

SDV

智能网联汽车服务接口规范

SDV/T 001-2022

第一部分：原子服务 API 接口

(Version 4 Beta1)

2022-12-28 发布

中国汽车工业协会软件分会
软件定义汽车委员会

目 录

1 前言	21
2 概述	23
2.1 行业现状	23
2.2 SDV 定义的服务软件架构	24
2.3 SDV 标准化 API 接口的意义	24
2.4 SDV API 版本规则和升级兼容说明	25
2.5 原子服务 API 范围	25
2.6 文档内容介绍	26
2.7 文档发布记录	26
3 BCM(车身控制)原子服务 API	27
3.1 BCM_Door(车门服务).....	27
3.1.1 unlock	28
3.1.2 lock	29
3.1.3 superLock	29
3.1.4 getLockStatus	30
3.1.5 notifyLockStatus	30
3.1.6 foldHandle	31
3.1.7 unfoldHandle	31
3.1.8 isHandleFolded	31
3.1.9 notifyHandleFoldedStatus	32
3.1.10 notifyHandleAntiPinch	32
3.1.11 notifyHandleIceBreaking	33
3.1.12 open	33
3.1.13 close	34
3.1.14 stop.....	34
3.1.15 getActuateStatus	35
3.1.16 notifyActuateStatus.....	35
3.1.17 getLearnedStatus.....	35
3.1.18 adjustPosition.....	36
3.1.19 getPosition	36
3.1.20 notifyPosition.....	37

3.1.21 notifyAntiPinch.....	37
3.1.22 notifyIceBreaking	38
3.1.23 getStatus.....	38
3.1.24 notifyStatus	39
3.1.25 lockChildLock	39
3.1.26 unlockChildLock	40
3.1.27 getChildLockStatus.....	40
3.1.28 getDoorType	40
3.2 BCM_Horn(鸣笛服务).....	41
3.2.1 stop.....	41
3.2.2 start	42
3.2.3 alert	42
3.2.4 setVolumeLevel	43
3.2.5 isRunning.....	43
3.2.6 getVolume	44
3.2.7 notifyStatus	44
3.3 BCM_Siren(汽笛服务)	45
3.3.1 stop.....	45
3.3.2 start	46
3.3.3 setVolumeLevel	46
3.3.4 isRunning.....	47
3.3.5 getVolume	47
3.3.6 notifyStatus	47
3.4 BCM_Key(钥匙服务)	48
3.4.1 getBodyAuthStatus	49
3.4.2 getLocation	49
3.4.3 getEngineAuthStatus	50
3.4.4 getKeyNumber.....	50
3.4.5 getKeySerialNo.....	51
3.4.6 notifyKey	51
3.5 BCM_RearView(后视镜服务).....	52
3.5.1 stopHeating.....	53
3.5.2 startHeating.....	53
3.5.3 setHeatingLevel	54
3.5.4 isHeating	54
3.5.5 getHeatingLevel.....	55
3.5.6 notifyHeatingStatus	55
3.5.7 adjustXAngle	56
3.5.8 getXAngle.....	56
3.5.9 startAdjustXAngle	57
3.5.10 stopAdjustXAngle	57
3.5.11 notifyXAngle	58

3.5.12 adjustYAngle.....	58
3.5.13 getYAngle	59
3.5.14 startAdjustYAngle.....	59
3.5.15 stopAdjustYAngle.....	60
3.5.16 notifyYAngle.....	60
3.5.17 unfold.....	61
3.5.18 fold.....	61
3.5.19 isFolded	61
3.5.20 notifyFoldedStatus	62
3.5.21 adjustAntiGlare.....	62
3.5.22 getAntiGlare	63
3.6 BCM_Seat(座椅服务).....	63
3.6.1 adjustMainXDir	65
3.6.2 getMainXDir.....	65
3.6.3 startAdjustMainXDir	66
3.6.4 stopAdjustMainXDir	66
3.6.5 notifyXActuateStatus	67
3.6.6 adjustMainZDir.....	67
3.6.7 getMainZDir	68
3.6.8 startAdjustMainZDir.....	68
3.6.9 stopAdjustMainZDir.....	69
3.6.10 notifyMZActuateStatus.....	69
3.6.11 adjustFrontZDir	70
3.6.12 getFrontZDir	70
3.6.13 startAdjustFrontZDir	71
3.6.14 stopAdjustFrontZDir.....	71
3.6.15 notifyFZActuateStatus	72
3.6.16 adjustBackRestAngle.....	72
3.6.17 getBackRestAngle	73
3.6.18 startAdjustBackRestAngle.....	73
3.6.19 stopAdjustBackRestAngle	74
3.6.20 notifyBackActuateStatus.....	74
3.6.21 isOccupied	74
3.6.22 getOccupiedStatus	75
3.6.23 notifyOccupiedStatus.....	75
3.7 BCM_SeatExtended(座椅扩展服务).....	76
3.7.1 adjustPanLength	78
3.7.2 getPanLength	78
3.7.3 startAdjustPanLength.....	79
3.7.4 stopAdjustPanLength.....	79
3.7.5 adjustPillowZDir.....	80
3.7.6 getPillowZDir	80

3.7.7 startAdjustPillowZDir.....	81
3.7.8 stopAdjustPillowZDir.....	81
3.7.9 adjustLumbarZDir.....	82
3.7.10 getLumbarZDir.....	82
3.7.11 startAdjustLumbarZDir.....	83
3.7.12 stopAdjustLumbarZDir.....	83
3.7.13 adjustLumbarXDir.....	84
3.7.14 getLumbarXDir.....	84
3.7.15 startAdjustLumbarXDir.....	85
3.7.16 stopAdjustLumbarXDir.....	85
3.7.17 adjustLegRestAngle.....	86
3.7.18 getLegRestAngle.....	86
3.7.19 startAdjustLegRestAngle.....	87
3.7.20 stopAdjustLegRestAngle.....	87
3.7.21 adjustLegRestXDir.....	88
3.7.22 getLegRestXDir.....	88
3.7.23 startAdjustLegRestXDir.....	89
3.7.24 stopAdjustLegRestXDir.....	89
3.7.25 adjustFootRestAngle.....	90
3.7.26 getFootRestAngle.....	90
3.7.27 startAdjustFootRestAngle.....	91
3.7.28 stopAdjustFootRestAngle.....	91
3.8 BCM_CushionTemp(座椅通风加热服务).....	92
3.8.1 stopVentilation.....	92
3.8.2 startVentilation.....	93
3.8.3 setVentilationLevel.....	93
3.8.4 isVentilating.....	94
3.8.5 getVentilationLevel.....	94
3.8.6 notifyVentilatingStatus.....	95
3.8.7 startHeating.....	95
3.8.8 stopHeating.....	95
3.8.9 setHeatingLevel.....	96
3.8.10 isHeating.....	96
3.8.11 getHeatingLevel.....	97
3.8.12 notifyHeatingStatus.....	97
3.9 BCM_Message(座椅按摩服务).....	98
3.9.1 stopMassage.....	98
3.9.2 startMassage.....	99
3.10 BCM_SteerWheel(方向盘辅助功能服务).....	99
3.10.1 adjustTilt.....	100
3.10.2 getTilt.....	101
3.10.3 startAdjustTilt.....	101

3.10.4 stopAdjustTilt	102
3.10.5 adjustHeight	102
3.10.6 getHeight	103
3.10.7 startAdjustHeight	103
3.10.8 stopAdjustHeight	104
3.10.9 notifyTilt	104
3.10.10 notifyHeight	104
3.10.11 startHeating	105
3.10.12 stopHeating	105
3.10.13 setHeatingLevel	106
3.10.14 getHeatingLevel	106
3.10.15 isHeating	107
3.10.16 notifyHeatingStatus	107
3.11 BCM_SunRoof(天窗服务)	108
3.11.1 adjustPosition	108
3.11.2 getPosition	109
3.11.3 adjustAngle	109
3.11.4 getAngle	110
3.11.5 open	110
3.11.6 close	111
3.11.7 stop	111
3.11.8 notifyActuateStatus	112
3.11.9 getLearnedStatus	112
3.11.10 notifyAntiPinch	112
3.11.11 notifyIceBreaking	113
3.11.12 notifyPosition	113
3.12 BCM_WelcomePedal(迎宾踏板服务)	114
3.12.1 fold	114
3.12.2 unfold	115
3.12.3 notifyPosition	115
3.12.4 adjustPosition	116
3.12.5 getPosition	116
3.13 BCM_Window(车窗服务)	117
3.13.1 lock	117
3.13.2 unlock	118
3.13.3 getLockStatus	118
3.13.4 notifyLockStatus	119
3.13.5 adjustPosition	119
3.13.6 getPosition	120
3.13.7 open	120
3.13.8 close	121
3.13.9 stop	121

3.13.10 getLearnedStatus.....	121
3.13.11 notifyAntiPinch.....	122
3.13.12 notifyActuateStatus.....	122
3.13.13 notifyIceBreaking.....	123
3.13.14 notifyPosition.....	123
3.14 BCM_Sunshade(遮阳服务).....	124
3.14.1 adjustShade.....	124
3.14.2 getShade.....	125
3.14.3 notifyShade.....	125
3.14.4 getShadeType.....	126
3.14.5 open.....	126
3.14.6 close.....	127
3.14.7 stop.....	127
3.14.8 notifyAntiPinch.....	128
3.14.9 notifyActuateStatus.....	128
3.15 BCM_WiperWash(雨刷清洗服务).....	128
3.15.1 stopWiping.....	129
3.15.2 startWiping.....	130
3.15.3 setMaintenanceMode.....	130
3.15.4 setWipingLevel.....	131
3.15.5 setWipingLevelImme.....	131
3.15.6 setWipingInterval.....	132
3.15.7 isWiping.....	132
3.15.8 getWipingLevel.....	133
3.15.9 getWipingInterval.....	133
3.15.10 stopSprayWashing.....	134
3.15.11 startSprayWashing.....	134
3.15.12 getWaterLevel.....	135
3.15.13 notifyWaterLevelStatus.....	135
3.15.14 isSprayWashing.....	135
3.15.15 notifySprayingStatus.....	136
3.15.16 notifyWipingStatus.....	136
3.16 BCM_BrakeLight(刹车灯服务).....	137
3.16.1 turnOff.....	137
3.16.2 turnOn.....	138
3.16.3 emergencyOn.....	138
3.16.4 getStatus.....	139
3.16.5 notifyStatus.....	139
3.17 BCM_SteerLight(转向灯服务).....	139
3.17.1 turnOff.....	140
3.17.2 turnOn.....	140
3.17.3 getStatus.....	141

3.17.4 notifyStatus	141
3.18 BCM_HazardLight(报警灯服务).....	142
3.18.1 turnOff	142
3.18.2 turnOn.....	143
3.18.3 setFlashMode.....	143
3.18.4 getStatus.....	144
3.18.5 notifyStatus	144
3.19 BCM_DaytimeLight(日行灯服务).....	145
3.19.1 turnOff	145
3.19.2 turnOn.....	145
3.19.3 getStatus.....	146
3.19.4 notifyStatus	146
3.20 BCM_FogLight(雾灯服务).....	147
3.20.1 turnOff	147
3.20.2 turnOn.....	148
3.20.3 getStatus.....	148
3.20.4 notifyStatus	149
3.21 BCM_HighBeam(远光灯服务).....	149
3.21.1 turnOff	150
3.21.2 turnOn.....	150
3.21.3 getStatus.....	151
3.21.4 notifyStatus	151
3.22 BCM_LowBeam(近光灯服务).....	151
3.22.1 turnOff	152
3.22.2 turnOn.....	152
3.22.3 getStatus.....	153
3.22.4 notifyStatus	153
3.23 BCM_OutlineLight(位置灯服务).....	154
3.23.1 turnOff	154
3.23.2 turnOn.....	155
3.23.3 getStatus.....	155
3.23.4 notifyStatus	155
3.24 BCM_ReverseLight(倒车灯服务).....	156
3.24.1 turnOff	156
3.24.2 turnOn.....	157
3.24.3 getStatus.....	157
3.24.4 notifyStatus	158
3.25 BCM_CommonLight(通用灯服务).....	158
3.25.1 turnOff	159
3.25.2 turnOn.....	159
3.25.3 flash	160

