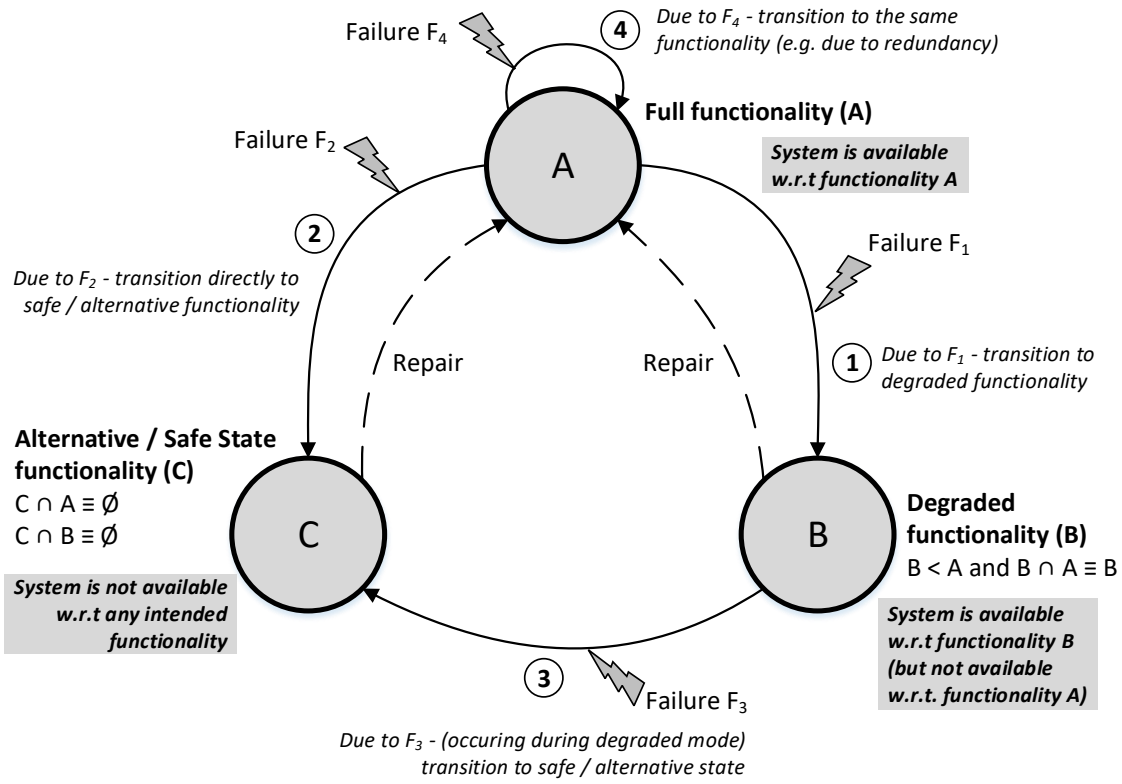
Definition	In AUTOSAR this is a serialized AUTOSAR model. In fact an AUTOSAR XML description is the XML representation of an `AUTOSAR Model`. The AUTOSAR XML description can consist of several files. Each individual file represents an `AUTOSAR Partial Model` and must validate successfully against the `AUTOSAR XML Schema`.
Initiator	Methodology and Templates
Further Explanations	–
Comment	–
Example	–
Reference	–

4.43 AUTOSAR XML Schema

Definition	The AUTOSAR XML Schema is an XML language definition for exchanging (AUTOSAR Model) and descriptions.
Initiator	Methodology and Templates
Further Explanations	The AUTOSAR XML Schema is a W3C XML schema that defines the language for exchanging AUTOSAR models. This Schema is derived from the AUTOSAR Meta-model. The AUTOSAR XML Schema defines the AUTOSAR data exchange format.
Comment	–
Example	–
Reference	–

4.44 Availability

Definition	1. Availability is the ability of the system to perform a Function A completely according to its specification. 2. The ratio of the total time the system is performing a function A (according to 1) during a given interval to the length of the interval. Alternative: The probability that the system is performing the function A at a specified time t. 3. In a degraded Mode the system has the ability to perform a subset B of A if full A is not available. In this case, the Functionality B is available.
Initiator	Safety
Further Explanations	see Figure 4.1 for more details.
Comment	1. Degraded modes are covered by this definition (see example)
Example	1. Power Steering: if the support Function fails it is not available while the steering as a base function has full availability. 2. From a security perspective availability is an attribute that ensures correct and timely access upon demand by an authorized entity.
Reference	See [7], ID 1.8



NOTES:
1. This diagram assumes the system is already in full intended functionality mode and neglects the start-up, shut-down, reset, etc. transitions which lead to this state.
2. The repair transitions are conceptual transitions for ilustration purposes.

Figure 4.1: Explanation of Availability

4.45 Basic Software

Definition	The Basic Software (BSW) provides the infrastructural (schematic dependent and schematic independent) functionalities of an `Electronic Control Unit`. It consists of `Integration Code` and `Standard Software`.
Initiator	Software and `Architecture`
Further Explanations	–
Comment	–
Example	`Microcontroller Abstraction Layer` (MCAL), `AUTOSAR Service`, Communication Layer
Reference	–

4.46 Basic Software Module

Definition	A collection of software files (code and description) that define a certain basic software `Functionality` present on an `Electronic Control Unit`.
Initiator	Software and `Architecture`
Further Explanations	`Standard Software` may be composed of several software modules that are developed independently. A software module may consist of `Integration Code` and/ or `Standard Software`.
Comment	–
Example	A Digital IO Driver, `Complex Driver`, OS are examples of Basic Software Modules.
Reference	–

4.47 Bit Position

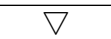
Definition	In AUTOSAR the bit position N within an `I-PDU` denotes the bit I, with I = N modulo 8, within the byte J, with J = N / 8. The byte J and bit position I is interpreted in accordance to the definition in ISO 17356-4 (OSEK/VDX Communication).
Initiator	Communication
Further Explanations	–
Comment	–
Example	–
Reference	–

4.48 Blueprint

Definition	This is a model from which other models can be derived by copy and refinement. Note that in contrast to meta-model resp. types, this process is not an instantiation.
Initiator	Methodology and Templates
Further Explanations	–
Comment	–
Example	Standardized Blueprint and `AUTOSAR Blueprint`.
Reference	–

4.49 Bulk Data

Definition	"Bulk Data" is a set of data such big in size, that standard mechanisms used to handle smaller data sets become inconvenient. This implies that bulk data in a software system are modeled, stored, accessed and transported by different mechanisms than smaller data sets.
Initiator	Software and `Architecture`
Further Explanations	Bulk data are typically handled by adding a level of abstraction (e.g. files) which separates the containment of the data from the internal structure.



△

Comment	The critical size, above which data must be regarded as bulk data depends on the technical infrastructure (e.g. bus system) and the considered `Use Case` (transport, storage etc.).
Example	Data on a persistent medium which has a capacity of a few kBytes (e.g. EEPROM) can be directly accessed via memory addresses, address offsets can be mapped to symbols of a programming language: No bulk data mechanisms are needed. For media with bigger capacity this becomes inconvenient or even impossible, so that a file system is used: The data are treated as bulk data.
Reference	–

4.50 Bus Mirroring

Definition	Forwarding information from an internal vehicle bus to an external bus, e.g. the diagnostic `Connector`.
Initiator	Communication
Further Explanations	Bus Mirroring is used to make internal buses accessible to external testers such that internal buses can be debugged in case of errors.
Comment	Because the external (or intermediary) buses typically do not have sufficient bandwidth to transport all information from an internal bus, filters have to be applied to select the frames that are relevant to the analysis.
Example	An `Electronic Control Unit` does not go to sleep in a vehicle. The engineer wants to check the NM traffic to check for irregular behavior, e.g. concerning partial networking information.
Reference	–

4.51 Bus Wake-Up

Definition	A bus wake-up is caused by a specific wake pulse on the bus defined within the specification of the dedicated communication standard (e.g. CAN, LIN, FR). A bus wake-up initiates that the transceiver and controller leave their energy saving `Mode` and enter normal mode to start bus communication again.
Initiator	Communication
Further Explanations	–
Comment	–
Example	–
Reference	–

4.52 Bypassing

Definition	The experimental incorporation of new `Functionality` within an ECU image.
Initiator	Runtime Environment
Further Explanations	Bypassing involves the incorporation of new `Functionality` or to replace existing functionality to an existing ECU image without requiring that the image be rebuilt.

▽

△

Comment	Bypassing can be either "internal" where the new/ replacement `Functionality` is present on the ECU image or "external" where an `Rapid Prototyping Tool` provides the functionality out with the ECU.
Example	An RP tool intercepts the output of a bypassed RunnableEntity via the RP Memory Interface and replaces the value with the bypass result. Subsequent RunnableEntitys then process the bypass value rather than the original result.
Reference	–

4.53 Calibration

Definition	Calibration is the adjustment of parameters of `Software Components` realizing the control functionality (namely parameters of AUTOSAR SW-Cs, ECU abstraction or `Complex Driver`.
Initiator	Software and `Architecture`
Further Explanations	Only those software modules can be calibrated, which are above RTE and ECU Abstraction and CDD. `Calibration` is always done at post-build time. Used techniques to set `Calibration Data` include end-of-line programming, garage programming and adaptive calibration (e.g. in the case of anti-pinch protection for power window).
Comment	–
Example	The `Calibration` of the engine control will take into account the production differences of the individual motor this `System` will control.
Reference	–

4.54 Call Point

Definition	A point in a `Software Component` where the SWC enforce an execution entity (`Entry Point`) in another SWC.
Initiator	Software and `Architecture`
Further Explanations	–
Comment	–
Example	Request Service Send Information
Reference	–

4.55 Callback

Definition	Functionality that is defined by an AUTOSAR module so that lower-level modules (i.e. lower in the Layered Software Architecture) can provide `Notification` as required (e.g. when certain events occur or asynchronous processing completes).
Initiator	Software and `Architecture`

▽

△

Further Explanations	In AUTOSAR, modules usually provide a register mechanism for callback functions which is set through `Configuration`. A module provides callbacks so that other modules can initiate its processing while the module calls `Callbacks` to execute functionality that could not be specified by AUTOSAR, i.e. `Integration Code`.
Comment	–
Example	(from the viewpoint of a particular SWS): The module being specified (Msws) should be informed about an `Event` in another module (Mexternal). In this example, Msws calls Mexternal to perform some processing and can only resume when Mexternal completes. Upon completion, Mexternal calls Msws's `Callback Function`. That is, the called module (Mexternal) CALLS the calling module (Msws) BACK when complete ==> a `Callback`.
Reference	–

4.56 Callout

Definition	`Function` stubs that the system designer can replace with code to add functionality to a module which could not be specified by AUTOSAR.
Initiator	Software and `Architecture`
Further Explanations	A module calls callouts to execute functionality that could not be specified by AUTOSAR, i.e. `Integration Code` while the module provides `Callbacks` so that other modules can initiate its processing. Callouts can be separated into two classes: 1) callouts that provide mandatory functionality and thus serve as a `Hardware Abstraction Layer` 2) callouts that provide optional functionality
Comment	–
Example	In the EcuM: For class 1): EcuM_EnableWakeupSources For class 2): The Init Lists (EcuM_AL_DriverInitZero)
Reference	–

4.57 CAN XL

Definition	CAN XL specifies a protocol and physical layer with higher `Data` rate and clearer separation of concerns, that builds on CAN 2.0/FD and features tunneling of Ethernet frames.
Initiator	Communication
Further Explanations	Besides supporting tunneled legacy CAN 2.0 / CAN FD communication between any type of `Electronic Control Unit` (ECU) it shall be also possible to directly tunnel IEEE 802.3 Ethernet frames for e.g. participation of IP communication. This leads to a vehicle wide possible communication with same semantic used everywhere regardless physical connection (CAN XL / Ethernet) or communication paradigm (Signal- and `Service`-based communication).
Comment	CAN XL will help bridge the gap between current CAN implementations and current 100 Mbit Ethernet solutions. On the same `Network` segment, both CAN 2.0/FD/XL and Ethernet traffic can coexist. Baudrate is not fixed to 10 Mbit like at 10BASE-T1S but can be adjusted flexible up to 20 Mbit
Example	Tunneling Ethernet frames to CAN XL connected ECUs.

▽

△

Reference	[14, CiA610-1], [15, CiA-611-1]

4.58 Cascaded Switch

Definition	A Cascaded Switch is an Ethernet switch that exists of at least two Ethernet switches: a `Master Switch` and a `Slave Switch`. The Master Switch and the Slave Switch are connected by `Uplink Ports`.
Initiator	Communication
Further Explanations	–
Comment	–
Example	Request Service Send Information
Reference	–

4.59 Cascading Failure

Definition	See ISO-26262 ([7]), ID 1.13
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	See [7], ID 1.13

4.60 Category 1 Interrupt

Definition	Category 1 (Cat1) Interrupts are supported by the OS but their code is only allowed to call a very small subset of OS functions. Furthermore they can bypass the OS. The code of Category 1 Interrupts depends (normally) on the used compiler and `Microcontroller`. Category 1 Interrupts are not allowed to use the ISR() macro. Category 1 Interrupts need to implement/ establish their own `Interrupt Frame`. Nevertheless they have to be configured in order to be included in the `Interrupt Vector Table`.
Initiator	Software and `Architecture`
Further Explanations	–
Comment	–
Example	–
Reference	–

4.61 Category 2 Interrupt

Definition	Category 2 (Cat2) Interrupts are supported by the OS and their code can call a subset of OS functions. The definition of the Cat2 Interrupt must use the ISR() macro in order to be recognized by the OS. The `Interrupt Frame` of a Category 2 Interrupt is managed by the OS.
Initiator	Software and `Architecture`
Further Explanations	–
Comment	–

Example	ISR(timer1)
	{
	/* here is the code which handles timer1 interrupts */
	...
	}

Reference	–

4.62 Causality of Transmission

Definition	Transmit order of PDUs with the same identifier (instances of `Protocol Data Units`, PDUs) from a source `Network` is preserved in the destination network.
Initiator	Communication
Further Explanations	Transmission of `Protocol Data Unit` (PDU) with the same identifier has a particular temporal order in a given source `Network`. After routing over a `Gateway` the temporal order of transmission of PDUs in a destination network may be changed. Only in case that the temporal order is the same, causality is given. Otherwise causality is violated. Causality can be in contradiction to prioritization of PDUs.
Comment	–
Example	–
Reference	–

4.63 Classic Platform

Definition	A `Software Platform` defined by AUTOSAR for deeply embedded systems and Application Software with high demands regarding predictability, `Safety` and responsivness.
Initiator	General
Further Explanations	–
Comment	–
Example	–
Reference	–

4.64 Client

Definition	Software entity which uses services of a `Server`.
Initiator	Software and `Architecture`
Further Explanations	The client and the `Server` might be located on one `Client-Server Communication` or distributed on different calculation units (e.g. `Electronic Control Unit` (ECU), external diagnostic tester).
Comment	Adapted from [16]
Example	–
Reference	[16]

4.65 Client-Server Communication

Definition	A specific form of communication in a possibly distributed system in which software entities act as `Clients`, `Servers`, or both, where 1...n clients are requesting services via a specific protocol from typically one Server.
Initiator	Communication
Further Explanations	Client-server communication can be realized by synchronous or asynchronous communication. • `Client` takes initiative: requesting that the `Server` performs a `Service`, e.g. client triggers action within server (server does not start action on its own) • Client is after service request blocked / non-blocked • Client expects response from server: `Data Flow` (and control flow, if blocked) One example for 1 client to n server communication (currently not supported) is a functional request by diagnosis. This has to be treated as a specific exception.
Comment	–
Example	Internet (TCP/IP)
Reference	–

4.66 Client-Server Interface

Definition	The Client-Server interface is a special kind of `Port Interface` used for the case of `Client-Server Communication`. The Client-Server interface defines the operations that are provided by the `Server` and that can be used by the `Client`.
Initiator	Communication
Further Explanations	–
Comment	–
Example	–
Reference	AUTOSAR Specification of `Virtual Functional Bus` ([11])

4.67 Cluster Signal

Definition	A cluster signal represents the aggregating `System Signal` on one specific communication cluster. Cluster signals can be defined independently of frames. This allows a development methodology where the signals are defined first, and are assigned to frames in a later stage.
Initiator	Communication
Further Explanations	–
Comment	–
Example	–
Reference	–

4.68 Code Generator

Definition	The Code Generator consumes complete and correctly formed XML for a `Basic Software Module` (BSW) and generates code and data that configures the module.
Initiator	Software and `Architecture`
Further Explanations	–
Comment	–
Example	–
Reference	–

4.69 Code Variant Coding

Definition	Adaptation of SW by selection of functional alternatives according to external requirements
Initiator	Software and `Architecture`
Further Explanations	Code Variant Coding might influences RTE (Runtime Environment) and Basic Software Modules, not only the application software modules. Code Variant Coding is always done at pre-compile time or at link time. Code Variant Coding also includes vehicle-specific (not user-specific) SW adaptation due to end-customer wishes (e.g. deactivation of speed dependent automatic locking).
Comment	In case of the C language the #if or #ifdef directive can be used for creating code variants. Code Variant Coding is a design time concept.
Example	The same window lifter ECU is used for cars with 2 and 4 doors, however different code segments have to be used in both cases.
Reference	

4.70 Common Cause Failure

Definition	See ISO-26262 ([7]), ID 1.14
Initiator	Safety
Further Explanations	–

▽

△

Comment	–
Example	–
Reference	See [7], ID 1.14

4.71 Communication Attribute

Definition	Communication attributes define, according to the development phase, behavioral as well as implementation aspects of the AUTOSAR communication patterns.
Initiator	Communication
Further Explanations	The exact characteristics of the communication patterns provided by AUTOSAR (`Client-Server Communication` and `Sender-Receiver Communication`) can be specified more precisely by communication attributes.
Comment	See AUTOSAR Specification of [11] Chapter 4 *"Communication on the VFB"*.
Example	–
Reference	AUTOSAR Specification of `Virtual Functional Bus` ([11])

4.72 Complex Driver

Definition	A software entity not standardized by AUTOSAR that can access or be accessed via `AUTOSAR Interfaces` and/or `Basic Software Module` APIs.
Initiator	Software and `Architecture`
Further Explanations	CDD used to be the acronym for Complex Device Driver, but is not limited to drivers.
Comment	–
Example	• Communication stack CDD to support the communication on a bus not supported by AUTOSAR • Reuse of legacy SW • Integration of software with high HW interaction requirements within a standardized AUTOSAR Architecture
Reference	–

4.73 Composition

Definition	An AUTOSAR Composition encapsulates a collaboration of `Software Components`, thereby hiding detail and allowing the creation of higher abstraction levels. Through Delegation `Connectors` a Composition explicitly specifies, which `Ports` of the internal components are visible from the outside. AUTOSAR Compositions are a type of Components, e.g. they can be part of further compositions.
Initiator	Software and `Architecture`
Further Explanations	–
Comment	See `Virtual Functional Bus` Specification, [11] Chapter 4 *"Communication on the VFB"*

▽

△

Example	–
Reference	AUTOSAR Specification of `Virtual Functional Bus` ([11])

4.74 Compositionality

Definition	Compositionality is given when the behavior of a `Software Component` or subsystem of a `Systems` is independent of the overall system load and `Configuration`.
Initiator	Software and `Architecture`
Further Explanations	Compositionality is an important property of deterministic systems. This property leads to a complete decoupling of systems. Smooth subsystem `Integration` without backlashes is then easily achievable.
Comment	–
Example	A new component or a subsystem can be added to a system without changing the behavior of the original components.
Reference	–

4.75 Conditioned Signal

Definition	The conditioned signal is the internal electrical representation of the `Electrical Signal` within the `Electronic Control Unit` (ECU). It is delivered to the processor and represented in voltage and time (or, in case of logical signals, by high or low level).
Initiator	General
Further Explanations	The `Electrical Signal` usually can not be processed by the peripherals directly, but has to be adopted. This includes amplification and limitation, conversion from a current into a voltage and so on. This conversion is performed by some electronic devices in the ECU and the result of the conversion is called the Conditioned Signal. The description means for the `Conditioned Signal` can also be the same as for `Technical Signals` and Electrical Signals, but limited to electrical voltage.
Comment	–
Example	–
Reference	–

4.76 Confidentiality

Definition	The property that data or information is not made available or disclosed to unauthorized persons or processes.
Initiator	Security
Further Explanations	–
Comment	–
Example	–
Reference	NIST SP 800-66 Rev. 1 under Confidentiality from 45 C.F.R., Sec. 164.304

4.77 Configuration

Definition	The arrangement of hardware and/or software elements in a system.
Initiator	Software and Architecture
Further Explanations	A configuration in general takes place before runtime.
Comment	–
Example	–
Reference	[5], [17]

4.78 Confirmation

Definition	Service primitive defined in the ISO/OSI Reference model ([18]). With the 'confirmation' service primitive a service provider informs a service user about the result of a preceding service request of the service user.
Initiator	Software and Architecture
Further Explanations	A confirmation is e.g. a specific notification generated by the underlying layer to inform about a Message Transmission Error.
Comment	–
Example	–
Reference	[19]

4.79 Connector

Definition	A Connector connects Ports of Software Components and represents the flow of information between those ports.
Initiator	Software and Architecture
Further Explanations	–
Comment	For more information see AUTOSAR Specification of VFB ([11])
Example	AssemblyConnector, DelegationConnector
Reference	[11]

4.80 Control Flow

Definition	The directed transmission of information between multiple entities, directly resulting in a state change of the receiving entity.
Initiator	Software and Architecture
Further Explanations	A state change could result in an activation of a schedulable entity.
Comment	–
Example	–
Reference	–

4.81 Coordinate

Definition	To control and harmonize two or more events or operations to act in an organized and predictable way.
Initiator	Communication
Further Explanations	–
Comment	–
Example	Two NM Channels can be coordinated to synchronize different stages of `Network` sleep.
Reference	

4.82 Data

Definition	A reinterpretable representation of information in a formalized manner suitable for communication, interpretation or processing.
Initiator	General
Further Explanations	–
Comment	–
Example	`Flag`, `Notification`, etc.
Reference	[20]

4.83 Data Element

Definition	Data elements are declared within the context of a `Sender-Receiver Interface`. They serve as the data units that are exchanged between sender and receiver.
Initiator	Software and `Architecture`
Further Explanations	–
Comment	–
Example	–
Reference	[21]

4.84 Data Flow

Definition	The directed transmission of `Data` between multiple entities. The transmissioned data are not directly related to a state change at the receiver side.
Initiator	Software and `Architecture`
Further Explanations	–
Comment	–
Example	Asynchronous communication.
Reference	–

4.85 Data Variant Coding

Definition	Adaptation of SW by setup of certain characteristic `Data` according to external requirements.
Initiator	Software and `Architecture`
Further Explanations	Data Variant Coding might influence RTE (Runtime Environment) and `Basic Software Modules`, not only the application software modules (Multiple `Configuration` parameter sets are needed). `Variant Coding` also includes vehicle-specific (not user-specific) SW adaptation due to end-customer wishes (e.g. deactivation of speed dependent automatic locking). Used techniques to select variants include end-of-line programming and garage programming.
Comment	The major difference with `Calibration` is that this later doesn't aim to adapt the SW `Functionality` itself but only aims to adjust the characteristic data of the SW to the HW/SW environment. Characteristic `Data` in the source code of a software `Function` have a significant impact on the functionality of the software.
Example	- Steering wheel controller adaptation to the left or right side can be done with `Variant Coding`. (Selection of the `Configuration`.) - Country related adaptation of MMI with respect to speed and/or temperature unit (km/h vs. mph, ℃ vs. F).
Reference	

4.86 Deadline

Definition	The point in time when an execution of an entity must be finished.
Initiator	Software and `Architecture`
Further Explanations	A deadline is calculated dependent on its local reference system.
Comment	–
Example	–
Reference	[22]

4.87 Debugging

Definition	Debugging is the process of gathering information in case of a software problem. The information is used to analyze the software problem.
Initiator	Software and `Architecture`
Further Explanations	To analyze and later fix a software problem, in many cases more information than the one provided by the software API is necessary. This can be for example the state of internal variables of the software or a trace of the communication. The information can be collected by different means, e.g. an emulator or a `Tracing` tool for the communication bus.
Comment	–
Example	–
Reference	–

4.88 Dependability

Definition	Dependability is defined as the trustworthiness of a computer system such that reliance can justifiable be placed on the `Service` it delivers.
Initiator	Software and `Architecture`
Further Explanations	–
Comment	–
Example	–
Reference	[5]

4.89 Dependent Failure

Definition	See ISO-26262 ([7]), ID 1.22
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	See [7], ID 1.22

4.90 Diagnostic Coverage

Definition	See ISO-26262 ([7]), ID 1.25
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	See [7], ID 1.25

4.91 Diagnostic Event

Definition	A diagnostic event defines the atomic unit that can be handled by the DEM module. The status of a diagnostic event represents the result of a monitor. The DEM receives the result of a monitor from a `Software Component` (SW-C) via the RTE or other `Basic Software Modules`.
Initiator	Diagnostics
Further Explanations	–
Comment	For definition of 'monitor' see chapter [23] Chapter 7.3.5 *"Diagnostic monitor definition"* in Specification of DEM.
Example	–

▽

△

Reference	[23]

4.92 Direct Memory Access

Definition	DMA (Direct Memory Access) is a hardware mechanism that allows data transfers between memory regions or between memory and peripherals without CPU involvement in each individual data movement step. A dedicated DMA controller manages the transfer autonomously, freeing the CPU for other processing tasks.
Initiator	–
Further Explanations	In AUTOSAR `Classic Platform`, the DMA Driver (module `Dma`) provides a standardized API to configure and control DMA transfers. Each transfer is described by a Transfer Control Descriptor (TCD) that specifies source address, destination address, transfer element width, and loop counts. The DMA Driver is part of the MCAL layer.
Comment	The term DMA refers both to the hardware peripheral (DMA controller) and to the data movement mechanism. This specification uses “DMA” in both contexts; the intended meaning is clear from context.
Example	An SPI driver uses a DMA channel to transfer a 256-byte frame from a memory buffer to the SPI transmit register without CPU intervention, triggering a completion notification via callback upon transfer end.
Reference	[24]

4.93 Diversity

Definition	See ISO-26262 ([7]), ID 1.28
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	See [7], ID 1.28

4.94 Dynamic PDU

Definition	`Protocol Data Unit` with dynamic identifier.
Initiator	Communication
Further Explanations	Dynamic PDUs are PDUs where the <bus> identifier (e.g. CAN ID) is dynamically assigned (transmission) or evaluated (reception) at run time.
Comment	AUTOSAR supports two types of dynamic PDUs in CanIf: CanIf_SetDynamicTxId (only transmission), and PDUs with Meta-data (reception and transmission).
Example	PDU with variable source address, encoded in the CAN ID, e.g. ISO15765 NormalFixed.
Reference	–

4.95 Dynamic Routing

Definition	The routing of signals or `Protocol Data Units` in a `Gateway` can be changed throughout operation without change of the operation `Mode` of the Gateway.
Initiator	Communication
Further Explanations	Dynamic routing requires the change of routing tables during operation. It is not intended to use dynamic routing in the `Gateway`.
Comment	–
Example	–
Reference	[5]

4.96 E2E Profile

Definition	A functional and complete description of a specific communication stack in terms of data structures, `Services`, behavioral state-machines, `Error` handling. E2E Profiles are defined in AUTOSAR E2E Library. An E2E Profile is configurable by runtime parameters. A specific set of runtime parameters is called E2E profile `Variant`. In order to reach interoperability, the application developers should use the E2E profile variants defined in the E2E library.
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	–

4.97 E2E Protection Alive Counter

Definition	An Alive Counter is a counter which is incremented in every transmission request. The counter value is checked at the receiver side, whether it changes at all, but not if the counter was incremented correctly.
Initiator	Safety
Further Explanations	With an alive counter, the receiver monitors if new values arrive, this means it checks if the sender is still alive
Comment	–
Example	[25] Annex D.2.4 EXAMPLE 3: Communication protocols can contain information such as identifiers for communication objects, keep-alive messages, alive counters, sequence numbers, error detection codes and error-correcting codes.
Reference	–

4.98 E2E Protection Sequence Counter

Definition	A Sequence Counter is a counter which is incremented in every transmission request. The value of the counter value is checked at the receiver side for each message, whether it was incremented correctly.
Initiator	Safety
Further Explanations	A Sequence Counter is used to check whether messages got lost or a message is repeated.
Comment	–
Example	[25] Annex D.2.4 EXAMPLE 3: Communication protocols can contain information such as identifiers for communication objects, keep-alive messages, alive counters, sequence numbers, error detection codes and error-correcting codes.
Reference	–

4.99 ECU Abstraction Layer

Definition	The `Electronic Control Unit` (ECU) Abstraction Layer is located above the `Microcontroller Abstraction Layer` and abstracts from the ECU schematic. It is implemented for a specific ECU and offers an API for access to peripherals and devices regardless of their location (onchip/offchip) and their connection to the `Microcontroller` (`Port` pins, type of interface). Task: Make higher software layers independent of the ECU hardware layout.
Initiator	Software and `Architecture`
Further Explanations	The ECU Abstraction Layer consists of the following parts: • I/O Hardware Abstraction • Communication Hardware Abstraction • Memory Hardware Abstraction • Crypto Hardware Abstraction • Onboard Device Abstraction Properties: • Implementation: μC independent, ECU hardware dependent • Upper Interface (API): μC and ECU hardware independent, dependent on signal type
Comment	–
Example	–
Reference	See Layered Software Architecture ([26])

4.100 ECU Configuration

Definition	Activity of integrating and configuring one ECU's software.
Initiator	Methodology and Templates
Further Explanations	Further Explanations: ECU Configuration denotes the activity when one ECU's software is set up for a specific usage inside the `Electronic Control Unit`. In AUTOSAR the ECU Configuration activity is divided into "Pre-compile time", "Link time" and "Post-build time" configuration.
Comment	–

▽

△

Example	–
Reference	`ECU Configuration Description`, `Pre-Compile Time Configuration`, `Link Time Configuration`, `Post-build Time Configuration`.

4.101 ECU Configuration Description

Definition	Output of the `ECU Configuration` activity containing the values of `Configuration` parameters and references.
Initiator	Methodology and Templates
Further Explanations	ECU Configuration Description holds the `Configuration` parameter values and references to other module's configurations which have been defined in the `ECU Configuration` activity.
Comment	ECU Configuration Description may contain the whole `ECU Configuration` information or only the parts relevant for a specific `Configuration` step (e.g. `Pre-Compile Time Configuration`).
Example	–
Reference	ECU Configuration Description (see [27]), `Pre-Compile Time Configuration`, `Link Time Configuration`, `Post-build Time Configuration`.

4.102 ECU HW

Definition	A container term for any combination of AP `Machines` and CP ECUs. In general terms, an ECU-HW is typically abstracted/independent from physical or virtual realization and does not convey any HW-Housing i.e. multiple physical ECU-HWs or a physical ECU-HW providing multiple virtual ECU-HWs (unless otherwise stated).
Initiator	Methodology and Templates
Further Explanations	–
Comment	–
Example	–
Reference	–

4.103 ECU Instance

Definition	An ECU Instance represents a single instantiation of a `Classic Platform` stack, that may run on physical ECU-HW (without `Hypervisor`) or on virtual ECU-HW (with hypervisor).
Initiator	Methodology and Templates
Further Explanations	–
Comment	–
Example	–
Reference	–

4.104 Electrical Signal

Definition	The electrical signal is the electrical representation of `Technical Signals`. Electrical signals can only be represented in voltage, current and time
Initiator	General
Further Explanations	When a sensor processes the `Technical Signal` it is converted into an Electrical Signal. The information can be provided in the current, the voltage or in the timely change of the signal (e.g. a pulse width modulation).
Comment	To describe the Electrical Signal the same means as for the `Technical Signal` can be used, limited to electrical current and voltage.
Example	–
Reference	–

4.105 Electronic Control Unit (ECU)

Definition	Embedded computer `System` consisting of at least one CPU and corresponding peripherals which are placed in one housing.
Initiator	General
Further Explanations	An ECU is typified by a connection to one or more in-vehicle networks, sensors and actuators.
Comment	–
Example	–
Reference	–

4.106 Empty Function

Definition	Any C function defined by an AUTOSAR specification which does not implement or alter behavior required to accomplish the assigned functional responsibility.
Initiator	Software and `Architecture`
Further Explanations	As such an empty function in the context of AUTOSAR can still have code but this code shall not impact the state `Machine` other than `Error` reporting. Auxiliary code like validating arguments to report to the DET does not constitute functional behavior because without the code and proper calling this code would still fulfill its architectural responsibility.
Comment	–
Example	–
Reference	–

4.107 Entry Point

Definition	A point in a `Software Component` where an execution entity of the SW-C begins.
Initiator	Software and `Architecture`
Further Explanations	–

▽

△

Comment	–
Example	• Service of the Server in Client-Server Communication • Reaction after receive Information (Notification)
Reference	–

4.108 Error

Definition	See ISO-26262 ([7]), ID 1.36
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	See [7], ID 1.36

4.109 Error Detection Rate

Definition	Ratio between detected lost/faulty words/symbols/blocks, divided by the total number of symbols/ words/blocks sent.
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	–

4.110 Ethernet Switch Port Groups

Definition	Ethernet Switch Port groups are Ethernet switch Ports of an Ethernet switch which are grouped to so called Port groups. Ethernet Switch Port groups are only relevant for the host Electronic Control Unit (ECU). Ethernet Switch Port Groups are derived from the model per VLAN and per PNC. The host port is participating in all port groups. A Ethernet Switch Port Group could be a mix of internal and external ports.
Initiator	Communication
Further Explanations	–
Comment	–
Example	–
Reference	–

4.111 Event

Definition	State change of a hardware and/or software entity.
Initiator	Software and Architecture
Further Explanations	See OS Event, RTE Event, Diagnostic Event and Event Message (SOME IP)
Comment	–
Example	–
Reference	–

4.112 Event Message (SOME/IP)

Definition	Event - a message sent by an Electronic Control Unit (ECU) implementing a Service Instance to an ECU using this service instance (Publish/Subscribe).
Initiator	Communication
Further Explanations	Eventgroup - a logical grouping of 1 or more events. An eventgroup is part of a Service.
Comment	–
Example	–
Reference	–

4.113 Executable

Definition	Part of an application which consists of either a file containing executable code with a defined entry point and suitable for the platform instance as the target of deployment (deployment time) or software code which is ready to be integrated for a specific platform.
Initiator	Execution Management
Further Explanations	In POSIX systems, an executable is typically running within a single Process. Therefore, intra-executable communication is different from inter-executable communication and should therefore be considered during design time of an executable.
Comment	–
Example	–
Reference	–

4.114 Executable Entity Cluster

Definition	A set of Executable Entities (EEs) and a reference to a set of Execution Order Constraints (EOCs) between these EEs. The Executable Entity Cluster is formed for the purpose of Mapping EEs to Logical Execution Time (LET) intervals and to tasks. Several EECs may be mapped to the same LET interval.
Initiator	Software and Architecture

▽

△

Further Explanations	An Executable Entity Cluster (EEC) groups EEs from any Software Component. EEs with different triggers/periods can be part of the same EEC, if the triggers/periods are harmonic (i.e. that all periods are integer multiples of the smallest period). The EEC can reference a `Logical Execution Time` (LET) interval specification. See Figure 4.2 for an example.
Comment	The set of EOCs is cycle-free.
Example	–
Reference	–

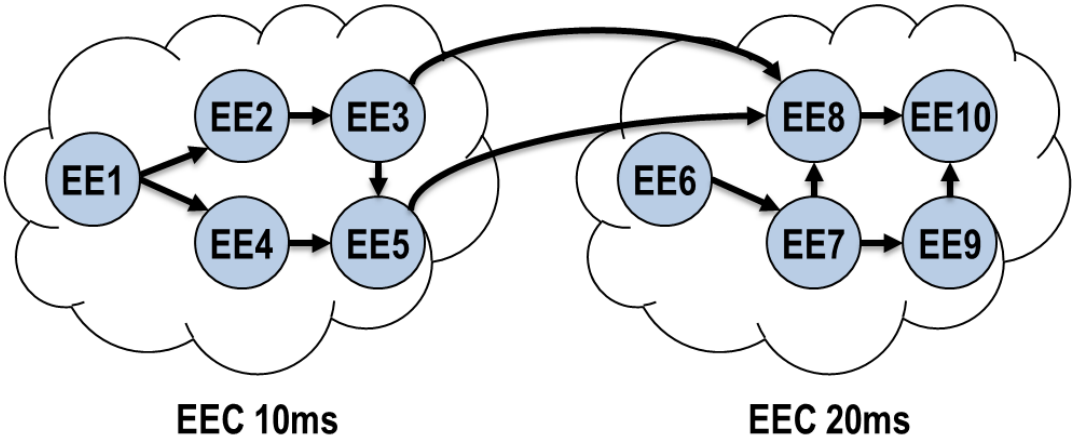


Figure 4.2: Executable Entity Cluster Example

4.115 Execution Order Constraint

Definition	Defines the execution order between instances of Executable Entities (EEs) within the same LET interval.
Initiator	Software and `Architecture`
Further Explanations	Example in Figure 4.3 shows an EE 2 that is marked as successor/directSuccessor of an EE 1, if the execution order of the instances of the respective Executable Entities is 12 (2 runs after 1) within the LET interval.
Comment	–
Example	–
Reference	[28]

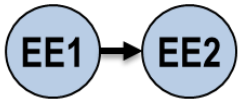


Figure 4.3: Example of an Execution Order Constraint

4.116 Execution Time

Definition	The time during which a program is actually executing, or more precisely during which a certain thread of execution is active.
Initiator	Software and Architecture
Further Explanations	The execution time of software is the time during which the CPU is executing its instructions. The time the CPU spends on Task switches or on the execution of other pieces of software is not considered here. See also: Response Time, Worst Case Execution Time, Worst Case Response Time.
Comment	–
Example	–
Reference	–

4.117 Exit Point

Definition	A point in a Software Component where an execution entity of the SW-C ends.
Initiator	Software and Architecture
Further Explanations	–
Comment	–
Example	Return point.
Reference	–

4.118 External Port

Definition	External Ports are ports of an automotive Ethernet switch used to communicate over an Ethernet physical connection with other Electronic Control Units (ECUs) (e.g. 100BASE-TX, 100BASE-T1).
Initiator	Communication
Further Explanations	–
Comment	–
Example	–
Reference	–

4.119 Fail-operational

Definition	Property of a system or Functional Unit. Describes the ability of a system or Functional Unit to continue normal operation at its output interfaces despite the presence of hardware or software faults.
Initiator	Safety

▽

△

Further Explanations	–
Comment	1. Typically, a fail-operational system or `Functional Unit` has no `Safe State`. 2. `Safety` means are not regarded as a part of the normal functionality respectively operation.
Example	Braking system
Reference	–

4.120 Fail-safe

Definition	Property of a system or `Functional Unit`. In case of a fault the system or `Functional Unit` transits to a `Safe State`.
Initiator	Safety
Further Explanations	Fail safe systems needs to have a `Safe State`. Note: not all the systems have a safe state.
Comment	–
Example	–
Reference	See also note of [7], ID 1.137

4.121 Fail-silent

Definition	Fail-silent is a property of a system in which no output is produced in the presence of a fault. In automotive domain, fail-silent systems are usually only used if the next hierarchical system level provides a safe-state.
Initiator	Safety
Further Explanations	Fail-silent is a special case of the fail-safe property.
Comment	–
Example	The fail-silent property can be used to avoid that "babbling idiots" disturb the overall communication.
Reference	–

4.122 Failure Mode

Definition	See ISO-26262 ([7]), ID 1.40
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	See [7], ID 1.40

4.123 Failure

Definition	See ISO-26262 ([7]), ID 1.39
Initiator	Safety
Further Explanations	Termination is a reduction in, or loss of, ability of an element or an item to perform a `Function` as required. There is a difference between "to perform a function as required" (stronger definition, use-oriented) and "to perform a function as specified", so a `Failure` can result from an incorrect specification.
Comment	–
Example	–
Reference	See [7], ID 1.39

4.124 Failure Rate

Definition	See ISO-26262 ([7]), ID 1.41
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	See [7], ID 1.41

4.125 Fault

Definition	See ISO-26262 ([7]), ID 1.42
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	See [7], ID 1.42