3.25.4 setBrightness	160
3.25.5 getStatus	161
3.25.6 getBrightness	161
3.25.7 notifyStatus	162
3.26 BCM_BeamDirection(光束方向调节服务).....	162
3.26.1 adjustZAngle.....	163
3.26.2 getZAngle	163
3.26.3 startAdjustZAngle.....	164
3.26.4 stopAdjustZAngle.....	164
3.26.5 adjustYAngle.....	165
3.26.6 getYAngle	165
3.26.7 startAdjustYAngle.....	166
3.26.8 stopAdjustYAngle.....	166
3.27 BCM_Buzzer(翁鸣器服务).....	167
3.27.1 start	167
3.27.2 stop.....	168
3.27.3 alert	168
3.27.4 setAlarmMode	169
3.27.5 getAlarmMode	169
3.27.6 notifyStatus	170
3.27.7 getStatus.....	170
3.28 BCM_Avas(行人警示器服务).....	171
3.28.1 start	171
3.28.2 stop.....	171
3.28.3 getStatus.....	172
3.29 BCM_RearSpoiler(尾翼服务).....	172
3.29.1 fold.....	173
3.29.2 unfold.....	174
3.29.3 stop.....	174
3.29.4 notifyActuateStatus.....	174
3.29.5 adjustPosition.....	175
3.29.6 getPosition	175
3.29.7 notifyPosition.....	176
3.29.8 isFolded	176
3.29.9 getLearnedStatus.....	177
3.29.10 notifyAntiPinch.....	177
3.29.11 notifyIceBreaking	178
3.30 BCM_SafetyBelt(安全带服务).....	178
3.30.1 getBuckleStatus	178
3.30.2 notifyBuckleStatus.....	179
3.31 BCM_ScreenAdjust(屏位置调节服务)	179

3.31.1 fold.....	180
3.31.2 unfold.....	180
3.31.3 stop.....	181
3.31.4 notifyActuateStatus.....	181
3.31.5 adjustPosition.....	182
3.31.6 getPosition.....	182
3.31.7 notifyPosition.....	183
3.31.8 getFoldedStatus.....	183
3.31.9 getLearnedStatus.....	184
3.31.10 notifyAntiPinch.....	184
3.32 数据类型定义.....	185
3.32.1 BCM_ReturnCode.....	185
3.32.2 BCM_LockStatus.....	186
3.32.3 BCM_ActuateStatus.....	186
3.32.4 BCM_LockStatus_Window.....	187
3.32.5 BCM_AuthStatus.....	188
3.32.6 BCM_KeyLocation.....	188
3.32.7 BCM_ShadeType.....	189
3.32.8 BCM_FogLightId.....	190
3.32.9 BCM_SteerLightId.....	190
3.32.10 BCM_WipingLevel.....	191
3.32.11 BCM_LightStatus.....	191
3.32.12 BCM_KeyContent.....	192
3.32.13 BCM_FlashMode.....	192
3.32.14 BCM_SteerLightingStatus.....	193
3.32.15 BCM_FogLightingStatus.....	193
3.32.16 BCM_LearnedStatus.....	194
3.32.17 BCM_StepActionType.....	194
3.32.18 BCM_AlarmMode.....	195
3.32.19 BCM_AntiGlareMode.....	195
3.32.20 BCM_AntiGlareActMode.....	196
3.32.21 BCM_PeriodMode.....	196
3.32.22 BCM_AvasStatus.....	197
3.32.23 BCM_BuzzerStatus.....	197
3.32.24 BCM_MessageMode.....	198
3.32.25 BCM_MessageStrength.....	198
3.32.26 BCM_MessageType.....	199
3.32.27 BCM_AntiPinchStatus.....	199
3.32.28 BCM_IceBreakingStatus.....	200
3.32.29 BCM_FoldedStatus.....	200
3.32.30 BCM_OpenStatus.....	201
3.32.31 BCM_BlowStatus.....	201

3.32.32 BCM_BuckleStatus	202
3.32.33 BCM_MaintenanceMode.....	202
3.32.34 BCM_OccupiedStatus.....	203
4 TMS(热管理)原子服务 API.....	204
4.1 TMS_Battery(电池包温控).....	204
4.1.1 setTargetTemp	204
4.1.2 getTargetTemp	205
4.1.3 getCurrentFlow	205
4.1.4 setMinFlow	206
4.1.5 setTargetWorkMode.....	206
4.1.6 getCurrentWorkMode	207
4.1.7 notifyCurrentWorkMode.....	207
4.2 TMS_Device(设备散热控制)	208
4.2.1 setTargetTemp.....	209
4.2.2 getTargetTemp	209
4.2.3 getCurrentTemp	210
4.2.4 setMinFlow	210
4.2.5 getCurrentFlow	211
4.2.6 notifyCurrentFlow	211
4.2.7 getCurrentWaterTemp.....	212
4.2.8 notifyCurrentWaterTemp	212
4.3 TMS_AC(座舱温控服务).....	213
4.3.1 turnOn.....	213
4.3.2 turnOff	214
4.3.3 isRunning	214
4.3.4 setTargetTemp.....	215
4.3.5 getTargetTemp	215
4.3.6 getAirOutletTemp	216
4.4 TMS_Ventilation(通风服务).....	216
4.4.1 turnOn.....	217
4.4.2 turnOff	218
4.4.3 isRunning	218
4.4.4 setDirection.....	219
4.4.5 getDirection	219
4.4.6 notifyDirection.....	220
4.4.7 setLevel.....	220
4.4.8 getLevel	221
4.4.9 notifyLevel.....	221
4.4.10 setMode	222
4.4.11 getMode	222
4.4.12 notifyMode	223

4.4.13 setCyclingMode	223
4.4.14 getCyclingMode	224
4.4.15 notifyCyclingMode.....	224
4.5 TMS_Purifier(空气净化服务)	225
4.5.1 turnOn.....	226
4.5.2 turnOff	226
4.5.3 isRunning	227
4.5.4 startDetecting	227
4.5.5 stopDetecting	228
4.5.6 isDetecting	228
4.5.7 setDetectMode	229
4.5.8 getDetectMode.....	229
4.5.9 getParticle	230
4.5.10 notifyParticle.....	230
4.5.11 startSterilization	231
4.5.12 stopSterilization	231
4.5.13 isSterilization	231
4.5.14 getUvLiveDetection.....	232
4.5.15 startAQSDetecting.....	232
4.5.16 stopAQSDetecting	233
4.5.17 isAQSDetecting	233
4.5.18 getAQI	234
4.5.19 notifyAQI.....	234
4.5.20 getCO2.....	234
4.5.21 notifyCO2	235
4.6 TMS_Fragrance(香氛服务).....	235
4.6.1 turnOn.....	236
4.6.2 turnOff	236
4.6.3 isRunning	237
4.6.4 setConcentration	237
4.6.5 getConcentration.....	238
4.6.6 setType.....	238
4.6.7 getType	239
4.7 TMS_Demister(除雾除霜服务).....	239
4.7.1 start	240
4.7.2 stop.....	240
4.7.3 isRunning	241
4.7.4 setLevel.....	241
4.7.5 getLevel	242
4.8 TMS_EnvMonitor(环境服务).....	242
4.8.1 getTemp	243
4.8.2 notifyTemp.....	243

4.8.3 getHumidity	244
4.8.4 notifyHumidity	244
4.8.5 getLightIntensity	245
4.8.6 notifyLightIntensity	245
4.9 数据类型定义	246
4.9.1 TMS_ReturnCode	246
4.9.2 TMS_ParticleDetectMode	246
4.9.3 TMS_ParticleType	247
4.9.4 TMS_AirMode	247
4.9.5 TMS_AirDirection	248
4.9.6 TMS_CyclingMode	248
4.9.7 TMS_FragranceType	249
4.9.8 TMS_DemistZoneId	250
4.9.9 TMS_EnvZone	250
4.9.10 TMS_VentilationDirectionAry	251
4.9.11 TMS_WindDirectionSt	251
4.9.12 TMS_VentilationLevelAry	251
4.9.13 TMS_WindLevelSt	252
4.9.14 TMS_VentilationModeAry	252
4.9.15 TMS_WindModeSt	253
4.9.16 TMS_PMConcentrationAry	253
4.9.17 TMS_PMConcentrationSt	254
4.9.18 TMS_EnvZoneTempAry	254
4.9.19 TMS_EnvZoneTempSt	254
4.9.20 TMS_EnvZoneHumidityAry	255
4.9.21 TMS_EnvZoneHumiditySt	255
4.9.22 TMS_EnvZoneLightIntensityAry	256
4.9.23 TMS_EnvZoneLightIntensitySt	256
4.9.24 TMS_BatteryMode	257
4.9.25 TMS_LoopPort	257
4.9.26 TMS_DeviceFlowAry	258
4.9.27 TMS_DeviceFlowSt	258
4.9.28 TMS_DeviceWaterTempAry	258
4.9.29 TMS_DeviceWaterTempSt	259
5 VCS(运动控制)原子服务 API	260
5.1 VCS_VehSpd(基础车速服务).....	260
5.1.1 notifyStatus	260
5.2 VCS_BrakePedal(制动状态服务).....	261
5.2.1 notifySwitch.....	261
5.2.2 notifyStatus	261
5.3 VCS_AccrPedal(加速踏板状态服务).....	262

5.3.1 notifyStatus	262
5.4 VCS_Gear(档位状态服务)	262
5.4.1 setTarget	263
5.4.2 notifyStatus	263
5.5 VCS_TqDmd(扭矩计算服务).....	264
5.5.1 notifyCreepTq	264
5.5.2 setCreepSpd	265
5.5.3 notifyOnePedalTq	265
5.5.4 notifyCrsTq	266
5.5.5 setCrsSpd	266
5.5.6 notifyCoastRgnTq	267
5.5.7 setCoastRgnTqCoeff	267
5.5.8 notifyDriveTq	268
5.5.9 notifyADSTq	268
5.5.10 setADSTarget	269
5.5.11 setDrivePedalPosn	269
5.5.12 notifyTotalBrakeTq	270
5.6 VCS_SteerWhl(方向盘状态和控制)	270
5.6.1 notifyAngle	271
5.6.2 setAngle	271
5.6.3 notifyAngleSpd	272
5.6.4 notifyTq	272
5.6.5 setOverlayTq	273
5.7 VCS_Steer(转向状态和控制).....	273
5.7.1 notifyFrntWhlAngle	274
5.7.2 setFrntWhlAngle	274
5.7.3 notifyReWhlAngle	275
5.7.4 setReWhlAngle	275
5.8 VCS_Motion(车辆运动状态)	276
5.8.1 notifyAx	276
5.8.2 notifyAy	276
5.8.3 notifyYawRate	277
5.9 VCS_Brake(制动状态和控制).....	277
5.9.1 notifyMCPrs	278
5.9.2 setTargetAx	278
5.9.3 setTotalForce	279
5.9.4 notifyTq	279
5.10 VCS_Whl(车轮状态服务)	280
5.10.1 notifySpd	280
5.10.2 notifySpdPls	280
5.10.3 notifyTirePrs	281

5.11 VCS_EPB(电子驻车服务).....	281
5.11.1 setSwitchStatus	282
5.11.2 notifyStatus	282
5.12 VCS_Slope(坡度服务).....	283
5.12.1 notifystaticSlope	283
5.13 VCS_MotCtrl(驱动电机控制服务)	283
5.13.1 notifyStatus	284
5.13.2 setMotTq.....	284
5.13.3 setMotSpd.....	285
5.13.4 setMotMode.....	285
5.13.5 setDampingMode.....	286
5.14 VCS_Crash(碰撞服务).....	286
5.14.1 notifyStatus	287
5.15 数据类型定义	287
5.15.1 VCS_VehSpd	287
5.15.2 VCS_BrakeSwitch	288
5.15.3 VCS_AccrPedal	289
5.15.4 VCS_Gear.....	290
5.15.5 VCS_ADSTarget.....	290
5.15.6 VCS_TotalBrakeTq.....	291
5.15.7 VCS_SteerAngle.....	292
5.15.8 VCS_SteerAngleSpd.....	292
5.15.9 VCS_SteerTq.....	293
5.15.10 VCS_WhlAngle	293
5.15.11 VCS_YawRate	294
5.15.12 VCS_Acceleration	294
5.15.13 VCS_MCPrs	295
5.15.14 VCS_BrakePosn	295
5.15.15 VCS_WhlSpd.....	296
5.15.16 VCS_EPBSwitch	298
5.15.17 VCS_EPB	299
5.15.18 VCS_LgtBrkFTotal.....	299
5.15.19 VCS_WhlSpdPls.....	300
5.15.20 VCS_TirePrs	301
5.15.21 VCS_TqCtrl.....	302
5.15.22 VCS_Slope	303
5.15.23 VCS_MotBaseStatus.....	303
5.15.24 VCS_MotStatus	305
5.15.25 VCS_Crash	306
6 EMS(能量管理)原子服务 API.....	307
6.1 EMS_ChargePort(充电接口服务)	307

6.1.1 getStatus.....	308
6.1.2 notifyStatus	308
6.1.3 notifyACStatus	309
6.1.4 notifyDCStatus	309
6.1.5 getActive.....	310
6.1.6 notifyACTemp	310
6.1.7 notifyDCTemp	310
6.1.8 getLockStatus	311
6.1.9 notifyLockStatus.....	311
6.1.10 lock	312
6.1.11 unlock	312
6.2 EMS_Charging(充电控制服务).....	313
6.2.1 getChrgrCap.....	314
6.2.2 getBattLimit.....	314
6.2.3 notifyBattInput.....	315
6.2.4 notifyChrgrOutp.....	315
6.2.5 getProgress.....	316
6.2.6 getChargedCp	316
6.2.7 getRemainCp	316
6.2.8 setMaxCurrent	317
6.2.9 start	317
6.2.10 stop.....	318
6.2.11 getStatus	318
6.2.12 notifyStatus	319
6.3 EMS_ChargeIndicator(充电指示服务).....	319
6.3.1 setIndicator	320
6.4 EMS_HVBatt(高压电池状态服务)	320
6.4.1 getSOC.....	321
6.4.2 getSOE.....	322
6.4.3 getSOH	322
6.4.4 notifySOC	322
6.4.5 notifySOE	323
6.4.6 getSOP	323
6.4.7 notifySOP	324
6.4.8 getRegenerate	324
6.4.9 notifyRegenerate.....	325
6.4.10 getAccuStat.....	325
6.4.11 getAccuChrgCp.....	326
6.4.12 getAccuDischrgCp.....	326
6.4.13 getAccuRgnCp.....	326
6.4.14 getAccuBalCp.....	327
6.4.15 getAccuChrgEgy.....	327

6.4.16 getAccuDischrgEgy	328
6.4.17 getAccuRgnEgy	328
6.4.18 getAccuBalEgy	329
6.5 EMS_LVBatt(低压电池状态服务)	329
6.5.1 getSOC	330
6.5.2 getSOE	330
6.5.3 getSOH	331
6.5.4 notifySOC	331
6.5.5 notifySOE	331
6.5.6 getVolt	332
6.5.7 notifyVolt	332
6.5.8 getAccuStat	333
6.5.9 getAccuChrgCp	333
6.5.10 getAccuDischrgCp	334
6.6 EMS_HvBattBalance(电池均衡服务)	334
6.6.1 start	335
6.6.2 stop	335
6.6.3 getStatus	335
6.6.4 getCp	336
6.6.5 getRemainTime	336
6.6.6 getRemainCp	337
6.7 EMS_PowerSupplier(电源管理服务)	337
6.7.1 notifyHVReady	338
6.7.2 startHV	338
6.7.3 stopHV	339
6.7.4 notifyHVStatus	339
6.7.5 notifyPower	339
6.7.6 getStatus	340
6.7.7 getAccuStat	340
6.7.8 getDischrgCp	341
6.7.9 getRegenerateCp	341
6.8 EMS_Circuit(电路配电服务)	342
6.8.1 notifyReady	342
6.8.2 turnOn	343
6.8.3 turnOff	343
6.8.4 notifyStatus	343
6.8.5 notifyPower	344
6.8.6 setVolt	344
6.8.7 getVolt	345
6.8.8 setPowerLimit	345
6.9 数据类型定义	346

6.9.1 EMS_PortConnectStatus.....	346
6.9.2 EMS_ChargePortId.....	346
6.9.3 EMS_PlugType.....	347
6.9.4 EMS_LockStatus.....	347
6.9.5 EMS_PlugCapability.....	348
6.9.6 EMS_ChargePortStatus.....	349
6.9.7 EMS_ChargePlugStatus.....	349
6.9.8 EMS_ReturnCode.....	350
6.9.9 EMS_PowerInfo.....	350
6.9.10 EMS_SupplyMode.....	351
6.9.11 EMS_ChargerOutput.....	351
6.9.12 EMS_ChargeStatus.....	352
6.9.13 EMS_BalanceStatus.....	352
6.9.14 EMS_PowerSupplyType.....	353
6.9.15 EMS_ChargeLockStatus.....	353
6.9.16 EMS_ChargingIndicator.....	354
6.9.17 EMS_HvReady.....	355
6.9.18 EMS_HvBattPowerLimit.....	355
6.9.19 EMS_HvBattAccuStat.....	356
6.9.20 EMS_LvBattAccuStat.....	357
6.9.21 EMS_PowerStat.....	357
6.9.22 EMS_HvResult.....	358
6.9.23 EMS_HvStatus.....	358
6.9.24 EMS_HvAbnormalCode.....	359
6.9.25 EMS_ChargePortTemp.....	360
7 ADAS(智驾域)原子服务 API.....	361
7.1 ADAS_Radar(毫米波雷达感知服务).....	361
7.1.1 notifyObjects.....	361
7.2 ADAS_USS(超声波雷达感知服务).....	362
7.2.1 notifyObjects.....	362
7.2.2 notifyObjectsDistance.....	362
7.3 ADAS_Lidar(激光雷达感知服务).....	363
7.3.1 notifyObjects.....	363
7.4 ADAS_Perception(视觉感知).....	364
7.4.1 getTrackObjects.....	364
7.4.2 getLaneline.....	365
7.4.3 getRoadMarking.....	365
7.4.4 getTrafficLight.....	366
7.4.5 getTrafficSign.....	366
7.4.6 getParkingSpace.....	367
7.4.7 getFreeSpace.....	367

7.4.8 getFusionTrackObjects	367
7.5 数据类型定义	368
7.5.1 ADAS_RadarObjects	368
7.5.2 ADAS_UssObjects	368
7.5.3 ADAS_LidarObjects	369
7.5.4 ADAS_DetectionObjects	369
7.5.5 ADAS_DetectionObjAry	370
7.5.6 ADAS_DetectionObj	370
7.5.7 ADAS_UssObjectsDistance	372
7.5.8 ADAS_UssObjDistanceAry	372
7.5.9 ADAS_UssObjDistance	373
7.5.10 ADAS_TrackObjects	373
7.5.11 ADAS_TrackObjectsAry	374
7.5.12 ADAS_TrackObject	374
7.5.13 ADAS_LaneLineInfo	376
7.5.14 ADAS_LaneLinesAry	377
7.5.15 ADAS_LaneLine	377
7.5.16 ADAS_RoadMarkingInfo	378
7.5.17 ADAS_RoadMarkingAry	379
7.5.18 ADAS_RoadMarking	379
7.5.19 ADAS_TrafficLightInfo	380
7.5.20 ADAS_TrafficLightAry	381
7.5.21 ADAS_TrafficLight	381
7.5.22 ADAS_TrafficSignInfo	383
7.5.23 ADAS_TrafficSignAry	383
7.5.24 ADAS_TrafficSign	383
7.5.25 ADAS_ParkingInfo	385
7.5.26 ADAS_ParkingSpaceAry	385
7.5.27 ADAS_ParkingSpace	386
7.5.28 ADAS_FreeSpaceInfo	387
7.5.29 ADAS_PerceptionFusionObjects	388
7.5.30 ADAS_Pose	389
7.5.31 ADAS_Dimension	389
7.5.32 ADAS_Velocity	390
7.5.33 ADAS_Acceleration	391
7.5.34 ADAS_Points2D	391
7.5.35 ADAS_Point2DAry	392
7.5.36 ADAS_Points3D	392
7.5.37 ADAS_Point3DAry	392
7.5.38 ADAS_Point2D	393
7.5.39 ADAS_Point3D	393
7.5.40 ADAS_Quaternion	394

7.5.41 ADAS_Coe	394
7.5.42 ADAS_TimeStamp	395
8 HMI(人机交互)原子服务 API.....	396
8.1 HMI_ClickSwitch(通用点击型按钮操作服务)	396
8.1.1 notifyMode	396
8.2 HMI_PressSwitch(通用按压型按钮按压服务).....	397
8.2.1 notifyStatus	397
8.3 HMI_RotarySwitch(通用旋钮型按钮操作服务)	397
8.3.1 notifyStatus	398
8.4 数据类型定义	398
8.4.1 HMI_ClickMode.....	398
8.4.2 HMI_PressedStatus.....	399
8.4.3 HMI_RotaryStatus	399
9 附录：基本数据类型说明	401

1 前言

近年来，随着汽车“新四化”浪潮的兴起，软件定义汽车已成为产业共识，软件将深度参与到整个汽车的定义、开发、验证、销售以及服务全过程。一方面确保汽车软件可升级，可跨车型、跨软件，甚至跨车企的软件重用。另一方面对于硬件来讲，要做到可扩展、可更换，甚至做到传感器的即插即用，让消费者享受到常用常新的汽车体验。

因此，中国汽车工业协会面向软件定义汽车的趋势，为协同整车厂、零部件企业，推动智能汽车软硬件接口标准化，降低智能汽车研发复杂度，成立软件定义汽车工作组，简称 SDV 工作组。其目标是达成：

- 构建一个公平开放的产业交流平台，联合产业力量开展软件定义汽车，包含乘用车、商用车相关应用场景、需求、技术、标准的研究、验证及应用推广，发挥中国标准化规模供应优势，减少低效的重复劳动和成本投入，避免产业碎片化，加速智能汽车发展进程，做大做强中国智能汽车产业链。
- 产业各方增进互相了解、加强协同合作，减少汽车电子产业链厂商之间互相对接的成本，通过联合创新、联合开发，促进场景快速商用落地。
- 帮助产业伙伴在软件定义汽车的发展趋势下，充分发挥自身优势进行业务持续创新，合作共赢。

本规范在编制过程中得到了 SDV 工作组诸多成员单位的支持，在此特别感谢参与编写及审阅的单位、组织和人员。

参与编写单位：（排名不分先后）

中国汽车工业协会、长城汽车股份有限公司、中国第一汽车集团有限公司、上汽通用五菱汽车股份有限公司、长安汽车软件科技公司、安徽江淮汽车集团股份有限公司、华为技术有限公司、北京新能源汽车股份有限公司、一汽解放汽车有限公司、奇瑞汽车股份有限公司、浙江吉利汽车研究总院、广州汽车集团股份有限公司、岚图汽车科技公司、东风汽车集团有限公司、比亚迪汽车工业有限公司、宝能汽车集团有限公司、江铃汽车股份有限公司、浙江亚太机电股份有限公司、圣戈班汽车玻璃系统（中国）、舍弗勒智能驾驶科技（长沙）有限公司、上海纳恩汽车技术股份有限公司、重庆睿博光电股份有限公司、艾泰斯热系统研发（上海）有限公司、华域视觉科技（上海）有限公司、北京海纳川汽车部件股份有限公司、上海海拉电子有限公司、弗迪科技有限公司、采埃孚汽车科技（上海）有限公司、博泽汽车技术企业管理(中国)有限公司、重庆延锋安道拓汽车部件系统有限公司、埃泰克汽车电子（芜湖）有限公司、福瑞泰克智能系统有限公司、科世达（上海）管理有限公司、浙江孔辉汽车科技有限公

司、苏州挚途科技有限公司、大陆投资（中国）有限公司、上海浙减汽车悬架有限公司、上海保隆汽车科技股份有限公司、上海亿凯软件科技有限公司、曼德电子电器有限公司、瑞昂汽车部件（苏州）有限公司、PIX Moving 像素智能、深圳南方德尔汽车电子有限公司、重庆集诚汽车电子有限责任公司、北京百度智行科技有限公司、浙江银轮机械股份有限公司、上海艾拉比智能科技有限公司、东方久乐汽车电子（上海）股份有限公司、爱达克车辆设计(上海)有限公司、上海悠跑网络科技有限公司、北京经纬恒润科技股份有限公司、中科创达软件股份有限公司、温州长江汽车电子有限公司、常州星宇车灯股份有限公司、上海汽车集团股份有限公司、惠州德赛西威汽车电子股份有限公司、深圳和而泰汽车电子科技有限公司、弗吉亚（中国）投资有限公司、上海拿森汽车电子有限公司、苏州天瞳威视电子科技有限公司（CalmCar）、北斗星通智联科技有限公司、上海寅家电子科技股份有限公司、广东东箭汽车科技股份有限公司、马瑞利汽车零部件（芜湖）有限公司上海分公司、中软国际科技服务软件公司、东风柳州汽车有限公司、麦格纳汽车系统（苏州）有限公司、宁德时代新能源科技股份有限公司、电装（中国）投资有限公司、武汉光庭信息技术股份有限公司、诚迈科技（南京）股份有限公司、上海映驰科技有限公司、星河智联汽车科技有限公司、纬湃科技投资(中国)有限公司、小米汽车有限公司、普华基础软件股份有限公司、安波福中国科技研发有限公司、奇瑞商用车（安徽）有限公司、杭州当虹科技有限公司、威马汽车科技集团有限公司、博世汽车部件（苏州）有限公司、科大讯飞股份有限公司、杭州三花汽车零部件有限公司、上海智驾汽车科技有限公司、盖瑞特动力科技（上海）有限公司、禾多科技（北京）有限公司、重庆长安新能源科技有限公司、深圳佑驾创新科技有限公司、镁佳（北京）科技有限公司、联合汽车电子有限公司、上海商泰汽车信息系统有限公司、领科汇智科技有限公司、亚信安全股份有限公司、润芯微科技（江苏）有限公司、合肥晟泰泰克汽车电子股份有限公司、东软睿驰汽车技术（上海）有限公司、江苏润和软件股份有限公司、重庆金康赛力斯新能源汽车设计院有限公司、国汽朴津智能科技（安庆）有限公司、上海怪星电子科技有限公司、中国汽车工程研究院股份有限公司、智慧协同（北京）科技有限公司、亿咖通（湖北）技术有限公司