4.126 Fault Detection

Definition	The action of monitoring errors and setting fault states to specific values is called fault detection.
Initiator	Software and Architecture
Further Explanations	The different states are called "not detected"/ "present"/ "intermittent or maturing"/... The names of the fault states are following the ISO/SAE norms; however there is a coordination step in between the states of the DTCs (Diagnostic Trouble Code see definition in ISO 15765/ ISO14229) and the states of the faults. The SW-C's Fault Detection is executed decentralized, e.g. each Software Component (SW-C) sets the state of a fault according to the defined fault qualification (SW-C Template). Therefore the Fault Detection is implemented in the SW-C (SW-C could be either Application Software Component or Basic Software Component). There are exceptions; these will be pointed out individually for each fault. The SW-C's developer will define the conditions (=fault qualification), when these conditions are fulfilled the SW-C notifies a fault to the Diagnostic Memory Management.
Comment	–
Example	–
Reference	[29], [30], [11]

4.127 Fault Reaction

Definition	In case of a Failure of a Software Component (SW-C) there is a specific action to be carried out. This action is called "Fault Reaction".
Initiator	Software and Architecture
Further Explanations	–
Comment	–
Example	Fault Reactions can be implemented decentralized in the SW-C. There might also be the need of coordinating the fault reactions since there are reactions excluding each other. This will be done by a central fault reaction manager.
Reference	–

4.128 Fault Reaction Time

Definition	See ISO-26262 ([7]), ID 1.44
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	See [7], ID 1.44

4.129 Fault Tolerance

Definition	Ability to deliver the specified functionality in the presence of one or more specified faults.
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	–

4.130 Fault Tolerant Time Interval

Definition	See ISO-26262 ([7]), ID 1.45
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	See [7], ID 1.45

4.131 Feature

Definition	A Feature is a notable characteristic of a system.
Initiator	General
Further Explanations	AUTOSAR defines and interacts with many entities where the term Feature can be applied (e.g. the AUTOSAR standard itself, its implementations, `Electronic Control Units` (ECUs) built with AUTOSAR, AUTOSAR Authoring Tools, AUTOSAR Feature Model). For each usage the term Feature may be used in a refined way - which is then defined for that specific usage (e.g. [TPS_FeatureModelExchangeFormat]).
Comment	–
Example	CAN FD support, Automatic windshield wiper, Editing of the FlexRay schedule
Reference	[5], [31]

4.132 Firewall

Definition	Functional element that inspects and filters `Network` traffic based on firewall rules.
Initiator	Security
Further Explanations	–
Comment	–
Example	–
Reference	–

4.133 Flag

Definition	A piece of data that can take on one of two values indicating whether a logical condition is true or false.
Initiator	Software and Architecture
Further Explanations	–
Comment	–
Example	Notification flag
Reference	–

4.134 FlexRay

Definition	Automotive time-triggered and fault-tolerant Network communication protocol that is standardized in the ISO 17458 ([32]) and provides options for deterministic data that arrives in a predictable time frame as well as dynamic Event-driven data.
Initiator	Communication
Further Explanations	FlexRay provides a communication cycle with a pre-defined space for static and dynamic data.
Comment	–
Example	–
Reference	For FlexRay specifications please see ISO 17458 ([32]). For more details about FlexRay in Classic AUTOSAR please see [33], [34], [35]

4.135 Foundation

Definition	Foundation contains the generic artifacts that are common for AUTOSAR Adaptive Platform and AUTOSAR Classic Platform to ensure compatibility between: • Classic- and AUTOSAR Adaptive Platform • Non-AUTOSAR platforms to AUTOSAR platforms In AUTOSAR the particular realization of those generic artifacts is described in AUTOSAR Adaptive Platform and Classic Platform and may differ.
Initiator	General
Further Explanations	The role of the Foundation in AUTOSAR is two fold: 1) To group those Artifacts which are agnostic of, or outside of, the immediate scope of the Classic and AUTOSAR Adaptive Platform. Example: Protocol specification (PRS), AUTOSAR Abstract Platform or Non-AUTOSAR platform artifacts. 2) To provide a repository for those artifacts, which have content applicable to both Classic and AUTOSAR Adaptive Platform. Example: Requirement Specification (RS), Model (MMOD or MOD), Template Specification (TPS), Adaptive Software Specification (ASWS) artifacts. This depth of this commonality between Classic and AUTOSAR Adaptive Platform is typically detailed further in the respective artifacts.
Comment	–
Example	

▽

△

Reference	–

4.136 Frame

Definition	`Data` unit according to the data link protocol specifying the arrangement and meaning of bits or bit fields in the sequence of transfer across the transfer medium.
Initiator	Communication
Further Explanations	–
Comment	–
Example	A CAN frame consists of up to 8 bytes of payload `Data` and additional protocol specific bits / bit fields (e.g. CAN-Identifier).
Reference	[36]

4.137 Frame PDU

Definition	A PDU that fits into 1 frame instance e.g. it does not need to be fragmented across more than 1 frame for transmission over a `Network`.
Initiator	Communication
Further Explanations	–
Comment	–
Example	–
Reference	–

4.138 Freedom from Interference

Definition	See ISO-26262 ([7]), ID 1.49
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	See [7], ID 1.49

4.139 Freshness

Definition	Data Freshness implies that the data is recent and it ensures that replayed messages in a replay attack are detected.
Initiator	Security
Further Explanations	–
Comment	–
Example	–
Reference	–

4.140 Function

Definition	1. A `Task`, action or activity that must be accomplished to achieve a desired outcome. 2. A part of programming code that is invoked by other parts of the program to fulfill a desired purpose. 3. In mathematics, a function is an association between two sets of values in which each element of one set has one assigned element in the other set so that any element selected becomes the independent variable and its associated element is the dependent variable.
Initiator	Software and `Architecture`
Further Explanations	–
Comment	Due to the different meanings in texts using the term application the appropriate meaning should be explained in detail or referenced.
Example	1. C-Code Function 2. Y=f(x)
Reference	[5]

4.141 Functional Cluster

Definition	A Functional Cluster is a high level logical grouping of functionality.
Initiator	Software and `Architecture`
Further Explanations	Note 1: Functional Clusters implement one or more `Features` or parts of features. Note 2: Functional Clusters are represented in our logical architecture.
Comment	–
Example	–
Reference	–

4.142 Functional Network

Definition	A logical structure of interconnections between defined functional parts of Features.
Initiator	Software and Architecture
Further Explanations	–
Comment	–
Example	–
Reference	–

4.143 Functional Safety Concept

Definition	See ISO-26262 ([7]), ID 1.52
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	See [7], ID 1.52

4.144 Functional Safety Requirement

Definition	See ISO-26262 ([7]), ID 1.53
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	See [7], ID 1.53

4.145 Functional Unit

Definition	An entity of software or hardware, or both, capable of accomplishing a specified purpose.
Initiator	Software and Architecture
Further Explanations	–
Comment	–
Example	ECU, Software Component, ...
Reference	[20]

4.146 Functionality

Definition	Functionality comprises User-visible and User-non-visible functional aspects of a `System`.
Initiator	Software and `Architecture`
Further Explanations	–
Comment	
Example	Functionality of a communication system is a user-non-visible aspect.
Reference	–

4.147 Gateway

Definition	A gateway is `Functionality` within a `Classic Platform` ECU that performs a `Frame` or signal `Mapping Function` between two communication systems. Communication system in this context means e.g. a CAN system or one channel of a FlexRay system.
Initiator	Communication
Further Explanations	–
Comment	–
Example	–
Reference	`Gateway ECU`

4.148 Gateway ECU

Definition	A Gateway ECU is an `Electronic Control Unit` that is connected to two or more communication channels, and performs `Gateway Functionality`.
Initiator	Communication
Further Explanations	–
Comment	–
Example	–
Reference	`Gateway`

4.149 Graceful Degradation

Definition	Graceful Degradation: The system continues to operate in the presence of errors, accepting a partial degradation of `Functionality` or `Performance` during `Recovery` or repair. Found in the literature also as "fail soft".
Initiator	Safety
Further Explanations	–
Comment	`Safety` means are not regarded as a part of the normal `Functionality` respectively operation. Also known as: Fail-reduced, Fail-soft

▽

△

Example	"Limp home" `Functionality` for `Electronic Control Unit` (ECU) (reduce torque to assure an arrival at home or service station)
Reference	See also: [7]: 3.181 - warning and degradation strategy.

4.150 Hardware Abstraction Layer

Definition	A hardware abstraction layer is a software layer that serves as an abstraction layer between the physical hardware and its software. It allows to access the hardware resources through programming interfaces.
Initiator	–
Further Explanations	In AUTOSAR `Classic Platform` the Hardware Abstraction Layer is realized by the `Microcontroller Abstraction Layer` and `ECU Abstraction Layer`.
Comment	–
Example	–
Reference	Layered Software Architecture

4.151 Hardware Connection

Definition	HW Connections are used to describe the connection of `Hardware Elements` among each other. It defines/characterizes the interrelationship among HW Elements (for abstract modeling). The `Hardware Ports` of HW Elements serve as connection points for this purpose.
Initiator	General
Further Explanations	In AUTOSAR are 2 kinds of HW Connections defined: • Assembly HW Connection • Delegation HW Connection
Comment	–
Example	–
Reference	[37]

4.152 Hardware Element

Definition	The HW Element is the main describing element of an `Hardware Connection`. It provides `Hardware Ports` for being interconnected among each others. A generic HW Element specifies definitions valid for all specific HW Elements.
Initiator	General
Further Explanations	A HW Element is the piece or a part of the piece to be described with the ECU `Resource` Template. It uses other elements as primitives: This means HW elements can be nested (through HW Containers, a hierarchical structure of HW Elements). At the lowest level a HW Element only uses primitives

▽

△

Comment	–
Example	–
Reference	[37]

4.153 Hardware Interrupt

Definition	Interrupt triggered by HW `Event`.
Initiator	Software and `Architecture`
Further Explanations	2 sorts of HW events • Processor-intern: events as for example division by zero, arithmetical overflow, non-implemented instruction • Processor-extern: events as for example response of peripheral device (e.g. PWM), Memory `Error`, Timer
Comment	–
Example	–
Reference	Translation/Adaptation from [38]

4.154 Hardware Port

Definition	The HW port exposes functionality to the exterior of the `Hardware Element`. HW elements can be connected via a `Hardware Connection`. It defines a connection Endpoint for the HW Element.
Initiator	Communication
Further Explanations	HW elements provide HW ports for being interconnected among each others. Each HW port has a name which is unique within the HW element it is located in.
Comment	–
Example	–
Reference	[37]

4.155 Health Indicator

Definition	Health Indicators (HIs) are an evaluation metric of current system `Performance` with regard to `Safety` requirement
Initiator	Safety
Further Explanations	Health Indicator format: <HI_ID, `Performance`, `Reliability`, SubsystemState>
Comment	–
Example	–
Reference	[39]

4.156 Hook

Definition	An intervention point within ECU software for the exchange of data.
Initiator	Runtime Environment
Further Explanations	–
Comment	–
Example	Hooks used to read Electronic Control Unit (ECU) variables and/ or overwrite ECU variables with values generated by Rapid Prototyping algorithm.
Reference	–

4.157 Host ECU

Definition	A Host ECU is a Electronic Control Unit (ECU) that controls one or more automotive Ethernet switches (e.g. switch on / off the Ethernet switch and its ports, read and write the Ethernet switch Configuration). For this purpose the Host ECU is connected to the Ethernet switch over a common control interface (e.g. SPI, MDIO). The Host ECU also take part in the Network communication. For this purpose the Host ECU is connected by a data interface (e.g. MII) to a specific Ethernet switch Port (host port). It transmits and receives Ethernet frames.
Initiator	Communication
Further Explanations	–
Comment	–
Example	–
Reference	–

4.158 Host Port

Definition	A host port is a port of an automotive Ethernet switch where the data interface (e.g. MII) of the `Host ECU` is connected to. The host port could either be an `Internal Port` or an `External Port`. The host port has a special role from the perspective of the software (see link accumulation and port groups).
Initiator	Communication
Further Explanations	–
Comment	–
Example	–
Reference	–

4.159 Hypervisor

Definition	Low-level software that provides and manages several virtual machines in one physical machine. Maybe an independent software or contained as an OS functionality.
Initiator	Execution Management
Further Explanations	Shared physical resources are either exclusively assigned to single virtual machine, or accessed through virtual device which is managed by Hypervisor. Various hardware and software mechanisms can support the efficient implementation of virtual devices.
Comment	–
Example	–
Reference	–

4.160 Identity and Access Management (IAM)

Definition	IAM is about managing access rights of `Adaptive Applications` to interfaces of the `Adaptive Platform Foundation` and `Services`.
Initiator	Security
Further Explanations	–
Comment	–
Example	–
Reference	–

4.161 Identity Information

Definition	The access control is decided / enforced upon the identity information which represents properties of the Adaptive Applications.
Initiator	Security
Further Explanations	–

▽

△

Comment	–
Example	An example for identity information are Capabilities.
Reference	–

4.162 Implementation Conformance Class 1 (ICC1)

Definition	An ICC1 cluster offers a `Software Component Interface` and/or `Network Interface`. The Software Component Interface and Network Interface of an ICC1 cluster provide the functional behavior as specified in the AUTOSAR specifications on ICC3 level.
Initiator	Software and `Architecture`
Further Explanations	In an ICC1 cluster the basic software is regarded as a black box. It allows legacy platforms to migrate to AUTOSAR: - to be integrated into an AUTOSAR `Network` - to support `Software Components` (SW-C). The features of an ICC1 cluster can be a subset of the ICC3 features (e.g. FlexRay not used). This has to be indicated in the Implementation Conformance Statement (ICS). The `Functionality` represented in AUTOSAR by the RTE must be a part of any ICC1 cluster that provides an SW-CI. Typically an ICC1 cluster - is not structured into Basic Software (BSW) modules (ICC3) or BSW module clusters (ICC2) - has a proprietary internal structure and might consist of legacy/proprietary or highly optimized code. An ICC1 cluster shall provide an interface to the boot loader. ICC1 shall support SWC compatible configuration for SW-CI and AUTOSAR `Network` compatible `Configuration` for NWI.
Comment	Up to Release 4.0 the boot loader architecture is not standardized in AUTOSAR. Therefore the term ICC1 is not applicable to the boot loader architecture itself.
Example	–
Reference	–

4.163 Implementation Conformance Class 2 (ICC2)

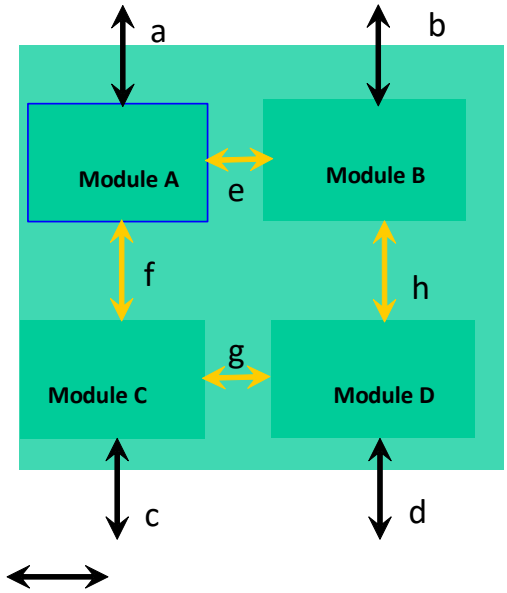
Definition	ICC2 clusters logically related ICC3 Basic Software (BSW) modules (2...N modules). The number of Cluster Features in an ICC2 cluster is a subset of the union of the number of features of the clustered ICC3 modules.
Initiator	Software and `Architecture`

▽

△

Further Explanations	Each ICC2 cluster presents a subset of the clustered ICC3 module's interfaces. ICC2 cluster provides the functional behavior as specified in the AUTOSAR specifications on ICC3 level. ICC2 cluster have a proprietary internal structure and might consist of proprietary or highly optimized code. ICC2 shall support AUTOSAR ECU `Configuration` description as an input for the Cluster Configuration It shall be possible to combine ICC2 Clusters and ICC3 Modules in a `Basic Software` (BSW) Architecture. Application interface Conformance (above RTE, `Software Component Interface`) and Bus Conformance (`Network Interface`) must be testable for a BSW which contain one or more ICC2 clusters.
Comment	–
Example	See Figure 4.4 for an example.
Reference	–

ICC2 Cluster Y ⊆ (ICC3 Module A **U** ICC3 Module B **U** ICC3 Module C **U** ICC3 Module D)



External Interfaces relevant for ICC2 clustering, subset of ICC3 interfaces to other BSW modules or clusters

Internal Interfaces not relevant for ICC2 clustering (can be proprietary).

Figure 4.4: ICC2 Cluster example

4.164 Implementation Conformance Class 3 (ICC3)

Definition	For ICC3 the AUTOSAR BSW consists of BSW modules as defined in the `Basic Software Module` List, including the RTE. ICC3 is the highest level of granularity.
Initiator	Software and `Architecture`
Further Explanations	All `Basic Software Modules` as defined in the CP SWS BSWGeneral including the RTE, must comply with the defined interfaces and `Functionality` as specified in their respective software specification document (SWS).
Comment	–
Example	–
Reference	–

4.165 Independence

Definition	See ISO-26262 ([7]), ID 1.61
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	See [7], ID 1.61

4.166 Independent Failures

Definition	See ISO-26262 ([7]), ID 1.62
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	See [7], ID 1.62

4.167 Indication

Definition	`Service` primitive defined in the ISO/OSI Reference Model ([18]). With the service primitive 'indication' a service provider informs a service user about the occurrence of either an internal `Event` or a service request issued by another service user.
Initiator	Software and `Architecture`
Further Explanations	An indication is e.g. a specific `Notification` generated by the underlying layer to inform about a Message Reception `Error`.
Comment	–

▽

△

Example	–
Reference	–

4.168 Integration

Definition	The progressive assembling of system components into the whole system.
Initiator	Software and Architecture
Further Explanations	–
Comment	–
Example	–
Reference	[40]

4.169 Integration Code

Definition	Code that the Integrator needs to add to an AUTOSAR System, to adapt non-standardized functionalities. Examples are Callouts of the ECU State Manager and Callbacks of various other Basic Software Modules.
Initiator	Software and Architecture
Further Explanations	–
Comment	–
Example	–
Reference	–

4.170 Integrity

Definition	The property that data or information have not been altered or destroyed in an unauthorized manner.
Initiator	Security
Further Explanations	–
Comment	–
Example	–
Reference	NIST SP 800-66 Rev. 1 under Integrity from 45 C.F.R., Sec. 164.304

4.171 Integrity Check Value

Definition	The result of a cryptographic `Function` used to ensure that unauthorized modifications of a message are detected.
Initiator	Global Time `Synchronization`
Further Explanations	–
Comment	See [41]
Example	The value might be the result of the keyed hash `Function` HMAC-SHA256.
Reference	[42]

4.172 Inter-Integrated Circuit I2C

Definition	I2C (Inter-Integrated Circuit) is a 2-wire serial data bus. It was developed by Philips Semiconductors (now NXP Semiconductors). I2C is a simply structured bus system and is widely used in automotive industry.
Initiator	Semiconductors
Further Explanations	–
Comment	–
Example	–
Reference	See [43]

4.173 Intermediate PNC Coordinator

Definition	An intermediate PNC (Partial `Network` Cluster) coordinator is located beneath a `Top-level PNC Coordinator` or a further intermediate PNC coordinator and above a `PNC Leaf Node` or a further intermediate PNC coordinator within a PNC network. The intermediate PNC coordinator forwards received PNC requests and own PNC requests to its `Top-level PNC Coordinator`. An intermediate PNC coordinator processes a received `PN shutdown message` immediately, forwards it to its subordinated ECUs (either a further intermediate PNC coordinator or a PNC leaf nodes), releases the indicated PNCs and resets the PN reset timer(s) for those PNCs.
Initiator	Partial Networking
Further Explanations	–
Comment	An intermediate PNC coordinator always immediately processes received PN shutdown messages
Example	–
Reference	–

4.174 Internal Port

Definition	Internal Ports are ports of an automotive Ethernet switch used for local communication (Host ECU or Cascaded Switch)
Initiator	Communication
Further Explanations	–
Comment	–
Example	–
Reference	–

4.175 Interrupt Frame

Definition	An interrupt frame is the code which handles the entering/leaving of (C written) Interrupt Service Routines. This code is Microcontroller specific and often written in assembly language. Interrupt frames are typically generated by the OS generation tool.
Initiator	Software and Architecture
Further Explanations	–
Comment	–
Example	–
Reference	

4.176 Interrupt Handler

Definition	In the case of a Category 2 interrupt, the ISR is synonymous with Interrupt Handler. In the case of Category 1 interrupt the Interrupt Handler is the Function called by the Hardware Interrupt vector. In both cases the Interrupt Handler is the user code that is normally a part of the Basic Software Module. So the Interrupt Handler is a user level piece of code.
Initiator	Software and Architecture
Further Explanations	–
Comment	–
Example	–
Reference	–

4.177 Interrupt Service Routine

Definition	A software routine called in case of an Interrupt.
Initiator	Software and Architecture
Further Explanations	ISRs have normally higher priority than normal processes and can only be suspended by another ISR which presents a higher priority than the one running.

▽

△

Comment	–
Example	–
Reference	[38]

4.178 Interrupt Vector Table

Definition	An interrupt vector table is a table of interrupt vectors that associates the `Interrupt Service Routine` with the corresponding interrupt request (typically by an array of jumps or similar mechanisms).
Initiator	Software and `Architecture`
Further Explanations	–
Comment	–
Example	–
Reference	–

4.179 Interrupt

Definition	`Event` that enforces the processor to change its state. This interruption causes the normal sequence of instructions to be stopped. Once an interrupt occurred, the running software entity is suspended and an `Interrupt Service Routine` (the one dedicated to this interrupt) is called.
Initiator	Software and `Architecture`
Further Explanations	Two sorts of interrupts exist: `Hardware Interrupts` and `Software Interrupts`.
Comment	–
Example	–
Reference	Translation/Adaptation from [38]

4.180 Intrusion Detection System

Definition	A system that is responsible for detecting, recording and reporting `Security` events.
Initiator	Security
Further Explanations	An AUTOSAR compliant IDS consist of `Security` sensors, Intrusion Detection System Manager (IdsM) and Intrusion Detection System Reporter (Idsr).
Comment	–
Example	–
Reference	–

4.181 Invalid Flag

Definition	For a signal in a Protocol Data Unit (PDU) an optional invalid flag can be added to the PDU payload layout. This Flag indicates the validity of other signals in the payload. In case the invalid flag of a signal is set to true in a PDU instance, the respective signal in the payload of the PDU instance does not contain a valid signal value.
Initiator	Communication
Further Explanations	This mechanism may be used in Gateways to indicate that parts of a PDU do not contain valid data.
Comment	–
Example	–
Reference	–

4.182 Invalid Value of Signal

Definition	For a signal in a Protocol Data Unit (PDU) an optional invalid value can be defined.
Initiator	Communication
Further Explanations	The invalid value is element of the signal value range that can be represented and transported by the signal. The invalid value is the value that is used in all situations where the receiver should be notified that the value in a signal is not valid.
Comment	–
Example	In case a PDU for a destination network of a Gateway is composed from two PDUs of two different source Networks, the Failure to receive one PDU can be indicated as invalid values in the respective signals of the transmitted PDU in the destination Network.
Reference	–

4.183 I-PDU

Definition	Interaction Layer Protocol Data Unit Collection of messages for transfer between nodes in a Network. At the sending node the Interaction Layer (IL) is responsible for packing messages into an I-PDU and then sending it to the Data Link Layer (DLL) for transmission. At the receiving node the DLL passes each I-PDU to the IL which then unpacks the messages sending their contents to the application.
Initiator	Communication
Further Explanations	–
Comment	–
Example	ISO 17356-4 (OSEK/VDX Communication) specifies an Interaction Layer and works on I-PDUs
Reference	[44]

4.184 Life Cycle

Definition	The course of development/evolutionary stages of a model element during its life time.
Initiator	Methodology and Templates
Further Explanations	A life cycle consists of a set of life cycle states. A life cycle state can be attached to an element in parallel to its version information. A typical life cycle is {valid, obsolete} and means that a valid element is up to date when first introduced but is substituted later by a new one and therefore gets the life cycle state "obsolete".
Comment	–
Example	–
Reference	–

4.185 LIN Bus Idle

Definition	Bus Idle is defined as no transition between recessive and dominant bit values on the LIN bus.
Initiator	Communication
Further Explanations	LIN slave nodes observe the LIN line for bus idle state. After a specific duration of bus idle (bus idle timeout), slave nodes enter bus sleep `Mode`.
Comment	Synonym "LIN Bus Inactivity"
Example	–
Reference	[45]

4.186 Link State Accumulation

Definition	The link state of a certain switch `Port` group is accumulated by embracing the link state of each port that is part of the port group. The rule how to embracing the link state is specified in the Ethernet Interface.
Initiator	Communication
Further Explanations	–
Comment	–
Example	–
Reference	–

4.187 Link Time Configuration

Definition	The configuration of the SW module is done until link time.
Initiator	Methodology and Templates
Further Explanations	The object code of the SW modules receives parts of its configuration from another object code file or it is defined by linker options.
Comment	–

▽

△

Example	Initial value of a signal.
Reference	–

4.188 Logical Execution Time (LET)

Definition	Is a fixed time interval. Input data is read at the beginning of this interval and output data is written at the end of the interval. Processing of the data is limited within the time interval.
Initiator	Software and `Architecture`
Further Explanations	The logical execution time (LET) is a real-time programming abstraction. Conceptually, a LET program execution has three steps: read the program input (in zero time), then execute, and finally write the output (in zero time) exactly when the time has elapsed since reading input. Hence, communication logically happens instantaneously at fixed points in time and the program executes within a time window (`Execution Time`) with a `Deadline` represented by the LET. When the execution of a program finishes before the `Deadline`, writing the output is (logically) delayed until the `Deadline`. This makes the `Deadline` a logical upper and lower bound for the execution. Thus, using a faster processor does not result in lower `Response Time`, but in decreased core utilization. Therefore, the behavior of the application is the same on any platform able to execute the program within the LET interval.
Comment	Practical implementation of the LET may use buffers to avoid write bursts at the end of a LET interval. Instead, only buffers are switched.
Example	See example in Figure 4.5.
Reference	Henzinger, T.A. et al: Giotto: A Time-Triggered Language for Embedded Programming. In: Proceedings of the IEEE, vol.91, 2003. Pp. 94-99

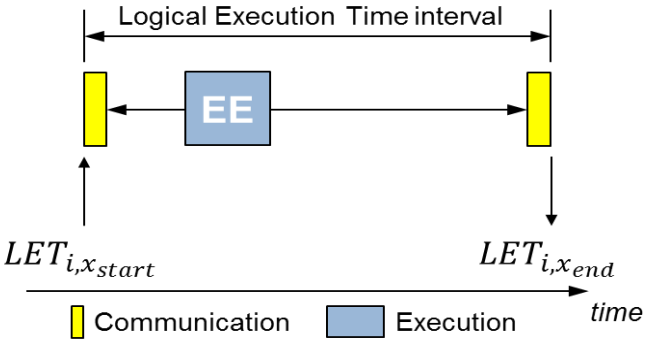


Figure 4.5: Example of Logical Execution Time

4.189 Log and Trace

Definition	Log and Trace provides interfaces for applications to forward logging and `Tracing` information onto the communication bus, the console, or to the file system.
Initiator	Communication
Further Explanations	–
Comment	–
Example	–
Reference	–

4.190 Logging

Definition	Logging is the activity of collecting information about arbitrary, not necessarily correlated events within a `System` during runtime.
Initiator	Software and `Architecture`
Further Explanations	The purpose of logging is to get information for finding and resolving defects of one or multiple programs running on a `System`. Therefore, it should also be usable in the field within a released product and the created logs are intended to be human readable.
Comment	Based on the currently active log level thresholds, logging may have a considerable timing and/or load impact on the `System`, which must be considered during further analysis. In contrast to measuring, logging does not focus on raw data values but on aggregated information within a `System` that is represented by events. In contrast to tracing, logging does not focus on recording internal program flow or state variables but focuses on collecting events that are explicitly added by a software developer on source code level. In contrast to diagnostics, logging does not make a statement on a system's health state but focuses on debug information that allows for detection and resolution of defects.
Example	Operation reporting, `Error` logging, printf output
Reference	–

4.191 Machine

Definition	A Machine is a single instantiation of an `AUTOSAR Adaptive Platform` stack, that may run on physical ECU-HW (without `Hypervisor`) or on virtual ECU-HW (with hypervisor).
Initiator	Methodology and Templates
Further Explanations	–
Comment	–
Example	–
Reference	–

4.192 Manifest

Definition	A Manifest represents a piece of AUTOSAR model description that is created to support the `Configuration` of an `AUTOSAR Adaptive Platform` product and which is uploaded to the AUTOSAR Adaptive Platform product, potentially in combination with other artifacts (like binary files) that contain `Executable` code to which the Manifest applies.
Initiator	Methodology and Templates
Further Explanations	Manifests are often used to denote a piece of `Configuration` content that ships along a given piece of software and is used to deploy the software in the field. Three examples of manifest are: - Execution Manifest - `Service` Instance Manifest - `Machine` Manifest
Comment	The Manifest may contain platform implementation dependent data, as well as generic data derived from Application System Description.
Example	–
Reference	–

4.193 Mappable Element

Definition	A mappable element is a part of a MCAL module which can be assigned to a partition via a reference parameter in the Base Software Module Description (BSWMD) of the module. Mappable elements allow the formal description on how MCAL modules are available respectively- distributed on partitions (and thus cores).
Initiator	General
Further Explanations	The type- and scope of the `Functionality` represented by a mappable element strongly depends on the MCAL module. A mappable element may closely represent the hardware (e.g. a channel) but can also represent subsets of HW units as well as groups of HW units.
Comment	Single partition MCAL driver define the complete driver as mappable element. Thus indicating that advanced multi-partition use-cases are not supported.
Example	Adc channel, CAN controller
Reference	–

4.194 Mapping

Definition	Mapping designates the distribution of elements in the logical view to elements in the physical view.
Initiator	Software and `Architecture`
Further Explanations	In general several entities may be allocated to one container but an entity may be allocated to only one container.
Comment	–
Example	a) Mapping of AUTOSAR Signals onto Frames (for inter-ECU communication). b) Mapping of SWC onto ECUs (Distribution of the SW-Components to the ECUs).
Reference	–

4.195 Master Switch

Definition	A Master Switch is an Ethernet switch where the host `Port` is located.
Initiator	Communication
Further Explanations	–
Comment	–
Example	–
Reference	–

4.196 MCAL Signal

Definition	The MCAL signal is the software representation of the `Conditioned Signal`. It is provided by the `Microcontroller Abstraction Layer` (MCAL) and is further processed by the ECU abstraction.
Initiator	General
Further Explanations	The processing unit is accessing the `Conditioned Signal` through some peripheral device that typically digitizes the `Conditioned Signal` into a software representation. The transformation from the `Conditioned Signal` to the MCAL Signal has to take the digitalization `Error` into account in order to provide information about the quality loss between the `Technical Signal` and the MCAL Signal.
Comment	–
Example	–
Reference	–

4.197 Meta-data

Definition	Meta-data is data about data
Initiator	Methodology and Templates
Further Explanations	Meta-data includes pertinent information about data, including information about the authorship, versioning, access-rights, timestamps etc..
Comment	–
Example	–
Reference	–

4.198 MetaDataItem

Definition	Defined item of Meta-data for a `Protocol Data Unit` e.g. a diagnostic address, a Message address, a CAN ID, or a J1939 node address.
Initiator	Communication

▽

△

Further Explanations	An ordered list of Meta Data Items defines the layout of PDU Meta Data. Each MetaDataItem has a fixed type and length, and enables the accessing modules to parse the PDU Meta Data, and to access items of the types that are relevant for the module.
Comment	Meta-data was revised with AUTOSAR 4.3.
Example	A PDU exchanged between CanIf and PduR can carry the CAN ID as a MetaDataItem to enable range routing of CAN messages.
Reference	–

4.199 Microcontroller

Definition	Hardware Element that integrates computing and communication resources as well as peripheral circuits in a single chip, including memories.
Initiator	General
Further Explanations	Microcontrollers are normally designed for small embedded systems and allow hardware designs with minimal amount of external parts. Microcontroller designs are normally optimized for silicon area and often support hard real-time and high-Integrity demands.
Comment	Classic AUTOSAR is intended for Microcontroller based embedded systems.
Example	–
Reference	–

4.200 Microcontroller Abstraction Layer

Definition	Software layer containing drivers to enable the access of onchip peripheral devices of a Microcontroller and offchip memory mapped peripheral devices by a defined Application Programming Interface. Task: make higher software layers independent of the Microcontroller.
Initiator	Software and Architecture
Further Explanations	The Microcontroller Abstraction Layer is the lowest software layer of the Basic Software. The Microcontroller Abstraction Layer consists of the following parts: • I/O Drivers • Communication Drivers • Memory Drivers • Crypto Drivers • Microcontroller Drivers Properties: • Implementation: μC dependent • Upper Interface (API): standardizable and μC independent
Comment	–
Example	Examples of drivers located in the Microcontroller Abstraction Layer are: • onchip eeprom driver • onchip adc driver • offchip flash driver
Reference	[26]

4.201 Middleware

Definition	A software layer that separates application SW from more technical layers like software stack, operating system or drivers. Middleware allows the communication of heterogenous applications with each other by providing an abstraction.
Initiator	Safety
Further Explanations	Middleware describes SW in a layered structure. It serves as a connection between at least two other software parts. In embedded software environments they usually consist of an upper layer, which is closer to application or user and a lower layer, which is closer to technical features or components. The upper layer passes commands or requests to the middleware whereas the middleware processes commands or requests and passes them to the lower layer. A Middleware can be seen as a proxy and allows a higher grade of abstraction, supports portability and maintainability of software.
Comment	The middleware approach is also used in non-embedded software systems.
Example	The complete AUTOSAR stack is middleware. It serves an application as upper layer and uses the MCAL (`Microcontroller Abstraction Layer`) software as lower layer (physical layer representation).
Reference	Refer to the description of the CP architecture in [26].

4.202 Minimum Send Interval

Definition	The Minimum Send Interval specifies the minimum amount of time that shall pass between two consecutive transmissions of an `Event`, field, trigger or method call, i.e., the Minimum Send Interval limits the maximum Tx frequency of `Service Interface` elements in the `Network`. The Minimum Send Interval can be configured individually for each event, field and method of a service interface, through the associated value of ServiceInterfaceDeploymentElement.minimumSendInterval, referenced by the ServiceEventDeployment, ServiceFieldDeployment and ServiceMethodDeployment.
Initiator	Communication
Further Explanations	For each field, event and method, ara::com shall monitor the frequency of transmission requests by the application. If a transmission is requested faster than allowed and specified by the configured ServiceInterfaceDeploymentElement.minimumSendInterval, ara::com shall discard the request and return with an error code. Minimum Send Interval monitoring shall be disabled for a given event, field or method, if the configured value ServiceInterfaceDeploymentElement.minimumSendInterval = 0. This shall also be the default value, to avoid unintended Tx limitation, unless the user explicitly configured a suitable value for its application.
Comment	–
Example	If a field has a configured Minimum Send Interval of 0.010 (10 ms), and the application Updates() the field every 50 ms, ara::com will accept the update and hand over the updated field to the lower transmission layers and will never discard a transmission request. However, if the application tries to update every 5 ms, ara::com will discard every second update and return with an error, because the configured Minimum Send Interval is higher than the time between request.
Reference	ara::com::ComErrc::kminimumSendIntervalViolationError ([SWS_CM_10432] of [46]) ServiceInterfaceDeploymentElement.minimumSendInterval in TPS Manifest ([47])

4.203 Mistake

Definition	Human Error
Initiator	Software and Architecture
Further Explanations	–
Comment	–
Example	–
Reference	[48]

4.204 Mode

Definition	A Mode is a certain set of states of the various state machines that are running in the vehicle that are relevant to a particular entity, e.g. a SWC, a BSW module, an application or a whole vehicle. In its lifetime, an entity changes between a set of mutually exclusive Modes. These changes are triggered by environmental data, e.g. signal reception, operation invocation.
Initiator	Runtime Environment
Further Explanations	–
Comment	–
Example	–
Reference	–

4.205 Multimedia Stream

Definition	A consistent sequence of digital data versus time which is suited as input for devices which transfer these data into a continuous visible or audible impression to humans. When transferred over a physical link, multimedia stream data typically are produced at the same rate (by the data source), as they are consumed (by the data sinks).
Initiator	General
Further Explanations	A multimedia stream usually follows a certain standard (e.g. MPEG-x). When transferred over a physical link, a multimedia stream needs a certain minimum bandwidth (in terms of bits/second) in order to allow continuous impressions. A multimedia stream in a car typically exists for several seconds (a warning signal, a navigation hint) up to several hours (a video film, a phone call, playing a radio program). Resources (e.g. bus system channels) needed by the stream have to be allocated continuously over this lifetime (this is a difference to e.g. file transfer, which may be split into several chunks of data). The source of a multimedia stream typically is a specialized device and/or software program (a tuner, a microphone, a text-to-speech engine, etc.). The same holds for the sinks (an audio amplifier or mixer, a voice recognition software, an MPEG decoder, etc.).
Comment	The term "visible or audible impression to humans" should not be taken too literally, because streams can also be used to transfer machine readable data (e.g. modem, encrypted signals). But it is this condition, which defines the standards and technology used in multimedia streams.
Example	Audio stream as output of or input to a telephone (mono, low bandwidth) Audio stream as output of a radio tuner (stereo, high bandwidth) Video stream as output of a television tuner An example for the physical implementation on a multimedia bus is the Firewire isochronous stream. see reference
Reference	[49]

4.206 Multiplexed PDU

Definition	A multiplexed PDU is a PDU with a configurable number of different payload layouts.
Initiator	Communication
Further Explanations	Each instance of a multiplexed PDU has a distinct layout. The set of possible layouts is statically defined. A selector signal defines which layout is used in a PDU instance. The selector signal must reside at the same position in all layouts. Each layout is identified by a unique selector value. The length of each instance of a multiplexed PDU is fixed.
Comment	–
Example	–
Reference	–

4.207 Network

Definition	Communication infrastructure between AUTOSAR ECUs/Machines.
Initiator	Methodology and Templates
Further Explanations	–
Comment	–
Example	–
Reference	–

4.208 Network Interface

Definition	A Network Interface is the sum of all interfaces offered by the `Basic Software` towards its connected `Network`.
Initiator	Communication
Further Explanations	The interface that the Basic Software shares via the communication lines with other systems that behave like AUTOSAR ECUs in order to - allow distributed Software Components) to exchange inter-ECU signals and to - operate the communication lines (the `Network`) is called Network Interface. A Network Interface (NWI) denotes the interface between the `Basic Software` and the physical network (OSI Layer 0) to which the ECU executing the Basic Software is connected to (e.g. CAN, LIN, FlexRay). The NWI therefore transports network data packets between the `Basic Software` and the physical network. The interfaces included within the term NWI are: - Logical interfaces, including • Network Management • Data Management • Data transmission/reception The interfaces excluded from the term NWI are: - The physical network interface (CAN, FlexRay etc).

△

	△ Note that, attention must be given to the physical form of the network, since it is not formally specified by AUTOSAR. The NWI provided by a given ECU supports the transfer of data to and from the ECU, and management of the Network. For the purposes of this definition, the Basic Software can be designed according to ICC1, ICC2 or ICC3.
Comment	The term has been introduced as a short-hand to aid in discussion of the conformance of the content of ICC1 / 2 and to define the backward compatibility between releases and revisions. However, since from the Network perspective, the clustering of the Basic Software is invisible, the Network Interface is applicable to all potential Basic Software conformance classes (ICC1, ICC2, ICC3) in the same way.
Example	–
Reference	Software Component Interface

4.209 NM Coordination Cluster

Definition	A discrete set of NM Channels on which shutdown is coordinated.
Initiator	Communication
Further Explanations	The NM Coordinator will keep all presently awake NM Channels of an NM Coordination Cluster awake until it is possible to Coordinate Network sleep on all the awake channels.
Comment	–
Example	–
Reference	[50]

4.210 NM Coordinator

Definition	A Functionality of the Generic NM Interface which allows coordination of Network sleep for multiple NM Channels.
Initiator	Communication
Further Explanations	Depending on Configuration, different level of synchronous Network sleep can be achieved. The NM Coordinator is using a generic coordination algorithm which, by means of individually configured Timeout and Synchronization indications can coordinate a synchronized shutdown of multiple NM Channels.
Comment	–
Example	–
Reference	[50]

4.211 Non-repudiation

Definition	Concept that is used in information Security that assures that the sender of information is provided with proof of delivery and the recipient is provided with proof of the sender's identity, so neither can later deny having processed the information.
Initiator	Security

▽

△

Further Explanations	–
Comment	–
Example	–
Reference	NIST SP 800-18 Rev. 1 under Non-repudiation from CNSSI 4009

4.212 Notification

Definition	Informing a software entity about a state change of a hardware and/or software entity which has occurred.
Initiator	Software and `Architecture`
Further Explanations	The informing about a state change can be done by an activation of a software part or by setting a `Flag`.
Comment	–
Example	–
Reference	–

4.213 Onboard Security Event

Definition	Analogous to ISO/IEC 27000:2018 ([51]) an onboard security event (SEv) is the identified occurrence of an onboard system, `Service` or `Network` state indicating a possible breach of information `Security` policy or `Failure` of controls, or a previously unknown situation that can be security relevant.
Initiator	Security
Further Explanations	–
Comment	–
Example	The failed `Verification` of a single secured PDU is typically an onboard security event.
Reference	

4.214 OS Application

Definition	A block of software including tasks, interrupts, hooks and user services that form a cohesive `Functional Unit`.
Initiator	Software and `Architecture`
x ***Further Explanations***	Trusted: An OS-Application that may be executed in privileged `Mode` and may have unrestricted access to the API and hardware resources. Only trusted applications can provide trusted functions. Non-trusted: An OS-Application that is executed in non-privileged mode has restricted access to the API and hardware resources.
Comment	The trusted / non-trusted attribute of an OS-Application is not related to `Automotive Safety Integrity Levels` (ASIL)/non-ASIL.
Example	–
Reference	[22]

4.215 OS Event

Definition	The event mechanism • is a means of `Synchronization` • is only provided for extended tasks • initiates state transitions of tasks to and from the waiting state.
Initiator	Software and `Architecture`
Further Explanations	–
Comment	–
Example	–
Reference	[4] (OSEK/VDX Operating System) [4] (OSEK/VDX Operating System)

4.216 Partitioning

Definition	See ISO-26262 ([7]), ID 1.85
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	See [7], ID 1.85

4.217 Protocol Control Information

Definition	Information which is needed to pass a `Service Data Unit` from one instance of a specific protocol layer to another instance. E.g. it contains source and target information.
Initiator	Communication
Further Explanations	The PCI is added by a protocol layer on the transmission side and is removed again on the receiving side.
Comment	–
Example	–
Reference	–

4.218 Protocol Data Unit (PDU)

Definition	The Protocol Data Unit (PDU) contains `Service Data Unit` and `Protocol Control Information`.
Initiator	Communication
Further Explanations	On the transmission side the PDU is passed from the upper layer to the lower layer, which interprets this PDU as its SDU (as shown in Figure 4.6).
Comment	–
Example	[44] (OSEK/VDX Communication)
Reference	[44] (OSEK/VDX Communication)

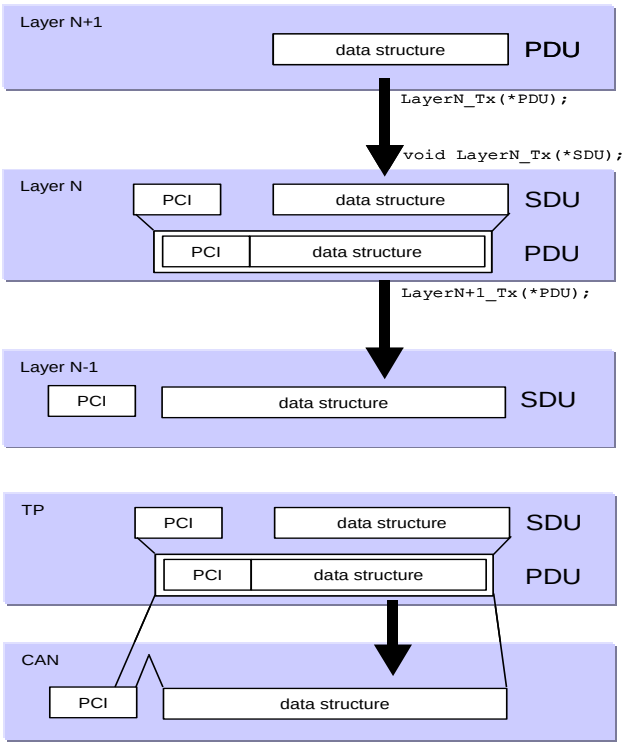


Figure 4.6: Explanation of Protocol Data Unit

4.219 PDU Meta-Data

Definition	Additional data of a `Protocol Data Unit`, which is not part of the payload.
Initiator	Communication
Further Explanations	Meta-data is placed alongside the PDU payload in a separate buffer. The layout of the Meta-data is determined by an ordered list of `MetaDataItems`.
Comment	Meta-data was introduced to transport parts of the CAN ID or addressing information alongside the data of a PDU.
Example	Diagnostics according to ISO 15765/14229, J1939 parameter group handling.
Reference	–

4.220 PDU Timeout

Definition	Maximum time between the receptions of two instances of one PDU is exceeded.
Initiator	Communication
Further Explanations	This `Timeout` indicates that the last reception of a PDU instance is too long in the past. As a consequence it can be concluded that the data in the last PDU instance is outdated.
Comment	–

▽

△

Example	–
Reference	–

4.221 Performance

Definition	Performance is a set of measurable characteristics (e.g. time, memory, resources usage, power consumption, etc.) which may be used to compare different system, SW element, algorithm, etc. implementations.
Initiator	Safety
Further Explanations	`Scalability` refers to the characteristic of a system to increase performance by adding additional resources. If the software performance requirements change (e.g more functions that impact the `Response Time`), scalability comes into play. Scalability is the ability of a system to continue to meet its response time or throughput objectives as the demand for the software functions increases.
Comment	–
Example	–
Reference	–

4.222 Peripheral Hardware

Definition	Hardware devices integrated in micro-controller architecture to interact with the environment.
Initiator	Software and `Architecture`
Further Explanations	–
Comment	–
Example	Memory, CAN-Controller, ADC, DIO, etc.
Reference	–

4.223 Personalization

Definition	User-specific and memorized adjustment of SW data or selection of functional alternatives.
Initiator	Software and `Architecture`
Further Explanations	–
Comment	–
Example	Seat parameters (position, activation status of drive-dynamic seat) can be stored in correlation to a user ID. For a given user ID the seat can be adjusted according to the stored position parameters and the drive-dynamic seat can be activated or deactivated.
Reference	–

4.224 Plausibility

Definition	Runtime Plausibility check is a method to verify during runtime if inputs for a computation/ algorithm or results of a computation/algorithm are reasonable against corresponding values of a simplified reference model.
Initiator	Safety
Further Explanations	Range checks are a subset of plausibility checks. The additional knowledge can be taken from various sources, e.g. the physical domain of the value or from a model representing the computation/algorithm more roughly and calculating in parallel to the actual computation/ algorithm.
Comment	–
Example	• Range Check: for determination that a value for a car velocity is plausible, the knowledge that a normal vehicle cannot be faster than 400km/h. • Plausibility: for determination that that a value for a car velocity is plausible, the history of the values can be used and the knowledge that a certain acceleration for a car cannot be exceeded. E.g. velocity was 10 km/h and increases within 1 Sec to 100 km/h.
Reference	–

4.225 PNC Leaf Node

Definition	A PNC leaf node is located beneath a `Top-level PNC Coordinator` or `Intermediate PNC Coordinator`. A PNC leaf node represents the lowest level within the hierarchy of a PN topology. A PNC leaf node does not coordinate PNC request across its local communication channels. It receives and transmits Nm frames with PN information, but it does not forward received PNC requests to other local communication channels. A PNC leaf node processes PN shutdown messages as usual Nm frames with PN information.
Initiator	Partial Networking
Further Explanations	–
Comment	A PNC leaf node represents the lowest level within a PN topology.
Example	–
Reference	–

4.226 PNC Request Bit Vector

Definition	A PNC Request Bit Vector consists of different PNC Request Bits. Each Bit represents a specific PNC. If the Bit is set to '1' then the PNC is requested by a node in the network. The receiving nodes in the network will bring the PNC to requested state. If the Bit is set to '0' in all NM messages on the Bus then the PNC is released because no node in the network requested the PNC.
Initiator	Partial Networking
Further Explanations	–
Comment	–
Example	–
Reference	–

4.227 PN shutdown message

Definition	A `Top-level PNC Coordinator` transmits the PN shutdown messages to indicate a PNC shutdown. A PN shutdown message is a NM message where the PNSR bit (resides in the control bit vector) and all PNC bits (resides in the PN info) which are indicated for a synchronized shutdown set to "1". A receiving `Intermediate PNC Coordinator` has to process the PN shutdown message immediately, reset the PN reset timer and forward the PN information as fast as possible to ensure a synchronized shutdown of the affected PNCs at nearly the same point in time. A `PNC Leaf Node` acts independently of the PNSR bit and resets the PN reset timer.
Initiator	Partial networking
Further Explanations	–
Comment	A PN shutdown message always indicates a shutdown of the indicated PNCs.
Example	–
Reference	–

4.228 Policy Decision Point (PDP)

Definition	The PDP represents the logic in which the access control decision is made. It determines if the application is allowed to perform the requested `Task`.
Initiator	Security
Further Explanations	The PDP provides an `Access Control Decision`.
Comment	–
Example	–
Reference	–

4.229 Policy Enforcement Point (PEP)

Definition	The PEP represents the logic in which the `Access Control Decision` is enforced. It communicates directly with the corresponding `Policy Decision Point` to receive the `Access Control Decision`.
Initiator	Security
Further Explanations	–
Comment	–
Example	–
Reference	–

4.230 Port

Definition	A port belongs to a `Software Component` and is the interaction point between the component and other components. The interaction between specific ports of specific components is modeled using `Connectors`. A port can either be a `Provide Port` or an `Require Port`.

▽

△

Initiator	Software and Architecture
Further Explanations	–
Comment	–
Example	–
Reference	For more information see [11]

4.231 Port Interface

Definition	A Port Interface characterizes the information provided or required by a Port of a Software Component.
Initiator	Software and Architecture
Further Explanations	A Port Interface is either a Client-Server Interface in case Client-Server Communication is chosen or a Sender-Receiver Interface in case Sender-Receiver Communication is used.
Comment	–
Example	–
Reference	For more information see [11]

4.232 Post-build Time Configuration

Definition	The Configuration of the SW module is possible after building the SW module.
Initiator	Methodology and Templates
Further Explanations	The SW may receive its Configuration file that can be downloaded to the Electronic Control Unit (ECU) separately, avoiding a re-compilation and re-build of the ECU SW modules. In order to make the post-build time re-configuration possible, the re-configurable elements shall be stored at a known position in the ECU storage area
Comment	–
Example	Identifiers of the CAN frames
Reference	–

4.233 Post-build Hooking

Definition	The insertion of Hooks to facilitate Rapid Prototyping support into a (complete) Electronic Control Unit (ECU) hex image.
Initiator	Runtime Environment
Further Explanations	–
Comment	–
Example	Detection of reads and/or writes of ECU variables by analysis of the instruction stream.
Reference	–

4.234 Pre-build Hooking

Definition	The insertion of Hooks to facilitate Rapid Prototyping support into software source prior to creating an Electronic Control Unit (ECU) hex image.
Initiator	Runtime Environment
Further Explanations	–
Comment	–
Example	–
Reference	–

4.235 Pre-Compile Time Configuration

Definition	The Configuration of the SW module is done at source code level and will be effective after compile time.
Initiator	Methodology and Templates
Further Explanations	The source code contains all the Electronic Control Unit (ECU) Configuration data and when compiled together, it produces the given SW.
Comment	–
Example	Preprocessor switch for enabling the development Error detection and reporting
Reference	–

4.236 Predictabiliy

Definition	Predictability is the degree to which a correct prediction or forecast of a system's state / behavior can be made either qualitatively or quantitatively.
Initiator	Safety
Further Explanations	Important type of predictability occurs in the design of systems that are subject to real-time requirements. A good overview of predictability criteria and how to achieve them can be found in
Comment	–
Example	–
Reference	John A. Stankovic, Krithi Ramamritham, What is predictability for real-time systems?, Journal of Real-Time Systems, Volume 2 Issue 4, Nov. 1990, 247-254

4.237 Pretended Networking

Definition	Method to reduce energy consumption in an existing active Network without changing network infrastructure.
Initiator	Communication
Further Explanations	–
Comment	–

▽

△

Example	–
Reference	–

4.238 Private Interface

Definition	A private interface is an interface within the Basic Software of AUTOSAR which is neither standardized nor defined within AUTOSAR.
Initiator	Software and Architecture
Further Explanations	The goal of the private interface is to enable a more efficient implementation of Basic Software Modules. Basic software modules sharing a private interface have to be distributed as one package. This package has to behave exactly the same as separate modules would. It must provide the same Standardized Interfaces to the rest of the Basic Software and/or RTE as separate modules would. It has to be configured exactly the same as separate modules would be configured.
Comment	Private interfaces contradict the goal of exchangeability of Standard Software modules and should be avoided.
Example	–
Reference	–

4.239 Probability of Failure

Definition	Probability of the occurrence of a Failure in a system or Functional Unit.
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	–

4.240 Procedure Call

Definition	A simple statement that provides the actual parameters for and invokes the execution of a procedure (software Function).
Initiator	Software and Architecture
Further Explanations	A Synchronous Communication mechanism can be implemented by a procedure call.
Comment	–
Example	–
Reference	[52]

4.241 Process

Definition	An Executable unit managed by an operating system Scheduler that has its own name space and resources (including memory) protected against use from other processes.
Initiator	Software and Architecture
Further Explanations	A process consists of n Task (n>=1)
Comment	–
Example	–
Reference	–

4.242 Processed Manifest

Definition	A Processed Manifest is a `Manifest` that is stored in the implementation specific format on the `AUTOSAR Adaptive Platform` product. Usually this is done in combination with other artifacts (like binary files) that contain `Executable` code to which the Manifest applies.
Initiator	Execution Management
Further Explanations	`Manifests` are often used to denote a piece of `Configuration` content that ships along a given piece of software and is used to deploy the software in the field. There are several kinds of manifest, this list includes but is not limited to: • Execution Manifest • `Machine` Manifest • `Service` Instance Manifest
Comment	The `Manifest` may contain platform implementation dependent data, as well as generic data derived from Application System Description.
Example	–
Reference	–

4.243 Profiling

Definition	Profiling refers to the process of evaluating `Performance`.
Initiator	Software and `Architecture`
Further Explanations	Profiling aggregates descriptive statistics of `Performance` measures of items running in an AUTOSAR system based on recordings, e.g. measurements or traces. The evaluation can be done online, i.e. during runtime, or offline.
Comment	Items subject to profiling could be functions, tasks, runnables, modules, buses etc.
Example	Statistics (min/max/average), worst case analysis
Reference	–

4.244 Proven In Use Argument

Definition	See ISO-26262 ([7]), ID 1.90
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	See [7], ID 1.90

4.245 Provide Port

Definition	Specific Port providing Data or providing a Service of a Server.
Initiator	Software and Architecture
Further Explanations	The Provide Port is sometimes abbreviated as PPort or P-Port.
Comment	–
Example	• Server Port • Server Port
Reference	–

4.246 Rapid Prototyping

Definition	The experimental incorporation of new Functionality.
Initiator	Runtime Environment
Further Explanations	Rapid Prototyping (RP) permits a user to quickly perform experiments to add new Functionality, or to replace/bypass existing functionality, without requiring an ECU image to be built.
Comment	–
Example	–
Reference	–

4.247 Rapid Prototyping Memory Interface

Definition	The memory access pattern necessary for Rapid Prototyping Tool.
Initiator	Runtime Environment
Further Explanations	The RP memory interface provides the well-defined memory access pattern required by RP tool to ensure consistent and complete access to bypass values.
Comment	–
Example	A mandated "write-read" cycle within RTE APIs provides the RP tool with an opportunity to bypass (i.e. substitute with value generated from an alternative algorithm) the written value before it is read and then subsequently used within the generated code.
Reference	–

4.248 Rapid Prototyping Tool

Definition	Software and/ or hardware tools to support Rapid Prototyping.
Initiator	Runtime Environment
Further Explanations	–
Comment	–
Example	Dedicated prototyping interfaces on ECUs accessed by PC-based software tools.

▽

△

Reference	–

4.249 Rate Conversion

Definition	Operation to change the timing between two transmissions of the same PduId on one physical Network.
Initiator	Communication
Further Explanations	–
Comment	–
Example	–
Reference	–

4.250 Raw Data Stream

Definition	A Raw Data Stream is a collection of data that is unprocessed, basically a series of bytes without any information on how to interpret it.
Initiator	Communication
Further Explanations	Raw Data Streams are used in communication links, mainly over ethernet reading streaming data from sensors. An API for Raw Data Streams is defined for AUTOSAR Adaptive Applications in SWS Communication Management.
Comment	–
Example	–
Reference	–

4.251 Recovery

Definition	Returning to intended Functionality after Fault Detection without violating the Safety goals.
Initiator	Safety

▽

△

Further Explanations	• Restart `Mode`: Restart Operation from the initial state of operation • Continue `Mode`: Restart Operation from the last known state of operation • Recovery by repetition: repeat until `Timeout` to cope i.e. random transmission errors. • Forward `Error` recovery: relies on continue from an erroneous state by making selective corrections to the system state. This includes making the controlled environment safe, which may be damaged because of the `Failure` • Backward `Error` recovery: relies on restoring the system to a previous `Safe State` and executing an alternative section of the program. This has the same `Functionality` but uses a different algorithm (c.f. N-Version Programming) • Recovery Point: The point to which a `Process` is restored is called a recovery point and the act of establishing it is termed check-pointing (saving appropriate system state) • Recovery testing is the forced `Failure` of the software in a variety of ways to verify that recovery is properly performed.
Comment	–
Example	–
Reference	–

4.252 Redundancy

Definition	Existence of means in addition to the means that would be sufficient for an element to perform a required `Function` or to represent information. `Hardware Element` redundancy includes replicated or additional hardware means added to the system to support `Fault Tolerance`. Software element redundancy includes the additional SW units and/or data used to support fault tolerance.
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	See [7], ID 1.94

4.253 Reentrancy

Definition	In AUTOSAR a `Function` is called reentrant if it can be interrupted in the middle of its execution and then safely called again ("re-entered") before its previous invocations complete execution. AUTOSAR differs between • (full) reentrancy • non reentrancy and • conditional reentrancy.
Initiator	Software and `Architecture`
Further Explanations	Reentrancy is always considered from the viewpoint of the caller. A `Function` which is conditional reentrant has to document the conditions for the reentrancy. Typical cases for conditional reentrancy are functions which are reentrant as long as a function parameter is different to (possible) ongoing calls.

▽

△

Comment	An implementation of a `Function` might be reentrant for one system but only conditional reentrant (or even non reentrant) for another one. It always depend how the reentrancy was realized (e.g. locks). As an example, just consider a function which uses `Interrupt` locks to realize full reentrancy on a single core `System`. If this implementation is used in a multi core system its reentrancy will only be conditional reentrant for calls from the same core.
Example	–
Reference	–

4.254 Reliability

Definition	Probability of a `System` or `Functional Unit` to perform as expected under specified conditions within a time interval.
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	–

4.255 Relocatability

Definition	Capability of a software part being executed on different hardware environments without changing the code of the software part.
Initiator	Software and `Architecture`
Further Explanations	–
Comment	–
Example	–
Reference	–

4.256 Require Port

Definition	Specific `Port` requiring `Data` or requiring a `Service` of a `Server`.
Initiator	Software and `Architecture`
Further Explanations	The Require `Port` is sometimes abbreviated as RPort or R-Port.
Comment	–
Example	• `Client` Port • Receiver Port
Reference	–

4.257 Required Property

Definition	A required property or quality of a design entity (e.g. `Software Component` or `System`) is a property or quality which has to be fulfilled by the environment of this design entity.
Initiator	Software and `Architecture`
Further Explanations	A property or quality can be required by a stakeholder (e.g. customer) or another design entity.
Comment	–
Example	1) In order to meet its `Functionality`, a `Software Component` A requires a minimum temporal resolution of a signal (information on a required `Port`) which has to be fulfilled by SW component B. 2) SW component requires to be activated by the runtime environment every 100ms with a jitter of 10ms.
Reference	Compare term `Asserted Property`

4.258 Residual Error Rate

Definition	The ratio of the number of bits, unit elements, or blocks incorrectly received and undetected, to the total number of bits, unit elements, characters, or blocks sent.
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	–

4.259 Resource

Definition	A resource is a required but limited hardware entity of an `Electronic Control Unit`, which in general can be accessed concurrently, but not simultaneously, by multiple software entities.
Initiator	Software and `Architecture`
Further Explanations	–
Comment	The definition from [4] (OSEK/VDX Operating System) cannot be used, due to the specific usage in ISO 17356.
Example	CPU-load, interrupts (mechanism itself and the resulting CPU-load), memory, peripheral hardware, communication, ...
Reference	–

4.260 Resource-Management

Definition	Entity which controls the use of `Resources`.
Initiator	Software and `Architecture`
Further Explanations	The main `Functionality` of resource management is the control of simultaneous use of a single `Resource` by several entities, e.g. scheduling of requests, multiple access protection.
Comment	–
Example	OS-`Scheduler` (CPU-load management)
Reference	–

4.261 Response Time

Definition	Time between receiving a stimulus and delivering an appropriate response or reaction.
Initiator	Software and `Architecture`
Further Explanations	The response time describes the time between a stimulus like e.g. the state change of hardware or software entity and the expected reaction of the system (e.g. response, actuator activation). Synonym: reaction time See also: `Execution Time`, `Worst Case Execution Time` and `Worst Case Response Time`.
Comment	–
Example	–
Reference	–

4.262 Risk

Definition	See ISO-26262 ([7]), ID 1.99
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	See [7], ID 1.99

4.263 Robustness

Definition	Ability of a `System` or `Functional Unit` to perform as expected also under unexpected conditions.
Initiator	Safety
Further Explanations	–
Comment	–

▽

△

Example	–
Reference	–

4.264 RTE Event

Definition	An RTE Event encompasses all possible situations that can trigger execution of a Runnable Entity by the RTE. Thus they can address timing, data sending and receiving, invoking operations, call Server returning, Mode switching, or external events. RTE Events can either activate a runnable entity or wake-up a runnable entity at its waitpoints.
Initiator	Runtime Environment
Further Explanations	Note: 'Event' in this context is not necessarily synonymous with 'RTEEvent' as defined in the Virtual Functional Bus (VFB) specification. In particular, RTE Events that result from communication are handled by communication-triggered runnable entities.
Comment	Events can have a variety of sources including time.
Example	Scheduling of runnable entities from angular position, e.g. a crankshaft, that are used to trigger an Interrupt and hence an RTE Notification. A software component needs to perform a regular interval, e.g. flash an LED, reset a watchdog, etc.
Reference	–

4.265 Runnable Entity

Definition	A Runnable Entity is a part of an Atomic Software Component which can be executed and scheduled independently from the other Runnable Entities of this Atomic Software-Component. It is described by a sequence of instructions that can be started by the RTE. Each runnable entity is associated with exactly one (1) Entry Point.
Initiator	Software and Architecture
Further Explanations	A Runnable Entity contains at least two points for the Scheduler: 1 Entry Point and 1 Exit Point. Due to the reason that an Atomic Software Component is not dividable, all its Runnable Entities are executed on the same Electronic Control Unit (ECU).
Comment	In general a Task in the runtime system consists out of n Runnable Entities of m Atomic Software-Components.
Example	Server Function of a Software Component.
Reference	–

4.266 SAE J1939

Definition	SAE J1939 is a vehicle bus standard created by the SAE (Society of Automotive Engineers, a USA standards body) for car and heavy duty truck industries.
Initiator	Communication

▽

△

Further Explanations	The J1939 standard encompasses the following areas: - bus physics (J1939/11, J1939/15) - CAN message layout (J1939/21) - request/response and multi packet transport protocols (J1939/21) - Network management used to assign a unique address to each node (J1939/81) - diagnostics layer comparable to UDS in complexity (J1939/73) - standardized application signals and messages (J1939/71)
Comment	The J1939 standard is used by most truck manufacturers worldwide and is prescribed for OBD in some states of the USA. It is also used as a base for other standards for maritime (NMEA 2000), agricultural (ISO 11783), and military (MilCAN A) applications.
Example	–
Reference	http://www.sae.org/

4.267 Safe State

Definition	See ISO-26262 ([7]), ID 1.102
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	See [7], ID 1.102

4.268 Safety

Definition	See ISO-26262 ([7]), ID 1.103
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	See [7], ID 1.103

4.269 Safety Analysis

Definition	The objective of Safety Analysis is to examine the consequences of faults and failures on items and elements considering their functions, behaviour and design.
Initiator	Safety
Further Explanations	–

▽

△

Comment	–
Example	–
Reference	ISO 26262 - 9: Automotive Safety Integrity Levels (ASIL)-oriented and safety-oriented analyses (see also: Safety Analysis)

4.270 Safety Case

Definition	See ISO-26262 ([7]), ID 1.106
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	See [7], ID 1.106

4.271 Safety Goal

Definition	See ISO-26262 ([7]), ID 1.108
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	See I[7], ID 1.108

4.272 Safety Measure

	See ISO-26262 ([7]), ID 1.110
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	See [7], ID 1.110

4.273 Safety Mechanism

Definition	See ISO-26262 ([7]), ID 1.111
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	See [7], ID 1.111

4.274 Safety Protocol

Definition	A communication protocol defining the necessary mechanisms to ensure the integrity of transmitted data and to detect any communication related `Error`.
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	–

4.275 Sample Application

Definition	Defined system used for evaluation purposes.
Initiator	Software and `Architecture`
Further Explanations	The application may be simplified for better understanding within the evaluation phase.
Comment	–
Example	Diagnosis Application Exterior Light Management
Reference	–

4.276 Scalability

Definition	The degree to which assets can be adapted to specific target environments for various defined measures.
Initiator	Software and `Architecture`
Further Explanations	–
Comment	Target environment introduced compared to [5].
Example	–
Reference	[5]

4.277 Scheduler

Definition	The scheduler handles the scheduling of the `Tasks` / `Runnable Entitys` according to the priority and scheduling policy (pre-defined or configurable). It has the responsibility to decide during run-time when which task can run on on the CPU of the `Electronic Control Unit` (ECU).
Initiator	Software and `Architecture`
Further Explanations	There are many strategies (priority-based, time-triggered, round-robin, ...) a scheduler can use, depending of the selected and/or implemented algorithms
Comment	–
Example	–
Reference	See AUTOSAR Specification of the `Virtual Functional Bus` ([11]

4.278 Service Data Unit

Definition	Service Data Unit is the data passed by an upper layer, with the request to transmit the data. It is as well the data, which is extracted after reception by the lower layer and passed to the upper layer.
Initiator	Communication
Further Explanations	A SDU is part of a `Protocol Data Unit`.
Comment	–
Example	–
Reference	–

4.279 Security

Definition	Protection of data, software entities or resources from accidental or malicious acts.
Initiator	Software and `Architecture`
Further Explanations	–
Comment	Slightly adapted norm.
Example	–
Reference	[53]

4.280 Secure Channel

Definition	A secure channel is a communication channel between two parties. A secure channel shall at least provide `Integrity` and `Authenticity` of the communication. It shall provide means to authenticate at least one participant of the communication. It may provide means to mutually authenticate both participants of the communication. A secure channel may additionally provide the `Confidentiality` of the communication.
Initiator	Security

▽

△

Further Explanations	–
Comment	–
Example	A TLS or IPsec channel for secure SOME/IP communication between two Adaptive Platform (AP) instances.
Reference	–

4.281 Security Event

Definition	An `Event` which is related to the `Security` of the `Electronic Control Unit` (ECU) and should be reported for later analysis.
Initiator	Security
Further Explanations	–
Comment	–
Example	Failed negotiation of a shared secret, failed `Verification` of a download signature, successful update of a certificate
Reference	–

4.282 Sender-Receiver Communication

Definition	A communication pattern which offers asyncronous distribution of information where a sender communicates information to one or more receivers, or a receiver receives information from one or several senders.
Initiator	Software and `Architecture`
Further Explanations	The process of sending data does not block the sender and the sender usually gets no response from the receivers
Comment	Often used for data or `Event` distribution
Example	–
Reference	[11]

4.283 Sender-Receiver Interface

Definition	A sender-receiver interface is a special kind of a `Port Interface` for the case of `Sender-Receiver Communication`. The sender-receiver interface defines the data-elements which are sent by a sending component (which has a p-`Port` providing the sender-receiver interface) or received by a receiving component (which has an r-port requiring the sender-receiver interface).
Initiator	Software and `Architecture`
Further Explanations	–
Comment	A special kind of `Port Interface`
Example	–

▽

△

Reference	[11]

4.284 Sensor-Actuator SW-Component

Definition	`Software Component` dedicated to the control of a sensor or actuator.
Initiator	Software and `Architecture`
Further Explanations	There will be several Sensor/ Actuator SW-Cs in each ECU. In general there will be one Sensor/ Actuator SW-C for each sensor and one for each actuator (=> number of Sensor/Actuator SW-C = number of sensors + number of actuators).
Comment	–
Example	–
Reference	–

4.285 Server

Definition	Software entity which provides services for Clients
Initiator	Software and `Architecture`
Further Explanations	The Server and the clients using its `Service` might be located on one ECU or distributed on different calculation units (e.g. ECU).
Comment	Adapted from [16].
Example	–
Reference	[16]

4.286 Service

Definition	A service is a type of operation that has a published specification of interface and behavior, involving a contract between the provider of the capability and the potential clients.
Initiator	Software and `Architecture`
Further Explanations	–
Comment	–
Example	Diagnosis service, ...
Reference	[5]

4.287 Service Discovery

Definition	Generic `Functionality` provided by the Communication Management to Applications that allows Applications at runtime to find locally or remotely available `Service Instances` providing the requested `Service`.
Initiator	Communication
Further Explanations	Based on Application query, the Communication Management provides list of compatible `Service Instances`. Compatibility is defined by compatibility rules and may consider version or QoS attributes.
Comment	–
Example	–
Reference	–

4.288 Service Instance

Definition	The properties of a service instance are described by a specific `Service Interface`. A service instance has a unique identity.
Initiator	Communication
Further Explanations	It is accessible by other Applications by using a Service Requester Proxy at runtime and is typed by a specific `Service Interface` It is addressable within the vehicle `Network` by its Service Instance ID, an abstraction from of the physical location. Optionally, authentication data is associated with a Service Instance for authentication at runtime.
Comment	–
Example	–
Reference	–

4.289 Service Interface

Definition	A service interface is a special kind of `Port Interface` used in the Adaptive platform. It defines both data elements for `Event`-based communication and operations that are provided by the service provider and that can be used by the service requester.
Initiator	Communication
Further Explanations	–
Comment	–
Example	–
Reference	–

4.290 Service-Oriented Communication

Definition	Communication, for which communication partners are generally not defined during design time, but dynamically discovered and bound during runtime.
Initiator	Communication

▽

△

Further Explanations	Communication partners are generally not defined during design time, but dynamically discovered and bound during runtime. `Adaptive Applications` are therefore developed agnostic to the concrete context, assuming model of loosely-coupled components.
Comment	This is the standard communication paradigm for communication between AUTOSAR Adaptive Applications.
Example	–
Reference	–

4.291 Service Port

Definition	Special kind of a `Port` from a `Atomic Software Component` used to describe `Service` communication.
Initiator	Software and `Architecture`
Further Explanations	The interface of a Service Port has to be a `Standardized AUTOSAR Interface`. A Service Port does not need to be connected to another Port in the `Virtual Functional Bus` View.
Comment	If a `Service` is provided by the ECU where a specific `Atomic Software Component` is located the `VFB View` is sufficient. If a service is provided by another ECU the connection of the service call to the service has to be done explicitly during the `Mapping` step.
Example	Write data to non volatile memory.
Reference	–

4.292 Service Proxy

Definition	A facade that represents a specific `Service` on code level from the perspective of the service consumer providing methods for all functionalities offered by the represented service.
Initiator	Communication
Further Explanations	The service consumer side Application code interacts with this local facade, which then knows how to propagate these calls to the real service implementation and back. The Service Proxy is typically an instance of a service proxy class which itself is potentially generated from Service Interface according to standardized patterns and implemented by platform-specific Communication Management. The Service Proxy provides placeholder for Service Instance ID, which is set at runtime by requesting Application implementation using Service Discovery or statically based on Planned Dynamics.
Comment	–
Example	–
Reference	–

4.293 Service Skeleton

Definition	A representation of a specific `Service` on code level from the perspective of the service implementation, which provides functionalities according to the service definition and allows to connect the service implementation to the Communication Management transport layer, so that the service implementation can be contacted by service consumers.

▽

△

Initiator	Communication
Further Explanations	The Service Skeleton is typically an instance of a service skeleton class which itself is potentially generated from a `Service Interface` according to standardized patterns and implemented by platform-specific Communication Management. The skeleton provides placeholder for the Service Instance ID, which is set by platform implementation, e.g. based on Application Description at design time, or by the Vehicle `Software Configuration` Manager at setup.
Comment	–
Example	–
Reference	–

4.294 Services Layer

Definition	The Services Layer is the highest layer of the `Basic Software` which also applies for its relevance for the application software: while access to I/O signals is covered by the `Hardware Abstraction Layer`, the Services Layer offers Operating system services Vehicle `Network` communication and management services Memory services (NVRAM management) Diagnosis Services (including KWP2000 interface and `Error` Memory) `Electronic Control Unit` (ECU) state management Task: Provide basic services for application and `Basic Software Modules`
Initiator	Software and `Architecture`
Further Explanations	The Services Layer consists of the following parts: Communication Services Memory Services System Services
Comment	–
Example	`Network` Management, NVRAM Manager, ECU State Manager
Reference	[26]

4.295 ShortName-path

Definition	ShortName paths are composed of sequences of shortNames separated by '/'. They can be `relative` or `absolute`.
Initiator	WG-MT
Further Explanations	–
Comment	–
Example	autosar/example
Reference	[54] [TPS_GST_00460]

4.296 ShortName-path (relative)

Definition	ShortName-path relative to a base.
Initiator	WG-MT
Further Explanations	–
Comment	–
Example	autosar/example
Reference	[54] [TPS_GST_00170]

4.297 ShortName-path (absolute)

Definition	ShortName-path starting at a root element and prepended with '/'.
Initiator	WG-MT
Further Explanations	–
Comment	–
Example	/autosar/example
Reference	[54] [TPS_GST_00169]

4.298 Signal Service Translation

Definition	Signal Service Translation is the standardized way to map the definition of signal-based communication to Service-Oriented Communication, and vice versa.
Initiator	Manifests & Templates
Further Explanations	AUTOSAR Adaptive Platform restricts communication paradigm to Service-Oriented Communication, a major part of the vehicle however still uses signal-based communication means - therefore a translation of these two approaches has to be performed.
Comment	One goal of AUTOSAR is to support the development of a whole vehicle in a seamless way. The translation of communication paradigms is essential for a holistic design. The AUTOSAR methodology supports the translation activity on either a Classic Platform Gateway ECU or on an AUTOSAR Adaptive Platform Machine.
Example	–
Reference	–

4.299 Slave Switch

Definition	A Slave Switch is an Ethernet switch which is connected to a Master Switch by uplink ports
Initiator	Communication
Further Explanations	–
Comment	–
Example	–

▽

△

Reference	–

4.300 Software Cluster (Adaptive Platform)

Definition	An Adaptive Platform Software Cluster groups all AUTOSAR artifacts which are relevant for deployment on an Adaptive Platform Machine. The content of a Software Cluster is dependent on the category and "may" (but is not limited to) contain artifacts such as: • Adaptive Platform Applications (Executables) • Adaptive Platform Functional Clusters (Modules) • Manifests • Persistent Storage Databases • Platform libraries, e.g. Transformers, Cryptographics • Bootloader
Initiator	Diagnostics
Further Explanations	A Software Cluster is deployed on a Machine via UCM inside a `Software Package` - multiple `Software Packages` are encapsulated inside a Vehicle Package to be deployed on multiple `Machines` in a vehicle. In the context of diagnostics a Software Cluster might be individually addressable via its own set of diagnostic addresses.
Comment	A Software Cluster is used to partition Applications running on a `Machine` into individually updateable clusters.
Example	–
Reference	[55], [47]

4.301 Software Cluster (Classic Platform)

Definition	The purpose of a Software Cluster for CP is to group AUTOSAR artifacts that are relevant for the static deployment of a software part on an ECU (`Electronic Control Unit`).
Initiator	WG-A-CP
Further Explanations	Each Software Cluster is an independent Build Unit and the result of the cluster specific build processes are the Binary Objects.
Comment	The term 'cluster' (or Software Cluster) is also used in the context of ICC2. In this case, it refers to a group of `Basic Software Module`s.
Example	–
Reference	see [56]

4.302 Software Component

Definition	Software Components are architectural elements that provide and/or require interfaces and are connected to each other through the `Virtual Functional Bus` to fulfill architectural responsibilities.