2 概述

2.1 行业现状

传统的分布式 E/E 架构下，汽车采用的是“面向信号”的软件结构，ECU 之间通过 LIN/CAN 等总线进行点对点静态通信。随着汽车智能化需求的快速增长，传统通讯网络及软件架构设计中扩展性差、升级和移植成本高等问题逐渐凸显，例如新增某项软件应用或服务，需从头建立一个新的基础软件环境，成本极高。为解决上述问题，汽车行业借鉴 IT 行业发展经验，提出 SOA（Service Oriented Architecture，面向服务的架构）软件架构。

SOA 是实现软件定义汽车的软件基础，本质是将原本相互分散的 ECU 及其对应的基础软件功能模块化、标准化，重新部署为分层式的软件架构，汽车可在不增加或更换硬件的条件下通过不同的软件配置为车主提供不同的服务，从而实现千车千面。

在 ICT 领域，SOA 已经在国内外得到广泛的应用，相对比较成熟。SOA 由 Gartner 于 20 世纪 90 年代中期提出，2002 年业界开始积极探索和研究，2005 年 SOA 开始推广和普及，2007 年行业开始制定标准推动实施，如 SCA 和 SDO 通过 OASIS 审核，WS-POLICY、W3C 成为 W3C 标准等，2010 年起开始迈向成熟期。而实现 SOA 架构落地的关键点就是：对已有系统中的功能进行提取和包装，形成标准化的“服务”。

在汽车行业，近年来在特斯拉的引领下，众多传统整车厂正通过成立子公司、软件研发部门、与软件供应商合作等模式加码车载软件领域布局。

- 在国外：大众、宝马、奔驰、奥迪、福特等汽车巨头已各自在车身和座舱域实践 SOA 服务化并逐步量产。
- 在国内：上汽、一汽、长城等部分头部整车厂已意识到 SOA 软件架构的重要性，尝试落地 SOA 服务化，且考虑不再高度依赖国外技术平台和标准规范。但目前多是少量服务化尝试，SOA 架构也未统一，未形成通用普适性规范，极大制约了车载 SOA 的发展，限制了软件推动创新。

软件架构技术和 API 接口定义为汽车行业共性技术和资产，属于开放式共性平台，目前国内缺少行业协同共享机制，在共性平台和生态建设方面发展较为缓慢，这也是 SDV 工作组亟待解决的关键问题。

2.2 SDV 定义的服务软件架构

SDV 工作组对 SOA 软件架构进行服务化分层解耦，这也是提升软件复用性、降低软件开发复杂度的关键手段。SDV 定义的服务软件架构分为 4 大层。

- 应用层：基于原子服务实现对整车服务、应用、体验等进行定义和组合增强，构建差异化竞争力的 APP。
- 原子服务层：实现一定的数据融合或控制逻辑的功能模块。作为服务的最小单位与单一执行实体，通过 API 向上为应用提供可按需编排的基础服务，使能一次开发多次重用，最大化提升开发效率。
- 设备抽象层：对传感器、执行器、Legacy ECU 等硬件资源进行抽象，通过 API 向上为服务提供设备访问接口，屏蔽设备功能实现差异（硬件差异&厂家差异），减少定制化与重复劳动。
- 基础平台层：包括硬件和操作系统，主要提供整车运行所需的基础运行环境。

其中，设备抽象层和原子服务层的 API 接口由 SDV 工作组联合定义并规范化，如图 2-1 所示。通过产业协同，使能软件与硬件解耦、应用与软件代码解耦，从而降低软件复杂度，加速智能汽车发展创新进程。

图2-1 SDV 服务软件架构



2.3 SDV 标准化 API 接口的意义

SDV 工作组本次发布的 API 参考文档，为定义智能汽车软硬件接口标准化的规范性文件。工作组通过对 API 接口的标准化定义，为各领域带来全新的体验。

对整车厂：

- 整车跨域打通，性能最优，整车级能力开放，按需调用
- 实现版本归一管理、快速开发、软硬件解耦、车型间架构重用
- 基于 SOA 的高效软件开发，获得预定义或预集成服务 API，加速车型上市速度

对零部件厂商：

- 自身领域 know-how 通过标准化服务 API 进行变现，同步带动硬件销售
- IO 设备标准化，减少定制化，降低面向不同车企车型新功能开发成本，缩短 TTM(Time to Market，上市时间)
- 共享新生态资源池不同服务能力，通过调用服务 API 优化及创新零部件产品

对开发者：

- 调用及组合不同服务 API，快速开发应用场景
- 开放生态，整合开发者资源，聚焦体验提升，实现千车千面
- 价值持续变现，后市场价值挖掘（基于大数据）

对消费者：

- 获得更好的汽车使用体验，更加丰富的后市场改装和常用常新感受
- 持续提高车辆保值率，享受个性化服务

2.4 SDV API 版本规则和升级兼容说明

为了更好的管理 API 版本，支持两类版本类型。

- **Beta**: 征集意见版本，供 Beta 试用，不承诺 API 的稳定性。版本规则为 Beta 1, 2, 3.....
- **Release**: 正式版本，面向开发者公开，承诺 API 的稳定性。版本规则为 Version 1, 2, 3.....

初始版本不涉及变更，后续版本 API 的发布会伴随着 API 版本编号的变更，此时会提供两个版本之间的变更说明。通过变更说明，可以直观的了解 API 的变更情况。

服务 API 一般支持前向兼容，新版本会兼容老版本的功能，故 API 版本升级后，不会影响老版本的功能。但是在个别情况下，API 的更改可能会影响老功能在新版本中的运行使用，这种属于不兼容性变更。针对不兼容性变更，会在变更说明中突出强调，用户在升级到新版本 API 时，要特别关注这些不兼容性变更。

2.5 原子服务 API 范围

本文档重点介绍原子服务 API。原子服务 API 按功能进行分类，当前包括：

- 车身控制服务，缩写为 BCM（Body Control Module）
- 热管理服务，缩写为 TMS（Thermal Management System）
- 车辆运动控制服务，缩写为 VCS（Vehicle Control System）
- 能量管理服务，缩写为 EMS（Energy Management System）

- 智驾域服务，缩写为 ADAS（Advanced Driver Assistance System）
- 人机交互服务，缩写为 HMI（Human Machine Interface）

📖 说明

- 不同功能的原子服务 API，发布计划有所不同。根据不同节奏，API 文档会单独功能或多个功能合一发布。
- 各功能的详细发布信息，请咨询 SDV 工作组。

2.6 文档内容介绍

本文档的 API 参考，按功能域分章节，每个章节包括两部分，第一部分描述服务的 API 接口的功能和参数解释，第二部分为按功能各自定义的数据类型解释。

API 描述

按功能分章节介绍每一个 API，API 描述包括如下几个维度：

- 功能说明：介绍该 API 的功能。
- 函数原型：介绍 API 调用的函数原型。
- 参数说明：API 包含的输入参数，包括输入参数的类型及含义说明。
- 返回值说明：API 调用后的返回信息，包括返回值的类型及含义说明。

数据类型定义

每个功能域会自定义数据类型，仅在该域范围内使用，包括 enum、struct 和 array 三种数据类型。

除了域各自定义的数据类型外，还有一部分基本数据类型，统一在 [9 附录：基本数据类型说明](#) 描述。

2.7 文档发布记录

介绍文档的历史发布情况。

Draft (2022-12-28)

本文档为征求意见稿，基于 API Version 4 版本的新增优化。详细的变更，请参见对应的变更说明。包括如下 API：

- 车身控制的原子服务 API Version 4 Beta 1
- 热管理的原子服务 API Version 4 Beta 1
- 运动控制的原子服务 API Version 4 Beta 1
- 能量管理的原子服务 API Version 4 Beta 1
- 智驾域的原子服务 API Version 4 Beta 1
- 人机交互的原子服务 API Version 4 Beta 1

3 BCM(车身控制)原子服务 API

3.1 BCM_Door(车门服务)

提供车门的控制和状态服务，每一个车门一个原子服务。

BCM_Door(车门服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
BCM_ReturnCode	unlock()	车门解锁。
BCM_ReturnCode	lock()	车门上锁。
BCM_ReturnCode	superLock()	车门超级锁锁闭。
BCM_LockStatus	getLockStatus()	返回车门门锁状态。
BCM_LockStatus	notifyLockStatus()	上报车门门锁状态。
BCM_ReturnCode	foldHandle()	折叠车门把手。
BCM_ReturnCode	unfoldHandle()	展开车门把手。
boolean	isHandleFolded()	返回车门把手是否处于折叠状态。
boolean	notifyHandleFoldedStatus()	上报车门把手状态。
BCM_AntiPinchStatus	notifyHandleAntiPinch()	上报门把手防夹状态。
BCM_IceBreakingStatus	notifyHandleIceBreaking()	上报门把手破冰模式。
BCM_ReturnCode	open()	打开车门。

类型	API 函数	功能说明
BCM_ReturnCode	close()	关闭车门。
BCM_ReturnCode	stop()	停止开启或关闭车门。
BCM_ActuateStatus	getActuateStatus()	返回车门执行状态。
BCM_ActuateStatus	notifyActuateStatus()	上报车门执行状态。
BCM_LearnedStatus	getLearnedStatus()	返回电动车门学习状态。
BCM_ReturnCode	adjustPosition(uint8 position)	调整车门开度大小。
uint8	getPosition()	返回车门开度信息。
uint8	notifyPosition()	上报车门开度变化。
BCM_AntiPinchStatus	notifyAntiPinch()	上报车门防夹状态。
BCM_IceBreakingStatus	notifyIceBreaking()	上报门破冰模式。
BCM_OpenStatus	getStatus()	返回门开启状态。
BCM_OpenStatus	notifyStatus()	上报门开启状态。
BCM_ReturnCode	lockChildLock()	儿童安全锁上锁。
BCM_ReturnCode	unlockChildLock()	儿童安全锁解锁。
BCM_LockStatus	getChildLockStatus()	返回儿童安全锁的状态。
String	getDoorType()	返回车门类型。

3.1.1 unlock

功能说明

车门解锁。

函数原型

BCM_ReturnCode unlock()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.1.2 lock

功能说明

车门上锁。

函数原型

BCM_ReturnCode lock()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.1.3 superLock

功能说明

车门超级锁锁闭。

函数原型

BCM_ReturnCode superLock()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.1.4 getLockStatus

功能说明

返回车门门锁状态。

函数原型

BCM_LockStatus getLockStatus()

参数说明

无。

返回值说明

类型	描述
BCM_LockStatus	当前车门门锁状态

3.1.5 notifyLockStatus

功能说明

上报车门门锁状态。

函数原型

BCM_LockStatus notifyLockStatus()

参数说明

无。

返回值说明

类型	描述
BCM_LockStatus	当前车门门锁状态

3.1.6 foldHandle

功能说明

折叠车门把手。

函数原型

```
BCM_ReturnCode foldHandle()
```

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.1.7 unfoldHandle

功能说明

展开车门把手。

函数原型

```
BCM_ReturnCode unfoldHandle()
```

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.1.8 isHandleFolded

功能说明

返回车门把手是否处于折叠状态。

函数原型

boolean isHandleFolded()

参数说明

无。

返回值说明

类型	描述
boolean	车门把手已折叠返回 TRUE，否则返回 FALSE

3.1.9 notifyHandleFoldedStatus

功能说明

上报车门把手状态。

函数原型

boolean notifyHandleFoldedStatus()

参数说明

无。

返回值说明

类型	描述
boolean	当前车门把手状态，折叠为 TRUE，否则为 FALSE

3.1.10 notifyHandleAntiPinch

功能说明

上报门把手防夹状态。

函数原型

BCM_AntiPinchStatus notifyHandleAntiPinch()

参数说明

无。

返回值说明

类型	描述
BCM_AntiPinchStat us	当前车门把手开关的防夹状态

3.1.11 notifyHandleIceBreaking

功能说明

上报门把手破冰模式。

函数原型

BCM_IceBreakingStatus notifyHandleIceBreaking()

参数说明

无。

返回值说明

类型	描述
BCM_IceBreakingSt atus	当前车门把手的破冰模式状态

3.1.12 open

功能说明

打开车门。

函数原型

BCM_ReturnCode open()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.1.13 close

功能说明

关闭车门。

函数原型

BCM_ReturnCode close()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS, 否则返回 FAIL

3.1.14 stop

功能说明

停止开启或关闭车门。

函数原型

BCM_ReturnCode stop()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS, 否则返回 FAIL