▽

△

Initiator	Software and Architecture
Further Explanations	A Software Component (SWC) has a formal description defined by the software component template. SWCs may be atomic components, parameter components or Compositions. Also the software modules providing the Software Component Interface of a Basic Software are called Software Components.
Comment	–
Example	–
Reference	–

4.303 Software Component Interface

Definition	A Software Component Interface (SW-CI) is the sum of all interfaces offered by the Basic Software towards the Software Component.
Initiator	Software and Architecture
Further Explanations	A SW-CI denotes the interface between an SWC and the underlying Basic Software cluster including the RTE. The SW-CI therefore comprises all Application Programming Interface Functions and Callbacks that the Software Component requires from and provides to the Basic Software (generally by means of RTE mechanisms). It includes also the mechanisms allowing Software Components sharing the SW-CI to communicate with one another. For the purposes of this definition, the Basic Software clustered on an Electronic Control Unit (ECU) can be designed according to ICC1, 2 and 3.
Comment	The term has been introduced as a short-hand to aid in discussion of the conformance of the content of Basic Software clusters of conformance class ICC1 / 2 and to define the backward compatibility between releases and revisions. However, since from the Software Component perspective, the clustering of the Basic Software is invisible, the Component Interface is applicable to all potential Basic Software conformance classes (ICC1, ICC2, ICC3) in the same way.
Example	–
Reference	Network Interface (NWI)

4.304 Software Configuration

Definition	The arrangement of software elements in a SW system.
Initiator	Software and Architecture
Further Explanations	A software element is a clearly definable software part. A software configuration is a selection version of software modules, Software Components, parameters and generator configurations. Calibration and Variant Coding can be regarded as subset of Software Configuration.
Comment	–
Example	–
Reference	[5]

4.305 Software Interrupt

Definition	`Interrupt` triggered by SW `Event`.
Initiator	Software and `Architecture`
Further Explanations	SW events are for example calling an operating system service, starting a `Process` with higher priority.
Comment	–
Example	–
Reference	Translation/Adaptation from [38]

4.306 Software Package

Definition	Unit for deployment of software onto AUTOSAR Adaptive Platform instances containing zero or more `Executables` and the `Meta-data` to install and execute it on the Machine.
Initiator	Software and `Architecture`
Further Explanations	Typically, a software package contains one or more `Executables`, however it is permitted to have no Executables to enable update of `Configuration Meta-data`.
Comment	–
Example	–
Reference	Translation/Adaptation from [38]

4.307 Software Platform

Definition	Software environment on which application software is executed.
Initiator	–
Further Explanations	`AUTOSAR Adaptive Platform`, AUTOSAR Classic Platform
Comment	–
Example	–
Reference	–

4.308 Software Signal

Definition	A Software Signal is an asynchronous `Event` transmitted between one `Process` and another.
Initiator	Software and `Architecture`
Further Explanations	A SW Signal is the software implementation of an (control-) information. Additionally it may have attributes (e.g. `Freshness`, data type, ...). It is exchanged between `Software Components`.
Comment	–
Example	–
Reference	–

4.309 Software Unit

Definition	See ISO-26262 ([7]), ID 1.125
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	See [7], ID 1.125

4.310 Special Periphery Access

Definition	Special functions to standard peripheral devices or special peripherals.
Initiator	Software and `Architecture`
Further Explanations	Is only used when, because of technical issues, no standard periphery access can be used
Comment	–
Example	–
Reference	–

4.311 Standard Periphery Access

Definition	Standard functions to typical standard peripheral devices that are available on an ECU (most `Microcontroller` integrated) used in automotive embedded applications.
Initiator	Software and `Architecture`
Further Explanations	–
Comment	–
Example	Digital Input/Output, Analog/Digital Converter, Pulse Width (De)Modulator, EEPROM, FLASH, Capture Compare Unit, Watchdog Timer
Reference	–

4.312 Standard Software

Definition	Standard Software is software which provides schematic independent infrastructural functionalities on an ECU. It contains only `Standardized AUTOSAR Interfaces`, `Standardized Interfaces` and/or `Private Interfaces`.
Initiator	Software and `Architecture`
Further Explanations	–
Comment	–
Example	ISO 17356, MCAL, Services
Reference	[36]

4.313 Standardized AUTOSAR Blueprint

Definition	A Standardized AUTOSAR Blueprint is a Blueprint standardized within the AUTOSAR project. Its derived objects are considered as being standardized within the AUTOSAR project as well.
Initiator	AUTOSAR Application Interfaces
Further Explanations	Blueprints were introduced within the AUTOSAR projects to enable standardization of ports without standardizing the static view of the architecture (i.e. the software components providing or requesting the ports). Sometimes it is not possible to standardize all attributes of an AUTOSAR element because the values of some attributes are project specific. Nevertheless it enables better collaboration if some of the attributes are standardized. Additionally Blueprints enable adding descriptions and long names in different languages.
Comment	–
Example	–
Reference	[57]

4.314 Standardized AUTOSAR Interface

Definition	This is an AUTOSAR Interface which is standardized within the AUTOSAR project.
Initiator	Software and Architecture
Further Explanations	AUTOSAR Services interact with other components through a Standardized AUTOSAR Interface. AUTOSAR Interfaces can be derived from AUTOSAR Application Interfaces.
Comment	–
Example	–
Reference	–

4.315 Standardized Blueprint

Definition	A Blueprint is called a Standardized Blueprint if the derived objects are considered as being standardized as well. It also includes that additionally concrete standardized rules exist how to specify the blueprint as well as how to derive an object from that blueprint. This is typically not done for a specific blueprint but for all blueprints of the same class.
Initiator	AUTOSAR Application Interfaces
Further Explanations	–
Comment	–
Example	–
Reference	–

4.316 Standardized Interface

Definition	A software interface is called Standardized Interface if a concrete standardized API exists.
Initiator	Software and Architecture
Further Explanations	Modules in the Basic Software interact which each other through Standardized Interfaces.
Comment	–
Example	ISO 17356-4 ([44]): COM Interface
Reference	–

4.317 Static Configuration

Definition	A setup where the routing Configuration cannot be changed during normal operation of the Gateway.
Initiator	Communication
Further Explanations	Static configuration doesn't allow reconfiguration of the routing during normal operation e.g. during driving. Static configuration does not restrict the update of the configuration in specific maintenance operation modes (e.g. programming Mode).
Comment	–
Example	A software update may change a routing Configuration such that a Protocol Data Unit (PDU) is routed into two instead of one destination Networks.
Reference	–

4.318 Synchronization Points

Definition	Used to synchronize the execution of EEs of different tasks that execute within the same Logical Execution Time (LET) interval. A synchronization point realizes the execution order of EEs that belong to different Executable Entity Clusters (EECs).
Initiator	Software and Architecture
Further Explanations	This pattern is used when EEs on multiple cores are synchronized, e.g. to ensure Execution Order Constraints as shown for example in Figure 4.7. Two kinds of Synchronization are in focus here. The execution of Executable Entity Cluster may be synchronized by actively waiting at a barrier or by advancing the execution until to relative point in time after LET interval start.
Comment	Synchronization points can be derived directly from the EOCs and the Mapping of EEs to tasks. The synchronization can be implemented by time (advance) or as a barrier. A synchronization point between EECs that are mapped to the same Task can already be fulfilled by the EE positions in the task. It is important to known that direct access before the synchronization point does not lead to interference, i.e. no other access to the same memory location can take place. The use of advance requires confident knowledge of the EE's Worst Case Execution Time (WCET).
Example	Execution Order Constraints between Executable Entities imposed by a data dependency. A Runnable Entity C waiting for input from Runnable Entity A and B. A synchronization point before C guarantees that A and B have produced the input for C when it starts execution.
Reference	–

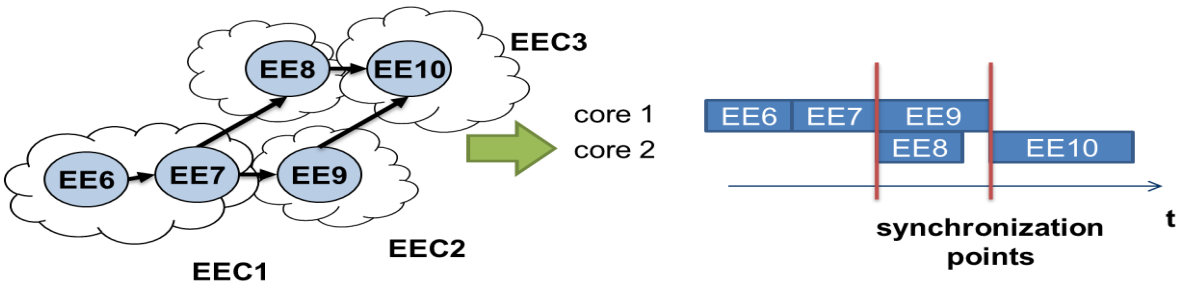


Figure 4.7: Example for Synchronization Points

4.319 Synchronization

Definition	To make two or more events or operations to occur at the same predefined moment in time.
Initiator	Communication
Further Explanations	–
Comment	–
Example	Two NM Channels can enter Bus Sleep `Mode` at the same time ("synchronized network sleep") or they can be ordered to go to sleep at the same time ("synchronized shutdown initiation").
Reference	[50]

4.320 Synchronous Communication

Definition	A communication is synchronous when the calling software entity is blocked until the called operation is evaluated. The calling software entity continues its operation by getting the result.
Initiator	Software and `Architecture`
Further Explanations	Synchronous communication between distributed functional units has to be implemented as Remote `Procedure Call` (RPC).
Comment	
Example	–
Reference	–

4.321 Synchronous Function

Definition	A `Function` is called synchronous if the described `Functionality` is guaranteed to be completed the moment the function returns to the caller.
Initiator	Software and `Architecture`

▽

△

Further Explanations	–
Comment	–
Example	–
Reference	–

4.322 System

Definition	Representation of software related parts in a vehicle and their means to communicate with each other.
Initiator	General
Further Explanations	–
Comment	–
Example	Representation of the `Network` topology of a vehicle with `Software Components` deployed on individual `Electronic Control Units` (ECUs) with a defined description of network communication between the ECUs.
Reference	[35], [47]

4.323 System Constraint

Definition	Boundary conditions that restrict the Design-Freedom of the (cars E/E-) System.
Initiator	Software and `Architecture`
Further Explanations	The design of `Electronic Control Unit` (ECU) Networks and the distribution of functionalities to ECUs are limited by several constrains. These constraints result mostly by the communication matrix and `Safety` requirements.
Comment	–
Example	An existing communication matrix that restricts the distribution of `Electrical Signals` to frames is a system constraint. Another system constraint is a `Safety` requirement that does not allow to map a specified `Software Component` to specific ECU.
Reference	–

4.324 System Health Monitor

Definition	System Health Monitor (SHM) analyzes the health of subsystems.
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	[39]

4.325 System Signal

Definition	The system signal represents the communication system's view of data exchanged between SW components which reside on different ECUs. The system signals allow to represent this communication in a flattened structure, with (at least) one system signal defined for eac`Data Element` sent or received by a `Software Component` instance. If data has to be sent over `Gateways`, there is still only one system signal representing this `Data`. The representation of the `Data` on the individual communication systems is done by the `Cluster Signals`.
Initiator	Communication
Further Explanations	–
Comment	–
Example	–
Reference	–

4.326 Systematic Fault

Definition	See ISO-26262 ([7]), ID 1.131
Initiator	Safety
Further Explanations	–
Comment	–
Example	–
Reference	See [7], ID 1.131

4.327 Task

Definition	A Task is the smallest schedulable unit managed by the OS. The OS decides when which task can run on the CPU of the ECU.
Initiator	Software and `Architecture`
Further Explanations	A `Runnable Entity` of a software component runs in the context of a task. Also the `Basic Software Modules` runs in the context of a task.
Comment	–
Example	–
Reference	[11]

4.328 Technical Signal

Definition	The technical signal is the physical value of an external `Event` coupled to an AUTOSAR system. Technical signals are represented in SI units (e.g. pressure in PA).
Initiator	General



△

Further Explanations	The term Technical Signal is used when we are referring to the "real world" signal that is under consideration. So typical Technical Signals are temperature, velocity, torque, force, electrical current and voltage, etc.
Comment	–
Example	–
Reference	–

4.329 Timed Communication

Definition	Implicit communication with logical publication at the end of a `Logical Execution Time` (LET) interval and logical read at the beginning of a LET interval. Access to the timed communication buffer is available for all EEs that reference the LET interval through EECs.
Initiator	Software and `Architecture`
Further Explanations	Data consistency is guaranteed between EECs which reference the same LET interval (regardless of their `Task` and core `Mapping`). Implicit communication crossing LET interval boundaries is redefined to timed communication. See also the example in Figure 4.8.
Comment	One OS `Task` can have several implicit communication buffers for several LET intervals. Implicit communication between EEs referencing the same LET interval remains unchanged.
Example	–
Reference	–

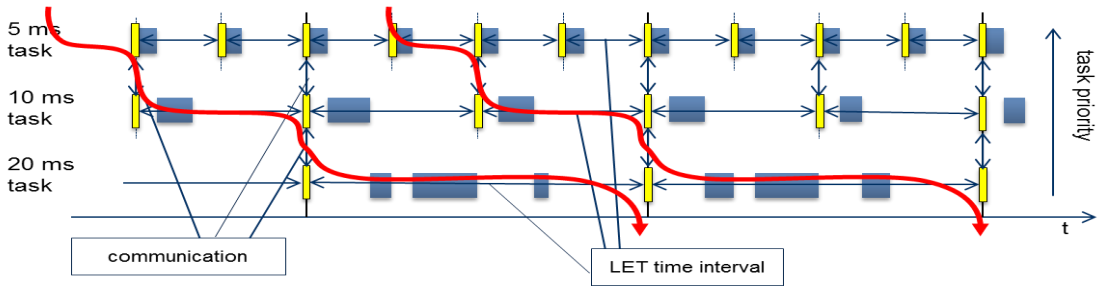


Figure 4.8: Example for Timed Communication

4.330 Timeout

Definition	`Notification` with respect to `Deadline` violation of an `Event` or `Task` (e.g. while working on/ with information: receiving, sending, processing, etc.).
Initiator	Software and `Architecture`
Further Explanations	–
Comment	–
Example	–

▽

△

Reference	–

4.331 Top-level PNC Coordinator

Definition	The top-level PNC (Partial Network Cluster) coordinator is the topmost coordinator in a PN topology, the single root of the directed acyclic graph forming the logical partial `Network` cluster for particular PNCs. The top-level PNC coordinator is responsible for the coordinated shutdown of all PNC members, because it is the only node which has the full overview of the states of all PNCs. Only the PNC top-level coordinator is allowed to send a PN shutdown messages. Note that for different PNCs it is possible to have different top-level PNC coordinators. For the same PNC only one top-level coordinator is supported.
Initiator	Partial Networking
Further Explanations	–
Comment	Partial networking is used to group nodes all over the `Network` topology into logical clusters. The clusters can be activated independently from each other. Each partial network cluster has a defined hierarchy, which spans from top level coordinator across multiple (0...N) levels of `Intermediate PNC Coordinators` to the `PNC Leaf Nodes`. PNC leaf nodes form the lowest level of the PNCs.
Example	–
Reference	–

4.332 Tracing

Definition	Tracing is the activity of recording a program's observable state and execution path within the `System` over a certain period of time.
Initiator	Software and `Architecture`
Further Explanations	Tracing collects events of selected types over time and stores the information in a so called "trace buffer", which may be located on-board or off-board. To enable timing measurement and `Performance` optimization, the events may be stored together with a timestamp. Tracing is intended to be mainly used during the development phase, with possibly added extra tracing software and/or hardware.
Comment	The recording may be done by software solutions, e.g. code instrumentation, hardware assisted solutions like CPU instruction flow tracing or Ethernet sniffers, or a combination of both. Depending on this, tracing may have a considerable impact on the system's timing, which must be considered when doing further analysis. In contrast to `Logging`, tracing does not target arbitrary but only correlating events of internal state transitions, variable content, and program flow.
Example	Program flow trace, `Scheduler Event` trace, Ethernet protocol trace, Stack usage trace.
Reference	–

4.333 Trusted Platform

Definition	An execution platform supporting a continuous chain of trust from boot through to application. The trust chain ensures that all execution is both authenticated (that all code executed is from the claimed source) and subjected to `Integrity Validation` (that prevents tampered code/`Data` from being executed)
Initiator	Software and `Architecture`
Further Explanations	–
Comment	–
Example	–
Reference	–

4.334 Uplink Port

Definition	An Uplink Port is a port of an automotive Ethernet switch which is connected to another Ethernet automotive switch (`Cascaded Switch`). An Uplink Port could either be an `Internal Port` or an `External Port`. One Uplink Port is connected to another `Uplink Port`. The Uplink Port has a special role from the perspective of the software.
Initiator	Communication
Further Explanations	–
Comment	–
Example	–
Reference	–

4.335 Use Case

Definition	A use case defines a list of actions defining the interactions between actors and a system to achieve a goal.
Initiator	General
Further Explanations	–
Comment	Use cases are used in the system analysis, for example, to identify and to clarify system requirements, and to define the behavior of a system.
Example	–
Reference	

4.336 Validation

Definition	Confirmation by examination and provisions of objective evidence that the particular requirements for a specific intended use are fulfilled.
Initiator	General

▽

△

Further Explanations	In design and development, validation concerns the process of examining a product to determine conformity with user needs. Validation is normally performed on the final product under defined operating conditions. It may be necessary in earlier stages. "Validated" is used to designate the corresponding status. Multiple validations may be carried out if there are different intended uses. [ISO 8402: 1994]
Comment	–
Example	–
Reference	[58]

4.337 Variability

Definition	Variability of a system is its quality to describe a set of variants. These variants are characterized by `Variant` specific property settings and / or selections.
Initiator	Methodology and Templates
Further Explanations	–
Comment	–
Example	As an example, such a system property selection manifests itself in a particular "receive `Port`" for a connection.
Reference	–

4.338 Variant

Definition	A system variant is a concrete realization of a system, so that all its properties have been set respectively selected. The software system has no `Variability` anymore with respect to the binding time.
Initiator	Methodology and Templates
Further Explanations	–
Comment	–
Example	–
Reference	–

4.339 Variant Coding

Definition	Adaptation of SW by selection of functional alternatives according to external requirements (e.g. country-dependent or legal restrictions).
Initiator	Software and `Architecture`

▽

△

Further Explanations	The major difference with `Calibration` is that this later doesn't aim to adapt the SW `Functionality` itself but only aims to adjust the SW to the HW/SW environment, e.g. the `Calibration` of engine control SW that is adjusted to the physical parameters of every engine. Variant Coding also includes vehicle-specific (not user-specific) SW adaptation due to end-customer wishes (e.g. deactivation of speed-dependent automatic locking). Variant Coding is always done after compile time. Used techniques to select variants include end-of-line programming and garage programming.
Comment	–
Example	Country related adaptation of MMI with respect to speed and/or temperature unit (km/h vs. mph, ℃ vs. F).
Reference	–

4.340 Variation Binding

Definition	A variant is the result of a variation binding `Process` that resolves the `Variability` of the system by assigning particular values/selections to all the system's properties.
Initiator	Methodology and Templates
Further Explanations	–
Comment	–
Example	–
Reference	–

4.341 Variation Binding Time

Definition	The variation binding time determines the step in the methodology at which the `Variability` given by a set of variable properties is resolved.
Initiator	Methodology and Templates
Further Explanations	–
Comment	–
Example	–
Reference	–

4.342 Variation Point

Definition	A variation point indicates that a property is subject to variation. Furthermore, it is associated with a condition and a binding time which define the system context for the selection / setting of a concrete `Variant`.
Initiator	Methodology and Templates
Further Explanations	–
Comment	–
Example	–

▽

△

Reference	–

4.343 Vehicle State Manager

Definition	An OEM specific application that controls vehicle level state and interacts with various entities in the vehicle system.
Initiator	Safety and UCM
Further Explanations	–
Comment	The OEMs may just have one such application and send the vehicle state on the communication bus or may have one Master Vehicle State Manager and `Client` Vehicle State Manager applications on each ECU.
Example	–
Reference	[55]

4.344 Vehicle Variant

Definition	A vehicle variant is a fully characterized product and a subset of the artifacts of the product line.
Initiator	–
Further Explanations	–
Comment	–
Example	–
Reference	–

4.345 Vendor ID

Definition	A vendor ID is a unique identification of the vendor of a `Software Component`. All `Basic Software Modules` conforming to the AUTOSAR Standard shall provide a readable vendor ID.
Initiator	General
Further Explanations	AUTOSAR Vendor IDs are used to determine vendors of Basic Software Modules before and during runtime. The mechanism is used to improve bug handling. AUTOSAR currently only provides Vendor IDs to members of the AUTOSAR partnership.
Comment	To apply for an AUTOSAR vendor ID the possible member has to send an E-Mail to request@autosar.org. Within the request name of the company, company address and contact person should be listed.
Example	Vendor ID for EEPROM driver is called: EEP_VENDOR_ID
Reference	SRS_BSW_00374

4.346 Verification

Definition	Confirmation by examination and provisions of objective evidence that specified requirements have been fulfilled.
Initiator	General
Further Explanations	In design and development, verification concerns the process of examining the result of a given activity to determine conformity with the stated requirement for that activity. "Verified" is used to designate the corresponding status. [ISO 8402: 1994]
Comment	–
Example	–
Reference	[58]

4.347 VFB View

Definition	The VFB View describes systems or subsystems in the car independently of these resources; in other words, independently of: • what kind of and how many `Electronic Control Units` (ECUs) are present in the car • on what ECUs the entities in the VFB-View run • how the ECUs are interconnected: what kind of `Network` technology (CAN, LIN, ...) and what kind of topology (presence of gateways) is used
Initiator	Software and `Architecture`
Further Explanations	In the VFB-View, the system or subsystem under consideration is a `Composition` which consists out of `Connectors` and Components.
Comment	–
Example	–
Reference	[11]

4.348 Virtual Functional Bus

Definition	The Virtual Functional Bus is an abstraction of the communication between `Atomic Software Components` and `AUTOSAR Services`. This abstraction is such that specification of the communication mechanisms is independent from the concrete technology chosen to realize the communication.
Initiator	Software and `Architecture`
Further Explanations	After compilation and linking of software for a dedicated `Electronic Control Unit` the Virtual Functional Bus interfaces are realized by the AUTOSAR Runtime Environment.
Comment	–
Example	–
Reference	–

4.349 Virtual Integration

Definition	The simulated, modeled and/or calculated (not real) combination of software entities forming a System.
Initiator	Software and Architecture
Further Explanations	By virtual integration several constraints and/or requirements are checked without the need of real hardware units, like needed CPU load, needed memory, completeness of interfaces, fulfillment of timing requirements etc.).
Comment	–
Example	–
Reference	–

4.350 Virtualization

Definition	Virtualization is a mechanism which hides the physical characteristics of a computing platform from the users, presenting instead another abstract computing platform. It can be used to fulfill functional Safety requirements like Availability, Partitioning, Resource conflict management, Recovery etc.
Initiator	Safety
Further Explanations	Different types of hardware virtualization include: • Full virtualization - almost complete simulation of the actual hardware to allow software, which typically consists of a guest operating system, to run unmodified. • Partial virtualization - some but not all of the target environment attributes are simulated. As a result, some guest programs may need modifications to run in such virtual environments. • Paravirtualization - a hardware environment is not simulated; however, the guest programs are executed in their own isolated domains, as if they are running on a separate System. Guest programs need to be specifically modified to run in this environment.
Comment	–
Example	–
Reference	–

4.351 Wake-up and Sleep on Dataline

Definition	Wake-up request or sleep request which is transmitted over a dataline of an Automotive Ethernet switched Network.
Initiator	Communication
Further Explanations	Wake-up on dataline is used for Ethernet switched Network and is defined for 100Base-T1. A wake-up on dataline without an established link between the communication partner is called a wake-up pulse (WUP). A wake-up on dataline with an established link between the communication partners is called a wake-up request (WUR). A sleep request is transmitted as a burst of continues LPS (low power signal)
Comment	–
Example	–
Reference	OPEN ALLIANCE Sleep/Wake-up specification for Automotive Ethernet

4.352 Worst Case Execution Time

Definition	Maximum possible time during which a program is actually executing
Initiator	Software and Architecture
Further Explanations	The worst case execution time of a piece of software is the maximum possible time during which the CPU is executing instructions which belong to this piece. The worst case execution time is often identified by analytical methods. It is required to determine if a schedule meets the overall timing requirements. Abbreviation: WCET See also: Response Time, Execution Time, Worst Case Response Time
Comment	This definition has been extended by WP COM.
Example	–
Reference	–

4.353 Worst Case Response Time

Definition	Maximum possible time between receiving a stimulus and delivering an appropriate response or reaction.
Initiator	Software and Architecture
Further Explanations	The worst case response time describes the maximum possible time between a stimulus like e.g. the state change of hardware or software entity and the expected reaction of the System (e.g. response, actuator activation). Typically: worst-case execution-time + infrastructure-overhead + scheduling-policy = worst-case reaction time Synonym: Worst Case Reaction Time (WCRT) See also: Response Time, Execution Time, Worst Case Execution Time
Comment	Worst case reaction time was renamed to worst case response time because response time is the more common terminology. This definition has been extended by WP COM.
Example	–
Reference	–

A Index of AUTOSAR entities

A.1 AUTOSAR Document categories

The categories of AUTOSAR documents are originally defined in [10].

Document category (abbr)	Document category (full)	General purpose	Detailed purpose
Software related			
RS	Requirement specification	Specification of requirements	Composed requirements intended to be decomposed into SWS/PRS requirements.
ASWS	Abstract software specification	Specification of platform software (abstract)	Platform independent description of SWS aspects.
SWS	Software specification	Specification of platform software (concrete)	Either: • Concretely describes the behavior, configuration and API of an AUTOSAR middleware (CP: BSW module, AP: Functional Cluster) • Describes generic language binding aspects independent of an AUTOSAR middleware
PRS	Protocol specification	Specification of protocol	Describes an AUTOSAR provided protocol. The protocol is typically implemented by an SWS or by a non-AUTOSAR software to inter-operate with an AUTOSAR platform.
SRC	Source	Source code header files	Source code for APIs specified in an SWS
Modeling related			
MMOD	Meta-model	n/a	Meta-description of a M3, M2, M1 model. Provided in two formats: • Graphical UML view of the M2 and M1 levels and SPEM view of the M3 level • Serialized XML schema (XSD) view of the M2 and M1 levels
MOD	Model	Platform configuration (M1 model)	Serialized ARXML view of a specific part of an AUTOSAR configuration. This is the agreed "exchange" format amongst vendors and is used directly by most AUTOSAR tools.
TPS	Template specification	Specification of platform configuration (M2 model)	Describes the methodology, structural modeling (meta-classes and their relations) which collectively build up the configuration of a specific part of an AUTOSAR.
General			
EXP	Explanation	Elaboration of specification (informative)	Informally elaborates content from other AUTOSAR specifications in a more illustrative exemplary approach. It does not use traceable items.
TR	Technical report	Elaboration of specification (normative)	Generic document describing arbitrary AUTOSAR related topics to provide additional information.

A.2 AUTOSAR Documents

The AUTOSAR documents are originally defined in [10].

Document standard (abbr)	Document category (abbr)	Document title	Document name (full)	Document name (abbr)
AP	EXP	Design guidelines for using parallel processing technologies on Adaptive Platform	ParallelProcessing Guidelines	
AP	EXP	Explanation of ARA Applications in Rust	ARARustApplications	
AP	EXP	Explanation of Adaptive Platform Design	PlatformDesign	
AP	EXP	Explanation of Adaptive Platform Software Architecture	SWArchitecture	
AP	EXP	Explanation of Automotive API	AutomotiveAPI	

▽

△

Document standard (abbr)	Document category (abbr)	Document title	Document name (full)	Document name (abbr)
AP	EXP	Explanation of IPsec Implementation Guidelines	IPsecImplementation Guidelines	
AP	EXP	Explanation of Identity and Access Management	IdentityAndAccess Management	
AP	EXP	Explanation of MACsec and MKA Protocols implementation and configuration guidelines	MACsec	
AP	EXP	Explanation of Safe API for hardware accelerators	SafeHardwareAcceleration API	
AP	EXP	Explanation of Sensor Interfaces	SensorInterfaces	
AP	EXP	Explanation of Service-Oriented Vehicle Diagnostics	SOVD	
AP	EXP	Explanation of ara::com API	ARAComAPI	
AP	EXP	Guidelines for using Adaptive Platform interfaces	InterfacesGuidelines	
AP	MOD	Collection of blueprints for AUTOSAR Adaptive Platform M1 models	GeneralBlueprints	
AP	MOD	Specification of Machine Configuration Parameters	MachineConfiguration Parameters	
AP	MOD	Specifications in ARXML format	SpecificationsARXML	
AP	RS	General Requirements specific to Adaptive Platform	General	AP
AP	RS	Requirements of State Management	StateManagement	SM
AP	RS	Requirements on Automated Driving Interfaces	AutomatedDriving Interfaces	ADI
AP	RS	Requirements on Automotive API Gateway	AutomotiveAPIGateway	AAG
AP	RS	Requirements on Execution Management	ExecutionManagement	EM
AP	RS	Requirements on HWTestManager	HWTestManager	HTM
AP	RS	Requirements on Operating System Interface	OperatingSystemInterface	OSI
AP	RS	Requirements on Platform Health Management	PlatformHealth Management	PHM
AP	RS	Requirements on Safe Hardware Acceleration	SafeHardwareAcceleration	SHWA
AP	RS	Requirements on Update and Configuration Management	UpdateAndConfiguration Management	UCM
AP	RS	Requirements on Vehicle Update and Configuration Management	VehicleUpdateAnd ConfigurationManagement	VUCM
AP	SRC	Source of Adaptive Platform ARAHeader Files	ARAHeaderFiles	
AP	SWS	General Specification of Adaptive Platform	General	GENERAL
AP	SWS	Specification of Adaptive Platform Core	Core	CORE
AP	SWS	Specification of Automotive API Gateway	AutomotiveAPIGateway	AAG
AP	SWS	Specification of Communication Management	Communication Management	CM
AP	SWS	Specification of Cryptography	Cryptography	CRYPT

▽

△

Document standard (abbr)	Document category (abbr)	Document title	Document name (full)	Document name (abbr)
AP	SWS	Specification of Demands and Constraints on Base Software	DemandsConstraintsBase SW	APBSW
AP	SWS	Specification of Diagnostics	Diagnostics	DM
AP	SWS	Specification of Execution Management	ExecutionManagement	EM
AP	SWS	Specification of Firewall for Adaptive Platform	Firewall	Fw
AP	SWS	Specification of Intrusion Detection System Manager for Adaptive Platform	IntrusionDetectionSystem Manager	AIDSM
AP	SWS	Specification of Language Binding for modeled AP data types	LanguageBindingFor ModeledAPdatatypes	LBAP
AP	SWS	Specification of Log and Trace	LogAndTrace	LOG
AP	SWS	Specification of Network Management	NetworkManagement	ANM
AP	SWS	Specification of Operating System Interface	OperatingSystemInterface	OSI
AP	SWS	Specification of Persistency	Persistency	PER
AP	SWS	Specification of Platform Health Management	PlatformHealth Management	PHM
AP	SWS	Specification of Platform Types for Adaptive Platform	PlatformTypes	APT
AP	SWS	Specification of Raw Data Stream	RawDataStream	RDS
AP	SWS	Specification of Remote Persistency	RemotePersistency	RPER
AP	SWS	Specification of Safe Hardware Acceleration	SafeHardwareAcceleration	SHWA
AP	SWS	Specification of Sensor Interfaces	SensorInterfaces	ADI
AP	SWS	Specification of State Management	StateManagement	SM
AP	SWS	Specification of Time Synchronization	TimeSynchronization	TS
AP	SWS	Specification of Update and Configuration Management	UpdateAndConfiguration Management	UCM
AP	SWS	Specification of Vehicle Update and Configuration Management	VehicleUpdateAnd ConfigurationManagement	VUCM
AP	TPS	Adaptive Platform Machine Configuration	MachineConfiguration	APMC
AP	TPS	Specification of Manifest	ManifestSpecification	MANI
AP	TR	AUTOSAR Adaptive Platform Specification Hashes	SpecificationHashes	
AP	TR	Adaptive Platform Change Documentation	ChangeDocumentation	
AP	TR	Adaptive Platform Release Overview	ReleaseOverview	
AP	TR	Integration of DDS Security	DDSSecurityIntegration	DDSS
AP	TR	Methodology for Adaptive Platform	Methodology	AMETH
AP	TR	Technical Report on Operating System Tracing Interface	OperatingSystemTracing Interface	OSTI
AP	TR	Technical Report on VSS Representation	VSSRepresentation	VSS
CP	EXP	Application Interfaces User Guide	AIUserGuide	AIUG
CP	EXP	Complex Driver design and integration guideline	CDDDesignAndIntegration Guideline	
CP	EXP	Description of the AUTOSAR standard errors	ErrorDescription	ED

▽

△

Document standard (abbr)	Document category (abbr)	Document title	Document name (full)	Document name (abbr)
CP	EXP	Explanation of Application Interface of AD/ADAS vehicle motion control	AIADASAndVMC	
CP	EXP	Explanation of Application Interfaces of Occupant and Pedestrian Safety Systems Domain	AIOccupantAndPedestrian Safety	AIOPS
CP	EXP	Explanation of Application Interfaces of the Body and Comfort Domain	AIBodyAndComfort	AIBC
CP	EXP	Explanation of Application Interfaces of the Chassis Domain	AIChassis	AICS
CP	EXP	Explanation of Application Interfaces of the HMI, Multimedia and Telematics Domain	AIHMIMultimediaAnd Telematics	
CP	EXP	Explanation of Application Interfaces of the Powertrain Engine Domain	AIPowertrain	AIPT
CP	EXP	Explanation of Error Handling on Application Level	ApplicationLevelError Handling	ALEH
CP	EXP	Explanation of Firmware Over-The-Air	FirmwareOverTheAir	
CP	EXP	Explanation of Software Cluster Design And Integration Guideline for Classic Platform	SwClusterDesignAnd IntegrationGuideline	
CP	EXP	Explanatory Document for Usage of AUTOSAR Run-Time Interface	ARTI	
CP	EXP	Guide to BSW Distribution	BSWDistributionGuide	
CP	EXP	Guide to Mode Management	ModeManagementGuide	MMG
CP	EXP	Layered Software Architecture	LayeredSoftware Architecture	
CP	EXP	Macro Encapsulation of Interpolation Calls	MacroEncapsulationof InterpolationCalls	
CP	EXP	Modeling Show Cases Examples	ModelingShowCases	
CP	EXP	NV Data Handling Guideline	NVDataHandling	
CP	EXP	Overview of Functional Safety Measures in AUTOSAR	FunctionalSafetyMeasures	
CP	EXP	Safety Use Case Example	SafetyUseCase	
CP	EXP	Utilization of Crypto Services	UtilizationOfCryptoServices	
CP	MOD	Application Interface Examples	AISpecificationExamples	
CP	MOD	Basic Software UML Model	BSWUMLModel	
CP	MOD	Specification of ECU Configuration Parameters (XML)	ECUConfiguration Parameters	
CP	MOD	Specifications in ARXML format	SpecificationsARXML	
CP	MOD	XML Specification of Application Interfaces	AISpecification	
CP	RS	General Requirements on Basic Software Modules	BSWGeneral	BSW
CP	RS	General Requirements on SPAL	SPALGeneral	SPAL
CP	RS	Requirements on ADC Driver	ADCDriver	Adc
CP	RS	Requirements on Core Test	CoreTest	CoreTst
CP	RS	Requirements on DIO Driver	DIODriver	Dio
CP	RS	Requirements on Ethernet Support in AUTOSAR	Ethernet	Eth

▽

△

Document standard (abbr)	Document category (abbr)	Document title	Document name (full)	Document name (abbr)
CP	RS	Requirements on Firmware Over-The-Air	FirmwareOverTheAir	FOTA
CP	RS	Requirements on Flash Test	FlashTest	FlsTst
CP	RS	Requirements on FlexRay	FlexRay	Fr
CP	RS	Requirements on Function Inhibition Manager	FunctionInhibitionManager	Fim
CP	RS	Requirements on GPT Driver	GPTDriver	Gpt
CP	RS	Requirements on I/O Hardware Abstraction	IOHWAbstraction	IoHwAb
CP	RS	Requirements on I2C Driver	I2CDriver	I2C
CP	RS	Requirements on ICU Driver	ICUDriver	Icu
CP	RS	Requirements on MCU Driver	MCUDriver	Mcu
CP	RS	Requirements on Memory Hardware Abstraction Layer	MemoryHWAbstraction Layer	MemHwAb
CP	RS	Requirements on Memory Services	MemoryServices	Mem
CP	RS	Requirements on Mode Management	ModeManagement	ModeMgm
CP	RS	Requirements on OCU Driver	OCUDriver	Ocu
CP	RS	Requirements on Operating System	OS	Os
CP	RS	Requirements on PWM Driver	PWMDriver	Pwm
CP	RS	Requirements on Port Driver	PortDriver	Port
CP	RS	Requirements on RAM Test	RAMTest	RamTst
CP	RS	Requirements on Runtime Environment	RTE	Rte
CP	RS	Requirements on SPI Handler/Driver	SPIHandlerDriver	Spi
CP	RS	Requirements on SW-C and System Modeling	SWCModeling	SWMG
CP	RS	Requirements on Software Cluster Connection module	SoftwareClusterConnection	SwCluC
CP	RS	Requirements on Time Service	TimeService	Tm
CP	RS	Requirements on Transformer	Transformer	Xfrm
CP	RS	Requirements on Watchdog Driver	WatchdogDriver	Wdg
CP	SWS	General Specification of Basic Software Modules	BSWGeneral	BSW
CP	SWS	General Specification of Transformers	TransformerGeneral	Xfrm
CP	SWS	Specification for CAN XL Driver	CANXLDriver	CanXL
CP	SWS	Specification of ADC Driver	ADCDriver	Adc
CP	SWS	Specification of AUTOSAR Run-Time Interface	ARTI	Arti
CP	SWS	Specification of Basic Software Mode Manager	BSWModeManager	BswM
CP	SWS	Specification of Basic Software Multicore Library	BSWMulticoreLibrary	BMC
CP	SWS	Specification of Bit Handling Library	BFXLibrary	Bfx
CP	SWS	Specification of Bulk NvData Manager	BulkNvDataManager	BndM
CP	SWS	Specification of Bus Mirroring	BusMirroring	Mirror
CP	SWS	Specification of CAN Driver	CANDriver	Can
CP	SWS	Specification of CAN Interface	CANInterface	CANIF

▽

△

Document standard (abbr)	Document category (abbr)	Document title	Document name (full)	Document name (abbr)
CP	SWS	Specification of CAN Network Management	CANNetworkManagement	CanNm
CP	SWS	Specification of CAN State Manager	CANStateManager	CanSM
CP	SWS	Specification of CAN Transceiver Driver	CANTransceiverDriver	CanTrcv
CP	SWS	Specification of CAN Transport Layer	CANTransportLayer	CanTp
CP	SWS	Specification of CAN XL Transceiver Driver	CANXLTransceiverDriver	CanXLTrcv
CP	SWS	Specification of COM Based Transformer	COMBasedTransformer	ComXf
CP	SWS	Specification of CRC Library	CRCLibrary	Crc
CP	SWS	Specification of Communication	COM	Com
CP	SWS	Specification of Communication Manager	COMManager	ComM
CP	SWS	Specification of Communication Stack Types	CommunicationStackTypes	Comtype
CP	SWS	Specification of Core Test	CoreTest	CorTst
CP	SWS	Specification of Crypto Driver	CryptoDriver	Crypto
CP	SWS	Specification of Crypto Interface	CryptoInterface	CryIf
CP	SWS	Specification of Crypto Service Manager	CryptoServiceManager	Csm
CP	SWS	Specification of DIO Driver	DIODriver	Dio
CP	SWS	Specification of DMA Driver	DMADriver	Dma
CP	SWS	Specification of Data Distribution Service Transformer	DataDistributionService Transformer	DdsXf
CP	SWS	Specification of Data Distribution Service for Classic Platform	DataDistributionService	Dds
CP	SWS	Specification of Default Error Tracer	DefaultErrorTracer	Det
CP	SWS	Specification of Diagnostic Communication Manager	DiagnosticCommunication Manager	Dcm
CP	SWS	Specification of Diagnostic Event Manager	DiagnosticEventManager	Dem
CP	SWS	Specification of Diagnostic Log and Trace	DiagnosticLogAndTrace	Dlt
CP	SWS	Specification of Diagnostic over IP	DiagnosticOverIP	DoIP
CP	SWS	Specification of ECU State Manager	ECUStateManager	EcuM
CP	SWS	Specification of EEPROM Abstraction	EEPROMAbstraction	Ea
CP	SWS	Specification of Ethernet Driver	EthernetDriver	Eth
CP	SWS	Specification of Ethernet Interface	EthernetInterface	EthIf
CP	SWS	Specification of Ethernet State Manager	EthernetStateManager	EthSM
CP	SWS	Specification of Ethernet Switch Driver	EthernetSwitchDriver	EthSwt
CP	SWS	Specification of Ethernet Transceiver Driver	EthernetTransceiverDriver	EthTrcv
CP	SWS	Specification of Extended Fixed Point Library	EFXLibrary	Efx
CP	SWS	Specification of Firewall for Classic Platform	Firewall	Fw
CP	SWS	Specification of Fixed Point Interpolation Library	IFXLibrary	Ifx
CP	SWS	Specification of Fixed Point Math Library	MFXLibrary	Mfx

▽

△

Document standard (abbr)	Document category (abbr)	Document title	Document name (full)	Document name (abbr)
CP	SWS	Specification of Flash EEPROM Emulation	FlashEEPROMEmulation	Fee
CP	SWS	Specification of Flash Test	FlashTest	FlsTst
CP	SWS	Specification of FlexRay Driver	FlexRayDriver	Fr
CP	SWS	Specification of FlexRay ISO Transport Layer	FlexRayISOTransportLayer	FrTp
CP	SWS	Specification of FlexRay Interface	FlexRayInterface	FrIf
CP	SWS	Specification of FlexRay Network Management	FlexRayNetwork Management	FrNm
CP	SWS	Specification of FlexRay State Manager	FlexRayStateManager	FrSM
CP	SWS	Specification of FlexRay Transceiver Driver	FlexRayTransceiverDriver	FrTrcv
CP	SWS	Specification of Floating Point Interpolation Library	IFLLibrary	Ifl
CP	SWS	Specification of Floating Point Math Library	MFLLibrary	Mfl
CP	SWS	Specification of Function Inhibition Manager	FunctionInhibitionManager	Fim
CP	SWS	Specification of GPT Driver	GPTDriver	Gpt
CP	SWS	Specification of I-PDU Multiplexer	IPDUMultiplexer	IpduM
CP	SWS	Specification of I/O Hardware Abstraction	IOHardwareAbstraction	IoHwAb
CP	SWS	Specification of I2C Driver	I2CDriver	I2C
CP	SWS	Specification of ICU Driver	ICUDriver	Icu
CP	SWS	Specification of IEEE1722 Transport Protocol Module	IEEE1722TransportLayer	IEEE1722Tp
CP	SWS	Specification of ISO15118 Charging	ISO15118Charging	ISO15118Chrg
CP	SWS	Specification of Intrusion Detection System Manager	IntrusionDetectionSystem Manager	IdsM
CP	SWS	Specification of Key Manager	KeyManager	KeyM
CP	SWS	Specification of LIN Driver	LINDriver	Lin
CP	SWS	Specification of LIN Interface	LINInterface	LinIf, LinTp
CP	SWS	Specification of LIN State Manager	LINStateManager	LinSM
CP	SWS	Specification of LIN Transceiver Driver	LINTransceiverDriver	LinTrcv
CP	SWS	Specification of Linklayer Sdu Routing Module	LSduRouter	LSduR
CP	SWS	Specification of MACsec Key Agreement	MACsecKeyAgreement	Mka
CP	SWS	Specification of MCU Driver	MCUDriver	Mcu
CP	SWS	Specification of MSFLibrary	MSFLibrary	Msf
CP	SWS	Specification of Memory Abstraction Interface	MemoryAbstraction Interface	MemIf
CP	SWS	Specification of Memory Access	MemoryAccess	MemAcc
CP	SWS	Specification of Memory Driver	MemoryDriver	Mem
CP	SWS	Specification of Memory Mapping	MemoryMapping	MemMap
CP	SWS	Specification of Module E2E Transformer	E2ETransformer	E2EXf
CP	SWS	Specification of Module XCP	XCP	Xcp
CP	SWS	Specification of NVRAM Manager	NVRAMManager	NvM