3.1.15 getActuateStatus

功能说明

返回车门执行状态。

函数原型

```
BCM_ActuateStatus getActuateStatus()
```

参数说明

无。

返回值说明

类型	描述
BCM_ActuateStatus	当前车门的车门开关的执行状态

3.1.16 notifyActuateStatus

功能说明

上报车门执行状态。

函数原型

```
BCM_ActuateStatus notifyActuateStatus()
```

参数说明

无。

返回值说明

类型	描述
BCM_ActuateStatus	当前车门开关的执行状态

3.1.17 getLearnedStatus

功能说明

返回电动车门学习状态。

函数原型

```
BCM_LearnedStatus getLearnedStatus()
```

参数说明

无。

返回值说明

类型	描述
BCM_LearnedStatus	当前车门的门开度的学习状态

3.1.18 adjustPosition

功能说明

调整车门开度大小。

函数原型

```
BCM_ReturnCode adjustPosition(uint8 position)
```

参数说明

参数名称	类型	描述
position	uint8	车门开度大小，单位：最大开度的 1%

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.1.19 getPosition

功能说明

返回车门开度信息。

函数原型

```
uint8 getPosition()
```

参数说明

无。

返回值说明

类型	描述
uint8	当前车门的车门开度信息，单位：%，255 代表未知

3.1.20 notifyPosition

功能说明

上报车门开度变化。

函数原型

uint8 notifyPosition()

参数说明

无。

返回值说明

类型	描述
uint8	当前车门的车门的开度，单位：%，255 代表未知

3.1.21 notifyAntiPinch

功能说明

上报车门防夹状态。

函数原型

BCM_AntiPinchStatus notifyAntiPinch()

参数说明

无。

返回值说明

类型	描述
BCM_AntiPinchStatus	当前车门开关的防夹状态

3.1.22 notifyIceBreaking

功能说明

上报门破冰模式。

函数原型

BCM_IceBreakingStatus notifyIceBreaking()

参数说明

无。

返回值说明

类型	描述
BCM_IceBreakingStatus	当前车门的破冰模式状态

3.1.23 getStatus

功能说明

返回门开启状态。

函数原型

BCM_OpenStatus getStatus()

参数说明

无。

返回值说明

类型	描述
BCM_OpenStatus	当前门的开启状态

3.1.24 notifyStatus

功能说明

上报门开启状态。

函数原型

BCM_OpenStatus notifyStatus()

参数说明

无。

返回值说明

类型	描述
BCM_OpenStatus	当前门的开启状态

3.1.25 lockChildLock

功能说明

儿童安全锁上锁。

函数原型

BCM_ReturnCode lockChildLock()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.1.26 unlockChildLock

功能说明

儿童安全锁解锁。

函数原型

```
BCM_ReturnCode unlockChildLock()
```

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.1.27 getChildLockStatus

功能说明

返回儿童安全锁的状态。

函数原型

```
BCM_LockStatus getChildLockStatus()
```

参数说明

无。

返回值说明

类型	描述
BCM_LockStatus	当前车门的儿童锁状态

3.1.28 getDoorType

功能说明

返回车门类型。

函数原型

String getDoorType()

参数说明

无。

返回值说明

类型	描述
String	车门的类型，例如手动门、自动剪刀门、自动翼门、自动滑门等

3.2 BCM_Horn(鸣笛服务)

提供喇叭的控制和状态服务。

BCM_Horn(鸣笛服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
BCM_ReturnCode	stop()	停止鸣笛。
BCM_ReturnCode	start()	启动鸣笛。
BCM_ReturnCode	alert(BCM_PeriodMode periodMode)	启动警示音。
BCM_ReturnCode	setVolumeLevel(uint8 volume)	设置鸣笛音量。
boolean	isRunning()	返回当前是否正在鸣笛。
uint8	getVolume()	返回鸣笛的音量。
boolean	notifyStatus()	上报鸣笛的当前状态。

3.2.1 stop

功能说明

停止鸣笛。

函数原型

BCM_ReturnCode stop()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.2.2 start

功能说明

启动鸣笛。

函数原型

BCM_ReturnCode start()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.2.3 alert

功能说明

启动警示音。

函数原型

BCM_ReturnCode alert(BCM_PeriodMode periodMode)

参数说明

参数名称	类型	描述
periodMode	BCM_PeriodMode	嗡鸣周期模式

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.2.4 setVolumeLevel

功能说明

设置鸣笛音量。

函数原型

```
BCM_ReturnCode setVolumeLevel(uint8 volume)
```

参数说明

参数名称	类型	描述
volume	uint8	鸣笛的音量大小，单位：最高音量的 1%

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.2.5 isRunning

功能说明

返回当前是否正在鸣笛。

函数原型

```
boolean isRunning()
```

参数说明

无。

返回值说明

类型	描述
boolean	当前正在鸣笛返回 TRUE，否则返回 FALSE

3.2.6 getVolume

功能说明

返回鸣笛的音量。

函数原型

```
uint8 getVolume()
```

参数说明

无。

返回值说明

类型	描述
uint8	当前鸣笛的音量，单位：%，255 代表未知

3.2.7 notifyStatus

功能说明

上报鸣笛的当前状态。

函数原型

```
boolean notifyStatus()
```

参数说明

无。

返回值说明

类型	描述
boolean	当前鸣笛的状态，正在鸣笛为 TRUE，否则为 FALSE

3.3 BCM_Siren(汽笛服务)

提供汽笛的控制和状态服务。

BCM_Siren(汽笛服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
BCM_ReturnCode	stop()	停止汽笛鸣笛。
BCM_ReturnCode	start()	启动汽笛鸣笛。
BCM_ReturnCode	setVolumeLevel(uint8 volume)	设置汽笛鸣笛音量。
boolean	isRunning()	返回当前是否正在鸣笛。
uint8	getVolume()	返回鸣笛的音量。
boolean	notifyStatus()	上报鸣笛的当前状态。

3.3.1 stop

功能说明

停止汽笛鸣笛。

函数原型

```
BCM_ReturnCode stop()
```

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.3.2 start

功能说明

启动汽笛鸣笛。

函数原型

BCM_ReturnCode start()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.3.3 setVolumeLevel

功能说明

设置汽笛鸣笛音量。

函数原型

BCM_ReturnCode setVolumeLevel(uint8 volume)

参数说明

参数名称	类型	描述
volume	uint8	鸣笛的音量大小，单位：最高音量的 1%

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.3.4 isRunning

功能说明

返回当前是否正在鸣笛。

函数原型

```
boolean isRunning()
```

参数说明

无。

返回值说明

类型	描述
boolean	当前正在鸣笛返回 TRUE，否则返回 FALSE

3.3.5 getVolume

功能说明

返回鸣笛的音量。

函数原型

```
uint8 getVolume()
```

参数说明

无。

返回值说明

类型	描述
uint8	当前鸣笛的音量，单位：%，255 代表未知

3.3.6 notifyStatus

功能说明

上报鸣笛的当前状态。

函数原型

```
boolean notifyStatus()
```

参数说明

无。

返回值说明

类型	描述
boolean	当前汽笛的状态，正在鸣笛为 TRUE，否则为 FALSE

3.4 BCM_Key(钥匙服务)

提供钥匙的控制和状态服务。

BCM_Key(钥匙服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
BCM_AuthStat us	getBodyAuthStatus(uint8 keyId)	返回钥匙与车身认证状态。原子服务内部不会保存认证状态，而是收到 get 命令后重新进行认证并将新的认证结果作为 get 的返回结果。
BCM_KeyLoca tion	getLocation(uint8 keyId)	返回检测到的钥匙的位置。
BCM_AuthStat us	getEngineAuthStatus(uint8 keyId)	返回钥匙与动力的认证状态。原子服务内部不会保存认证状态，而是收到 get 命令后重新进行认证并将新的认证结果作为 get 的返回结果。
uint8	getKeyNumber()	获得钥匙的数量，发现的钥匙被顺序分配一个数字作为钥匙的临时逻辑标识，即 keyId。
uint32	getKeySerialNo(uint8 keyId)	获得钥匙的物理串号，物理串号唯一标识一个物理钥匙。
BCM_KeyCont ent	notifyKey()	上报钥匙对象的变化。每当发现新的合法钥匙，或发现已有的合法钥匙的状态发生变化时上报。例如钥匙离开时上报对应钥匙的位置状态为未知，离开-UNKNOWN。

3.4.1 getBodyAuthStatus

功能说明

返回钥匙与车身认证状态。原子服务内部不会保存认证状态，而是收到 get 命令后重新进行认证并将新的认证结果作为 get 的返回结果。

函数原型

```
BCM_AuthStatus getBodyAuthStatus(uint8 keyId)
```

参数说明

参数名称	类型	描述
keyId	uint8	钥匙的唯一标识

返回值说明

类型	描述
BCM_AuthStatus	当前钥匙与车身的认证状态。上层服务查询该状态时，将触发钥匙与车身的认证

3.4.2 getLocation

功能说明

返回检测到的钥匙的位置。

函数原型

```
BCM_KeyLocation getLocation(uint8 keyId)
```

参数说明

参数名称	类型	描述
keyId	uint8	钥匙的唯一标识

返回值说明

类型	描述
----	----

类型	描述
BCM_KeyLocation	当前钥匙位置

3.4.3 getEngineAuthStatus

功能说明

返回钥匙与动力的认证状态。原子服务内部不会保存认证状态，而是收到 `get` 命令后重新进行认证并将新的认证结果作为 `get` 的返回结果。

函数原型

```
BCM_AuthStatus getEngineAuthStatus(uint8 keyId)
```

参数说明

参数名称	类型	描述
keyId	uint8	钥匙的唯一标识

返回值说明

类型	描述
BCM_AuthStatus	当前钥匙与动力的认证状态。上层服务查询该状态时，将触发钥匙与发动机间的认证

3.4.4 getKeyNumber

功能说明

获得钥匙的数量，发现的钥匙被顺序分配一个数字作为钥匙的临时逻辑标识，即 `keyId`。

函数原型

```
uint8 getKeyNumber()
```

参数说明

无。

返回值说明

类型	描述
uint8	曾经检测到的不同的合法钥匙的个数

3.4.5 getKeySerialNo

功能说明

获得钥匙的物理串号，物理串号唯一标识一个物理钥匙。

函数原型

```
uint32 getKeySerialNo(uint8 keyId)
```

参数说明

参数名称	类型	描述
keyId	uint8	钥匙的唯一标识

返回值说明

类型	描述
uint32	钥匙的物理串号

3.4.6 notifyKey

功能说明

上报钥匙对象的变化。每当发现新的合法钥匙，或发现已有的合法钥匙的状态发生变化时上报。例如钥匙离开时上报对应钥匙的位置状态为未知，离开-UNKNOWN。

函数原型

```
BCM_KeyContent notifyKey()
```

参数说明

无。

返回值说明

类型	描述
BCM_KeyContent	当前钥匙的状态信息

3.5 BCM_RearView(后视镜服务)

提供后视镜的控制和状态服务，每一个后视镜一个原子服务。

BCM_RearView(后视镜服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
BCM_ReturnCode	stopHeating()	停止后视镜加热。
BCM_ReturnCode	startHeating()	启动后视镜加热。
BCM_ReturnCode	setHeatingLevel(uint8 heatingLevel)	设置后视镜加热的强度。
boolean	isHeating()	返回后视镜是否正在加热。
uint8	getHeatingLevel()	返回后视镜当前加热的强度。
boolean	notifyHeatingStatus()	上报后视镜加热的当前状态。
BCM_ReturnCode	adjustXAngle(uint8 xAngle)	调整后视镜的横向角度。
uint8	getXAngle()	返回后视镜的横向角度。
BCM_ReturnCode	startAdjustXAngle(BCM_StepActionType actionType)	启动后视镜横向角度的步进式调节。
BCM_ReturnCode	stopAdjustXAngle()	停止后视镜横向角度调节。
uint8	notifyXAngle()	上报后视镜横向角度变化。
BCM_ReturnCode	adjustYAngle(uint8 yAngle)	调整后视镜的纵向角度。
uint8	getYAngle()	返回后视镜的纵向角度。
BCM_ReturnCode	startAdjustYAngle(BCM_StepActionType actionType)	启动后视镜纵向角度的步进式调节。
BCM_ReturnCode	stopAdjustYAngle()	停止后视镜纵向角度调节。

类型	API 函数	功能说明
uint8	notifyYAngle()	上报后视镜纵向角度变化。
BCM_ReturnCode	unfold()	展开后视镜。
BCM_ReturnCode	fold()	折叠后视镜。
boolean	isFolded()	返回后视镜是否处于折叠状态。
BCM_FoldedStatus	notifyFoldedStatus()	上报后视镜折叠完成状态。
BCM_ReturnCode	adjustAntiGlare(BCM_AntiGlareMode antiGlareMode)	调整后视镜的防眩程度。
uint8	getAntiGlare()	返回后视镜的防眩程度。

3.5.1 stopHeating

功能说明

停止后视镜加热。

函数原型

BCM_ReturnCode stopHeating()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.5.2 startHeating