▽

△

Document standard (abbr)	Document category (abbr)	Document title	Document name (full)	Document name (abbr)
CP	SWS	Specification of Network Management Interface	NetworkManagement Interface	Nm
CP	SWS	Specification of Network Management for SAE J1939	SAEJ1939Network Management	J1939Nm
CP	SWS	Specification of OCU Driver	OCUDriver	Ocu
CP	SWS	Specification of Operating System	OS	Os
CP	SWS	Specification of PDU Router	PDURouter	PduR
CP	SWS	Specification of PWM Driver	PWMDriver	Pwm
CP	SWS	Specification of Platform Types for Classic Platform	PlatformTypes	Platform
CP	SWS	Specification of Port Driver	PortDriver	Port
CP	SWS	Specification of RAM Test	RAMTest	RamTst
CP	SWS	Specification of RTE Software	RTE	Rte
CP	SWS	Specification of SOME/IP Transformer	SOMEIPTransformer	SomeIpXf
CP	SWS	Specification of SPI Handler/Driver	SPIHandlerDriver	Spi
CP	SWS	Specification of SW-C End-to-End Communication Protection Library	E2ELibrary	E2E
CP	SWS	Specification of Secure Onboard Communication	SecureOnboard Communication	SecOC
CP	SWS	Specification of Service Discovery	ServiceDiscovery	Sd
CP	SWS	Specification of Socket Adaptor	SocketAdaptor	SoAd
CP	SWS	Specification of Software Cluster Connection module	SoftwareClusterConnection	SwCluC
CP	SWS	Specification of Standard Types	StandardTypes	Std
CP	SWS	Specification of Synchronized Time-Base Manager	SynchronizedTimeBase Manager	StbM
CP	SWS	Specification of TCP/IP Stack	TcpIp	TcpIp, IKE
CP	SWS	Specification of Time Service	TimeService	Tm
CP	SWS	Specification of Time Synchronization over CAN	TimeSyncOverCAN	CanTSyn
CP	SWS	Specification of Time Synchronization over Ethernet	TimeSyncOverEthernet	EthTSyn
CP	SWS	Specification of Time Synchronization over FlexRay	TimeSyncOverFlexRay	FrTSyn
CP	SWS	Specification of UDP Network Management	UDPNetworkManagement	UdpNm
CP	SWS	Specification of VDP Communication Module Remote	VDPCMRemote	VdpCmR
CP	SWS	Specification of Watchdog Driver	WatchdogDriver	Wdg
CP	SWS	Specification of Watchdog Interface	WatchdogInterface	WdgIf
CP	SWS	Specification of Watchdog Manager	WatchdogManager	WdgM
CP	SWS	Specification of a Diagnostic Communication Manager for SAE J1939	SAEJ1939Diagnostic CommunicationManager	J1939Dcm
CP	SWS	Specification of a Functional Safety Communication Protocol Handler for SAE J1939	SAEJ1939FunctionalSafety CommProtocol	J1939Fscp
CP	SWS	Specification of a Request Manager for SAE J1939	SAEJ1939Request Manager	J1939Rm

▽

△

Document standard (abbr)	Document category (abbr)	Document title	Document name (full)	Document name (abbr)
CP	SWS	Specification of a Transport Layer for SAE J1939	SAEJ1939TransportLayer	J1939Tp
CP	SWS	Specification on SOME/IP Transport Protocol	SOMEIPTransportProtocol	SomeIpTp
CP	TPS	Basic Software Module Description Template	BSWModuleDescription Template	BSWMDT
CP	TPS	Diagnostic Extract Template	DiagnosticExtractTemplate	DEXT
CP	TPS	Software Component Template	SoftwareComponent Template	SWCT
CP	TPS	Specification of ECU Configuration	ECUConfiguration	ECUC, Cdd
CP	TPS	Specification of ECU Resource Template	ECUResourceTemplate	ECUR
CP	TPS	Specification of Timing Extensions for Classic Platform	TimingExtensions	TIMEX
CP	TPS	System Template	SystemTemplate	SYST
CP	TR	AUTOSAR Classic Platform Specification Hashes	SpecificationHashes	
CP	TR	Application Design Patterns Catalogue	AIDesignPatterns Catalogue	
CP	TR	Classic Platform Change Documentation	ChangeDocumentation	
CP	TR	Classic Platform Release Overview	ReleaseOverview	
CP	TR	Methodology for Classic Platform	Methodology	METH
CP	TR	Modeling Guidelines of Basic Software EA UML Model	BSWUMLModelModeling Guide	BSWMG
CP	TR	Modeling Show Cases Report	ModelingShowCases	
CP	TR	Modeling and Naming Aspects for Documentation, Measurement, and Calibration	AIMeasurementCalibration Diagnostics	MCR, MCA, MCG, MCM
CP	TR	SW-C and System Modeling Guide	SWCModelingGuide	SWMG, SWNR
CP	TR	Supplementary material of general blueprints for AUTOSAR	GeneralBlueprints Supplement	
CP	TR	Technical Report on Supplementary Material of Workflow Example	WorkflowExample	
CP	TR	Virtual Functional Bus	VFB	VFB
FO	ASWS	Specification of Health Monitoring	HealthMonitoring	HM
FO	EXP	Explanation of Adaptive and Classic Platform Software Architectural Decisions	SWArchitecturalDecisions	
FO	EXP	Explanation of Diagram Source	DiagramSource	
FO	EXP	Explanation of QoS Policies in the scope of SOME/IP	QoSPoliciesInTheScopeOf SOMEIP	
FO	EXP	Explanation of Safety Overview	SafetyOverview	
FO	EXP	Explanation of Security Overview	SecurityOverview	
FO	EXP	Explanation of System Health Monitoring	SystemHealthMonitoring	
FO	EXP	Explanation of Time Sensitive Network features	TimeSensitiveNetwork Features	
FO	MMOD	Meta Model	MetaModel	
FO	MMOD	Meta Model-generated XML Schema	XMLSchema	
FO	MOD	AUTOSAR Feature Model	Features	
FO	MOD	AUTOSAR Miscellaneous Support Files	MiscSupport	

▽

△

Document standard (abbr)	Document category (abbr)	Document title	Document name (full)	Document name (abbr)
FO	MOD	Collection of blueprints for AUTOSAR M1 models	GeneralBlueprints	
FO	MOD	Specifications in ARXML format	SpecificationsARXML	
FO	MOD	Standardized M1 Models used for the Definition of AUTOSAR	GeneralDefinitions	
FO	PRS	E2E Protocol Specification	E2EProtocol	E2E
FO	PRS	Log and Trace Protocol Specification	LogAndTraceProtocol	Dlt
FO	PRS	SOME/IP Protocol Specification	SOMEIPProtocol	SOMEIP
FO	PRS	SOME/IP Service Discovery Protocol Specification	SOMEIPServiceDiscovery Protocol	SOMEIPSD
FO	PRS	Specification of Bus Mirroring Protocol	BusMirroringProtocol	Mirror
FO	PRS	Specification of DDS Service Communication Protocol	DDSCommunication Protocol	DDS
FO	PRS	Specification of DDS Service Discovery Protocol	DDSServiceDiscovery Protocol	DDSSD
FO	PRS	Specification of Intrusion Detection System Protocol	IntrusionDetectionSystem	Ids
FO	PRS	Specification of Secure Onboard Communication Protocol	SecOCProtocol	SecOc
FO	PRS	Specification of the AUTOSAR Network Management Protocol	NetworkManagement Protocol	Nm
FO	PRS	Time Synchronization over CAN Protocol Specification	TimeSyncOver CANProtocol	CanTSyn
FO	PRS	Time Synchronization over Ethernet Protocol Specification	TimeSyncOverEthernet Protocol	TS
FO	PRS	Vehicle Data Protocol Specification	VDP	VDP
FO	PRS	Vehicle-2-X Remote Access Layer Protocol Specification	V2XRemoteAccessLayer	V2xRAL
FO	RS	Requirements on Data Distribution Service	DataDistributionService	Dds
FO	RS	Requirements on Diagnostics	Diagnostics	Diag
FO	RS	Requirements on E2E	E2E	E2E
FO	RS	Requirements on Health Monitoring	HealthMonitoring	HM
FO	RS	Requirements on IPsec Protocol	IPsecProtocol	IPSEC
FO	RS	Requirements on MACsec	MACsec	MACsec
FO	RS	Requirements on Time Synchronization	TimeSync	TS
FO	RS	Requirements on Vehicle Data Protocol	VDP	VDP
FO	RS	Safety Requirements for AUTOSAR Adaptive Platform and AUTOSAR Classic Platform	Safety	SAF
FO	TPS	ARXML Serialization Rules	ARXMLSerializationRules	ASR
FO	TPS	AUTOSAR Feature Model Exchange Format	FeatureModelExchange Format	FMDT
FO	TPS	AUTOSAR XML Schema Production Rules	XMLSchemaProduction Rules	XMLSPR
FO	TPS	Generic Structure Template	GenericStructureTemplate	GST
FO	TPS	Log And Trace Template	LogAndTrace	DLTXT
FO	TPS	Security Extract Template	SecurityExtractTemplate	SECXT

▽

△

Document standard (abbr)	Document category (abbr)	Document title	Document name (full)	Document name (abbr)
FO	TPS	Specification of Abstract Platform	AbstractPlatform Specification	APSD
FO	TPS	Standardization Template	StandardizationTemplate	STDT
FO	TR	AUTOSAR Foundation Specification Hashes	SpecificationHashes	
FO	TR	Collection of constraints on AUTOSAR M1 models	AutosarModelConstraints	
FO	TR	Foundation Change Documentation	ChangeDocumentation	
FO	TR	Foundation Release Overview	ReleaseOverview	
FO	TR	Glossary	Glossary	
FO	TR	List of known Issues of Secure Hardware Extensions	ListOfKnownIssuesSecure HardwareExtensions	
FO	TR	Specification of Secure Hardware Extensions	SecureHardware Extensions	
FO	TR	Supplementary material of the AUTOSAR XML Schema	XMLSchemaSupplement	
FO	TR	Technical Report on AUTOSAR Features	Features	FEAT
FO	TR	Technical Report on Security Events Specification	SecurityEvents Specification	
FO	TR	Timing Analysis and Design	TimingAnalysis	

Table A.1: AUTOSAR Document Names and Abbreviations for Trace Prefixes

A.3 BSW modules

The BSW modules are originally defined in [59].

[SWS_BSW_91001] List of Basic Software Modules

Module name	*Module abbreviation (API service prefix)*	*Module ID (uint16)*	*AUTOSAR SW Layer*
ADC Driver	Adc	123	I/O Drivers
AUTOSAR Run-Time Interface	Arti	005	System Services
BSW Mode Manager	BswM	042	System Services
BSW Scheduler Module	SchM	130	System Services
Bulk NvData Manager	BndM	023	Memory Services
Bus Mirroring	Mirror	048	Communication Services
CAN Driver	Can	080	Communication Drivers
CAN Interface	CanIf	060	Communication Hardware Abstraction
CAN Network Management	CanNm	031	Communication Services
CAN State Management	CanSM	140	Communication Services
CAN Transceiver Driver	CanTrcv	070	Communication Hardware Abstraction
CAN Transport Layer	CanTp	035	Communication Services
CAN XL Driver	CanXL	085	Communication Drivers

▽

△

Module name	*Module abbreviation (API service prefix)*	*Module ID (uint16)*	*AUTOSAR SW Layer*
CAN XL Transceiver Driver	CanXLTrcv	072	Communication Hardware Abstraction
Charging Manager	ISO15118 Chrg	215	Communication Services
COM	Com	050	Communication Services
COM Based Transformer	ComXf	175	Communication Services
COM Manager	ComM	012	System Services
Complex Drivers	Cdd	255; 2048-4095	Complex Drivers
Core Test	CorTst	103	Microcontroller Drivers
Crypto Driver	Crypto	114	Crypto Drivers
Crypto Interface	CryIf	112	Crypto Hardware Abstraction
Crypto Service Manager	Csm	110	Crypto Services
Data Distribution Service	Dds	047	Communication Services
Data Distribution Service Transformer	DdsXf	187	Communication Services
Default Error Tracer	Det	015	System Services
Diagnostic Communication Manager	Dcm	053	Communication Services
Diagnostic Event Manager	Dem	054	System Services
Diagnostic Log and Trace	Dlt	055	Communication Services
Diagnostic over IP	DoIP	173	Communication Services
DIO Driver	Dio	120	I/O Drivers
DMA Driver	Dma	216	Microcontroller Drivers
E2E Transformer	E2EXf	176	Communication Services
ECU State Manager	EcuM	010	System Services
EEPROM Abstraction	Ea	040	Memory Hardware Abstraction
Ethernet Driver	Eth	088	Communication Drivers
Ethernet Interface	EthIf	065	Communication Hardware Abstraction
Ethernet State Manager	EthSM	143	Communication Services
Ethernet Switch Driver	EthSwt	089	Communication Hardware Abstraction
Ethernet Transceiver Driver	EthTrcv	073	Communication Hardware Abstraction
Firewall	Fw	111	Communication Services
Flash EEPROM Emulation	Fee	021	Memory Hardware Abstraction
Flash Test	FlsTst	104	Memory Drivers
FlexRay Driver	Fr	081	Communication Drivers
FlexRay Interface	FrIf	061	Communication Hardware Abstraction
FlexRay ISO Transport Layer	FrTp	036	Communication Services
FlexRay Network Management	FrNm	032	Communication Services
FlexRay State Manager	FrSM	142	Communication Services
FlexRay Transceiver Driver	FrTrcv	071	Communication Hardware Abstraction
Function Inhibition Manager	FiM	011	System Services
GPT Driver	Gpt	100	Microcontroller Drivers
I2C Driver	I2C	113	Communication Drivers
ICU Driver	Icu	122	I/O Drivers
IEEE1722 Transport Layer	IEEE1722Tp	131	Communication Services
Intrusion Detection System Manager	IdsM	108	Crypto Services

▽

△

Module name	*Module abbreviation (API service prefix)*	*Module ID (uint16)*	*AUTOSAR SW Layer*
IO HW Abstraction	IoHwAb	254	I/O Hardware Abstraction
IPDU Multiplexer	IpduM	052	Communication Services
Key Manager	KeyM	109	Crypto Services
LIN Driver	Lin	082	Communication Drivers
LIN Interface	LinIf	062	Communication Hardware Abstraction
LIN State Manager	LinSM	141	Communication Services
LIN Transceiver Driver	LinTrcv	064	Communication Hardware Abstraction
LSDU Router	LSduR	132	Communication Hardware Abstraction
MACsec Key Agreement	Mka	151	Communication Services
MCU Driver	Mcu	101	Microcontroller Drivers
Memory Abstraction Interface	MemIf	022	Memory Hardware Abstraction
Memory Access	MemAcc	041	Memory Hardware Abstraction
Memory Driver	Mem	091	Memory Drivers
Network Management Interface	Nm	029	Communication Services
NVRAM Manager	NvM	020	Memory Services
OCU Driver	Ocu	125	I/O Drivers
OS	Os	001	System Services - OS
PDU Router	PduR	051	Communication Services
Port Driver	Port	124	I/O Drivers
PWM Driver	Pwm	121	I/O Drivers
RAM Test	RamTst	093	Memory Drivers
RTE	Rte	002	RTE
SAE J1939 Diagnostic Communication Manager	J1939Dcm	058	Communication Services
SAE J1939 Functional Safety Communication Protocol	J1939Fscp	194	Communication Services
SAE J1939 Network Management	J1939Nm	034	Communication Services
SAE J1939 Request Manager	J1939Rm	059	Communication Services
SAE J1939 Transport Layer	J1939Tp	037	Communication Services
Secure Onboard Communication	SecOC	150	Communication Services
Service Discovery	Sd	171	Communication Services
Socket Adaptor	SoAd	056	Communication Services
Software Cluster Connection	SwCluC	016	System Services
SOME/IP Transformer	SomeIpXf	174	Communication Services
SOME/IP Transport Protocol	SomeIpTp	177	Communication Services
SPI Handler Driver	Spi	083	Communication Drivers
Synchronized Time-Base Manager	StbM	160	System Services
TCP/IP Stack	TcpIp	170	Communication Services
Time Service	Tm	014	System Services
Time Sync over CAN	CanTSyn	161	Communication Services
Time Sync over Ethernet	EthTSyn	164	Communication Services
Time Sync over FlexRay	FrTSyn	163	Communication Services
UDP Network Management	UdpNm	033	Communication Services
VDP Communication Module Remote	VdpCmR	188	Communication Services

▽

△

Module name	*Module abbreviation (API service prefix)*	*Module ID (uint16)*	*AUTOSAR SW Layer*
Watchdog Driver	Wdg	102	Microcontroller Drivers
Watchdog Interface	WdgIf	043	Onboard Device Abstraction
Watchdog Manager	WdgM	013	System Services
XCP	Xcp	212	Communication Services

[SWS_BSW_91000] List of Libraries

Library name	*Library (module) abbreviation (API service prefix)*	*Library (module) ID (uint16)*
BFx Library	Bfx	205
BSW Multicore Library	Bmc	214
CRC Library	Crc	201
E2E Library	E2E	207
EFx Library	Efx	208
IFl Library	Ifl	209
IFx Library	Ifx	213
Memory Library	Msf	202
MFl Library	Mfl	210
MFx Library	Mfx	211

[SWS_BSW_91002] List of special files

AUTOSAR name	*Short name (API service prefix)*	*File name*	*Module ID*
Communication Stack Types	Comtype	ComStack_Types.h	196
Memory Mapping	MemMap	(Prefix_)MemMap.h	195
Platform Types	Platform	Platform_Types.h	199
Standard Types	Std	Std_Types.h	197

A.4 Functional Clusters

The functional clusters are originally defined in [60].

[AP_SWS_GENERAL_90025] AUTOSAR-standardized Functional Cluster names

Name	*Description*	*C++ namespace*
`aag`	`Automotive API Gateway`	`ara::aag`
`adi`	`Sensor Interfaces`	`ara::adi`
`apbsw`	`Adaptive Platform Base Software`	n/a

▽

△

Name	Description	C++ namespace
com	Communication Management	ara::com
core	Core	ara::core
crypto	Cryptography	ara::crypto
diag	Diagnostics	ara::diag
exec	Execution Management	ara::exec
fw	Firewall	ara::fw
idsm	Intrusion Detection System Manager	ara::idsm
log	Log and Trace	ara::log
nm	Network Management	ara::nm
osi	Operating System Interface	ara::osi
per	Persistency	ara::per
phm	Platform Health Management	ara::phm
rds	Raw Data Stream	ara::rds
rper	Remote Persistency	ara::rper
shwa	Safe Hardware Acceleration	ara::shwa
sm	State Management	ara::sm
tsync	Time Synchronization	ara::tsync
ucm	Update and Configuration Management	ara::ucm
vucm	Vehicle Update and Configuration Management	ara::vucm

A.5 M2 Constraint Imposition Times

The imposition times on AUTOSAR model constraints are originally defined in [10].

Imposition Time	Description	Motivation
IT_Apsd	Creation of the Abstract Platform System Description is finished	This imposition time indicates when the Abstract Platform System Description is complete.
IT_BefAraApiGen	Before the generation of the ara API starts	This imposition time is aimed at the time when a software-component is ready for generating the header files such that the implementation of the software-component can be started.
IT_BinObjMetaData	The definition of binary object meta-data is finished	This imposition time denotes the step in the workflow where the description of CpSoftwareClusterBinaryManifestDescriptor is considered complete so that that the Software Cluster Binary Manifest can be defined during the integration.
IT_BlueprintCreation	Blueprint is created	This imposition time denotes the step in the workflow when a Blueprint is created.
IT_BswMD	Configuration of the BSW module is finished	This imposition time is applicable at the time when the BSW module description is complete.
IT_CfgFcd	Creation of APMC parameter definition model is complete	This imposition time denotes the step in the workflow where the APMC parameter definition of a functional cluster is considered complete. This means that the project development on a given target platform can be started. This imposition time is special in the sense that this happens during AUTOSAR Standard development or Stack Vendor Product development.

▽

△

Imposition Time	*Description*	*Motivation*
IT_CfgFcVal	Creation of APMC configuration values of a Functional Cluster is complete.	This imposition time denotes the step in the workflow, where the configuration of a functional cluster is considered complete such that the installation on a target platform can be started.
IT_CompSwcT	Creation of the CompositionSw ComponentType is finished	This imposition time applies to the creation of compositions of software-components. This imposition time is considered optional. In other words, there may be use cases to deliver `CompositionSwComponentType`s that violate constraints with this imposition time to another party. But it may also make sense in some cases to make sure, that a `CompositionSwComponentType` that is going to be delivered to another party fulfills the constraints associated with this binding time.
IT_CpgExe	Contract Phase generation is executed	This imposition time is aimed at the time when a software-component is ready for generating the contract phase header files such that the implementation of the software-component can be started.
IT_DesExe	Design of the Executable is completed	This imposition time is aimed at the time when an `Executable` is finished, i.e. it shall be used in constraints that target the consistency of the modeling of `Executable`.
IT_DiagDes	Diagnostic design is completed	This imposition time denotes the step in the workflow, where the diagnostic design is about to be finished independent of the standard where it is applied. This includes the finalization, e.g., of the DEXT.
IT_EcucDef	ECU Configuration Parameter Definition is complete	This imposition time denotes the step in the workflow where the ECU Configuration Parameter Definition (either a STANDARDIZED_MODULE_DEFINITION or a VENDOR_SPECIFIC_MODULE_DEFINITION) of a Basic Software Module is considered complete. This imposition time is special in the sense that this happens during AUTOSAR Standard development or Stack Vendor Product development.
IT_EcucVal	ECU Configuration Values is complete	This imposition time denotes the step in the workflow where the ECU Configuration Values of a Basic Software Module are considered complete such that they can be used to generate code.
IT_EcuExt	ECU_EXTRACT is completed	This imposition time denotes the step in the workflow where the ECU_EXTRACT model is considered complete such that it can be used as input for the generation of the RTE.
IT_FeatMod	Feature Model is completed	This imposition time denotes the step in the workflow, where the feature model is about to be finished.
IT_GraDes	GrantDesign is completed	This imposition time is aimed at the time when a `GrantDesign` is finished, i.e. it shall be used in constraints that target the consistency of the modeling of `GrantDesign`.
IT_InstRef	Creation of InstanceRef is completed	This imposition time indicates when the configuration of instance ref is complete.
IT_LogTrace	Log and Trace Extract is complete	This imposition time denotes the step in the workflow, where the Log and Trace Extract is about to be finished.
IT_MachDes	Machine design is completed	This imposition time denotes the step in the workflow, where the machine design is about to be finished.
IT_Mani	Creation of the manifest is finished	This imposition time denotes the step in the workflow, where the manifest is considered complete such that the installation on a target platform can be started.
IT_Name	Naming of model elements is completed	This imposition time indicates when the naming of model elements is complete.
IT_ProDes	ProcessDesign is completed	This imposition time is aimed at the time when a `ProcessDesign` is finished, i.e. it shall be used in constraints that target the consistency of the modeling of `ProcessDesign`.

▽

△

Imposition Time	*Description*	*Motivation*
IT_RefBase	Configuration of reference bases is completed	This imposition time indicates when the configuration of reference bases is complete.
IT_ResPool	Definition of the resource pool is finished	This imposition time denotes the step in the workflow where the pool of resources which can be provided or required by Software Clusters is considered complete such that such that the development and integration of the Software Cluster can start.
IT_RteGen	RTE is generated	This imposition time denotes the step in the workflow where the model is considered complete such that the generation of the RTE can be executed. This means that *at the time when the RTE is generated*, all constraints that need to be imposed *at the time when the contract phase generation is executed* **and** those that are imposed *at any time in the workflow* also need to be observed. In other words, a constraint that is imposed *at the time when the contract phase generation is executed* shall also be imposed *at the time when the RTE is generated*.
IT_SecEx	Security Extract is completed	This imposition time indicates when the Security Extract is complete.
IT_SovdSysDes	SOVD System Design is complete	This imposition time denotes the step in the workflow, where the SOVD system design is about to be finished.
IT_SubSysDes	Sub-system design is completed	This imposition time denotes the step in the workflow, where the sub-system design is about to be finished.
IT_SwCluSysDesc	SW_CLUSTER_SYSTEM_ DESCRIPTION is completed	This imposition time denotes the step in the workflow where the SW_CLUSTER_SYSTEM_DESCRIPTION model is considered complete such that the development and integration of the Software Cluster can start.
IT_SysDes	System design is completed	This imposition time denotes the step in the workflow, where the system design is about to be finished.
IT_SysDesc	SYSTEM_DESCRIPTION is completed	This imposition time is aimed at the time when a system description (e.g. SYSTEM_DESCRIPTION or SYSTEM_ EXTRACT or ECU_SYSTEM_DESCRIPTION) is complete for exchange between parties and is ready as input for Ecu Extract and Com Stack generators.
IT_TimingDescription Ready	The constraint shall be imposed when the TimingDescription is ready for validation	This can be either on the TimingDescriptionEventChain or the sub-class of TimingDescriptionEvent depending on the condition of the constraint.
IT_TimingView Ready	The constraint shall be imposed when the TimingExtension (Timing View) is ready for validation	-
IT_ValSpec	ValueSpecification is applied	This imposition time is aimed at the point in time where a `ValueSpecification` is applied to data object and consistency requirements between the `ValueSpecification` and the data object can be checked.
IT_Variant	Variant configuration is completed	This imposition time indicates when the configuration of variant model elements is complete.

Table A.2: Overview Imposition Times

A.6 Modeling keywords

The modeling keywords are originally defined in [61].

Short Name	Abbreviation	Long Name	Description	Classification
A	A	Acceleration	This keyword is used to give the physical value of acceleration. It is the value of gaining speed.	Action-PhysicalType
Abnorm	Abnorm	Abnormal		Condition-Qualifier
Abort	Abort	Abort		Action-PhysicalType
Abs	Abs	ABS	Antilock Braking System	Mean-Environment-Device
Abslt	Abslt	Absolute	Absolute value	Condition-Qualifier
Absnt	Absnt	Absent		Condition-Qualifier
Abv	Abv	Above		Preposition
Ac	Ac	AC	Alternating Current / Voltage	Condition-Qualifier
Ac1	Ac	AC	Air Conditioner or Air Conditioning (A/C).	Mean-Environment-Device
Acc	Acc	ACC	Adaptive Cruise Control	Mean-Environment-Device
Acco	Acco	Accommodation		Condition-Qualifier
Accr	Accr	Accelerator		Mean-Environment-Device
Accy	Accy	Accuracy	The accuracy of a measurement method respectively the quality of signals are defined in ISO 5725: The "accuracy" (Genauigkeit) is described by the two terms "trueness" (Richtigkeit) and "precision" (Präzision).With the condition, that the "Precision" is homogenously spread over it's whole range, the fomula "Accuracy = Trueness + 0,5 Precision" decribes the accuracy.	Condition-Qualifier
Ack	Ack	Acknowledge		Action-PhysicalType
Ackd	Ackd	Acknowledged		Condition-Qualifier
Acos	Acos	Arccosine	Inverse function of cosine	Condition-Qualifier
Acoust	Acoust	Acoustic		Condition-Qualifier
Acpt	Acpt	Accept		Action-PhysicalType
Acptd	Acptd	Accepted		Condition-Qualifier
Acq	Acq	Acquire		Action-PhysicalType
Acqd	Acqd	Acquired		Condition-Qualifier
Acqn	Acqn	Acquisition	e.g. signal acquisition	Condition-Qualifier
Acrt	Acrt	Accurate		Condition-Qualifier
Acry	Acry	Accuracy		Condition-Qualifier
Acs	Acs	Access		Action-PhysicalType
Acsy	Acsy	Accessory		Mean-Environment-Device
Act	Act	Actual		Condition-Qualifier
Actn	Actn	Action		Condition-Qualifier
Actr	Actr	Actuator		Mean-Environment-Device
Acttn	Acttn	Actuation		Action-PhysicalType
Actv	Actv	Active		Condition-Qualifier
Actvd	Actvd	Activated		Condition-Qualifier
Actvn	Actvn	Activation		Action-PhysicalType
Actvt	Actvt	Activate		Action-PhysicalType

△

Short Name	Abbreviation	Long Name	Description	Classification
Adas	Adas	Adas	Advanced driver assistance systems (ADAS) are systems developed to automate/adapt/enhance vehicle systems for safety and better driving.	Mean-Environment-Device
Adbtc	Adbtc	Adiabatic		Condition-Qualifier
Adc	Adc	ADC	Analog Digital Converter	Mean-Environment-Device
Add	Add	Additive		Condition-Qualifier
Addl	Addl	Additional		Condition-Qualifier
Adh	Adh	Adhesion		Action-PhysicalType
Adj	Adj	Adjust		Action-PhysicalType
Adjmt	Adjmt	Adjustment		Action-PhysicalType
Admin	Admin	Administration	administration, administrator	Condition-Qualifier
Adpd	Adpd	Adapted		Condition-Qualifier
Adpn	Adpn	Adaptation		Condition-Qualifier
Adpr	Adpr	Adapter		Mean-Environment-Device
Adpv	Adpv	Adaptive		Condition-Qualifier
Adr	Adr	Address		Condition-Qualifier
Adsorb	Adsorb	Adsorb		Action-PhysicalType
Adsorpn	Adsorpn	Adsorption		Condition-Qualifier
Adv	Adv	Advance		Condition-Qualifier
Aecd	Aecd	AECD	Auxiliary Emission Control Devices	Condition-Qualifier
Afim	Afim	Air-Fuel Imbalance Monitor (AFIM)	The air to fuel imbalance monitor is designed to detect differences in the air to fuel ratio between cylinders per engine bank.	Mean-Environment-Device
Aft	Aft	After		Preposition
Ag	Ag	Angle		Action-PhysicalType
Age	Age	Age		Condition-Qualifier
Aged	Aged	Aged		Condition-Qualifier
Aggrvd	Aggrvd	Aggravated		Condition-Qualifier
Agi	Agi	Aging		Condition-Qualifier
Agr	Agr	Angular		Condition-Qualifier
Agt	Agt	Agent	E.g. Reducing Agent, Cooling Agent	Mean-Environment-Device
Ahd	Ahd	Ahead		Condition-Qualifier
Ai	Ai	AI	Application Interfaces	Mean-Environment-Device
Aid	Aid	Aid		Action-PhysicalType
Air	Air	Air	Also used as definition of "fresh gas"	Mean-Environment-Device
Airb	Airb	Airbag	airbag	Mean-Environment-Device
Aj	Aj	Antijerk	Avoiding high-frequent small or abrupt movements, quick changes of torque applied.	Mean-Environment-Device
Ajar	Ajar	Ajar		Condition-Qualifier
Alg	Alg	Algorithm		Condition-Qualifier

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Algn	Algn	Align		Action-PhysicalType
Algnd	Algnd	Aligned		Condition-Qualifier
All	All	All		Condition-Qualifier
Allw	Allw	Allow		Action-PhysicalType
Allwbl	Allwbl	Allowable		Condition-Qualifier
Allwd	Allwd	Allowed		Condition-Qualifier
Alpha	Alpha	Alpha		Condition-Qualifier
Alrm	Alrm	Alarm		Condition-Qualifier
Alrmd	Alrmd	Alarmed		Condition-Qualifier
Alt	Alt	Alternator	alternator	Mean-Environment-Device
Alti	Alti	Altitude		Action-PhysicalType
Altv	Altv	Alternative		Mean-Environment-Device
Alv	Alv	Alive		Condition-Qualifier
Alwys	Alwys	Always	At all times	Condition-Qualifier
Amb	Amb	Ambient		Mean-Environment-Device
Ammonia	Nh3	Ammonia	A pungent colorless gaseous alkaline compound of nitrogen and hydrogen NH3 that is very soluble in water and can easily be condensed to a liquid by cold and pressure	Action-PhysicalType
Amnt	Amnt	Amount	a total number or quantity, e.g. amount of substance.	Action-PhysicalType
Amp	Amp	Amplitude		Condition-Qualifier
Amplfr	Amplfr	Amplifier		Mean-Environment-Device
Ampr	Ampr	Ampere		Mean-Environment-Device
Amt	Amt	AMT	Automated Manual Transmission	Mean-Environment-Device
An	An	Analog		Condition-Qualifier
Anchrfitg	Anchrfitg	Anchorfitting		Mean-Environment-Device
Ancllry	Ancllry	Ancillary		Mean-Environment-Device
And	And	And	Preposition	Preposition
Ande	Ande	Anode	The electrode of an electrochemical cell at which oxidation occurs.	Mean-Environment-Device
Antcpt	Antcpt	Anticipate		Action-PhysicalType
Antcptd	Antcptd	Anticipated		Condition-Qualifier
Antcptn	Antcptn	Anticipation		Condition-Qualifier
Anti	Anti	Anti		Condition-Qualifier
Antisub	Antisub	Antisubmarine	Term used in restraint systems for automotive applications related to an Airbag that upon deployment prevents a front row passenger from diving below the dashboard by lifting the person's lower body. Thus, the person is brougth into a better position relative to the main Front Airbag.	Mean-Environment-Device
Antithft	Antithft	Antitheft		Mean-Environment-Device

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Any	Any	Any		Condition-Qualifier
Ape	Ape	APE	Outer Pump Electrode. Electrode of a lambda sensor facing directly the exhaust gas.	Mean-Environment-Device
Api	Api	Application Programming Interface	An Application Programming Interface is a set of definitions and protocols to build and integrate application sofftware.	Mean-Environment-Device
Apil	Apil	Apil	A-Pillar of a car	Mean-Environment-Device
Appch	Appch	Approach		Action-PhysicalType
Appl	Appl	Application		Mean-Environment-Device
Appld	Appld	Applied		Condition-Qualifier
Apply	Apply	Apply		Action-PhysicalType
Approx	Approx	Approximate		Condition-Qualifier
Approxn	Approxn	Approximation	A number, amount, etc. that is not exact, but almost correct.	Action-PhysicalType
Aprvd	Aprvd	Approved		Condition-Qualifier
Ar	Ar	Area		Action-PhysicalType
Ar1	Ar	AUTOSAR		Mean-Environment-Device
Arbd	Arbd	Arbitrated		Condition-Qualifier
Arbn	Arbn	Arbitration		Action-PhysicalType
Arch	Arch	Architecture	A description of system, software or hardware elements, of the relations among them, and of the properties of both elements and relations.	Mean-Environment-Device
Area	Area	Area		Action-PhysicalType
Arg	Arg	Argument		Condition-Qualifier
Arm	Arm	Arm	Intended as the action of arming some device	Action-PhysicalType
Armed	Armed	Armed		Condition-Qualifier
Arming	Arming	Arming	Intended as the action of arming some device	Action-PhysicalType
Armtr	Armtr	Armature	This keyword described the constructional element of an armature. It should be used e.g. for calculations of the armature movement of a coil actuator.	Mean-Environment-Device
Arthm	Arthm	Arithmetic		Condition-Qualifier
Ary	Ary	Array		Condition-Qualifier
Asc	Asc	ASC	Acceleration Slip Control	Mean-Environment-Device
Ascd	Ascd	Ascending	Ascending, increasing	Condition-Qualifier
Ascn	Ascn	Antiscanning		Mean-Environment-Device

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Asep	Asep	Additional Sound Emission Provisions (ASEP)	Additional Sound Emission Provisions - extra sound emitting elements built in, e.g. to make electric vehicles receptable for humans	Mean-Environment-Device
Asgn	Asgn	Assign		Action-PhysicalType
Ash	Ash	Ash	e.g. in Ash Soot Factor	Mean-Environment-Device
Ashtray	Ashtray	Ashtray		Mean-Environment-Device
Asic	Asic	ASIC	Application Specific Integrated Circuit	Mean-Environment-Device
Asin	Asin	Arcsine	Inverse function of sine	Condition-Qualifier
Asr	Asr	ASR	Anti Slip Regulation	Mean-Environment-Device
Assc	Assc	Assistance	assistance. this keyword expresses the availability of an electronic assistance for a component or system	Mean-Environment-Device
Assi	Assi	Assist		Action-PhysicalType
Assid	Assid	Assisted		Condition-Qualifier
Asst	Asst	Assistant	assistant. For example: the electronic brake assistant.	Mean-Environment-Device
Asw	Asw	ASW	Application Software	Mean-Environment-Device
Asymc	Asymc	Asymmetrical		Condition-Qualifier
Asymtry	Asymtry	Asymmetry		Condition-Qualifier
Async	Async	Asynchronous		Condition-Qualifier
At	At	At	Preposition for location, time, presence	Preposition
At1	At	AT	Automatic Transmission	Mean-Environment-Device
Ata	Ata	Ata	American Trucking Associations Diagnostics Bus	Condition-Qualifier
Atan	Atan	Arctangent	The inverse function of the tangent function.	Condition-Qualifier
Atch	Atch	Attach		Action-PhysicalType
Atm	Atm	Atmosphere		Mean-Environment-Device
Atmpt	Atmpt	Attempt		Condition-Qualifier
Atnms	Atnms	Autonomous	independent	Condition-Qualifier
Attn	Attn	Attenuator	The attenuator switch is a part of the power stage which drives the piezo injectors.	Condition-Qualifier
Audio	Audio	Audio		Action-PhysicalType
Australia	Australia	Australia		Condition-Qualifier
Aut	Aut	Automatic		Condition-Qualifier
Auth	Auth	Authorize		Action-PhysicalType
Authent	Authent	Authentication		Condition-Qualifier
Authn	Authn	Authorization		Action-PhysicalType
Autrk	Autrk	Autarkic	self-sustaining	Condition-Qualifier
Aux	Aux	Auxiliary		Condition-Qualifier
Avd	Avd	Avoid		Action-PhysicalType
Avdbl	Avdbl	Avoidable		Condition-Qualifier

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Avl	Avl	Available		Condition-Qualifier
Avrg	Avrg	Average		Condition-Qualifier
Avs	Avs	AVS	Adjustment Value Service	Condition-Qualifier
Awd	Awd	AWD	All Wheel Drive	Mean-Environment-Device
Axis	Axis	Axis	E. g. might be used in Axis1st, Axis2nd, Axis3rd	Mean-Environment-Device
Axle	Axle	Axle		Mean-Environment-Device
Axles	Axles	Axles	Front and rear axle	Mean-Environment-Device
B	B	Magnetic Flux Density	Amount of magnetic flux through Unit Area taken perpendicular to the direction of magnetic flux (B=μH with μ= permeability of the medium and H = magnetic field)	Action-PhysicalType
Back	Back	Back	anatomic human back	Mean-Environment-Device
Backed	Backed	Backed		Condition-Qualifier
Backl	Backl	Backlash	A sudden backward movement or reaction/the play between adjacent movable parts (as in a series of gears)	Condition-Qualifier
Backlite	Backlite	Backlite		Mean-Environment-Device
Backlt	Backlt	Backlit		Condition-Qualifier
Backw	Backw	Backward		Condition-Qualifier
Bal	Bal	Balance		Action-PhysicalType
Ballstc	Ballstc	Ballistic		Condition-Qualifier
Band	Band	Band		Condition-Qualifier
Bar	Bar	Bar	As unit of measure	Mean-Environment-Device
Baro	Baro	Barometric		Condition-Qualifier
Bas	Bas	Base	Pre-processed sensor data, model free, abstraction of the sensor hardware.	Condition-Qualifier
Basc	Basc	Basic		Condition-Qualifier
Batt	Batt	Battery		Mean-Environment-Device
Bay	Bay	Bay	Space , e.g. engine bay, parking bay	Mean-Environment-Device
Bcu	Bcu	BCU	(High Voltage) Battery Control Unit	Mean-Environment-Device
Bd	Bd	Board		Mean-Environment-Device
Bdc	Bdc	BDC	Bottom Dead Center	Condition-Qualifier
Beam	Beam	Beam		Action-PhysicalType
Bel	Bel	Bel	bel (decibel)	Condition-Qualifier
Belgium	Belgium	Belgium		Condition-Qualifier
Bench	Bench	Bench	bench mode. During OBD I / OBD II diagnoses and error management validation and tuning, the bench mode has to be used so as to force some failures.	Mean-Environment-Device
Beta	Beta	Beta	ß, Greek character, offently used for indexing	Condition-Qualifier
Bfr	Bfr	Before		Preposition

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Bgn	Bgn	Begin		Action-PhysicalType
Bhv	Bhv	Behave		Action-PhysicalType
Bhvr	Bhvr	Behaviour		Condition-Qualifier
Bicycle	Bicycle	Bicycle		Mean-Environment-Device
Bif	Bif	Bi-fuel	Bi-fuel	Mean-Environment-Device
Bilnr	Bilnr	Bilinear	e.g. in bilinear interpolation	Condition-Qualifier
Bio	Bio	Biological		Condition-Qualifier
Bit	Bits	Bit	Smallest unit of data that a computer can process and store	Mean-Environment-Device
Bitws	Bitws	Bitwise	Used for bitwise functions, e.g. Bitwise And (&)	Condition-Qualifier
Bkt	Bkt	Bucket	E.g. for excavator or wheel loader applications	Mean-Environment-Device
Bkup	Bkup	Backup		Condition-Qualifier
Blb	Blb	Bulb		Mean-Environment-Device
Bli	Bli	Blind		Condition-Qualifier
Blink	Blink	Blink		Action-PhysicalType
Blk	Blk	Block		Condition-Qualifier
Blkd	Blkd	Blocked		Condition-Qualifier
Blnk	Blnk	Blank	e.g. blank out engine fuel injection, for specifying a gap	Condition-Qualifier
Blow	Blow	Blow		Action-PhysicalType
Blowr	Blowr	Blower		Mean-Environment-Device
Bls	Bls	BLS	Brake Light Switch	Mean-Environment-Device
Blstr	Blstr	Bolster		Mean-Environment-Device
Blt	Blt	Belt	seat belt. this keyword is used to express the seat belt, which is a kind of security rope to keep the passenger in the seat in case of an accident.	Mean-Environment-Device
Blueprint	Blueprint	Blueprint		Condition-Qualifier
Bmp	Bmp	Bumper		Mean-Environment-Device
Bms	Bms	BMS	Battery Management System	Mean-Environment-Device
Bnce	Bnce	Bounce		Mean-Environment-Device
Bncg	Bncg	Bouncing		Condition-Qualifier
Bnd	Bnd	Bound		Condition-Qualifier
Bndls	Bndls	Boundless		Condition-Qualifier
Bnk	Bnk	Bank		Mean-Environment-Device
Body	Body	Body		Mean-Environment-Device
Bol	Bol	Begin of live	Beginning of Life - e.g. in the context of battery: System state of a battery shortly after manufacturing	Condition-Qualifier
Bolt	Bolt	Bolt	e. g. as in Lock Bolt	Mean-Environment-Device

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Boost	Boost	Boost	A general expression to increase performance for a short period of time or to give a higher output. e.g. to give high voltage output or to support the combustion engine with additional torque (e.g. with a Integrated Starter Generator) for acceleration	Action-PhysicalType
Boot	Boot	Boot		Condition-Qualifier
Botm	Botm	Bottom		Mean-Environment-Device
Box	Box	Box		Mean-Environment-Device
Bp	Bp	Bandpass		Condition-Qualifier
Bpil	Bpil	Bpil	B-Pillar of a car	Mean-Environment-Device
Brazil	Brazil	Brazil		Condition-Qualifier
Brdg	Brdg	Bridge	bridge contact	Condition-Qualifier
Break	Break	Break		Action-PhysicalType
Brg	Brg	Bearing	e.g. in compressor bearing, engine bearing	Mean-Environment-Device
Bri	Bri	Brightness		Action-PhysicalType
Brk	Brk	Brake		Mean-Environment-Device
Brkg	Brkg	Braking		Action-PhysicalType
Brkn	Brkn	Broken		Condition-Qualifier
Brkt	Brkt	Bracket		Mean-Environment-Device
Brn	Brn	Burn		Action-PhysicalType
Brng	Brng	Burning		Action-PhysicalType
Brnr	Brnr	Burner		Mean-Environment-Device
Bsw	Bsw	BSW	Basic Software	Mean-Environment-Device
Btn	Btn	Button		Mean-Environment-Device
Buc	Buc	Buckle	Part of supplement restraint system to protect occupant in case of crash	Mean-Environment-Device
Bucd	Bucd	Buckled	Buckling of occupant	Condition-Qualifier
Buf	Buf	Buffer		Condition-Qualifier
Burst	Burst	Burst	To emerge or spring suddenly or Describes, that a signal is considered as a sequence of events. Burst is an established term in signal processing	Condition-Qualifier
Bus	Bus	Bus	used for the vehicle and not as a data bus	Mean-Environment-Device
Busy	Busy	Busy		Condition-Qualifier
Buz	Buz	Buzzer		Condition-Qualifier
Bw	Bw	Bandwidth		Condition-Qualifier
By	By	By	Preposition	Preposition
Byp	Byp	Bypass		Action-PhysicalType
Byp1	Byp	Bypass		Mean-Environment-Device
Cache	Cache	Cache	Memory for temporary storage of data for fast access	Mean-Environment-Device
Cal	Cal	Calibration		Condition-Qualifier

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Calc	Calc	Calculate		Action-PhysicalType
Calcd	Calcd	Calculated		Condition-Qualifier
Calcn	Calcn	Calculation		Action-PhysicalType
Call	Call	Call		Condition-Qualifier
Calrc	Calrc	Caloric		Condition-Qualifier
Camr	Camr	Camera	A device that records an image of an object	Mean-Environment-Device
Camsft	Camsft	Camshaft		Mean-Environment-Device
Can	Can	CAN	Controller Area Network	Mean-Environment-Device
Canada	Canada	Canada		Condition-Qualifier
Canng	Canng	Canning	e.g: Catalyst Canning	Mean-Environment-Device
Cap	Cap	Capacitor	Capacitor used to store energy in an electric field (AUTOSAR BSW)	Mean-Environment-Device
Car	Car	Car		Mean-Environment-Device
Carb	Carb	CARB	Californian Air Resource Board	Condition-Qualifier
Carbon	Carbon	Carbon		Mean-Environment-Device
Case	Case	Case		Mean-Environment-Device
Cat	Cat	Catalyst		Mean-Environment-Device
Cavtn	Cavtn	Cavitation		Mean-Environment-Device
Cbk	Cbk	Callback	Callback + Callback functions	Action-PhysicalType
Cbl	Cbl	Cable	An assembly of electrical conductors insulated from each other but laid up together (as by being twisted around a central core), for example in electrical cable	Mean-Environment-Device
Cbn	Cbn	Cabin		Mean-Environment-Device
Cco	Cco	Cylinder cut-off	Deactivation of single cylinders	Mean-Environment-Device
Ccp	Ccp	CCP	CAN Calibration Protocol	Mean-Environment-Device
Cd	Cd	Candela	Unit of measure of luminous intensity	Mean-Environment-Device
Cdn	Cdn	Condition		Condition-Qualifier
Cdng	Cdng	Conditioning		Action-PhysicalType
Cdnr	Cdnr	Conditioner		Mean-Environment-Device
Cds	Cds	Coast Down Shift	A Coast Down Shift is an automatic transmission behavior where the gearbox shifts to a lower gear while the vehicle is decelerating (coasting), without the driver pressing the accelerator.	Condition-Qualifier
Cell	Cell	Cell		Mean-Environment-Device
Cen	Cen	Central		Condition-Qualifier
Centi	Centi	Centi		Mean-Environment-Device
Centr	Centr	Center		Condition-Qualifier
Centrfgl	Centrfgl	Centrifugal		Condition-Qualifier
Ceramic	Ceramic	Ceramic		Mean-Environment-Device
Cfg	Cfg	Configuration		Condition-Qualifier

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Cfgd	Cfgd	Configured		Condition-Qualifier
Cfm	Cfm	Confirmation		Condition-Qualifier
Cfmd	Cfmd	Confirmed		Condition-Qualifier
Cgrd	Cgrd	Centigrade		Mean-Environment-Device
Ch	Ch	Channel	This keyword is used to express a channel. A channel is a defined port (of a bundle of ports) of a data line.	Condition-Qualifier
Chain	Chain	Chain		Mean-Environment-Device
Chartc	Chartc	Characteristic		Condition-Qualifier
Chassis	Chassis	Chassis		Mean-Environment-Device
Chd	Chd	Child		Mean-Environment-Device
Chemcl	Chem	Chemical	Chemical in "chemical catalyst heating"	Condition-Qualifier
Chg	Chg	Change		Action-PhysicalType
Chill	Chill	Chill		Condition-Qualifier
Chillr	Chillr	Chiller		Mean-Environment-Device
Chimer	Chimer	Chimer		Mean-Environment-Device
China	China	China		Condition-Qualifier
Chirp	Chirp	Chirp		Action-PhysicalType
Chk	Chk	Check		Condition-Qualifier
Chks	Chks	Checksum		Condition-Qualifier
Chmb	Chmb	Chamber		Mean-Environment-Device
Choke	Choke	Choke	Choking, choke line. this keyword is used to describe the appearance of choking effects in an (air-)stream, e.g. at the choke line of a turbocharger (which is a characteristic parameter)	Mean-Environment-Device
Chrg	Chrg	Charge		Condition-Qualifier
Chrgn	Chrgn	Charging		Action-PhysicalType
Chrgr	Chrgr	Charger		Mean-Environment-Device
Cigarette	Cigarette	Cigarette		Mean-Environment-Device
Cilc	Cilc	Cylinder-Individual Lambda Control	Cylinder-Individual Lambda Control	Mean-Environment-Device
Circ	Circ	Circuit		Mean-Environment-Device
Circum	Circum	Circumference	Used in the meaning of the wheel circumference	Mean-Environment-Device
Circuml	Circuml	Circumferential		Condition-Qualifier
City	City	City		Condition-Qualifier
Class	Class	Class		Condition-Qualifier
Classn	Classn	Classification		Action-PhysicalType
Claw	Claw	Claw	e.g. claw couplings are used within automated transmission instead of clutches	Mean-Environment-Device
Clima	Clima	Climatisation		Action-PhysicalType
Climb	Climb	Climb	e.g. climb mode	Condition-Qualifier

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Clk	Clk	Clock		Condition-Qualifier
Cllsn	Cllsn	Collision		Action-PhysicalType
Clmg	Clmg	Claiming	Address Claiming	Action-PhysicalType
Clmp	Clmp	Clamp		Action-PhysicalType
Clmpd	Clmpd	Clamped		Condition-Qualifier
Cln	Cln	Clean		Condition-Qualifier
Clng	Clng	Cleaning		Condition-Qualifier
Clnr	Clnr	Cleaner		Mean-Environment-Device
Clnt	Clnt	Client	A computer in a network that uses the services (such as access to files or shared peripherals) provided by a server	Condition-Qualifier
Clog	Clog	Clog	Similar to dirty, e.g. clog filter.	Condition-Qualifier
Clr	Clr	Clear		Action-PhysicalType
Clrd	Clrd	Cleared		Condition-Qualifier
Cls	Cls	Close		Action-PhysicalType
Clsd	Clsd	Closed		Condition-Qualifier
Clstr	Clstr	Cluster		Mean-Environment-Device
Clu	Clu	Clutch		Mean-Environment-Device
Cmb	Cmb	Combustion		Mean-Environment-Device
Cmd	Cmd	Command		Condition-Qualifier
Cmft	Cmft	Comfort	comfort. this keyword is used to express something as comfortable or convenient	Condition-Qualifier
Cmn	Cmn	Common	common. for common environmental data (stored with all failures of one project).	Condition-Qualifier
Cmng	Cmng	Coming		Action-PhysicalType
Cmp	Cmp	Compensation		Action-PhysicalType
Cmpbl	Cmpbl	Compatible		Condition-Qualifier
Cmpblty	Cmpblty	Compatibility		Condition-Qualifier
Cmpl	Cmpl	Complete		Condition-Qualifier
Cmplx	Cmplx	Complex		Condition-Qualifier
Cmpn	Cmpn	Compression		Condition-Qualifier
Cmpr	Cmpr	Compressor	This keyword is used to express a component producing a compressed (= with raised pressure) working medium, e.g. air, refrigerant.	Mean-Environment-Device
Cmptmt	Cmptmt	Compartment		Mean-Environment-Device
Cmu	Cmu	Commutation		Condition-Qualifier
Cncl	Cncl	Cancel		Action-PhysicalType
Cncld	Cncld	Cancelled		Condition-Qualifier
Cnct	Cnct	Connect		Action-PhysicalType
Cnctn	Cnctn	Connection		Condition-Qualifier
Cnctr	Cnctr	Connector		Mean-Environment-Device
Cndctvty	Cndctvty	Conductivity		Action-PhysicalType

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Cng	Cng	CNG	Compressed Natural Gas	Mean-Environment-Device
Cns	Cns	Consumption		Condition-Qualifier
Cnse	Cnse	Consume		Action-PhysicalType
Cnsr	Cnsr	Consumer		Mean-Environment-Device
Cnt	Cnt	Count		Action-PhysicalType
Cntct	Cntct	Contact	used in the sense of contact or contactor e.g. in "close contact", "contactor open", "contact defect" or "high-voltage contactors"	Mean-Environment-Device
Cntnr	Cntnr	Container		Mean-Environment-Device
Cntr	Cntr	Counter	Also applicable to running indexes.	Condition-Qualifier
Cntxt	Cntxt	Context	e.g. CAN frame context	Condition-Qualifier
Cnv	Cnv	Convert		Action-PhysicalType
Cnvc	Cnvc	Convection		Condition-Qualifier
Cnvn	Cnvn	Conversion		Action-PhysicalType
Cnvr	Cnvr	Converter		Condition-Qualifier
Cnvtb	Cnvtb	Convertible		Condition-Qualifier
Cnvtn	Cnvtn	Convention		Condition-Qualifier
Co	Co	CO	Carbon Monoxide	Mean-Environment-Device
Co2	Co2	CO2	Carbon dioxide, Chemical molecule CO2	Mean-Environment-Device
Coach	Coach	Coach	motor coach, often simply called a bus	Mean-Environment-Device
Coast	Coast	Coasting	situation of vehicle, where the clutch is kept open and the vehicle is moving by its own inertia, e.g. rolling downhill (coast driving state of a vehicle is a typical transmission / clutch function).	Action-PhysicalType
Cod	Cod	Code		Condition-Qualifier
Coeff	Coeff	Coefficient		Condition-Qualifier
Cog	Cog	COG	Center of Gravity	Mean-Environment-Device
Cohrncy	Cohrncy	Coherency		Action-PhysicalType
Cohrnt	Cohrnt	Coherent		Condition-Qualifier
Coil	Coil	Coil		Mean-Environment-Device
Cok	Cok	Coke	Coke, Coking Deposits on devices, which are very often in contact with the burning remainders of carbon based fuels, caused e.g. by incompletely burned fuel. There are special strategies to recognise or to prevent the building of these deposits.	Condition-Qualifier
Col	Col	Column		Condition-Qualifier
Cold	Cold	Cold		Condition-Qualifier
Coll	Coll	Collector		Mean-Environment-Device
Collct	Collct	Collect		Action-PhysicalType

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Collctd	Collctd	Collected		Condition-Qualifier
Collctn	Collctn	Collection		Condition-Qualifier
Com	Com	Communication		Condition-Qualifier
Comp	Comp	Comparison		Condition-Qualifier
Compu	Compu	Computation		Condition-Qualifier
Con	Con	Constant		Condition-Qualifier
Conc	Conc	Concentration		Action-PhysicalType
Conds	Conds	Condense		Action-PhysicalType
Consctv	Consctv	Consecutive		Condition-Qualifier
Consist	Consist	Consist		Action-PhysicalType
Consistnc	Consistnc	Consistence		Condition-Qualifier
Consold	Consold	Consolidated	A consolidated physical value is either the measured physical value or an estimated one.	Action-PhysicalType
Constr	Constr	Constraint		Condition-Qualifier
Contmn	Contmn	Contamination	soiled, stained, corrupted, or infected	Condition-Qualifier
Contns	Contns	Continuous		Condition-Qualifier
Contrb	Contrb	Contribute		Action-PhysicalType
Contrbn	Contrbn	Contribution		Condition-Qualifier
Conv	Conv	Convolution	Complication or intricacy of form, design, or structure	Condition-Qualifier
Cool	Cool	Cool		Mean-Environment-Device
Coolg	Coolg	Cooling	This keyword is used to describe the cooling system e.g. of the engine. A cooling request can be fulfilled by air, coolant, or other cooling systems or refrigerants.	Action-PhysicalType
Coolr	Coolr	Cooler		Mean-Environment-Device
Coolt	Coolt	Coolant		Mean-Environment-Device
Coor	Coor	Coordinate		Condition-Qualifier
Coord	Coord	Coordinated		Condition-Qualifier
Coords	Coords	Coordinates	Numerical values, which describe the position of a point.	Condition-Qualifier
Coorn	Coorn	Coordination		Condition-Qualifier
Coorr	Coorr	Coordinator		Condition-Qualifier
Copy	Copy	Copy		Condition-Qualifier
Core	Core	Core		Condition-Qualifier
Cornr	Cornr	Corner		Mean-Environment-Device
Cornrg	Cornrg	Cornering		Action-PhysicalType
Corr	Corr	Correct		Condition-Qualifier
Corrd	Corrd	Corrected		Condition-Qualifier
Corrl	Corrl	Correlate		Action-PhysicalType
Corrln	Corrln	Correlation		Condition-Qualifier
Corrn	Corrn	Correction		Condition-Qualifier
Cot	Cot	Coated		Condition-Qualifier

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Coulomb	Coulomb	Coulomb	Unit of measure for Electric charge	Mean-Environment-Device
Coupl	Coupl	Coupling	It is the keyword for any linkage of software modules or functionalities or to express a kind of strict connection between two otherwise not linked parts.	Condition-Qualifier
Coupld	Coupld	Coupled		Condition-Qualifier
Couplr	Couplr	Coupler		Mean-Environment-Device
Course	Course	Course		Mean-Environment-Device
Cover	Cover	Cover		Mean-Environment-Device
Cp	Cp	Capacity	this keyword is used to express the Physical Meaning of a capacity (e.g. thermal capacity of oil)	Action-PhysicalType
Cpby	Cpby	Capability		Condition-Qualifier
Cpil	Cpil	Cpil	C-Pillar of a car	Mean-Environment-Device
Cpl	Cpl	Complement		Condition-Qualifier
Cpt	Cpt	Component		Condition-Qualifier
Cpv	Cpv	CPV	Canister Purge Valve	Mean-Environment-Device
Crash	Crash	Crash		Mean-Environment-Device
Crawl	Crawl	Crawl		Action-PhysicalType
Crc	Crc	CRC	Cyclic Redundancy Check	Condition-Qualifier
Crd	Crd	Cardan	A vehicle component for transmitting mechanical power and torque and rotation, usually used to connect other components of a drivetrain that cannot be connected directly because of distance or the need to allow for relative movement between them.	Mean-Environment-Device
Creat	Creat	Create		Action-PhysicalType
Crit	Crit	Critical		Condition-Qualifier
Crk	Crk	Crank		Action-PhysicalType
Crkcs	Crkcs	Crankcase		Mean-Environment-Device
Crksft	Crksft	Crankshaft		Mean-Environment-Device
Crp	Crp	Creep	example: creep mode of automatic transmission	Action-PhysicalType
Crs	Crs	Cruise		Mean-Environment-Device
Crv	Crv	Curve		Mean-Environment-Device
Crvt	Crvt	Curvature		Action-PhysicalType
Crypto	Crypto	Cryptography		Mean-Environment-Device
Cs	Cs	Chip-select	Chip select or slave select is the name of a control line in digital electronics used to select one (or a set) of integrated circuits (commonly called "chips") out of several connected to the same computer bus	Mean-Environment-Device

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Csa	Csa	Current Sense Amplifier	Current sense amplifiers accurately amplify small voltages.	Mean-Environment-Device
Cstr	Cstr	Canister		Mean-Environment-Device
Cta	Cta	Criteria		Condition-Qualifier
Ctde	Ctde	Cathode	The electrode of an electrochemical cell at which reduction occurs.	Mean-Environment-Device
Ctrl	Ctrl	Control		Condition-Qualifier
Ctrld	Ctrld	Controlled		Condition-Qualifier
Ctrlr	Ctrlr	Controller		Mean-Environment-Device
Ctrsft	Ctrsft	Countershaft		Mean-Environment-Device
Ctry	Ctry	Country		Mean-Environment-Device
Cub	Cub	Cubic		Condition-Qualifier
Cubd	Cubd	Cubed		Condition-Qualifier
Cum	Cum	Cumulative	increasing by successive additions	Condition-Qualifier
Cur	Cur	Current		Condition-Qualifier
Curb	Curb	Curb	American English for Kerb	Mean-Environment-Device
Curt	Curt	Curtain		Mean-Environment-Device
Cush	Cush	Cushion		Mean-Environment-Device
Cust	Cust	Customer		Mean-Environment-Device
Cut	Cut	Cut		Action-PhysicalType
Cv	Cv	Commercial Vehicle	A commercial vehicle is any type of motor vehicle used for transporting goods or paying passengers.	Mean-Environment-Device
Cvt	Cvt	CVT	Continuous Variable Transmission	Mean-Environment-Device
Cyc	Cyc	Cycle		Condition-Qualifier
Cyl	Cyl	Cylinder		Mean-Environment-Device
D	D	D	Differential part of a control structure	Condition-Qualifier
Dampg	Dampg	Damping		Condition-Qualifier
Dampr	Dampr	Damper		Mean-Environment-Device
Das	Das	DAS	Driver Assistance System	Mean-Environment-Device
Dash	Dash	Dashpot		Condition-Qualifier
Data	Data	Data		Action-PhysicalType
Date	Date	Date	date, point of time. This keyword is used to refer to a precise moment or a specific date. This can be a "time stamp" or a calendar date as well.	Action-PhysicalType
Day	Day	Day		Mean-Environment-Device
Dayti	Dayti	Daytime		Mean-Environment-Device
Dbg	Dbg	Debug		Condition-Qualifier
Dbl	Dbl	Double		Condition-Qualifier
Dbnd	Dbnd	Deadband		Action-PhysicalType
Dc	Dc	DC	Direct Current/Voltage	Condition-Qualifier

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Dcc	Dcc	DCC	Distance Cruise Control. In case of cruise control on, the distance to the vehicle ahead is measured. If the distance falls below a defined limit, the vehicle speed is reduced automatically.	Mean-Environment-Device
Dcdc	Dcdc	Converter (DCDC)	A DC-to-DC converter is an electronic circuit or electromechanical device that converts a source of direct current (DC) from one voltage level to another. A power inverter, inverter or invertor is a power electronic device or circuitry that changes direct current (DC) to alternating current (AC).	Mean-Environment-Device
Dcha	Dcha	Discharge		Condition-Qualifier
Dcm	Dcm	DCM	Diagnostic Communication Manager	Mean-Environment-Device
Dcop	Dcop	Decoupling	to eliminate an interrelationship	Condition-Qualifier
Dct	Dct	DCT	Dual Clutch Transmission	Mean-Environment-Device
Dcyc	Dcyc	Driving Cycle	A driving cycle is a series of data points representing the speed of a vehicle versus time.	Condition-Qualifier
De	De	Deviation		Condition-Qualifier
Deactvd	Deactvd	Deactivated		Condition-Qualifier
Deactvn	Deactvn	Deactivation		Condition-Qualifier
Deactvt	Deactvt	Deactivate		Action-PhysicalType
Dead	Dead	Dead		Condition-Qualifier
Deb	Deb	Debounce		Action-PhysicalType
Dec	Dec	Decrease	decrease, decrementation. this keyword is used to express the stepwise descent of a value or counter.	Action-PhysicalType
Decel	Decel	Deceleration		Condition-Qualifier
Deci	Deci	Deci	Unit extension use: decibel (DeciBel, display name dB)	Condition-Qualifier
Decod	Decod	Decode		Action-PhysicalType
Deep	Deep	Deep		Condition-Qualifier
Def	Def	Define		Action-PhysicalType
Defd	Defd	Defined		Condition-Qualifier
Definite	Definite	Definite	e.g. definite path from navigation system	Condition-Qualifier
Defn	Defn	Definition		Condition-Qualifier
Defrst	Defrst	Defrost		Action-PhysicalType
Defu	Defu	Defueling	defueling. It shall be used to describe the action of taking fuel out of the tank.	Condition-Qualifier
Deg	Deg	Degree		Mean-Environment-Device

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Degraded	Degraded	Degraded		Condition-Qualifier
Del	Del	Delete		Action-PhysicalType
Delta	Delta	Delta		Condition-Qualifier
Dem	Dem	DEM	Diagnostic Event Manager	Mean-Environment-Device
Den	Den	Deny	Deny, Denied, Not Accepted	Condition-Qualifier
Denox	Denox	DENOX	System Reduction of Nox Emission	Action-PhysicalType
Dens	Dens	Density		Action-PhysicalType
Dep	Dep	Depression		Condition-Qualifier
Deploy	Deploy	Deploy		Action-PhysicalType
Deploymt	Deploymt	Deployment		Condition-Qualifier
Deriv	Deriv	Derive		Action-PhysicalType
Dern	Dern	Derating	In electronics, derating is the operation of a device at less than its rated maximum capability to prolong its life. Typical examples include operations below the maximum power rating, current rating, or voltage rating.	Condition-Qualifier
Des	Des	Desired		Condition-Qualifier
Desat	Desat	Desaturation	The effect of desaturation which, e.g. can occur in electric / electronic /magnetic components.	Condition-Qualifier
Desc	Desc	Descending	to proceed in a sequence or gradation from higher to lower	Condition-Qualifier
Descr	Descr	Description		Condition-Qualifier
Desorpn	Desorpn	Desorption		Condition-Qualifier
Desox	Desox	Desulphurization		Mean-Environment-Device
Dest	Dest	Destination		Condition-Qualifier
Det	Det	Detect		Action-PhysicalType
Detn	Detn	Detection		Action-PhysicalType
Detrm	Detrm	Determined		Condition-Qualifier
Dev	Dev	Device		Mean-Environment-Device
Devlp	Devlp	Develop		Action-PhysicalType
Dew	Dew	Dew	Dew Point	Mean-Environment-Device
Df	Df	Dual-fuel	Engines that can operate using a mixture of two different fuels are called dual fuel engines.	Condition-Qualifier
Dfct	Dfct	Defect		Condition-Qualifier

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Dfi	Dfi	Direct Fuel Injection (DFI)	Direct Fuel Injection is a fuel injection system for internal combustion engines. It injects fuel directly into the combustion chamber, unlike manifold injection systems, which inject fuel into the intake manifold so that it mixes with the incoming airstream before reaching the combustion chamber.	Mean-Environment-Device
Dft	Dft	Default	default	Condition-Qualifier
Dftl	Dftl	Differential	A special transmission, used in drive train context	Mean-Environment-Device
Dht	Dht	Denavit-Hartenberg Transformation	The Denavit-Hartenberg transformation is a systematic method used in robotics to represent the kinematic relationships between joints and links of robotic arms using a set of parameters.	Condition-Qualifier
Di	Di	Disable		Condition-Qualifier
Diag	Diag	Diagnosis		Action-PhysicalType
Diagc	Diagc	Diagnostic		Mean-Environment-Device
Diam	Diam	Diameter	This name describes the component size, e.g. of driving wheels	Condition-Qualifier
Dif	Dif	Difference		Condition-Qualifier
Difsn	Difsn	Diffusion		Condition-Qualifier
Dig	Dig	Digital		Condition-Qualifier
Dil	Dil	Dilution		Condition-Qualifier
Dim	Dim	Dim	used as compound with Light	Action-PhysicalType
Dimension	Dimension	Dimension		Action-PhysicalType
Dio	Dio	DIO	Digital Input/Output	Mean-Environment-Device
Dir	Dir	Direction		Condition-Qualifier
Dire	Dire	Directory		Condition-Qualifier
Dis	Dis	Discrete		Condition-Qualifier
Disa	Disa	Disappear	disappear. used, in order to define a state of failure: "disappeared failure"	Action-PhysicalType
Disad	Disad	Disabled		Action-PhysicalType
Disarm	Disarm	Disarm		Action-PhysicalType
Disarmed	Disarmed	Disarmed		Condition-Qualifier
Discnct	Discnct	Disconnect		Action-PhysicalType
Disengmt	Disengmt	Disengagement		Condition-Qualifier
Disp	Disp	Display	In both meanings, the display (Component) and to display (Characteristic)	Mean-Environment-Device
Dispd	Dispd	Displayed		Condition-Qualifier
Distbn	Distbn	Distribution		Condition-Qualifier
Div	Div	Division		Condition-Qualifier

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Dly	Dly	Delay	Delay, delayed - to express a postponing, hindering, or causing something to occur more slowly than normal	Condition-Qualifier
Dma	Dma	Direct memory access (DMA)	Direct memory access (DMA) is a feature of many computer systems that allows certain hardware subsystems to access main system memory independently of the central processing unit (CPU). [University of Colorado Boulder]	Mean-Environment-Device
Dman	Dman	Deadman's switch	A dead man's switch is a switch that is designed to be activated or deactivated if the human operator becomes incapacitated, such as through abandonment, drowsiness, loss of consciousness, death, or being bodily removed from control. Originally applied to switches on a vehicle or machine, it has since come to be used to describe other intangible uses, as in computer software.	Condition-Qualifier
Dmd	Dmd	Demand		Condition-Qualifier
Dod	Dod	Depth of discharge (DOD)	DoD represents the percentage of the total battery capacity that has already been used.	Action-PhysicalType
Dom	Dom	Domain		Mean-Environment-Device
Done	Done	Done	e.g. plausibility check done	Condition-Qualifier
Door	Door	Door		Mean-Environment-Device
Dpd	Dpd	Dependency	dependency, dependence, dependent	Condition-Qualifier
Dpl	Dpl	Displacement		Condition-Qualifier
Drain	Drn	Drain	To draw off liquid gradually or completely	Condition-Qualifier
Drg	Drg	Drag	resistance, caused by mechanical, aerodynamical of topographical influences, being effective against a movement	Condition-Qualifier
Drift	Drift	Drift		Condition-Qualifier
Droop	Droop	Droop		Condition-Qualifier
Drp	Drp	Drop		Condition-Qualifier
Drv	Drv	Drive	drive (gear engaged)	Condition-Qualifier

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Drvblty	Drvblty	Drivability	The term driveability describes the driver's complex subjective perception of the interactions between himself and the vehicle, e.g. associated to longitudinal acceleration aspects, like, e.g. jerk-free acceleration or deceleration.	Condition-Qualifier
Drvg	Drvg	Driving	Action of driving the vehicle.	Action-PhysicalType
Drvn	Drvn	Driven		Condition-Qualifier
Drvr	Drvr	Driver	Driver of the vehicle.	Mean-Environment-Device
Drvr1	Drvr	Driver	Example: as in device driver.	Mean-Environment-Device
Dry	Dry	Dry	dry. dry is the opposite of lubricated in case of friction	Condition-Qualifier
Ds	Ds	Down-stream	down-stream or down stream or downstream	Condition-Qualifier
Dsgn	Dsgn	Designate		Action-PhysicalType
Dshb	Dshb	Dashboard	can also be used in the sense of instrument cluster, instrument panel	Mean-Environment-Device
Dsk	Dsk	Disk		Mean-Environment-Device
Dsl	Dsl	Diesel		Mean-Environment-Device
Dst	Dst	Distance		Action-PhysicalType
Dstbc	Dstbc	Disturbance		Condition-Qualifier
Dstcn	Dstcn	Distinction	Perceiving someone or something as being not the same or treating sb./st.as separate or different.	Condition-Qualifier
Dstr	Dstr	Destruct		Action-PhysicalType
Dstrn	Dstrn	Destruction		Condition-Qualifier
Dt	Dt	Drivetrain		Mean-Environment-Device
Dtc	Dtc	DTC	Diagnostic Trouble Code	Condition-Qualifier
Dtd	Dtd	DTD	Drivetrain Torque Distribution	Mean-Environment-Device
Dthr	Dthr	Dither	Dither used in control and signal processing	Action-PhysicalType
Dtr	Dtr	Diagnostic Test Result (DTR)	Status produced after diagnostic tests	Condition-Qualifier
Dual	Dual	Dual		Condition-Qualifier
Dummy	Dummy	Dummy		Condition-Qualifier
Dump	Dump	Dump		Action-PhysicalType
Durg	Durg	During		Preposition
Durn	Durn	Duration		Condition-Qualifier
Duty	Duty	Duty		Condition-Qualifier
Dwell	Dwell	Dwell	Dwell, Dwelling	Condition-Qualifier
Dwn	Dwn	Down		Condition-Qualifier
Dwnld	Dwnld	Download		Condition-Qualifier
Dyn	Dyn	Dynamic		Condition-Qualifier
Dyno	Dyno	Dynamometer		Mean-Environment-Device
Easy	Easy	Easy		Condition-Qualifier
Ec	Ec	EC	European Community	Condition-Qualifier

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Eco	Eco	Economy	economy. related to the detection of driver type either an economy/comfort oriented or a sportive/power requesting driver can be distinguished.	Condition-Qualifier
Ecu	Ecu	ECU	Electronic Control Unit	Mean-Environment-Device
Edge	Edge	Edge		Condition-Qualifier
Edit	Edit	Edit		Action-PhysicalType
Eep	Eep	EEPROM	Electrically Erasable Programmable Read-Only Memory	Mean-Environment-Device
Efc	Efc	Effective		Condition-Qualifier
Eff	Eff	Efficiency		Action-PhysicalType
Effort	Effort	Effort		Condition-Qualifier
Egr	Egr	EGR	Exhaust Gas Recirculation	Mean-Environment-Device
Egy	Egy	Energy		Action-PhysicalType
Egypt	Egypt	Egypt		Condition-Qualifier
Ehb	Ehb	EHB	Electro Hydraulic Brake	Mean-Environment-Device
Elec	Elec	Electric	Used in the sense of electric or electrical	Condition-Qualifier
Electc	Electc	Electronic		Condition-Qualifier
Elevn	Elevn	Elevation	Measured angle from horizon	Action-PhysicalType
Elm	Elm	Element		Mean-Environment-Device
Elpd	Elpd	Elapsed		Condition-Qualifier
Eltpnm	Eltpnm	Electropneumatic		Condition-Qualifier
Em	Em	EM	Electrical Machine	Mean-Environment-Device
Emc	Emc	Electromagnetic Compatibility	Electromagnetic compatibility (EMC) is the ability of electrical equipment and systems to function acceptably in their electromagnetic environment, by limiting the unintentional generation, propagation and reception of electromagnetic energy which may cause unwanted effects.	Condition-Qualifier
Emf	Emf	EMF	Electromotive "force" is not a force (measured in newtons) but a potential, or energy per unit of charge, measured in volts.	Action-PhysicalType
Emgy	Emgy	Emergency		Mean-Environment-Device
Emi	Emi	Emission		Mean-Environment-Device
Empc	Empc	Empirical	Originating in or based on observation or experience, like e.g. for expressing empirical model, empirical modelling etc.	Condition-Qualifier
Empty	Empty	Empty		Condition-Qualifier
Emul	Emul	Emulate		Action-PhysicalType

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Emuld	Emuld	Emulated		Condition-Qualifier
Emuln	Emuln	Emulation		Condition-Qualifier
Ena	Ena	Enable		Condition-Qualifier
Enad	Enad	Enabled		Condition-Qualifier
Enag	Enag	Enabling		Condition-Qualifier
Enchmt	Enchmt	Enrichment		Condition-Qualifier
End	End	End		Condition-Qualifier
Enf	Enf	Enforced		Condition-Qualifier
Eng	Eng	Engine		Mean-Environment-Device
Engd	Engd	Engaged		Condition-Qualifier
Engh	Engh	Enough		Condition-Qualifier
England	England	England		Condition-Qualifier
Engmt	Engmt	Engagement	engagement. to express the situation of engaging e.g. a gear from neutral position / transmission gutter until synchronisation performed.	Mean-Environment-Device
Enhd	Enhd	Enhanced		Condition-Qualifier
Enhmt	Enhmt	Enhancement		Condition-Qualifier
Enth	Enth	Enthalpy	The sum of the internal energy of a body or system	Action-PhysicalType
Entr	Entr	Enter		Action-PhysicalType
Entrc	Entrc	Entrance		Mean-Environment-Device
Entry	Entry	Entry		Action-PhysicalType
Env	Env	Environment		Mean-Environment-Device
Envtl	Envtl	Environmental		Condition-Qualifier
Eoi	Eoi	EOI	End of Injection	Condition-Qualifier
Eol	Eol	EOL	End of Line	Condition-Qualifier
Epb	Epb	EPB	Electric Parking Brake	Mean-Environment-Device
Equ	Equ	Equal		Condition-Qualifier
Equid	Equid	Equipped		Condition-Qualifier
Equip	Equip	Equipment		Mean-Environment-Device
Equiv	Equiv	Equivalent		Condition-Qualifier
Erase	Erase	Erase		Action-PhysicalType
Err	Err	Error	information of a defect	Condition-Qualifier
Esc	Esc	ESC	Electronic Stability Control	Mean-Environment-Device
Escl	Escl	Electronic Steering Column Lock (ESCL)	Electrical Steering Column Lock (ESCL) is a device to electrically lock/unlock the steering handle. Steering Column Lock is superior in vehicle theft prevention and passenger safety.	Mean-Environment-Device
Estim	Estim	Estimate		Action-PhysicalType
Estimd	Estimd	Estimated		Condition-Qualifier
Estimn	Estimn	Estimation		Condition-Qualifier
Estimr	Estimr	Estimator		Condition-Qualifier

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Eta	Eta	Eta	Typically used for eta factor, efficiency, factor depending on viscosity	Action-PhysicalType
Etc	Etc	ETC	Electronic Throttle Control	Mean-Environment-Device
Eth	Eth	Ethernet		Mean-Environment-Device
Ethnl	Ethnl	Ethanol	A simple alcohol with the chemical formula C2-H5-OH.	Condition-Qualifier
Ev	Ev	EV	Electric Vehicle	Mean-Environment-Device
Evap	Evap	Evaporative		Condition-Qualifier
Evapr	Evapr	Evaporator		Mean-Environment-Device
Eve	Eve	Event		Condition-Qualifier
Even	Even	Even		Condition-Qualifier
Ever	Ever	Ever		Condition-Qualifier
Evl	Evl	Evaluate		Action-PhysicalType
Evln	Evln	Evaluation		Action-PhysicalType
Evlr	Evlr	Evaluator		Mean-Environment-Device
Exact	Exact	Exact		Condition-Qualifier
Excd	Excd	Exceed		Action-PhysicalType
Excdd	Excdd	Exceeded		Condition-Qualifier
Exch	Exch	Exchange		Action-PhysicalType
Excl	Excl	Exclude		Action-PhysicalType
Exclsn	Exclsn	Exclusion		Action-PhysicalType
Exclv	Exclv	Exclusive	Restricted in distribution, use, or appeal	Condition-Qualifier
Excpn	Excpn	Exception		Condition-Qualifier
Excr	Excr	Excenter		Condition-Qualifier
Exct	Exct	Excitation	the keyword is used to express the technical effect of excitation, e.g. in connection with a current or a voltage in a generator	Action-PhysicalType
Excvr	Excvr	Excavator	Power-operated shovel	Mean-Environment-Device
Exe	Exe	Execute	To carry out fully or put completely into effect	Condition-Qualifier
Exh	Exh	Exhaust	The escape of used gas or vapor from an engine, or the escaped gas.	Condition-Qualifier
Exist	Exist	Exist		Action-PhysicalType
Exotherm	Exotherm	Exothermal		Condition-Qualifier
Exp	Exp	Expansion	This keyword can be used for the expansion phase of the engine cycle.	Condition-Qualifier
Expc	Expc	Expect	Expect, Expected; to consider probable, certain, reasonable, due, necessary, bound in duty, or obligated	Condition-Qualifier
Expi	Expi	Expire	To exceed its period of validity	Condition-Qualifier