功能说明

启动后视镜加热。

函数原型

BCM_ReturnCode startHeating()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.5.3 setHeatingLevel

功能说明

设置后视镜加热的强度。

函数原型

BCM_ReturnCode setHeatingLevel(uint8 heatingLevel)

参数说明

参数名称	类型	描述
heatingLevel	uint8	加热的强度， 单位： 最大功率的 1%

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.5.4 isHeating

功能说明

返回后视镜是否正在加热。

函数原型

boolean isHeating()

参数说明

无。

返回值说明

类型	描述
boolean	后视镜正在加热返回 TRUE，否则返回 FALSE

3.5.5 getHeatingLevel

功能说明

返回后视镜当前加热的强度。

函数原型

```
uint8 getHeatingLevel()
```

参数说明

无。

返回值说明

类型	描述
uint8	当前后视镜当前加热的强度，单位：%，255 代表未知

3.5.6 notifyHeatingStatus

功能说明

上报后视镜加热的当前状态。

函数原型

```
boolean notifyHeatingStatus()
```

参数说明

无。

返回值说明

类型	描述
boolean	当前后视镜加热的状态，正在加热为 TRUE，否则为 FALSE

3.5.7 adjustXAngle

功能说明

调整后视镜的横向角度。

函数原型

```
BCM_ReturnCode adjustXAngle(uint8 xAngle)
```

参数说明

参数名称	类型	描述
xAngle	uint8	后视镜横轴旋转的目标位置，单位：最大转角的 1%

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.5.8 getXAngle

功能说明

返回后视镜的横向角度。

函数原型

```
uint8 getXAngle()
```

参数说明

无。

返回值说明

类型	描述
uint8	当前后视镜的横向角度，单位：%，255 代表未知

3.5.9 startAdjustXAngle

功能说明

启动后视镜横向角度的步进式调节。

函数原型

```
BCM_ReturnCode startAdjustXAngle(BCM_StepActionType actionType)
```

参数说明

参数名称	类型	描述
actionType	BCM_StepActionType	步进调节类型

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.5.10 stopAdjustXAngle

功能说明

停止后视镜横向角度调节。

函数原型

```
BCM_ReturnCode stopAdjustXAngle()
```

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.5.11 notifyXAngle

功能说明

上报后视镜横向角度变化。

函数原型

```
uint8 notifyXAngle()
```

参数说明

无。

返回值说明

类型	描述
uint8	当前后视镜横向角度，单位：%，255 代表未知

3.5.12 adjustYAngle

功能说明

调整后视镜的纵向角度。

函数原型

```
BCM_ReturnCode adjustYAngle(uint8 yAngle)
```

参数说明

参数名称	类型	描述
yAngle	uint8	后视镜纵轴旋转的目标位置，单位：最大转角的 1%

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.5.13 getYAngle

功能说明

返回后视镜的纵向角度。

函数原型

```
uint8 getYAngle()
```

参数说明

无。

返回值说明

类型	描述
uint8	指定后视镜的纵向角度，单位：%，255 代表未知

3.5.14 startAdjustYAngle

功能说明

启动后视镜纵向角度的步进式调节。

函数原型

```
BCM_ReturnCode startAdjustYAngle(BCM_StepActionType actionType)
```

参数说明

参数名称	类型	描述
actionType	BCM_StepActionType	步进调节类型

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.5.15 stopAdjustYAngle

功能说明

停止后视镜纵向角度调节。

函数原型

```
BCM_ReturnCode stopAdjustYAngle()
```

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.5.16 notifyYAngle

功能说明

上报后视镜纵向角度变化。

函数原型

```
uint8 notifyYAngle()
```

参数说明

无。

返回值说明

类型	描述
uint8	当前后视镜纵向角度， 单位： %， 255 代表未知

3.5.17 unfold

功能说明

展开后视镜。

函数原型

BCM_ReturnCode unfold()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.5.18 fold

功能说明

折叠后视镜。

函数原型

BCM_ReturnCode fold()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.5.19 isFolded

功能说明

返回后视镜是否处于折叠状态。

函数原型

boolean isFolded()

参数说明

无。

返回值说明

类型	描述
boolean	后视镜已折叠返回 TRUE， 否则返回 FALSE

3.5.20 notifyFoldedStatus

功能说明

上报后视镜折叠完成状态。

函数原型

BCM_FoldedStatus notifyFoldedStatus()

参数说明

无。

返回值说明

类型	描述
BCM_FoldedStatus	当前后视镜折叠的状态

3.5.21 adjustAntiGlare

功能说明

调整后视镜的防眩程度。

函数原型

BCM_ReturnCode adjustAntiGlare(BCM_AntiGlareMode antiGlareMode)

参数说明

参数名称	类型	描述
antiGlareMode	BCM_AntiGlareMode	后视镜防眩和执行模式

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.5.22 getAntiGlare

功能说明

返回后视镜的防眩程度。

函数原型

```
uint8 getAntiGlare()
```

参数说明

无。

返回值说明

类型	描述
uint8	后视镜防眩程度，单位：%，255 代表未知

3.6 BCM_Seat(座椅服务)

提供座椅的控制和状态服务，每一个座椅一个原子服务。

BCM_Seat(座椅服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
BCM_ReturnCode	adjustMainXDir(uint8 mainXDir)	调整座椅的前后位置。
uint8	getMainXDir()	返回座椅的前后位置。

类型	API 函数	功能说明
BCM_ReturnCode	startAdjustMainXDir(BCM_StepActionType actionType)	启动座椅前后方向的步进式调节。
BCM_ReturnCode	stopAdjustMainXDir()	停止座椅前后方向调节。
BCM_ActuateStatus	notifyXActuateStatus()	上报座椅前后方向调节的执行状态。
BCM_ReturnCode	adjustMainZDir(uint8 mainZDir)	调整座椅主高度位置。
uint8	getMainZDir()	返回座椅主高度位置。
BCM_ReturnCode	startAdjustMainZDir(BCM_StepActionType actionType)	启动座椅主高度的步进式调节。
BCM_ReturnCode	stopAdjustMainZDir()	停止座椅主高度方向调节。
BCM_ActuateStatus	notifyMZActuateStatus()	上报座椅主高度调节的执行状态。
BCM_ReturnCode	adjustFrontZDir(uint8 frontZDir)	调整座椅的坐垫前部位高度。
uint8	getFrontZDir()	返回座椅的坐垫前部位高度。
BCM_ReturnCode	startAdjustFrontZDir(BCM_StepActionType actionType)	启动座椅坐垫前高度的步进式调节。
BCM_ReturnCode	stopAdjustFrontZDir()	停止座椅坐垫前高度方向调节。
BCM_ActuateStatus	notifyFZActuateStatus()	上报座椅坐垫前高度调节的执行状态。
BCM_ReturnCode	adjustBackRestAngle(uint8 backRestAngle)	调整座椅的座椅靠背倾角。
uint8	getBackRestAngle()	返回座椅的座椅靠背倾角。
BCM_ReturnCode	startAdjustBackRestAngle(BCM_StepActionType actionType)	启动座椅靠背角度的步进式调节。
BCM_ReturnCode	stopAdjustBackRestAngle()	停止座椅靠背角度调节。
BCM_ActuateStatus	notifyBackActuateStatus()	上报座椅靠背角度调节的执行状态。
boolean	isOccupied()	返回座椅是否被占用。

类型	API 函数	功能说明
BCM_OccupiedStatus	getOccupiedStatus()	返回座椅占用状态。
BCM_OccupiedStatus	notifyOccupiedStatus()	上报座椅被占用状态。

3.6.1 adjustMainXDir

功能说明

调整座椅的前后位置。

函数原型

BCM_ReturnCode adjustMainXDir(uint8 mainXDir)

参数说明

参数名称	类型	描述
mainXDir	uint8	座椅前后方向上的目标位置，单位：最大行程的 1%

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.6.2 getMainXDir

功能说明

返回座椅的前后位置。

函数原型

uint8 getMainXDir()

参数说明

无。

返回值说明

类型	描述
uint8	当前座椅前后方向上的位置，单位：%，255 代表未知

3.6.3 startAdjustMainXDir

功能说明

启动座椅前后方向的步进式调节。

函数原型

```
BCM_ReturnCode startAdjustMainXDir(BCM_StepActionType actionType)
```

参数说明

参数名称	类型	描述
actionType	BCM_StepActionType	步进调节类型

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.6.4 stopAdjustMainXDir

功能说明

停止座椅前后方向调节。

函数原型

```
BCM_ReturnCode stopAdjustMainXDir()
```

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.6.5 notifyXActuateStatus

功能说明

上报座椅前后方向调节的执行状态。

函数原型

```
BCM_ActuateStatus notifyXActuateStatus()
```

参数说明

无。

返回值说明

类型	描述
BCM_ActuateStatus	当前座椅前后方向调节的执行状态

3.6.6 adjustMainZDir

功能说明

调整座椅主高度位置。

函数原型

```
BCM_ReturnCode adjustMainZDir(uint8 mainZDir)
```

参数说明

参数名称	类型	描述
mainZDir	uint8	座椅上下方向上的目标位置，单位：最大行程的 1%

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.6.7 getMainZDir

功能说明

返回座椅主高度位置。

函数原型

```
uint8 getMainZDir()
```

参数说明

无。

返回值说明

类型	描述
uint8	当前座椅上下方向上的位置，单位：%，255 代表未知

3.6.8 startAdjustMainZDir

功能说明

启动座椅主高度的步进式调节。

函数原型

```
BCM_ReturnCode startAdjustMainZDir(BCM_StepActionType actionType)
```

参数说明

参数名称	类型	描述
actionType	BCM_StepActionType	步进调节类型

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.6.9 stopAdjustMainZDir

功能说明

停止座椅主高度方向调节。

函数原型

```
BCM_ReturnCode stopAdjustMainZDir()
```

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.6.10 notifyMZActuateStatus

功能说明

上报座椅主高度调节的执行状态。

函数原型

```
BCM_ActuateStatus notifyMZActuateStatus()
```

参数说明

无。

返回值说明

类型	描述
BCM_ActuateStatus	当前座椅主高度调节的执行状态

3.6.11 adjustFrontZDir

功能说明

调整座椅的坐垫前部位高度。

函数原型

```
BCM_ReturnCode adjustFrontZDir(uint8 frontZDir)
```

参数说明

参数名称	类型	描述
frontZDir	uint8	座椅坐垫前部位高度方向的目标位置，单位：最大行程的 1%

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.6.12 getFrontZDir

功能说明

返回座椅的坐垫前部位高度。

函数原型

```
uint8 getFrontZDir()
```

参数说明

无。

返回值说明

类型	描述
uint8	当前座椅坐垫前部位高度方向的位置，单位：%

3.6.13 startAdjustFrontZDir

功能说明

启动座椅坐垫前高度的步进式调节。

函数原型

BCM_ReturnCode startAdjustFrontZDir(BCM_StepActionType actionType)

参数说明

参数名称	类型	描述
actionType	BCM_StepActionType	步进调节类型

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.6.14 stopAdjustFrontZDir

功能说明

停止座椅坐垫前高度方向调节。

函数原型

BCM_ReturnCode stopAdjustFrontZDir()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.6.15 notifyFZActuateStatus

功能说明

上报座椅坐垫前高度调节的执行状态。

函数原型

```
BCM_ActuateStatus notifyFZActuateStatus()
```

参数说明

无。

返回值说明

类型	描述
BCM_ActuateStatus	当前座椅坐垫前高度调节的执行状态

3.6.16 adjustBackRestAngle

功能说明

调整座椅的座椅靠背倾角。

函数原型

```
BCM_ReturnCode adjustBackRestAngle(uint8 backRestAngle)
```

参数说明

参数名称	类型	描述
backRestAngle	uint8	靠背的目标倾角，单位：最大行程的 1%

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.6.17 getBackRestAngle

功能说明

返回座椅的座椅靠背倾角。

函数原型

```
uint8 getBackRestAngle()
```

参数说明

无。

返回值说明

类型	描述
uint8	当前靠背的角度位置，单位：%，255 代表未知

3.6.18 startAdjustBackRestAngle

功能说明

启动座椅靠背角度的步进式调节。

函数原型

```
BCM_ReturnCode startAdjustBackRestAngle(BCM_StepActionType actionType)
```

参数说明

参数名称	类型	描述
actionType	BCM_StepActionType	步进调节类型

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.6.19 stopAdjustBackRestAngle

功能说明

停止座椅靠背角度调节。

函数原型

```
BCM_ReturnCode stopAdjustBackRestAngle()
```

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.6.20 notifyBackActuateStatus

功能说明

上报座椅靠背角度调节的执行状态。

函数原型

```
BCM_ActuateStatus notifyBackActuateStatus()
```

参数说明

无。

返回值说明

类型	描述
BCM_ActuateStatus	当前座椅靠背角度调节的执行状态

3.6.21 isOccupied

功能说明

返回座椅是否被占用。

函数原型

boolean isOccupied()