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Expl	Expl	Explicit	Fully revealed or expressed without vagueness, implication, or ambiguity; leaving no question as to meaning or intent	Condition-Qualifier
Exprt	Exprt	Export		Action-PhysicalType
Expt	Expt	Experiment	experiment. In contrast to the keyword Test, it is not used to check an existing functionality, but to try something new.	Condition-Qualifier
Ext	Ext	External		Condition-Qualifier
Extd	Extd	Extended		Condition-Qualifier
Extn	Extn	Extend		Action-PhysicalType
Extr	Extr	Exterior		Mean-Environment-Device
Extrpn	Extrpn	Extrapolation		Action-PhysicalType
F	F	Force	describes the physical parameter 'force'	Action-PhysicalType
Fac	Fac	Factor	factor, coefficient	Action-PhysicalType
Facd	Facd	Faced		Condition-Qualifier
Factry	Factry	Factory		Condition-Qualifier
Fade	Fade	Fade		Action-PhysicalType
Fail	Fail	Fail		Condition-Qualifier
Faild	Faild	Failed		Condition-Qualifier
Failr	Failr	Failure		Condition-Qualifier
Fall	Fall	Falling		Condition-Qualifier
Fan	Fan	Fan		Mean-Environment-Device
Far	Far	Far	To be found a long distance away, opposite to near	Condition-Qualifier
Fast	Fast	Fast		Condition-Qualifier
Fav	Fav	Favourable		Condition-Qualifier
Fb	Fb	Feedback		Action-PhysicalType
Fct	Fct	Function	Purpose	Condition-Qualifier
Feat	Feat	Feature	A prominent part or characteristic	Mean-Environment-Device
Fft	Fft	Fast Fourier Transform	A fast Fourier transform (FFT) is an algorithm that computes the discrete Fourier transform (DFT) of a sequence, or its inverse (IDFT). A Fourier transform converts a signal from its original domain (often time or space) to a representation in the frequency domain and vice versa.	Condition-Qualifier
Fid	Fid	FID	Functional Identifier	Condition-Qualifier
Fie	Fie	Field		Condition-Qualifier
Fil	Fil	Filter		Condition-Qualifier
Fild	Fild	Filtered		Condition-Qualifier
File	File	File		Mean-Environment-Device
Fillg	Fillg	Filling		Action-PhysicalType

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Film	Film	Film		Mean-Environment-Device
Fim	Fim	FIM	Function Inhibition Manager	Mean-Environment-Device
Final	Final	Final		Condition-Qualifier
Findr	Findr	Finder		Mean-Environment-Device
Fine	Fine	Fine		Condition-Qualifier
Finshd	Finshd	Finished		Condition-Qualifier
First	First	First		Condition-Qualifier
Fixa	Fixa	Fixation		Action-PhysicalType
Fixd	Fixd	Fixed		Condition-Qualifier
Flank	Flank	Flank		Mean-Environment-Device
Flap	Flap	Flap		Mean-Environment-Device
Flash	Flash	Flash	Flash used as verb in the sense of blinking e.g. of an indicator	Action-PhysicalType
Fld	Fld	Fluid		Mean-Environment-Device
Flexfu	Flexfu	Flex-Fuel	flexible fuel vehicle, flex-fuel engines etc., sometimes also Flex Fuel or FlexFuel	Condition-Qualifier
Flg	Flg	Flag	Flag is a binary value.	Condition-Qualifier
Flo	Flo	Float	A computer number format that (...) represents a wide dynamic range of values by using a floating point.	Condition-Qualifier
Flood	Flood	Flood		Condition-Qualifier
Flow	Flow	Flow		Condition-Qualifier
Flpr	Flpr	Flipper		Mean-Environment-Device
Fls	Fls	Flash	Flash as device used for flash programming	Mean-Environment-Device
Flsg	Flsg	Flashing		Condition-Qualifier
Flt	Flt	Fault		Mean-Environment-Device
Fluc	Fluc	Fluctuation		Condition-Qualifier
Flx	Flx	Flux	Flux like e.g. in physical quantity "magnetic flux"	Action-PhysicalType
Fly	Fly	Fly		Condition-Qualifier
Flybck	Flybck	Flybackconverter	DC/DC converter based on Flyback topology	Mean-Environment-Device
Flywh	Flywh	Flywheel		Mean-Environment-Device
Fm	Fm	FM	Frequency Modulation	Mean-Environment-Device
Fmale	Fmale	Female		Condition-Qualifier
Fmt	Fmt	Format		Condition-Qualifier
Fob	Fob	Fob	Transponder device in the Key	Mean-Environment-Device
Fog	Fog	Fog		Mean-Environment-Device
Fol	Fol	Follower		Condition-Qualifier
Fold	Fold	Fold		Action-PhysicalType
Folw	Folw	Follow		Action-PhysicalType
Footwell	Footwell	Footwell	Space for vehicle occupants legs	Mean-Environment-Device
For	For	For	Preposition	Preposition

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Foulg	Foulg	Fouling	e.g. used in detection of the fouling of the exhaust gas recirculation (EGR) line	Condition-Qualifier
Fr	Fr	Flexray		Mean-Environment-Device
Frd	Frd	Farad	Unit of measure for Electric capacitance	Mean-Environment-Device
Free	Free	Free		Condition-Qualifier
Fric	Fric	Friction	Resistance against a movement	Condition-Qualifier
Frm	Frm	Frame		Condition-Qualifier
Frnt	Frnt	Front		Condition-Qualifier
From	From	From	Preposition	Preposition
Frq	Frq	Frequency		Action-PhysicalType
Frsh	Frsh	Fresh		Condition-Qualifier
Frz	Frz	Freeze		Action-PhysicalType
Frzn	Frzn	Frozen		Condition-Qualifier
Fu	Fu	Fuel		Mean-Environment-Device
Full	Full	Full		Condition-Qualifier
Fuse	Fuse	Fuse		Mean-Environment-Device
Fusn	Fusn	Fusion		Condition-Qualifier
Fw	Fw	Freewheeling		Condition-Qualifier
Fwd	Fwd	Forward		Condition-Qualifier
Gain	Gain	Gain		Condition-Qualifier
Gap	Gap	Gap		Action-PhysicalType
Gas	Gas	Gas		Mean-Environment-Device
Gate	Gate	Gate		Mean-Environment-Device
Gatewy	Gatewy	Gateway		Mean-Environment-Device
Gauss	Gauss	Gaussian	Gaussian functions are often used to represent the probability density function of a normally distributed random variable	Condition-Qualifier
Gbx	Gbx	Gearbox		Mean-Environment-Device
Gdi	Gdi	GDI	Gasoline Direct Injection	Mean-Environment-Device
Gear	Gear	Gear		Mean-Environment-Device
Gen	Gen	Generic		Condition-Qualifier
Gen1	Gen	Generate		Action-PhysicalType
Gend	Gend	Generated		Condition-Qualifier
Generic	Generic	Generic		Condition-Qualifier
Genn	Genn	Generation		Condition-Qualifier
Genr	Genr	Generator		Mean-Environment-Device
Germany	Germany	Germany		Condition-Qualifier
Get	Get	Get		Action-PhysicalType
Giga	Giga	Giga		Mean-Environment-Device
Glass	Glass	Glass		Mean-Environment-Device
Glb	Glb	Global		Condition-Qualifier
Glv	Glv	Glove		Mean-Environment-Device
Glw	Glw	Glow		Mean-Environment-Device
Gnd	Gnd	Ground		Mean-Environment-Device

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Govr	Govr	Governor		Mean-Environment-Device
Gpio	Gpio	General Purpose Input/Output (GPIO)	A general-purpose input/ output (GPIO) is an uncommitted digital signal pin on an integrated circuit or electronic circuit (e.g. MCUs/ MPUs) board that can be used as an input or output, or both, and is controllable by software.	Mean-Environment-Device
Gpt	Gpt	GPT	General Purpose Timer	Mean-Environment-Device
Gr	Gr	Gram	Unit of measure for mass	Mean-Environment-Device
Grd	Grd	Grade		Mean-Environment-Device
Grdt	Grdt	Gradient	gradient/slope	Action-PhysicalType
Grid	Grid	Grid		Mean-Environment-Device
Grip	Grip	Grip	example: Maximum desired engine torque for grip adaptation	Action-PhysicalType
Group	Group	Group		Condition-Qualifier
Grvy	Grvy	Gravity	This keyword describes the force of attraction between to masses, well known as gravity.	Mean-Environment-Device
Gsl	Gsl	Gasoline	This keyword is used to express the fuel type "gasoline".	Mean-Environment-Device
Gsm	Gsm	GSM	Global System for Mobile Communications	Mean-Environment-Device
Gulf	Gulf	Gulf		Condition-Qualifier
H2	H2	H2	Chemical element Hydrogen diatomic (H2)	Mean-Environment-Device
HC	HC	Hydrocarbon		Preposition
HEV	HEV	Hev	Hybrid Electric Vehicle	Mean-Environment-Device
Half	Half	Half	half, semi-. This keyword is used to characterize something for being only one part of two that make a whole item. Example: the time for a half engine segment.	Condition-Qualifier
Hall	Hall	Hall	as used in Hall Sensor	Mean-Environment-Device
Halt	Halt	Halt		Condition-Qualifier
Hard	Hard	Hard		Condition-Qualifier
Hbrnt	Hbrnt	Hibernate		Action-PhysicalType
Hc	Hc	Hydrocarbon		Mean-Environment-Device
Hd	Hd	Head		Mean-Environment-Device
Hdlamp	Hdlamp	Headlamp		Mean-Environment-Device
Hdphn	Hdphn	Headphone		Mean-Environment-Device
Hdr	Hdr	Header		Condition-Qualifier
Hdrest	Hdrest	Headrest	Headrest of seats	Mean-Environment-Device

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Heal	Heal	Heal	This keyword is needed for functions that try to heal a failure. E.g. an actuator or a control valve is blocked due to a dirt particle. A free-shaking control signal can lead to releasing the particle from the actuator.	Action-PhysicalType
Heat	Heat	Heat		Mean-Environment-Device
Heatd	Heatd	Heated		Condition-Qualifier
Heatg	Heatg	Heating		Action-PhysicalType
Heatr	Heatr	Heater		Mean-Environment-Device
Hecto	Hecto	Hecto	Unit of measure for pressure	Mean-Environment-Device
Hei	Hei	Height		Action-PhysicalType
Hev	Hev	HEV	Hybrid Electric Vehicle	Mean-Environment-Device
Hi	Hi	High		Condition-Qualifier
Hil	Hil	Hardware in the Loop	Hardware in the Loop for the testing of embedded systems	Condition-Qualifier
Hill	Hill	Hill		Mean-Environment-Device
Hist	Hist	History	for data stored in history memory	Condition-Qualifier
Hitch	Hitch	Hitch		Action-PhysicalType
Hld	Hld	Hold		Action-PhysicalType
Hmi	Hmi	HMI	Human-Machine Interface	Mean-Environment-Device
Hndbrk	Hndbrk	Handbrake		Mean-Environment-Device
Hndl	Hndl	Handle	The physical device for opening a door	Mean-Environment-Device
Hndlg	Hndlg	Handling		Mean-Environment-Device
Hom	Hom	Homogeneous	e. g. used in homogeneous injection	Condition-Qualifier
Home	Home	Home		Mean-Environment-Device
Hood	Hood	Hood	e. g. engine hood	Mean-Environment-Device
Hoop	Hoop	Hoop		Mean-Environment-Device
Horn	Horn	Horn		Mean-Environment-Device
Hot	Hot	Hot		Condition-Qualifier
Hozl	Hozl	Horizontal		Condition-Qualifier
Hp	Hp	Highpass		Condition-Qualifier
Hr	Hr	Hour	Unit of measure for time	Mean-Environment-Device
Htl	Htl	Hotel		Mean-Environment-Device
Hud	Hud	HUD	Head Up Display. This keyword is used to express the feature of show information right in the field of view of the driver, e.g. by mirroring it onto the wind screen.	Mean-Environment-Device
Hum	Hum	Humidity	the keyword is used to assign the amount of water in a gas or a gaseous mixture. Example: the humidity in the exhaust gas.	Condition-Qualifier

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Hvac	Hvac	HVAC	Heating Ventilation and Air Conditioning	Mean-Environment-Device
Hvy	Hvy	Heavy		Condition-Qualifier
Hw	Hw	Hardware	This keyword is used to express a relationship to Hardware within a functionality or an environmental tools, e.g. Labcar.	Mean-Environment-Device
Hyb	Hyb	Hybrid	Any vehicle that combines two or more sources of power that can directly or indirectly provide propulsion power is a hybrid	Condition-Qualifier
Hyd	Hyd	Hydraulic	used to characterize hydraulic components	Condition-Qualifier
Hyddyn	Hyddyn	Hydrodynamic		Condition-Qualifier
Hys	Hys	Hysteresis		Condition-Qualifier
Hz	Hz	Hertz	Unit of measure for frequency	Mean-Environment-Device
Hzrd	Hzrd	Hazard		Mean-Environment-Device
I	I	I	Electric current	Action-PhysicalType
Ice	Ice	Ice	This keyword expresses an interrelation with ice or with an icy environment, e.g. a functionality to improve the drivability on frosted streets.	Mean-Environment-Device
Ico	Ico	Intercooler		Mean-Environment-Device
Icu	Icu	ICU	Input Capture Unit	Mean-Environment-Device
Id	Id	Identifier		Mean-Environment-Device
Ideal	Ideal	Ideal	e.g. ideal computed vehicle speed for curve	Condition-Qualifier
Identcl	Identcl	Identical		Condition-Qualifier
Idle	Idle	Idle		Condition-Qualifier
Idn	Idn	Identification		Condition-Qualifier
Idx	Idx	Index		Condition-Qualifier
Idx0	0	0	Index 0. This keyword is used to express the number zero in form of an index	Index
Idx0th	0th	0th	This keyword is used to express the order in form of an index.	Index
Idx1	1	1	Index 1. This keyword is used to express the number one in form of an index	Index
Idx1d	1d	1d	index 1D. used for location of oxygen sensors (4 banks and 2 sensors) - Parameter Identification 1D according with ISO15031-5. 2 banks and 4 sensors are identified by the index 13	Index
Idx1st	1st	1st	This keyword is used to express the order in form of an index.	Index

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Idx2	2	2	Index 2. This keyword is used to express the number two in form of an index	Index
Idx2nd	2nd	2nd	Second. This keyword is used to express the order in form of an index.	Index
Idx3	3	3	Index 3. This keyword is used to express the number three in form of an index	Index
Idx3rd	3rd	3rd	This keyword is used to express the order in form of an index.	Index
Idx4	4	4	Index 4. This keyword is used to express the number four in form of an index	Index
Idx4th	4th	4th	This keyword is used to express the order in form of an index.	Index
Idx5	5	5	Index 5. This keyword is used to express the number five in form of an index	Index
Idx5th	5th	5th	This keyword is used to express the order in form of an index.	Index
Idx6	6	6	Index 6. This keyword is used to express the number six in form of an index	Index
Idx6th	6th	6th	This keyword is used to express the order in form of an index.	Index
Idx7	7	7	Index 7. This keyword is used to express the number seven in form of an index	Index
Idx7th	7th	7th	This keyword is used to express the order in form of an index.	Index
Idx8	8	8	Index 8. This keyword is used to express the number eight in form of an index	Index
Idx8th	8th	8th	This keyword is used to express the order in form of an index.	Index
Idx9	9	9	Index 9. This keyword is used to express the number nine in form of an index	Index
Idx9th	9th	9th	This keyword is used to express the order in form of an index.	Index
If	If	Interface		Condition-Qualifier
Ign	Ign	Ignition		Mean-Environment-Device
Illmn	Illmn	Illumination		Action-PhysicalType
Imdt	Imdt	Immediate	this keyword is used to assign a requirement of minimum delay time or the need to take action as soon as possible.	Condition-Qualifier
Img	Img	Image		Mean-Environment-Device

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Imob	Imob	Immobilizer		Mean-Environment-Device
Imobn	Imobn	Immobilisation	Immobilisation	Action-PhysicalType
Imp	Imp	Impulse		Mean-Environment-Device
Impct	Impct	Impact		Action-PhysicalType
Impo	Impo	Impossible		Condition-Qualifier
Imprt	Imprt	Import		Action-PhysicalType
Imps	Imps	Implausible		Condition-Qualifier
Impy	Impy	Implausibility		Condition-Qualifier
In	In	In	Preposition for location or condition	Preposition
Inacrt	Inacrt	Inaccurate		Condition-Qualifier
Inc	Inc	Increase	This keyword is used to express a stepwise accretion of a value or counter	Action-PhysicalType
Incl	Incl	Incline		Action-PhysicalType
Inclg	Inclg	Including		Action-PhysicalType
Incln	Incln	Inclination		Action-PhysicalType
Inct	Inct	Increment	This keyword is used to express a stepwise accretion of a value or counter	Condition-Qualifier
Ind	Ind	Individual		Condition-Qualifier
Indcd	Indcd	Indicated		Condition-Qualifier
Indcn	Indcn	Indication		Condition-Qualifier
Indcr	Indcr	Indicator		Mean-Environment-Device
India	India	India		Condition-Qualifier
Indonesia	Indonesia	Indonesia		Condition-Qualifier
Indt	Indt	Induction	The process by which an electrical conductor becomes electrified when near a charged body, by which a magnetizable body becomes magnetized when in a magnetic field or in the magnetic flux set up by a magnetomotive force, or by which an electromotive force is produced in a circuit by varying the magnetic field linked with the circuit.	Condition-Qualifier
Indu	Indu	Inductive		Condition-Qualifier
Inert	Inert	Inert		Condition-Qualifier
Info	Info	Information		Condition-Qualifier
Infra	Infra	Infra	e.g. in infra-red, infra-structure, infra-sound, infrasonic sound	Condition-Qualifier
Inhb	Inhb	Inhibit		Action-PhysicalType
Inhbn	Inhbn	Inhibition		Action-PhysicalType
Ini	Ini	Initial		Condition-Qualifier
Inin	Inin	Initialization		Condition-Qualifier
Inj	Inj	Injection		Action-PhysicalType
Injr	Injr	Injector		Mean-Environment-Device
Innr	Innr	Inner		Condition-Qualifier

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Inp	Inp	Input		Condition-Qualifier
Insd	Insd	Inside		Mean-Environment-Device
Insert	Insert	Insert		Action-PhysicalType
Inst	Inst	Instance	This keyword is used for the characterisation e.g. of a diagnosis, instance of the intake air temperature diagnosis	Condition-Qualifier
Int	Int	Internal		Condition-Qualifier
Intd	Intd	Intended		Condition-Qualifier
Inten	Inten	Intensity		Action-PhysicalType
Inter	Inter	Intermediate		Condition-Qualifier
Interdom	Interdom	Interdomain		Mean-Environment-Device
Intery	Intery	Intermediary		Condition-Qualifier
Intgl	Intgl	Integral		Condition-Qualifier
Intgr	Intgr	Integer		Condition-Qualifier
Intk	Intk	Intake	Keyword Intake will also be used in the sense of Inlet	Mean-Environment-Device
Intl	Intl	Interval		Condition-Qualifier
Intm	Intm	Intermittent		Condition-Qualifier
Intr	Intr	Interior		Mean-Environment-Device
Intrpn	Intrpn	Interpolation		Action-PhysicalType
Intrpt	Intrpt	Interrupt		Action-PhysicalType
Intrsn	Intrsn	Intrusion	The act of wrongfully entering a protected area	Condition-Qualifier
Intv	Intv	Intervention		Action-PhysicalType
Intvg	Intvg	Intervening		Condition-Qualifier
Intxcn	Intxcn	Intoxication	User under influence of drugs (e.g. alcohol)	Condition-Qualifier
Invld	Invld	Invalid		Condition-Qualifier
Io	Io	IO	Input/Output	Mean-Environment-Device
Ion	Ion	Ion	as used in Lithium Ion Battery	Mean-Environment-Device
Irrv	Irrv	Irreversible	this keyword is used to express that something can not return to a previous status, e.g. in Cruise Control deactivation it means in this case that cruise can not be activated again in the same driving cycle once a deactivation occurred	Condition-Qualifier
Italy	Italy	Italy		Condition-Qualifier
Itp	Itp	Interpreter	For the network layer of KWP2000	Condition-Qualifier
Ivs	Ivs	Inverse		Condition-Qualifier
Ivt	Ivt	Invert		Action-PhysicalType
Ivtr	Ivtr	Inverter	e.g. inverter in electric cars	Mean-Environment-Device

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
J	J	Inertia	this keyword is used to express the physical effect that the mass of a body is interacting against its acceleration	Action-PhysicalType
Jam	Jam	Jam		Action-PhysicalType
Japan	Japan	Japan		Condition-Qualifier
Jerk	Jerk	Jerk		Mean-Environment-Device
Jmp	Jmp	Jump		Action-PhysicalType
Jou	Jou	Joule	Unit of measure for energy, work	Mean-Environment-Device
Kd	Kd	Kickdown		Mean-Environment-Device
Keep	Keep	Keep	e.g. for active Safety Systems like Lane Keep Aid	Action-PhysicalType
Kelvin	Kelvin	Kelvin	Unit of measure for temperature	Mean-Environment-Device
Key	Key	Key		Mean-Environment-Device
Keyls	Keyls	Keyless		Condition-Qualifier
Keyword	Keyword	Keyword		Mean-Environment-Device
Ki	Ki	Ki	integral gain, Integral amplification, used in software control loops e.g. damper control or lambda control	Action-PhysicalType
Kilo	Kilo	Kilo		Mean-Environment-Device
Knee	Knee	Knee	knee of human body	Mean-Environment-Device
Knk	Knk	Knock		Mean-Environment-Device
Knob	Knob	Knob		Mean-Environment-Device
Korea	Korea	Korea		Condition-Qualifier
Kp	Kp	Kp	proportional gain, proportional amplification, used in software control loops e.g. damper control or lambda control	Action-PhysicalType
L	L	Liter	unit of measure for volume	Mean-Environment-Device
Lam	Lam	Lambda	Air/Fuel Ratio	Action-PhysicalType
Lamp	Lamp	Lamp	this keyword is used do describe the component of a lamp, e.g. in the display of the instrument panel	Mean-Environment-Device
Lane	Lane	Lane		Mean-Environment-Device
Language	Language	Language		Mean-Environment-Device
Lat	Lat	Lateral		Condition-Qualifier
Lati	Lati	Latitude	e.g. position from navigation system	Condition-Qualifier
Ldn	Ldn	Loudness		Action-PhysicalType
Le	Le	Left		Condition-Qualifier
Lead	Lead	Lead	e.g. lead cylinder	Mean-Environment-Device

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Leak	Leak	Leak	The power stage for the piezo injectors has a special hardware (leakage compensation) to avoid discharging of capacities.	Condition-Qualifier
Lean	Lean	Lean		Condition-Qualifier
Led	Led	LED	Light Emitting Diode	Mean-Environment-Device
Len	Len	Length		Action-PhysicalType
Lft	Lft	Lift	Movement from a lower to a higher position, or the distance of the movement.	Condition-Qualifier
Lgt	Lgt	Longitudinal	longitudinal	Condition-Qualifier
Li	Li	Light		Mean-Environment-Device
Lib	Lib	Library		Condition-Qualifier
Lid	Lid	Lid		Mean-Environment-Device
Lighter	Lighter	Lighter		Mean-Environment-Device
Lim	Lim	Limit		Condition-Qualifier
Limd	Limd	Limited		Condition-Qualifier
Limn	Limn	Limitation		Action-PhysicalType
Limp	Limp	Limp	example: Timeout camshaft sensor limp home	Action-PhysicalType
Limr	Limr	Limiter		Mean-Environment-Device
Lin	Lin	LIN	Local Interconnect Network	Mean-Environment-Device
Line	Line	Line		Mean-Environment-Device
List	List	List		Mean-Environment-Device
Lnch	Lnch	Launch		Action-PhysicalType
Lnr	Lnr	Linear	used for description of linear qualities	Condition-Qualifier
Lo	Lo	Low		Condition-Qualifier
Load	Load	Load	Unit must be percent	Action-PhysicalType
Lobe	Lobe	Lobe	as used in the case of single lobe cam	Mean-Environment-Device
Lock	Lock	Lock		Action-PhysicalType
Lockd	Lockd	Locked		Condition-Qualifier
Lockg	Lockg	Locking		Action-PhysicalType
Locn	Locn	Location	Position	Action-PhysicalType
Log	Log	Logarithmic	Logarithmic, Logarithm (AUTOSAR BSW)	Condition-Qualifier
Logl	Logl	Logical		Condition-Qualifier
Long	Long	Long		Condition-Qualifier
Longi	Longi	Longitude	e.g. position from navigation system	Condition-Qualifier
Loop	Loop	Loop		Mean-Environment-Device
Loss	Loss	Loss		Condition-Qualifier
Lost	Lost	Lost		Condition-Qualifier
Lowr	Lowr	Lower		Condition-Qualifier
Lp	Lp	Lowpass		Condition-Qualifier
Lpg	Lpg	LPG	Liquified Petroleum Gas	Mean-Environment-Device
Lrg	Lrg	Large		Condition-Qualifier

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Lrn	Lrn	Learn		Action-PhysicalType
Lrng	Lrng	Learning		Condition-Qualifier
Lst	Lst	Last		Condition-Qualifier
Ltch	Ltch	Latch		Mean-Environment-Device
Ltcy	Ltcy	Latency		Condition-Qualifier
Ltg	Ltg	Lighting		Condition-Qualifier
Lum	Lum	Lumbar		Mean-Environment-Device
Lumi	Lumi	Lumi	Luminous Intensity, used for measuring luminous Intensity	Action-PhysicalType
Lvl	Lvl	Level		Action-PhysicalType
Lvlg	Lvlg	Leveling		Action-PhysicalType
Lvng	Lvng	Leaving		Action-PhysicalType
Lvr	Lvr	Lever		Mean-Environment-Device
M	M	Mass		Action-PhysicalType
Mac	Mac	Machine		Mean-Environment-Device
Mag	Mag	Magnetic		Condition-Qualifier
Mai	Mai	Main		Condition-Qualifier
Malaysia	Malaysia	Malaysia		Condition-Qualifier
Male	Male	Male		Condition-Qualifier
Man	Man	Manual		Condition-Qualifier
Map	Map	Map		Mean-Environment-Device
Mask	Mask	Mask	this keyword is used to characterize a masking of a certain value to separate it from the non-masked one.	Condition-Qualifier
Massg	Massg	Massage		Action-PhysicalType
Mat	Mat	Mathematical		Condition-Qualifier
Max	Max	Maximum		Condition-Qualifier
Mcu	Mcu	MCU	Microcontroller Unit	Mean-Environment-Device
Mdl	Mdl	Model		Mean-Environment-Device
Meas	Meas	Measure		Condition-Qualifier
Measd	Measd	Measured		Condition-Qualifier
Mec	Mec	Mechanic		Condition-Qualifier
Mecl	Mecl	Mechanical		Condition-Qualifier
Med	Med	Medium		Condition-Qualifier
Mega	Mega	Mega		Mean-Environment-Device
Mem	Mem	Memory		Mean-Environment-Device
Memd	Memd	Memorized		Condition-Qualifier
Memr	Memr	Memorise		Action-PhysicalType
Method	Method	Method		Condition-Qualifier
Mexico	Mexico	Mexico		Condition-Qualifier
Mgn	Mgn	Magnitude		Condition-Qualifier
Mgr	Mgr	Manager		Condition-Qualifier
Micro	Micro	Micro		Mean-Environment-Device
Micro1	Micro	Micro		Condition-Qualifier
Mid	Mid	Middle		Condition-Qualifier
Mil	Mil	MIL	Malfunction Indication Lamp	Mean-Environment-Device

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Milg	Milg	Mileage		Action-PhysicalType
Milli	Milli	Milli		Mean-Environment-Device
Min	Min	Minimum		Condition-Qualifier
Mins	Mins	Minute	Unit of measure for time	Mean-Environment-Device
Mirr	Mirr	Mirror		Mean-Environment-Device
Misc	Misc	Miscellaneous		Condition-Qualifier
Misf	Misf	Misfire		Action-PhysicalType
Misfg	Misfg	Misfiring		Condition-Qualifier
Miss	Miss	Missing		Condition-Qualifier
Mkr	Mkr	Marker		Condition-Qualifier
Mkt	Mkt	Market	This keyword is used to express market in the meaning of delivery areas, e.g. to characterize deviances in a component or function due to different customer market requirements.	Mean-Environment-Device
Mmed	Mmed	Multimedia		Mean-Environment-Device
Mnfld	Mnfld	Manifold		Mean-Environment-Device
Mng	Mng	Manage		Action-PhysicalType
Mngt	Mngt	Management		Condition-Qualifier
Mntd	Mntd	Mounted		Condition-Qualifier
Mntg	Mntg	Mounting		Action-PhysicalType
Mntn	Mntn	Maintenance		Condition-Qualifier
Mod	Mod	Mode		Condition-Qualifier
Modln	Modln	Modulation		Action-PhysicalType
Mol	Mol	Mole	is used as unit for the amount of substance	Mean-Environment-Device
Mom	Mom	Moment		Mean-Environment-Device
Momtm	Momtm	Momentum		Action-PhysicalType
Mon	Mon	Monitoring		Action-PhysicalType
Monr	Monr	Monitor		Mean-Environment-Device
Mot	Mot	Motor		Mean-Environment-Device
Motorcycle	Motorcycle	Motorcycle		Mean-Environment-Device
Move	Move	Move		Action-PhysicalType
Movg	Movg	Moving		Action-PhysicalType
Movmt	Movmt	Movement		Condition-Qualifier
Mpg	Mpg	Mapping		Action-PhysicalType
Mpl	Mpl	Multiple		Condition-Qualifier
Msg	Msg	Message		Condition-Qualifier
Msr	Msr	MSR	Engine drag torque control (also known in german language as "Motorschlepp-momentregelung" from which the acronym is derived).	Condition-Qualifier
Mst	Mst	Master		Condition-Qualifier
Mt	Mt	MT	Manual shifted Transmission	Mean-Environment-Device
Mtgtn	Mtgtn	Mitigation		Condition-Qualifier

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Mth	Mth	Month		Mean-Environment-Device
Mtn	Mtn	Motion		Action-PhysicalType
Mtr	Mtr	Meter		Mean-Environment-Device
Mtrac	Mtrac	Memory access tracer	Tracer to log memory accesses and other system events to help analyze and understand the memory access patterns and cache line behavior of code.	Condition-Qualifier
Mtrc	Mtrc	Metric		Condition-Qualifier
Mtrx	Mtrx	Matrix		Condition-Qualifier
Multi	Multi	Multi		Condition-Qualifier
Mux	Mux	Multiplexer		Mean-Environment-Device
N	N	Speed	Rotatory speed, also referred to as speed of rotation measured in rad/s, rpm or equivalents. Examples: EngN = Engine speed [rpm], WhlN = Wheel speed [rpm]	Action-PhysicalType
N2	N2	Nitrogen	Chemical element Nitrogen molecule (N2)	Mean-Environment-Device
Nano	Nano	Nano		Mean-Environment-Device
Nat	Nat	Natural		Condition-Qualifier
Nav	Nav	Navigation		Mean-Environment-Device
Near	Near	Near		Condition-Qualifier
Neg	Neg	Negative		Condition-Qualifier
Negn	Negn	Negation		Condition-Qualifier
Net	Net	Network	describes all kind of networks in general, e. g. an electric network consisting of any type of consumers and generators	Mean-Environment-Device
Neut	Neut	Neutral		Condition-Qualifier
New	New	New		Condition-Qualifier
Night	Night	Night		Mean-Environment-Device
Nlnr	Nlnr	Nonlinear	nonlinear. used for description of non linear characteristics (e.g. for an optional additional nonlinear correction)	Condition-Qualifier
Nm	Nm	NM	Network Management	Mean-Environment-Device
No	No	No	Opposite of "yes"	Preposition
Nod	Nod	Node	keyword for a junction or joining of lines or ways	Mean-Environment-Device
Noise	Noise	Noise		Condition-Qualifier
Nom	Nom	Nominal		Condition-Qualifier

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Norm	Norm	Normal	this keyword is used to describe something as being in the most common mode, e.g. a state machine has a state normal when running properly without disturbance. Do not mix it with Basic for description of fulfilling a basic or minimum requirement.	Condition-Qualifier
Not	Not	Not		Preposition
Notif	Notif	Notification		Condition-Qualifier
Now	Now	Now	in this moment	Condition-Qualifier
Nox	Nox	NOx	Nitrogen oxides, part of the emissions of a combustion engine.	Mean-Environment-Device
Noz	Noz	Nozzle		Mean-Environment-Device
Nr	Nr	Number	this keyword is used to express a numbering in physical meaning	Action-PhysicalType
Nrdc	Nrdc	Nordic		Condition-Qualifier
Nv	Nv	Nv	Non volatile	Condition-Qualifier
Nvm	Nvm	NVM	Non Volatile Memory	Mean-Environment-Device
Nwt	Nwt	Newton		Mean-Environment-Device
O2	O2	O2	Oxygen molecule	Mean-Environment-Device
Obd	Obd	OBD	On Board Diagnosis	Mean-Environment-Device
Obj	Obj	Object		Mean-Environment-Device
Obs	Obs	Observe		Action-PhysicalType
Obslt	Obslt	Obsolete		Condition-Qualifier
Obstcl	Obstcl	Obstacle		Mean-Environment-Device
Obstrcn	Obstrcn	Obstruction		Condition-Qualifier
Obsvr	Obsvr	Observer		Mean-Environment-Device
Occpt	Occpt	Occupant		Mean-Environment-Device
Odd	Odd	Odd		Condition-Qualifier
Of	Of	Of	Preposition	Preposition
Off	Off	Off		Condition-Qualifier
Offroad	Offroad	Offroad		Condition-Qualifier
Offs	Offs	Offset		Action-PhysicalType
Ohm	Ohm	Ohm	Unit of measure for electric resistance	Mean-Environment-Device
Oil	Oil	Oil		Mean-Environment-Device
Ok	Ok	OK		Condition-Qualifier
Old	Old	Old	in the calculation period n, the keyword OLD assigns a value from the period (n-1).	Condition-Qualifier
On	On	On		Preposition
On1	On	On		Condition-Qualifier
Only	Only	Only		Preposition
Open	Open	Open		Action-PhysicalType
Opend	Opend	Opened		Condition-Qualifier
Opener	Opener	Opener		Mean-Environment-Device

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Oper	Oper	Operation		Condition-Qualifier
Opt	Opt	Optional		Condition-Qualifier
Optic	Optic	Optical		Condition-Qualifier
Optm	Optm	Optimum		Condition-Qualifier
Optn	Optn	Option		Condition-Qualifier
Or	Or	Or		Preposition
Orntn	Orntn	Orientation		Action-PhysicalType
Os	Os	OS	Operating System	Mean-Environment-Device
Osc	Osc	Oscillate		Action-PhysicalType
Oscn	Oscn	Oscillation	This keyword is used to describe occurrences of pulsation or oscillation	Condition-Qualifier
Out	Out	Out		Preposition
Outd	Outd	Outside	Used for outdoor and outside	Mean-Environment-Device
Outl	Outl	Outlet	Outlet (e.g. camshaft)	Condition-Qualifier
Outp	Outp	Output		Condition-Qualifier
Outr	Outr	Outer		Condition-Qualifier
Over	Over	Over		Condition-Qualifier
Ovf	Ovf	Overflow		Condition-Qualifier
Ovld	Ovld	Overload	The keyword is used for an electrical actuator, if the requested power exceeds the allowed long-time power limit. The overload must be possible for a certain time.	Condition-Qualifier
Ovrd	Ovrd	Override		Action-PhysicalType
Ovrdn	Ovrdn	Overridden		Condition-Qualifier
Ovrheatd	Ovrheatd	Overheated		Condition-Qualifier
Ovrheatg	Ovrheatg	Overheating		Action-PhysicalType
Ovrrn	Ovrrn	Overrun	Exceeding a certain limit	Condition-Qualifier
Ovs	Ovs	Oversteering	Understeer and oversteer are vehicle dynamics terms used to describe the sensitivity of the vehicle to changes in steering angle ... in any given steady-state maneuver beyond the Ackermann steer angle. The Ackermann steer angle is the steer angle at which the vehicle would travel about a curve when there is no lateral acceleration required (at negligibly low speed). [Wikipedia] Oversteering is when the rear of the vehicle slides outward, causing the vehicle to turn more sharply than intended.	Condition-Qualifier
Own	Own	Own		Condition-Qualifier
Oxd	Oxd	Oxidation		Condition-Qualifier
P	P	Pressure		Action-PhysicalType
Pa	Pa	Pascal	Unit of measure for pressure	Mean-Environment-Device

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Pad	Pad	Pad		Mean-Environment-Device
Paddl	Paddl	Paddle		Mean-Environment-Device
Page	Page	Page		Condition-Qualifier
Pah	Pah	Path		Mean-Environment-Device
Pan	Pan	Panel		Mean-Environment-Device
Panic	Panic	Panic		Action-PhysicalType
Par	Par	Parity		Condition-Qualifier
Parent	Parent	Parent	e.g. "Parent street segment" used in context of route guidance from navigation system	Condition-Qualifier
Park	Park	Park		Action-PhysicalType
Part	Part	Part		Condition-Qualifier
Pas	Pas	Passive		Condition-Qualifier
Pase	Pase	PASE	Passive Start and Entry (PASE), keyless access.	Mean-Environment-Device
Pass	Pass	Passenger		Mean-Environment-Device
Passd	Passd	Passed		Condition-Qualifier
Pat	Pat	Pattern		Condition-Qualifier
Pck	Pck	package, pack	Hierarchical element, representing a collection of coherent entities/ functionalities/instances.	Condition-Qualifier
Pdu	Pdu	PDU	Protocol Data Unit	Mean-Environment-Device
Peak	Peak	Peak		Condition-Qualifier
Ped	Ped	Pedestrian		Mean-Environment-Device
Pedl	Pedl	Pedal		Mean-Environment-Device
Pelv	Pelv	Pelvis	pelvis of human body	Mean-Environment-Device
Pen	Pen	Personalization		Condition-Qualifier
Per	Per	Per	Preposition	Preposition
Perc	Perc	Percent		Action-PhysicalType
Perd	Perd	Period		Condition-Qualifier
Perml	Perml	Permille		Condition-Qualifier
Permtn	Permtn	Permutation		Condition-Qualifier
Perox	Perox	Peroxide		Mean-Environment-Device
Pfi	Pfi	PFI	Port Fuel Injection	Mean-Environment-Device
Pha	Pha	Phase		Condition-Qualifier
Phev	Phev	PHEV	Plug-in Hybrid Electric Vehicle	Mean-Environment-Device
Philippine	Philippine	Philippine		Condition-Qualifier
Phy	Phy	Physical		Condition-Qualifier
Pi	Pi	Pi		Condition-Qualifier
Pico	Pico	Pico		Mean-Environment-Device
Pid	Pid	PID	Proportional Integral Derivative, as widely used in "PID Controller".	Condition-Qualifier
Pin	Pin	Pin		Mean-Environment-Device
Pinion	Pinion	Pinion		Mean-Environment-Device
Pipe	Pipe	Pipe		Mean-Environment-Device

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Pist	Pist	Piston	central component in a combustion engine. It serves to transform the gas pressure within the cylinder into a movement. The keyword is used e.g. to express the name for piston temperature	Mean-Environment-Device
Pitch	Pitch	Pitch		Action-PhysicalType
Pitchovr	Pitchovr	Pitchover		Action-PhysicalType
Pla	Pla	Plant		Mean-Environment-Device
Plate	Plate	Plate		Mean-Environment-Device
Plaus	Plaus	Plausible		Condition-Qualifier
Pll	Pll	PLL	Phase Locked Loop	Mean-Environment-Device
Plow	Plow	Plow	for example for snow plow, etc.	Mean-Environment-Device
Pls	Pls	Pulse	a pulse is a simple electrical signal with a defined shape	Mean-Environment-Device
Pm	Pm	Particulate matter	Microscopic particles of solid or liquid matter suspended in the air.	Condition-Qualifier
Pmp	Pmp	Pump		Mean-Environment-Device
Pnd	Pnd	Pending	This keyword is used to build the "pending" fault used in the error management aggregate defined by the ISO 15031-5.5 service $07.	Condition-Qualifier
Pnm	Pnm	Pneumatic	used e.g. for description of pneumatic actuators (such as recirculation valve actuator or waste gate actuator)	Condition-Qualifier
Polarity	Polarity	Polarity		Condition-Qualifier
Pole	Pole	Pole	e.g. magnetic pole	Condition-Qualifier
Polling	Polling	Polling		Action-PhysicalType
Pool	Pool	Pool		Mean-Environment-Device
Port	Port	Port		Condition-Qualifier
Pos	Pos	Positive		Condition-Qualifier
Posn	Posn	Position		Action-PhysicalType
Posng	Posng	Positioning		Action-PhysicalType
Post	Post	Post		Preposition
Poti	Poti	Potentiometer		Mean-Environment-Device
Ppty	Ppty	Property		Condition-Qualifier
Pr	Pr	Pr	Prandtl number	Action-PhysicalType
Pre	Pre	Pre	used to express any kind of pre-processing like e.g. for pre-injection	Preposition
Prec	Prec	Precharge		Condition-Qualifier
Pred	Pred	Predicted		Condition-Qualifier
Prel	Prel	Preliminary		Condition-Qualifier
Prem	Prem	Premature		Condition-Qualifier
Prep	Prep	Prepare		Action-PhysicalType