参数说明

无。

返回值说明

类型	描述
boolean	座椅被占用返回 TRUE，否则返回 FALSE

3.6.22 getOccupiedStatus

功能说明

返回座椅占用状态。

函数原型

BCM_OccupiedStatus getOccupiedStatus()

参数说明

无。

返回值说明

类型	描述
BCM_OccupiedStatus	座椅被占用状态

3.6.23 notifyOccupiedStatus

功能说明

上报座椅被占用状态。

函数原型

BCM_OccupiedStatus notifyOccupiedStatus()

参数说明

无。

返回值说明

类型	描述
BCM_OccupiedStatus	座椅被占用状态

3.7 BCM_SeatExtended(座椅扩展服务)

提供座椅的扩展功能的控制和状态服务，每一个座椅一个原子服务。

BCM_SeatExtended(座椅扩展服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
BCM_ReturnCode	adjustPanLength(uint8 frontZDir)	调整座椅坐盆长度。
uint8	getPanLength()	返回座椅坐盆长度。
BCM_ReturnCode	startAdjustPanLength(BCM_StepActionType actionType)	启动座盆长度的步进式调节。
BCM_ReturnCode	stopAdjustPanLength()	停止座椅座盆长度调节。
BCM_ReturnCode	adjustPillowZDir(uint8 pillowZDir)	调整座椅枕托高度。
uint8	getPillowZDir()	返回座椅枕托高度。
BCM_ReturnCode	startAdjustPillowZDir(BCM_StepActionType actionType)	启动座椅枕托高度的步进式调节。
BCM_ReturnCode	stopAdjustPillowZDir()	停止座椅枕托高度方向调节。
BCM_ReturnCode	adjustLumbarZDir(uint8 lumbarZDir)	调整座椅腰托高度。
uint8	getLumbarZDir()	返回座椅腰托高度。
BCM_ReturnCode	startAdjustLumbarZDir(BCM_StepActionType actionType)	启动座椅腰托上下高度的步进式调节。
BCM_ReturnCode	stopAdjustLumbarZDir()	停止座椅腰托上下高度方向调节。

类型	API 函数	功能说明
ode		
BCM_ReturnCode	adjustLumbarXDir(uint8 lumbarXDir)	调整座椅腰托突出高度。
uint8	getLumbarXDir()	返回座椅腰托突出高度。
BCM_ReturnCode	startAdjustLumbarXDir(BCM_StepActionType actionType)	启动座椅腰托突出高度的步进式调节。
BCM_ReturnCode	stopAdjustLumbarXDir()	停止座椅腰托突出高度调节。
BCM_ReturnCode	adjustLegRestAngle(uint8 legRestAngle)	调整座椅的腿托倾斜角度。
uint8	getLegRestAngle()	返回座椅腿托倾斜角度。
BCM_ReturnCode	startAdjustLegRestAngle(BCM_StepActionType actionType)	启动座椅腿托倾斜角度的步进式调节。
BCM_ReturnCode	stopAdjustLegRestAngle()	停止座椅腿托倾斜角度调节。
BCM_ReturnCode	adjustLegRestXDir(uint8 legRestXDir)	调整座椅腿托长度。
uint8	getLegRestXDir()	返回座椅腿托长度。
BCM_ReturnCode	startAdjustLegRestXDir(BCM_StepActionType actionType)	启动座椅腿托长度的步进式调节。
BCM_ReturnCode	stopAdjustLegRestXDir()	停止座椅腿托长度调节。
BCM_ReturnCode	adjustFootRestAngle(uint8 footRestAngle)	调整座椅脚托角度。
uint8	getFootRestAngle()	返回座椅脚托角度。
BCM_ReturnCode	startAdjustFootRestAngle(BCM_StepActionType actionType)	启动座椅脚托角度的步进式调节。
BCM_ReturnCode	stopAdjustFootRestAngle()	停止座椅脚托角度调节。

3.7.1 adjustPanLength

功能说明

调整座椅坐盆长度。

函数原型

```
BCM_ReturnCode adjustPanLength(uint8 frontZDir)
```

参数说明

参数名称	类型	描述
frontZDir	uint8	坐垫长度的目标位置，单位：最大行程的 1%

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.7.2 getPanLength

功能说明

返回座椅坐盆长度。

函数原型

```
uint8 getPanLength()
```

参数说明

无。

返回值说明

类型	描述
uint8	当前座椅坐盆长度的位置，单位：%，255 代表未知

3.7.3 startAdjustPanLength

功能说明

启动座盆长度的步进式调节。

函数原型

```
BCM_ReturnCode startAdjustPanLength(BCM_StepActionType actionType)
```

参数说明

参数名称	类型	描述
actionType	BCM_StepActionType	步进调节类型

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.7.4 stopAdjustPanLength

功能说明

停止座椅座盆长度调节。

函数原型

```
BCM_ReturnCode stopAdjustPanLength()
```

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.7.5 adjustPillowZDir

功能说明

调整座椅枕托高度。

函数原型

```
BCM_ReturnCode adjustPillowZDir(uint8 pillowZDir)
```

参数说明

参数名称	类型	描述
pillowZDir	uint8	定义枕托高度的调整角度，单位：最大行程的百分比

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.7.6 getPillowZDir

功能说明

返回座椅枕托高度。

函数原型

```
uint8 getPillowZDir()
```

参数说明

无。

返回值说明

类型	描述
uint8	当前座椅枕托高度的位置，单位：%，255 代表未知

3.7.7 startAdjustPillowZDir

功能说明

启动座椅枕托高度的步进式调节。

函数原型

```
BCM_ReturnCode startAdjustPillowZDir(BCM_StepActionType actionType)
```

参数说明

参数名称	类型	描述
actionType	BCM_StepActionType	步进调节类型

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.7.8 stopAdjustPillowZDir

功能说明

停止座椅枕托高度方向调节。

函数原型

```
BCM_ReturnCode stopAdjustPillowZDir()
```

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.7.9 adjustLumbarZDir

功能说明

调整座椅腰托高度。

函数原型

```
BCM_ReturnCode adjustLumbarZDir(uint8 lumbarZDir)
```

参数说明

参数名称	类型	描述
lumbarZDir	uint8	腰托高度调整的目标位置，单位：最大行程的百分比

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.7.10 getLumbarZDir

功能说明

返回座椅腰托高度。

函数原型

```
uint8 getLumbarZDir()
```

参数说明

无。

返回值说明

类型	描述
uint8	当前腰托高度的位置，单位：%，255 代表未知

3.7.11 startAdjustLumbarZDir

功能说明

启动座椅腰托上下高度的步进式调节。

函数原型

BCM_ReturnCode startAdjustLumbarZDir(BCM_StepActionType actionType)

参数说明

参数名称	类型	描述
actionType	BCM_StepActionType	步进调节类型

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.7.12 stopAdjustLumbarZDir

功能说明

停止座椅腰托上下高度方向调节。

函数原型

BCM_ReturnCode stopAdjustLumbarZDir()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.7.13 adjustLumbarXDir

功能说明

调整座椅腰托突出高度。

函数原型

```
BCM_ReturnCode adjustLumbarXDir(uint8 lumbarXDir)
```

参数说明

参数名称	类型	描述
lumbarXDir	uint8	腰托突出调整的目标高度，单位：最大行程的百分比

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.7.14 getLumbarXDir

功能说明

返回座椅腰托突出高度。

函数原型

```
uint8 getLumbarXDir()
```

参数说明

无。

返回值说明

类型	描述
uint8	当前座椅腰托的突出高度，单位：%，255 代表未知

3.7.15 startAdjustLumbarXDir

功能说明

启动座椅腰托突出高度的步进式调节。

函数原型

BCM_ReturnCode startAdjustLumbarXDir(BCM_StepActionType actionType)

参数说明

参数名称	类型	描述
actionType	BCM_StepActionType	步进调节类型

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.7.16 stopAdjustLumbarXDir

功能说明

停止座椅腰托突出高度调节。

函数原型

BCM_ReturnCode stopAdjustLumbarXDir()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.7.17 adjustLegRestAngle

功能说明

调整座椅的腿托倾斜角度。

函数原型

```
BCM_ReturnCode adjustLegRestAngle(uint8 legRestAngle)
```

参数说明

参数名称	类型	描述
legRestAngle	uint8	腿拖调整的目标倾斜角度

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.7.18 getLegRestAngle

功能说明

返回座椅腿托倾斜角度。

函数原型

```
uint8 getLegRestAngle()
```

参数说明

无。

返回值说明

类型	描述
uint8	当前座椅腿托的倾斜角度，单位：%，255 代表未知

3.7.19 startAdjustLegRestAngle

功能说明

启动座椅腿托倾斜角度的步进式调节。

函数原型

BCM_ReturnCode startAdjustLegRestAngle(BCM_StepActionType actionType)

参数说明

参数名称	类型	描述
actionType	BCM_StepActionType	步进调节类型

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.7.20 stopAdjustLegRestAngle

功能说明

停止座椅腿托倾斜角度调节。

函数原型

BCM_ReturnCode stopAdjustLegRestAngle()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.7.21 adjustLegRestXDir

功能说明

调整座椅腿托长度。

函数原型

```
BCM_ReturnCode adjustLegRestXDir(uint8 legRestXDir)
```

参数说明

参数名称	类型	描述
legRestXDir	uint8	腿拖调整的目标长度

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.7.22 getLegRestXDir

功能说明

返回座椅腿托长度。

函数原型

```
uint8 getLegRestXDir()
```

参数说明

无。

返回值说明

类型	描述
uint8	当前座椅腿托的长度，单位：%，255 代表未知

3.7.23 startAdjustLegRestXDir

功能说明

启动座椅腿托长度的步进式调节。

函数原型

BCM_ReturnCode startAdjustLegRestXDir(BCM_StepActionType actionType)

参数说明

参数名称	类型	描述
actionType	BCM_StepActionType	步进调节类型

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.7.24 stopAdjustLegRestXDir

功能说明

停止座椅腿托长度调节。

函数原型

BCM_ReturnCode stopAdjustLegRestXDir()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.7.25 adjustFootRestAngle

功能说明

调整座椅脚托角度。

函数原型

```
BCM_ReturnCode adjustFootRestAngle(uint8 footRestAngle)
```

参数说明

参数名称	类型	描述
footRestAngle	uint8	脚托调整的目标角度

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.7.26 getFootRestAngle

功能说明

返回座椅脚托角度。

函数原型

```
uint8 getFootRestAngle()
```

参数说明

无。

返回值说明

类型	描述
uint8	当前座椅脚托的角度，单位：%

3.7.27 startAdjustFootRestAngle

功能说明

启动座椅脚托角度的步进式调节。

函数原型

BCM_ReturnCode startAdjustFootRestAngle(BCM_StepActionType actionType)

参数说明

参数名称	类型	描述
actionType	BCM_StepActionType	步进调节类型

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.7.28 stopAdjustFootRestAngle

功能说明

停止座椅脚托角度调节。

函数原型

BCM_ReturnCode stopAdjustFootRestAngle()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.8 BCM_CushionTemp(座椅通风加热服务)

提供座椅通风加热的控制和状态服务，每一个通风加热座椅一个原子服务。

BCM_CushionTemp(座椅通风加热服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
BCM_ReturnCode	stopVentilation()	停止座椅通风。
BCM_ReturnCode	startVentilation()	启动座椅通风。
BCM_ReturnCode	setVentilationLevel(uint8 ventilationLevel)	设置座椅通风等级。
boolean	isVentilating()	返回座椅是否正在通风。
uint8	getVentilationLevel()	返回座椅通风等级。
boolean	notifyVentilatingStatus()	上报坐垫通风的当前状态。
BCM_ReturnCode	startHeating()	启动座椅加热。
BCM_ReturnCode	stopHeating()	停止座椅加热。
BCM_ReturnCode	setHeatingLevel(uint8 heatingLevel)	设置座椅加热的加热等级。
boolean	isHeating()	返回座椅是否正在加热。
uint8	getHeatingLevel()	返回座椅加热的加热等级。
boolean	notifyHeatingStatus()	上报座椅加热的当前状态。