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Prepn	Prepn	Preparation	this characteristic is used to express a general status of preparation, e.g. processing of certain tasks before an activation of a certain component or functionality	Condition-Qualifier
Pres	Pres	Presence		Condition-Qualifier
Prev	Prev	Previous		Condition-Qualifier
Prevn	Prevn	Prevention		Condition-Qualifier
Prfmnc	Prfmnc	Performance		Condition-Qualifier
Prim	Prim	Primary		Condition-Qualifier
Prio	Prio	Priority		Condition-Qualifier
Prj	Prj	Project		Condition-Qualifier
Prjn	Prjn	Projection		Condition-Qualifier
Prkg	Prkg	Parking		Mean-Environment-Device
Prm	Prm	Parameter		Condition-Qualifier
Prmnt	Prmnt	Permanent		Condition-Qualifier
Probl	Probl	Probable	e.g. probable path from nagivation system	Condition-Qualifier
Problty	Problty	Probability		Condition-Qualifier
Proc	Proc	Process		Action-PhysicalType
Prof	Prof	Profile		Action-PhysicalType
Prog	Prog	Program		Condition-Qualifier
Progd	Progd	Programmed		Condition-Qualifier
Progm	Progm	Programming		Condition-Qualifier
Progs	Progs	Progress		Condition-Qualifier
Progsn	Progsn	Progression		Condition-Qualifier
Prohb	Prohb	prohibit		Action-PhysicalType
Prop	Prop	Proportional		Condition-Qualifier
Prot	Prot	Protect		Action-PhysicalType
Protd	Protd	Protected		Condition-Qualifier
Protn	Protn	Protection		Action-PhysicalType
Prototype	Prototype	Prototype		Condition-Qualifier
Prp	Prp	Propulsive	opposite of drag resistance. Notion of positive forces generated by a system. Example: for a car, we can list internal combustion engine + starter/generator + electrical engine	Condition-Qualifier
Prpn	Prpn	Propulsion		Condition-Qualifier
Prsnl	Prsnl	Personal		Condition-Qualifier
Prsnt	Prsnt	Present		Condition-Qualifier
Psbl	Psbl	Possible		Condition-Qualifier
Psd	Psd	Pressed		Condition-Qualifier
Pt	Pt	Powertrain		Mean-Environment-Device
Ptc	Ptc	PTC	Power Train Controller	Mean-Environment-Device

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Ptco	Ptco	Positive temperature coefficient (heater)	Positive temperature coefficient (heater): An electronic element which provides temperature dependend resistance, the higher the temperature the higher the resistance. This is used for heating purpose.	Mean-Environment-Device
Ptcol	Ptcol	Protocol	Set of rules and guidelines, e.g. for communication purpose, like, e.g. CAN communication, Charging communication	Mean-Environment-Device
Ptl	Ptl	Partial	this keyword expresses the characterization of "only partly, not completely"	Condition-Qualifier
Ptr	Ptr	Pointer		Condition-Qualifier
Ptra	Ptra	Pointer to array	A pointer is a programming language object whose value refers directly to (or points to) another value stored elsewhere in the computer memory using its address (in this case: address of an array)	Condition-Qualifier
Puddle	Puddle	Puddle		Mean-Environment-Device
Pull	Pull	Pull		Action-PhysicalType
Purfr	Purfr	Purifier	Filter which cleans the air used in an HVAC of an vehicle	Mean-Environment-Device
Purg	Purg	Purge		Action-PhysicalType
Push	Push	Push	push (start), towing, power on driving, press. keyword for push or active towing (e.g. if a vehicle has to be started by towing, a button is pressed, if there is positive torque inside powertrain)	Action-PhysicalType
Put	Put	Put		Action-PhysicalType
Pwd	Pwd	Password		Condition-Qualifier
Pwm	Pwm	PWM	Pulse Width Modulation	Condition-Qualifier
Pwr	Pwr	Power		Action-PhysicalType
Pwr1	Pwr	Power		Mean-Environment-Device
Pyro	Pyro	Pyrometer	this keyword is used to express a relationship to a pyrometer which is a contactless instrument to measure high temperatures	Mean-Environment-Device
Q	Q	Q	Electric charge	Action-PhysicalType
Qlfr	Qlfr	Qualifier		Condition-Qualifier
Qly	Qly	Quality	This keyword is used to express the word 'quality'	Condition-Qualifier
Qnty	Qnty	Quantity		Condition-Qualifier
Query	Query	Query		Condition-Qualifier
Quick	Quick	Quick		Condition-Qualifier

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Quo	Quo	Quotient		Condition-Qualifier
R	R	Resistance	Unit must be Ohm	Action-PhysicalType
Rad	Rad	Radian	As unit of measure for angles	Mean-Environment-Device
Radio	Radio	Radio		Mean-Environment-Device
Radr	Radr	Radiator	a radiator is a component, that senses out heat. It can be found e.g. in the cooling circuit of a car.	Mean-Environment-Device
Rail	Rail	Rail	Examples: Common Rail, Fuel Pressure Rail. The common rail is a core component in some fuel system configurations (esp. diesel). It is a high pressure fuel container mostly in form of a tube and located close to the engine.	Mean-Environment-Device
Rain	Rain	Rain		Mean-Environment-Device
Ram	Ram	RAM	Random Access Memory	Mean-Environment-Device
Ramp	Ramp	Ramp		Condition-Qualifier
Rat	Rat	Ratio	RATIO is used to describe a value gained from a proportion of other values. The unit of RATIO must be [-] (no unit).	Action-PhysicalType
Rate	Rate	Rate	this keyword is used to express a rate in physical meaning	Action-PhysicalType
Raw	Raw	Raw	Unprocessed sensor data.	Condition-Qualifier
Rbd	Rbd	Rebound		Condition-Qualifier
Rbf	Rbf	Radial Basis Function	In the field of mathematical modeling, a radial basis function network is an artificial neural network that uses radial basis functions as activation functions for have many uses, including function approximation, time series prediction, classification, and system control.	Condition-Qualifier
Rcnd	Rcnd	Recognized		Condition-Qualifier
Rcrltn	Rcrltn	Ricirculation		Condition-Qualifier
Rcv	Rcv	Recovery	A healing or an upturn, e.g. as after a depression	Action-PhysicalType
Rd	Rd	Radius		Action-PhysicalType
Rdnt	Rdnt	Redundant		Condition-Qualifier
Rdr	Rdr	Radar		Mean-Environment-Device
Rdy	Rdy	Ready	readiness identification	Condition-Qualifier
Re	Re	Rear		Condition-Qualifier
Re1	Re	Re	Reynolds Number	Mean-Environment-Device
Reactn	Reactn	Reaction		Action-PhysicalType
Reacvn	Reacvn	Reactivation		Condition-Qualifier
Read	Read	Read		Action-PhysicalType

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Reading	Reading	Reading		Action-PhysicalType
Real	Real	Real	Real part	Condition-Qualifier
Rec	Rec	Record		Condition-Qualifier
Recalcn	Recalcn	Recalculation		Action-PhysicalType
Recall	Recall	Recall		Condition-Qualifier
Recirc	Recirc	Recirculation		Action-PhysicalType
Recpr	Recpr	Reciprocal		Condition-Qualifier
Red	Red	Reduce		Condition-Qualifier
Redn	Redn	Reduction		Condition-Qualifier
Ref	Ref	Reference		Condition-Qualifier
Refgnt	Refgnt	Refrigerant	A refrigerant is a substance or mixture, usually a fluid, used in a heat pump and refrigeration cycle.	Condition-Qualifier
Refrsh	Refrsh	Refresh		Action-PhysicalType
Refu	Refu	Refuel		Action-PhysicalType
Regn	Regn	Region		Condition-Qualifier
Reinf	Reinf	Reinforce		Action-PhysicalType
Rejctd	Rejctd	Rejected		Condition-Qualifier
Rel	Rel	Relative		Condition-Qualifier
Reld	Reld	Released		Condition-Qualifier
Rels	Rels	Release		Action-PhysicalType
Relzd	Relzd	Realized		Condition-Qualifier
Rem	Rem	Remote		Condition-Qualifier
Rep	Rep	Repeat		Action-PhysicalType
Replcmt	Replcmt	Replacement		Condition-Qualifier
Repn	Repn	Repetition		Condition-Qualifier
Reprog	Reprog	Reprogramming	Software update	Action-PhysicalType
Req	Req	Request		Condition-Qualifier
Reqd	Reqd	Requested		Condition-Qualifier
Res	Res	Result		Condition-Qualifier
Resd	Resd	Reserved		Condition-Qualifier
Resi	Resi	Residual		Condition-Qualifier
Resl	Resl	Resolution		Condition-Qualifier
Resp	Resp	Response		Condition-Qualifier
Rest	Rest	Rest		Condition-Qualifier
Restore	Restore	Restore		Action-PhysicalType
Restrctn	Restrctn	Restriction		Action-PhysicalType
Restrnt	Restrnt	Restraint		Mean-Environment-Device
Resu	Resu	Resume		Condition-Qualifier
Resv	Resv	Reserve		Condition-Qualifier
Rev	Rev	Revolution	As unit of measure	Mean-Environment-Device
Rex	Rex	REX	Range Extender	Mean-Environment-Device

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Rgl	Rgl	Regular	this keyword is used to describe e.g. a process as being in a normal status in sense of correctness. It is behaving like wanted (it behaves like wanted to treat a disturbance), it is not out of order.	Condition-Qualifier
Rgln	Rgln	Regulation		Condition-Qualifier
Rgn	Rgn	Regeneration		Condition-Qualifier
Rho	Rho	Rho	density	Action-PhysicalType
Ri	Ri	Right	right	Condition-Qualifier
Rich	Rich	Rich		Condition-Qualifier
Ride	Ride	Ride		Action-PhysicalType
Rise	Rise	Rise		Action-PhysicalType
Risk	Risk	Risk		Condition-Qualifier
Risng	Risng	Rising		Condition-Qualifier
Rke	Rke	RKE	Remote Keyless Entry	Mean-Environment-Device
Rlv	Rlv	Relevant		Condition-Qualifier
Rlvc	Rlvc	Relevance		Condition-Qualifier
Rly	Rly	Relay		Mean-Environment-Device
Rmn	Rmn	Reminder		Mean-Environment-Device
Rnd	Rnd	Random		Condition-Qualifier
Rng	Rng	Range		Action-PhysicalType
Road	Road	Road	this keyword is used to express the conditions a street can apply to the vehicle	Mean-Environment-Device
Roll	Roll	Roll		Action-PhysicalType
Rollbar	Rollbar	Rollbar	(Active) safety roll bar to protect passengers	Mean-Environment-Device
Rollg	Rollg	Rolling		Action-PhysicalType
Rollovr	Rollovr	Rollover	Info from airbag control unit	Action-PhysicalType
Rollr	Rollr	Roller		Mean-Environment-Device
Rom	Rom	ROM	Read Only Memory	Mean-Environment-Device
Ron	Ron	RON	Research Octane Number	Action-PhysicalType
Roof	Roof	Roof		Mean-Environment-Device
Rot	Rot	Rotation	This keyword is used to express circular motions.	Condition-Qualifier
Rotl	Rotl	Rotational		Condition-Qualifier
Rotor	Rotor	Rotor		Mean-Environment-Device
Roty	Roty	Rotary		Condition-Qualifier
Roughns	Roughns	Roughness		Action-PhysicalType
Route	Route	Route	e.g. used in context of route guidance from navigation system	Condition-Qualifier
Row	Row	Row		Mean-Environment-Device
Rpm	Rpm	RPM	Revolutions per Minute, used as unit of measure of frequency	Mean-Environment-Device

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Rqrd	Rqrd	Required		Condition-Qualifier
Rsa	Rsa	RSA	Republic of South Africa	Condition-Qualifier
Rsc	Rsc	RSC	Roll Stability Control	Mean-Environment-Device
Rst	Rst	Reset		Condition-Qualifier
Rstrt	Rstrt	Restart		Condition-Qualifier
Rt	Rt	Runtime		Condition-Qualifier
Rtd	Rtd	Retard		Condition-Qualifier
Rtdr	Rtdr	Retarder		Condition-Qualifier
Rte	Rte	RTE	Run-Time Environment	Mean-Environment-Device
Rti	Rti	Real Time	In computing and technology, real-time (or Echtzeit in German) describes systems that process data and provide results without noticeable delay, ensuring that outputs are available within a predetermined time window. [Wikipedia]	Condition-Qualifier
Rtmd	Rtmd	Runtime Manipulation Detection	Detection of SW manipulation during runtime, e.g., using Cipher-Based Message Authentication Code (CMAC)	Condition-Qualifier
Rtrctr	Rtrctr	Retractor	Part of supplement restraint system to protect occupant in case of crash	Mean-Environment-Device
Rule	Rule	Rule		Condition-Qualifier
Run	Run	Run		Action-PhysicalType
Runng	Runng	Running		Condition-Qualifier
Rvs	Rvs	Reverse		Condition-Qualifier
Rvsb	Rvsb	Reversible	This keyword is used to express that something can return to a previous status, e.g. in Cruise Control deactivation: It means in this case that cruise can be activated again in the same driving cycle once a deactivation occurred.	Condition-Qualifier
Rvsg	Rvsg	Reversing		Action-PhysicalType
Rx	Rx	Receive		Action-PhysicalType
Rxn	Rxn	Reception		Condition-Qualifier
Rxr	Rxr	Receiver		Mean-Environment-Device
S	S	Siemens	Unit of measure for conductance	Mean-Environment-Device

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Sae	Sae	Society of Automotive Engineers (SAE)	SAE International is a global professional association and standards organization. The scope of its activities beyond automotive engineering and the automotive industry includes aerospace and other transport industries, as well as commercial vehicles including autonomous vehicles such as self-driving cars, trucks, surface vessels, drones, and related technologies.	Condition-Qualifier
Safe	Safe	Safe		Condition-Qualifier
Sail	Sail	Sailing	Situation of vehicle, where the clutch is kept open and the vehicle is moving by its own inertia, e.g. rolling downhill (coast driving state of a vehicle is a typical transmission / clutch function). The engine might be shut of.	Condition-Qualifier
Sale	Sale	Sale		Action-PhysicalType
Sample	Sample	Sample		Condition-Qualifier
Sat	Sat	Satellite	For position data obtained from satellite systems	Mean-Environment-Device
Satn	Satn	Saturation		Action-PhysicalType
Save	Save	Save		Action-PhysicalType
Saved	Saved	Saved		Condition-Qualifier
Sca	Sca	Scale		Condition-Qualifier
Scan	Scan	Scan		Action-PhysicalType
Scanr	Scanr	Scanner		Mean-Environment-Device
Scav	Scav	Scavenging	The process of pushing fresh gas from the inlet into the exhaust by having big overlap of opened inlet and outlet valve, especially at charged engines is called scavenging.	Condition-Qualifier
Scr	Scr	SCR	Selective Catalytic Reduction (System)	Mean-Environment-Device
Scrn	Scrn	Screen		Mean-Environment-Device
Sdl	Sdl	Scheduler		Condition-Qualifier
Seat	Seat	Seat		Mean-Environment-Device
Seatg	Seatg	Seating		Condition-Qualifier
Sec	Sec	Second	Used as "2nd"	Condition-Qualifier
Sec1	Sec	Second	Used as time unit	Mean-Environment-Device
Sector	Sector	Sector		Condition-Qualifier
Secu	Secu	Security		Condition-Qualifier
Seed	Seed	Seed		Condition-Qualifier
Seg	Seg	Segment		Mean-Environment-Device
Segd	Segd	Segmented		Condition-Qualifier

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Sel	Sel	Select		Action-PhysicalType
Seld	Seld	Selected		Condition-Qualifier
Self	Self	Self		Condition-Qualifier
Seln	Seln	Selection	Seln is used to assign e.g. the selection of a certain functionality (one out of several strategies) or component (one of several gears, sensors,..) as well as in simulation to select a single entry of an Array by putting Seln at the end of the name	Condition-Qualifier
Sen	Sen	Sensing		Action-PhysicalType
Sent	Sent	SENT	Single Edge Nibble Transmission (Protocol)	Mean-Environment-Device
Seq	Seq	Sequence		Condition-Qualifier
Seql	Seql	Sequential		Condition-Qualifier
Ser	Ser	Serie		Mean-Environment-Device
Serl	Serl	Serial		Condition-Qualifier
Servo	Servo	Servo		Condition-Qualifier
Session	Session	Session		Condition-Qualifier
Set	Set	Set		Action-PhysicalType
Sev	Sev	Severity		Action-PhysicalType
Sfty	Sfty	Safety		Mean-Environment-Device
Shaft	Shaft	Shaft	a shaft is a rotating rod, carrying out a certain task or transmitting a force in a powertrain. Example: outbox gearbox shaft.	Condition-Qualifier
Shd	Shd	Shield		Condition-Qualifier
Shift	Shift	Shift	Used e.g. in up shift (Shift Up) or down shift (ShiftDwn) or gear shift (GearShift) (synonyms: upshift, downshift, gearshift, UpShift, DownShift etc.).	Action-PhysicalType
Shiftr	Shiftr	Shifter		Mean-Environment-Device
Sho	Sho	Short	short trip	Condition-Qualifier
Shong	Shong	Shortening		Action-PhysicalType
Sht	Sht	Shut	e.g. to shut down; do not introduce a special keyword for the noun "Shutdown" but the content can be expressed using the combination of "Sht" and "Dwn"	Action-PhysicalType
Shttr	Shttr	Shutter		Mean-Environment-Device
Side	Side	Side	This keyword is intended for common consideration of inlet and exhaust side.	Condition-Qualifier
Sig	Sig	Signal		Condition-Qualifier
Sill	Sill	Sill		Mean-Environment-Device
Simd	Simd	Simulated		Condition-Qualifier

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Simn	Simn	Simulation		Condition-Qualifier
Simple	Simple	Simple		Condition-Qualifier
Sin	Sin	Sinus		Condition-Qualifier
Size	Size	Size		Condition-Qualifier
Skip	Skip	Skip		Action-PhysicalType
Slave	Slave	Slave		Condition-Qualifier
Sld	Sld	Slide		Action-PhysicalType
Sldg	Sldg	Sliding		Action-PhysicalType
Sleep	Sleep	Sleep		Condition-Qualifier
Slip	Slip	Slip		Action-PhysicalType
Slop	Slop	Slope		Condition-Qualifier
Slow	Slow	Slow		Condition-Qualifier
Small	Small	Small		Condition-Qualifier
Smk	Smk	Smoke		Mean-Environment-Device
Smotng	Smotng	Smoothing		Action-PhysicalType
Smt	Smt	Smart	more and more electronic components take over tasks that had been done before by a central processing unit. Such components are seen as "intelligent" or "smart", whereas the second expression is the one used more generally.	Condition-Qualifier
Snd	Snd	Send		Action-PhysicalType
Sng	Sng	Single		Condition-Qualifier
Snow	Snow	Snow	for example for snow mode or snow plow, etc.	Condition-Qualifier
Snsr	Snsr	Sensor		Mean-Environment-Device
Snvty	Snvty	Sensitivity		Condition-Qualifier
Soc	Soc	State of Charge (SOC)	Critical aspect, e.g. of battery management systems (BMS), providing insights into the remaining charge of a battery	Condition-Qualifier
Sofc	Sofc	Solid Oxide Fuel Cell	A Solid Oxide Fuel Cell (SOFC) is a high temperature fuel cell. It is highly resistant regarding carbon.	Mean-Environment-Device
Soft	Soft	Soft		Condition-Qualifier
Soi	Soi	SOI	Start of Injection. Unit must be angle	Condition-Qualifier
Soln	Soln	Solenoid		Mean-Environment-Device
Soon	Soon	Soon		Condition-Qualifier
Soot	Soot	Soot	used in the sense of soot or smut	Mean-Environment-Device
Sound	Sound	Sound		Action-PhysicalType
Soundr	Soundr	Sounder		Mean-Environment-Device
Sox	Sox	SOx	Sulphur Oxide	Mean-Environment-Device
Sp	Sp	Setpoint		Condition-Qualifier

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Spare	Spare	Spare		Condition-Qualifier
Spc	Spc	Specific	for specific environmental data (stored individually with particular failures of one project).	Condition-Qualifier
Spcl	Spcl	Special		Condition-Qualifier
Spd	Spd	Speed	Speed only for use cases when neither N for rotatory speeds, nor V for translatory speeds are applicable	Action-PhysicalType
Spi	Spi	SPI	Serial Peripheral Interface	Condition-Qualifier
Spkr	Spkr	Speaker	e.g. loudspeaker	Mean-Environment-Device
Split	Split	Split		Condition-Qualifier
Sply	Sply	Supply		Action-PhysicalType
Spo	Spo	Spontaneous	This keyword characterizes a sudden, unexpected occurrence. E.g. a not pre-arranged catalyst regeneration phase.	Condition-Qualifier
Spol	Spol	Spoiler	A device to modify air flow in order to increase fuel efficiency or improve handling in automobiles.	Mean-Environment-Device
Sppl	Sppl	Supplementary		Condition-Qualifier
Sprg	Sprg	Spring		Mean-Environment-Device
Sprk	Sprk	Spark		Action-PhysicalType
Spt	Spt	Sport	Related to the (short- and long term) detection of driver type either an economic/ comfort oriented or a sportive/ power requesting driver can be distinguished.	Condition-Qualifier
Sq	Sq	Square	the Keyword is used to describe the relationship of a Functional Data to the arithmetic procedure of multiplication of a value with itself.	Condition-Qualifier
Sqd	Sqd	Squared		Condition-Qualifier
Sqrt	Sqrt	Sqrt	square root	Condition-Qualifier
Srad	Srad	Steradian	As a unit of measure for solid angle	Mean-Environment-Device
Src	Src	Source		Condition-Qualifier
Srv	Srv	Service	a service is a processing routine within a certain functionality. This keyword is used to nominate such services.	Condition-Qualifier
Ssm	Ssm	SSM	Stand Still Manager	Mean-Environment-Device

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Ssr	Ssr	Solid State Relay	A solid state relay (SSR) is an electronic switching device that switches on or off when an external voltage (AC or DC) is applied across its control terminals. They serve the same function as an electromechanical relay, but solid-state electronics contain no moving parts and have a longer operational lifetime.	Mean-Environment-Device
St	St	State	Used for both, "state" and "status".	Condition-Qualifier
Stab	Stab	Stable		Condition-Qualifier
Stabn	Stabn	Stabilization		Action-PhysicalType
Stabr	Stabr	Stabilizer		Mean-Environment-Device
Staby	Staby	Stability		Condition-Qualifier
Stall	Stall	Stall		Action-PhysicalType
Stallg	Stallg	Stalling		Condition-Qualifier
Stamp	Stamp	Stamp		Action-PhysicalType
Stand	Stand	Stand		Action-PhysicalType
Standg	Standg	Standing		Action-PhysicalType
Stat	Stat	Static		Condition-Qualifier
Stator	Stator	Stator		Mean-Environment-Device
Staty	Staty	Stationary	This keyword is used to describe a situation, where a movable part is at rest or a certain condition does not change	Condition-Qualifier
Stb	Stb	Standby		Condition-Qualifier
Stc	Stc	Statistic		Condition-Qualifier
Stck	Stck	Stack	An orderly pile or heap	Mean-Environment-Device
Std	Std	Standard	e.g. as in standard deviation	Condition-Qualifier
Stdn	Stdn	Standardization		Condition-Qualifier
Steer	Steer	Steering	This keyword is used to express the steering of a car.	Mean-Environment-Device
Step	Step	Step		Condition-Qualifier
Stfn	Stfn	Stiffness	stiffness is the grade of hardness of a material, e.g. in a damper: it is composed of a spring (stiffness part: force proportional to displacement) and a fluid friction (viscous part: force proportional to speed)	Action-PhysicalType
Stg	Stg	Stage	Examples: Power Stage / Powerstage	Mean-Environment-Device
Stgy	Stgy	Strategy		Condition-Qualifier
Sti	Sti	Stimulation	The stimulation describes the inputs for testing of functions.	Condition-Qualifier
Still	Still	Still		Condition-Qualifier

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Stk	Stk	Stroke		Condition-Qualifier
Stn	Stn	Station	Equipment or place used usually for performing a particular job, e.g. charging station, railway station	Mean-Environment-Device
Stoi	Stoi	Stoichiometric		Condition-Qualifier
Stop	Stop	Stop		Action-PhysicalType
Store	Store	Store		Action-PhysicalType
Storg	Storg	Storage		Action-PhysicalType
Strat	Strat	Stratified	e. g. used in stratified injection	Condition-Qualifier
Strike	Strike	Strike		Action-PhysicalType
Strt	Strt	Start		Action-PhysicalType
Strtd	Strtd	Started		Condition-Qualifier
Strtg	Strtg	Starting		Condition-Qualifier
Strtr	Strtr	Starter		Mean-Environment-Device
Struct	Struct	Structure		Condition-Qualifier
Sts	Sts	Status		Condition-Qualifier
Stuck	Stuck	Stuck	the keyword STUCK is used to express e.g. a "stuck reading" coolant temperature sensor. This is an official expression of the CARB (Californian Air Resources Board) and is used in term of OBD II requirements.	Condition-Qualifier
Su	Su	Sulphur		Mean-Environment-Device
Sub	Sub	Substitute		Condition-Qualifier
Subt	Subt	Subtractive		Condition-Qualifier
Subtyp	Subtyp	Subtype		Condition-Qualifier
Suctn	Suctn	Suction		Action-PhysicalType
Sum	Sum	Sum		Condition-Qualifier
Sun	Sun	Sun		Mean-Environment-Device
Sup	Sup	Super	example: super charger	Preposition
Supp	Supp	Suppress		Action-PhysicalType
Suppsn	Suppsn	Superposition		Action-PhysicalType
Supr	Supr	Superior		Condition-Qualifier
Surf	Surf	Surface	The exterior or upper boundary of an object, fluid or body.	Mean-Environment-Device
Surge	Surge	Surge	surging, surge line. this keyword is used to describe the appearance of surging effects in an (air-)stream, e.g. at the surge line of a turbocharger (which is a characteristic parameter)	Condition-Qualifier
Surrndgs	Surrndgs	Surroundings		Mean-Environment-Device
Susp	Susp	Suspension		Mean-Environment-Device
Suspc	Suspc	Suspicion		Condition-Qualifier
Suv	Suv	SUV	Sport Utility Vehicle	Mean-Environment-Device

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Sw	Sw	Software		Mean-Environment-Device
Swil	Swil	Switchable		Condition-Qualifier
Swirl	Swirl	Swirl		Mean-Environment-Device
Switzerland	Switzerland	Switzerland		Condition-Qualifier
Swt	Swt	Switch		Mean-Environment-Device
Sym	Sym	Symptom		Condition-Qualifier
Symc	Symc	Symmetrical		Condition-Qualifier
Symtry	Symtry	Symmetry		Condition-Qualifier
Sync	Sync	Synchronize		Action-PhysicalType
Syncd	Syncd	Synchronized		Condition-Qualifier
Syncn	Syncn	Synchronization		Condition-Qualifier
Sys	Sys	System		Mean-Environment-Device
T	T	Temperature		Action-PhysicalType
Tacho	Tacho	Tachometer	Tachometer or Speedometer. A component displaying the current speed of the vehicle	Mean-Environment-Device
Tachog	Tachog	Tachograph	Component recording or registering the values of the tachometer or speedometer	Mean-Environment-Device
Taiwan	Taiwan	Taiwan		Condition-Qualifier
Tank	Tank	Tank		Mean-Environment-Device
Tar	Tar	Target		Condition-Qualifier
Tau	Tau	Tau	Time constant of a control system (P, I, D).	Condition-Qualifier
Tbl	Tbl	Table		Condition-Qualifier
Tcs	Tcs	TCS	Traction Control System	Mean-Environment-Device
Tdc	Tdc	TDC	Top Dead Center	Condition-Qualifier
Telgrm	Telgrm	Telegram		Mean-Environment-Device
Telltl	Telltl	Telltale		Mean-Environment-Device
Telm	Telm	Telematics		Mean-Environment-Device
Tensnr	Tensnr	Tensioner		Mean-Environment-Device
Term	Term	Term		Condition-Qualifier
Terminal	Terminal	Terminal		Mean-Environment-Device
Test	Test	Test		Condition-Qualifier
Tgtnss	Tgtnss	Tightness		Action-PhysicalType
Th	Th	Thermostat		Mean-Environment-Device
Thailand	Thailand	Thailand		Condition-Qualifier
Thd	Thd	Threshold		Condition-Qualifier
Theo	Theo	Theoretical		Condition-Qualifier
Therm	Therm	Thermal		Condition-Qualifier
Thermo	Thermo	Thermo		Condition-Qualifier
Thorax	Thorax	Thorax	thorax of human body	Mean-Environment-Device
Thr	Thr	Throttle		Mean-Environment-Device
Thrd	Thrd	Third		Condition-Qualifier
Ti	Ti	Time	Time, duration	Action-PhysicalType
Tilt	Tilt	Tilt		Action-PhysicalType
Tintg	Tintg	Tinting	Action of changing the color of an object	Condition-Qualifier

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Tip	Tip	Tip		Condition-Qualifier
Tire	Tire	Tire		Mean-Environment-Device
Tit	Tit	Titanium		Mean-Environment-Device
Tmp	Tmp	Temporary		Condition-Qualifier
Tmr	Tmr	Timer	this keyword is used to express a timer. A timer is counter for a defined time span.	Condition-Qualifier
Tn	Tn	Tn	time constant for discrete control structures. This keyword is used in control engineering context as time constant for discrete control structures	Condition-Qualifier
To	To	To	toward something,.....	Preposition
Tog	Tog	Toggle		Condition-Qualifier
Tolr	Tolr	Tolerance		Action-PhysicalType
Tooth	Tooth	Tooth		Condition-Qualifier
Top	Top	Top		Mean-Environment-Device
Tors	Tors	Torsion		Condition-Qualifier
Tot	Tot	Total		Condition-Qualifier
Touch	Touch	Touch	To cause to be briefly in contact or conjunction with something, e.g. soft-touch (soft landing)	Action-PhysicalType
Touchd	Touchd	Touched		Condition-Qualifier
Tout	Tout	Tout	time out	Condition-Qualifier
Tow	Tow	Tow		Action-PhysicalType
Tp	Tp	TP	Transport Protocol	Mean-Environment-Device
Tq	Tq	Torque		Action-PhysicalType
Tr	Tr	Trunk		Mean-Environment-Device
Tra	Tra	Transient		Condition-Qualifier
Trac	Trac	Traction		Action-PhysicalType
Tran	Tran	Transition		Condition-Qualifier
Transpt	Transpt	Transport	e.g. to indicate a specific car/ ecu mode (transport mode)	Condition-Qualifier
Trap	Trap	Trap		Action-PhysicalType
Trapg	Trapg	Trapping	In case of scavenging the total fresh gas flowing into the cylinder is bigger than the mass air flow staying in the cylinder. The ratio of the mass air flow in the cylinder to the total mass air flow is called trapping efficiency	Condition-Qualifier
Trb	Trb	Turbine	To describe a turbine (turbine wheel) in general, e.g. in a hydraulic torque converter or in a turbocharger	Mean-Environment-Device
Trbl	Trbl	Turbulence	Flow regime characterized by chaotic stochastic property changes	Condition-Qualifier
Trbo	Trbo	Turbo		Mean-Environment-Device

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Trck	Trck	Tracking, Track	Awareness of a fact, progression, or condition	Condition-Qualifier
Trcv	Trcv	Transceiver		Mean-Environment-Device
Treat	Treat	Treatment		Condition-Qualifier
Trf	Trf	Transfer		Condition-Qualifier
Trig	Trig	Trigger		Condition-Qualifier
Trip	Trip	Trip		Condition-Qualifier
Trlr	Trlr	Trailer	a wheeled but not motorized vehicle which can be pulled by a motorized vehicle	Mean-Environment-Device
Tro	Tro	Trouble		Condition-Qualifier
Trpz	Trpz	Trapezoid, Trapezium	In geometry, a trapezoid is a quadrilateral that has at least one pair of parallel sides.	Condition-Qualifier
Trsm	Trsm	Transmission	In a mechanical context	Mean-Environment-Device
Truck	Truck	Truck		Mean-Environment-Device
Trv	Trv	Transversal	transversal in opposite to longitudinal. e.g. used for acceleration	Condition-Qualifier
Trvl	Trvl	Travel		Action-PhysicalType
Trvld	Trvld	Travelled		Condition-Qualifier
Tube	Tube	Tube	the keyword TUBE expresses any kind of a pipe, for instance the exhaust .	Condition-Qualifier
Tumb	Tumb	Tumble		Condition-Qualifier
Tun	Tun	Tuning		Condition-Qualifier
Turn	Turn	Turn		Action-PhysicalType
Twli	Twli	Twilight		Mean-Environment-Device
Tx	Tx	Transmit	In a communication protocol context	Action-PhysicalType
Txt	Txt	Text		Condition-Qualifier
Typ	Typ	Type		Condition-Qualifier
U	U	Voltage		Action-PhysicalType
Udp	Udp	UDP	User Datagram Protocol	Mean-Environment-Device
Uds	Uds	UDS	Unified Diagnostic Services. Unified Diagnostic Services is a standard for Diagnostics according to ISO14229-1	Mean-Environment-Device
Uego	Uego	UEGO	Universal Exhaust Gas Oxygen Sensor	Mean-Environment-Device
Ukwn	Ukwn	Unknown		Condition-Qualifier
Ults	Ults	UltraSound	Ultrasonic, Ultrasound is sound with frequencies greater than 20 kilohertz.	Condition-Qualifier
Undefd	Undefd	Undefined		Condition-Qualifier
Under	Under	Under		Condition-Qualifier
Unfav	Unfav	Unfavourable		Condition-Qualifier
Unique	Unique	Unique		Condition-Qualifier
Unit	Unit	Unit		Condition-Qualifier

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Univ	Univ	Universal	Including or covering all or a whole collectively or distributively without limit or exception	Condition-Qualifier
Unlck	Unlck	Unlock		Action-PhysicalType
Unlckd	Unlckd	Unlocked		Condition-Qualifier
Unlimd	Unlim	Unlimited		Condition-Qualifier
Up	Up	Up		Action-PhysicalType
Upd	Upd	Update		Action-PhysicalType
Uppr	Uppr	Upper		Condition-Qualifier
Upw	Upw	Upward		Condition-Qualifier
Urb	Urb	Urban		Condition-Qualifier
Urea	Urea	Urea		Mean-Environment-Device
Urg	Urg	Urgent	e.g. like in urgent or urgency signal	Condition-Qualifier
Us	Us	Up-stream	up-stream or up stream or upstream	Condition-Qualifier
Usa	Usa	USA	United States of America	Condition-Qualifier
Use	Use	Use		Action-PhysicalType
Used	Used	Used		Condition-Qualifier
Usg	Usg	Usage		Condition-Qualifier
Usr	Usr	User		Mean-Environment-Device
V	V	Velocity	Translatory speed, also referred to as speed of translation or translational speed. Value at this instant of time, no direction, measured in m/s, km/h or equivalents. Examples: Veh V = Vehicle speed [km/h]	Action-PhysicalType
Vac	Vac	Vacuum		Condition-Qualifier
Val	Val	Value		Condition-Qualifier
Van	Van	Van		Mean-Environment-Device
Vanity	Vanity	Vanity		Mean-Environment-Device
Vap	Vap	Vapor		Condition-Qualifier
Vari	Vari	Variance		Action-PhysicalType
Vcc	Vcc	Vcc	supply voltage	Action-PhysicalType
Vcos	Vcos	Viscosity	Viscosity describes the thickness of a fluid, e.g. in a damper: it is composed of a spring (stiffness part: force proportional to displacement) and a fluid friction (viscous part: force proportional to speed)	Action-PhysicalType
Vcv	Vcv	VCV	Volume Control Valve, e.g. used in "high pressure pump volume control valve".	Mean-Environment-Device
Vect	Vect	Vector		Condition-Qualifier
Veh	Veh	Vehicle		Mean-Environment-Device
Vent	Vent	Vent		Mean-Environment-Device
Ventn	Ventn	Ventilation		Action-PhysicalType

▽

△

Short Name	Abbreviation	Long Name	Description	Classification
Vers	Vers	Version		Condition-Qualifier
Vert	Vert	Vertical		Condition-Qualifier
Vietnam	Vietnam	Vietnam		Condition-Qualifier
Vin	Vin	VIN	Vehicle Identification Number. This keyword is used to describe the vehicle identification number. This is certain code which is unique for every vehicle and is used for its identification.	Mean-Environment-Device
Virt	Virt	Virtual		Condition-Qualifier
Vis	Vis	Visual		Condition-Qualifier
Visy	Visy	Visibility		Condition-Qualifier
Vlc	Vlc	VLC	Vehicle Longitudinal Control	Mean-Environment-Device
Vld	Vld	Valid		Condition-Qualifier
Vldt	Vldt	Validate		Action-PhysicalType
Vlv	Vlv	Valve		Condition-Qualifier
Vol	Vol	Volume		Action-PhysicalType
Volmc	Volmc	Volumetric		Condition-Qualifier
Volt	Volt	Volt	As unit of measure for voltage	Mean-Environment-Device
Vrnt	Vrnt	Variant		Condition-Qualifier
Vvt	Vvt	VVT	Variable Valve Timing	Mean-Environment-Device
W	W	Work	Energy	Action-PhysicalType
Wait	Wait	Wait		Action-PhysicalType
Wall	Wall	Wall		Mean-Environment-Device
Warn	Warn	Warning		Condition-Qualifier
Watt	Watt	Watt	As unit of measure for power	Mean-Environment-Device
Way	Way	Way		Condition-Qualifier
Wdg	Wdg	Watchdog		Mean-Environment-Device
Wear	Wear	Wear		Condition-Qualifier
Wedge	Wedge	Wedge		Mean-Environment-Device
Wet	Wet	Wet	Containing, covered with, or soaked with liquid	Condition-Qualifier
Wght	Wght	Weight		Mean-Environment-Device
Whl	Whl	Wheel		Mean-Environment-Device
Whls	Whls	Wheels		Mean-Environment-Device
Wide	Wide	Wide	keyword is used to specify gear box type, which is distinguished via variant coding (wide ratio: long transmission ratio for lower rpm. closed ratio: sports transmission ratio)	Condition-Qualifier
Width	Width	Width	Used in the sense of width or breadth, e.g. wheel width or wheel breadth	Condition-Qualifier
Win	Win	Window		Mean-Environment-Device
Wind	Wind	Windscreen		Mean-Environment-Device
Wind1	Wind	Wind	wind in the sense of breeze	Mean-Environment-Device
Wipg	Wipg	Wiping		Action-PhysicalType

▽

△

Short Name	*Abbreviation*	*Long Name*	*Description*	*Classification*
Wipr	Wipr	Wiper		Mean-Environment-Device
Wire	Wire	Wire	Electrical connection media	Mean-Environment-Device
Wirels	Wirels	Wireless	Used for Wireless connection	Condition-Qualifier
Wish	Wish	Wish		Action-PhysicalType
Wishd	Wishd	Wished		Condition-Qualifier
With	With	With	preposition	Preposition
Within	Within	Within	preposition	Preposition
Wntr	Wntr	Winter		Condition-Qualifier
Wo	Wo	Without	preposition	Preposition
Word	Word	Word		Condition-Qualifier
World	World	World		Condition-Qualifier
Wr	Wr	Write		Action-PhysicalType
Wrg	Wrg	Wrong	is used to express a mismatch which does not have to be treated by error management diagnosis	Condition-Qualifier
Wrm	Wrm	Warm		Condition-Qualifier
Wrpr	Wrpr	Wrapper		Mean-Environment-Device
Wrst	Wrst	Worst		Condition-Qualifier
Wshng	Wshng	Washing		Action-PhysicalType
Wshr	Wshr	Washer		Mean-Environment-Device
Wthr	Wthr	Weather		Condition-Qualifier
Wtr	Wtr	Water		Mean-Environment-Device
Xcp	Xcp	XCP	Universal Measurement and Calibration Protocol XCP / extended universal calibration protocol (ASAM standard)	Mean-Environment-Device
Xpos	Xpos	Xposition	X position	Action-PhysicalType
Yaw	Yaw	Yaw		Action-PhysicalType
Yes	Yes	Yes		Condition-Qualifier
Ypos	Ypos	Yposition	Y position	Action-PhysicalType
Yr	Yr	Year		Mean-Environment-Device

Table A.3: Overview AI Specification Keyword Sets