3.8.1 stopVentilation

功能说明

停止座椅通风。

函数原型

```
BCM_ReturnCode stopVentilation()
```

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.8.2 startVentilation

功能说明

启动座椅通风。

函数原型

```
BCM_ReturnCode startVentilation()
```

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.8.3 setVentilationLevel

功能说明

设置座椅通风等级。

函数原型

```
BCM_ReturnCode setVentilationLevel(uint8 ventilationLevel)
```

参数说明

参数名称	类型	描述
ventilationLevel	uint8	座椅通风的目标等级，单位：最大风量的 1%

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.8.4 isVentilating

功能说明

返回座椅是否正在通风。

函数原型

```
boolean isVentilating()
```

参数说明

无。

返回值说明

类型	描述
boolean	座椅正在通风返回 TRUE， 否则返回 FALSE

3.8.5 getVentilationLevel

功能说明

返回座椅通风等级。

函数原型

```
uint8 getVentilationLevel()
```

参数说明

无。

返回值说明

类型	描述
uint8	当前座椅的通风级别， 单位：%， 255 代表未知

3.8.6 notifyVentilatingStatus

功能说明

上报坐垫通风的当前状态。

函数原型

```
boolean notifyVentilatingStatus()
```

参数说明

无。

返回值说明

类型	描述
boolean	当前坐垫通风状态，正在通风为 TRUE，否则为 FALSE

3.8.7 startHeating

功能说明

启动座椅加热。

函数原型

```
BCM_ReturnCode startHeating()
```

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.8.8 stopHeating

功能说明

停止座椅加热。

函数原型

BCM_ReturnCode stopHeating()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.8.9 setHeatingLevel

功能说明

设置座椅加热的加热等级。

函数原型

BCM_ReturnCode setHeatingLevel(uint8 heatingLevel)

参数说明

参数名称	类型	描述
heatingLevel	uint8	座椅加热的目标等级，单位：最大功率的 1%

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.8.10 isHeating

功能说明

返回座椅是否正在加热。

函数原型

boolean isHeating()

参数说明

无。

返回值说明

类型	描述
boolean	座椅正在加热返回 TRUE， 否则返回 FALSE

3.8.11 getHeatingLevel

功能说明

返回座椅加热的加热等级。

函数原型

uint8 getHeatingLevel()

参数说明

无。

返回值说明

类型	描述
uint8	当前座椅的加热级别， 单位：%， 255 代表未知

3.8.12 notifyHeatingStatus

功能说明

上报座椅加热的当前状态。

函数原型

boolean notifyHeatingStatus()

参数说明

无。

返回值说明

类型	描述
boolean	当前座椅加热状态，正在加热为 TRUE，否则为 FALSE

3.9 BCM_Message(座椅按摩服务)

提供座椅按摩的控制和状态服务，每一个座椅一个原子服务。

BCM_Message(座椅按摩服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
BCM_ReturnCode	stopMessage()	停止座椅按摩。
BCM_ReturnCode	startMessage(BCM_MessageMode messageMode)	启动座椅按摩。

3.9.1 stopMessage

功能说明

停止座椅按摩。

函数原型

```
BCM_ReturnCode stopMessage()
```

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.9.2 startMassage

功能说明

启动座椅按摩。

函数原型

BCM_ReturnCode startMassage(BCM_MessageMode messageMode)

参数说明

参数名称	类型	描述
messageMode	BCM_MessageMode	定义本次按摩操作的实施模式

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.10 BCM_SteerWheel(方向盘辅助功能服务)

提供方向盘的控制和状态服务。

BCM_SteerWheel(方向盘辅助功能服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
BCM_ReturnCode	adjustTilt(uint8 Tilt)	调整方向盘倾斜角度。
uint8	getTilt()	返回方向盘倾斜角度。
BCM_ReturnCode	startAdjustTilt(BCM_SteerWheelActionType actionType)	启动方向盘倾斜角度的步进式调节。
BCM_ReturnCode	stopAdjustTilt()	停止方向盘倾斜角度调节。
BCM_ReturnCode	adjustHeight(uint8 height)	调整方向盘高度。
uint8	getHeight()	返回方向盘高度。

类型	API 函数	功能说明
BCM_ReturnCode	startAdjustHeight(BCM_StepActionType actionType)	启动方向盘高度的步进式调节。
BCM_ReturnCode	stopAdjustHeight()	停止方向盘高度调节。
uint8	notifyTilt()	上报方向盘倾斜角度。
uint8	notifyHeight()	上报方向盘高度。
BCM_ReturnCode	startHeating()	启动方向盘加热。
BCM_ReturnCode	stopHeating()	停止方向盘加热。
BCM_ReturnCode	setHeatingLevel(uint8 heatingLevel)	设置方向盘加热等级。
uint8	getHeatingLevel()	返回方向盘加热等级。
boolean	isHeating()	返回方向盘是否正在加热。
boolean	notifyHeatingStatus()	上报方向盘的加热状态。

3.10.1 adjustTilt

功能说明

调整方向盘倾斜角度。

函数原型

BCM_ReturnCode adjustTilt(uint8 Tilt)

参数说明

参数名称	类型	描述
Tilt	uint8	方向盘的目标倾斜角度，单位：最大倾斜度的 1%

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.10.2 getTilt

功能说明

返回方向盘倾斜角度。

函数原型

```
uint8 getTilt()
```

参数说明

无。

返回值说明

类型	描述
uint8	当前方向盘的倾斜位置 单位：%，255 代表未知

3.10.3 startAdjustTilt

功能说明

启动方向盘倾斜角度的步进式调节。

函数原型

```
BCM_ReturnCode startAdjustTilt(BCM_StepActionType actionType)
```

参数说明

参数名称	类型	描述
actionType	BCM_StepActionType	步进调节类型

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.10.4 stopAdjustTilt

功能说明

停止方向盘倾斜角度调节。

函数原型

```
BCM_ReturnCode stopAdjustTilt()
```

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.10.5 adjustHeight

功能说明

调整方向盘高度。

函数原型

```
BCM_ReturnCode adjustHeight(uint8 height)
```

参数说明

参数名称	类型	描述
height	uint8	方向盘的目标高度，单位：最大高度的 1%

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.10.6 getHeight

功能说明

返回方向盘高度。

函数原型

```
uint8 getHeight()
```

参数说明

无。

返回值说明

类型	描述
uint8	当前方向盘的高度，单位：%，255 代表未知

3.10.7 startAdjustHeight

功能说明

启动方向盘高度的步进式调节。

函数原型

```
BCM_ReturnCode startAdjustHeight(BCM_StepActionType actionType)
```

参数说明

参数名称	类型	描述
actionType	BCM_StepActionType	步进调节类型

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.10.8 stopAdjustHeight

功能说明

停止方向盘高度调节。

函数原型

```
BCM_ReturnCode stopAdjustHeight()
```

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.10.9 notifyTilt

功能说明

上报方向盘倾斜角度。

函数原型

```
uint8 notifyTilt()
```

参数说明

无。

返回值说明

类型	描述
uint8	当前方向盘的倾斜角度， 单位： %

3.10.10 notifyHeight

功能说明

上报方向盘高度。

函数原型

uint8 notifyHeight()

参数说明

无。

返回值说明

类型	描述
uint8	当前方向盘的高度，单位：%

3.10.11 startHeating

功能说明

启动方向盘加热。

函数原型

BCM_ReturnCode startHeating()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.10.12 stopHeating

功能说明

停止方向盘加热。

函数原型

BCM_ReturnCode stopHeating()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.10.13 setHeatingLevel

功能说明

设置方向盘加热等级。

函数原型

```
BCM_ReturnCode setHeatingLevel(uint8 heatingLevel)
```

参数说明

参数名称	类型	描述
heatingLevel	uint8	方向盘的加热等级，单位：最大功率的 1%

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.10.14 getHeatingLevel

功能说明

返回方向盘加热等级。

函数原型

```
uint8 getHeatingLevel()
```

参数说明

无。

返回值说明

类型	描述
uint8	当前方向盘的加热级别

3.10.15 isHeating

功能说明

返回方向盘是否正在加热。

函数原型

```
boolean isHeating()
```

参数说明

无。

返回值说明

类型	描述
boolean	方向盘正在加热返回 TRUE，否则返回 FALSE

3.10.16 notifyHeatingStatus

功能说明

上报方向盘的加热状态。

函数原型

```
boolean notifyHeatingStatus()
```

参数说明

无。

返回值说明

类型	描述
boolean	当前方向盘加热的状态，正在加热为 TRUE，否则为 FALSE

3.11 BCM_SunRoof(天窗服务)

提供天窗的控制和状态服务。

BCM_SunRoof(天窗服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
BCM_ReturnCode	adjustPosition(uint8 position)	调整天窗的天窗开启位置。
uint8	getPosition()	返回天窗的天窗开启位置。
BCM_ReturnCode	adjustAngle(uint8 angle)	调整天窗的天窗翘起位置。
uint8	getAngle()	返回天窗的天窗翘起位置。
BCM_ReturnCode	open()	打开天窗。
BCM_ReturnCode	close()	关闭天窗。
BCM_ReturnCode	stop()	停止开启或关闭天窗。
BCM_ActuateStatus	notifyActuateStatus()	上报天窗执行状态。
BCM_LearnedStatus	getLearnedStatus()	返回电动天窗的学习状态。
BCM_AntiPinchStatus	notifyAntiPinch()	上报防夹状态。
BCM_IceBreakingStatus	notifyIceBreaking()	上报破冰模式。
uint8	notifyPosition()	上报天窗开度变化。

3.11.1 adjustPosition

功能说明

调整天窗的天窗开启位置。

函数原型

BCM_ReturnCode adjustPosition(uint8 position)

参数说明

参数名称	类型	描述
position	uint8	天窗开启位置，单位：最大行程的 1%

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.11.2 getPosition

功能说明

返回天窗的天窗开启位置。

函数原型

```
uint8 getPosition()
```

参数说明

无。

返回值说明

类型	描述
uint8	输出天窗开启位置，单位：%，255 代表未知

3.11.3 adjustAngle

功能说明

调整天窗的天窗翘起位置。

函数原型

```
BCM_ReturnCode adjustAngle(uint8 angle)
```

参数说明

参数名称	类型	描述
angle	uint8	天窗的翘起位置，单位：最大行程的 1%

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.11.4 getAngle

功能说明

返回天窗的天窗翘起位置。

函数原型

```
uint8 getAngle()
```

参数说明

无。

返回值说明

类型	描述
uint8	当前天窗的翘起位置，单位：%，255 代表未知

3.11.5 open

功能说明

打开天窗。

函数原型

```
BCM_ReturnCode open()
```

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.11.6 close

功能说明

关闭天窗。

函数原型

BCM_ReturnCode close()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.11.7 stop

功能说明

停止开启或关闭天窗。

函数原型

BCM_ReturnCode stop()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.11.8 notifyActuateStatus

功能说明

上报天窗执行状态。

函数原型

```
BCM_ActuateStatus notifyActuateStatus()
```

参数说明

无。

返回值说明

类型	描述
BCM_ActuateStatus	当前天窗开关的执行状态

3.11.9 getLearnedStatus

功能说明

返回电动天窗的学习状态。

函数原型

```
BCM_LearnedStatus getLearnedStatus()
```

参数说明

无。

返回值说明

类型	描述
BCM_LearnedStatus	当前电动车窗的学习状态

3.11.10 notifyAntiPinch

功能说明

上报防夹状态。

函数原型

BCM_AntiPinchStatus notifyAntiPinch()

参数说明

无。

返回值说明

类型	描述
BCM_AntiPinchStatus	当前的防夹状态

3.11.11 notifyIceBreaking

功能说明

上报破冰模式。

函数原型

BCM_IceBreakingStatus notifyIceBreaking()

参数说明

无。

返回值说明

类型	描述
BCM_IceBreakingStatus	当前的破冰模式状态

3.11.12 notifyPosition

功能说明

上报天窗开度变化。

函数原型

uint8 notifyPosition()

参数说明

无。

返回值说明

类型	描述
uint8	当前天窗的开度，单位：%，255 代表未知

3.12 BCM_WelcomePedal(迎宾踏板服务)

提供迎宾踏板的控制和状态服务，每一个迎宾踏板一个原子服务。

BCM_WelcomePedal(迎宾踏板服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
BCM_ReturnCode	fold()	收起迎宾踏板。
BCM_ReturnCode	unfold()	展开迎宾踏板。
uint8	notifyPosition()	上报迎宾踏板的折叠程度。
BCM_ReturnCode	adjustPosition(uint8 pedalPosition)	调整迎宾踏板的伸出位置。
uint8	getPosition()	返回迎宾踏板的踏板伸出位置。

3.12.1 fold

功能说明

收起迎宾踏板。

函数原型

```
BCM_ReturnCode fold()
```

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.12.2 unfold

功能说明

展开迎宾踏板。

函数原型

BCM_ReturnCode unfold()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.12.3 notifyPosition

功能说明

上报迎宾踏板的折叠程度。

函数原型

uint8 notifyPosition()

参数说明

无。

返回值说明

类型	描述
uint8	当前迎宾踏板的折叠程度， 单位：%， 255 代表 unkown

3.12.4 adjustPosition

功能说明

调整迎宾踏板的伸出位置。

函数原型

```
BCM_ReturnCode adjustPosition(uint8 pedalPosition)
```

参数说明

参数名称	类型	描述
pedalPosition	uint8	迎宾踏板调整的目标位置，单位：最大行程的 1%

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.12.5 getPosition

功能说明

返回迎宾踏板的踏板伸出位置。

函数原型

```
uint8 getPosition()
```

参数说明

无。

返回值说明

类型	描述
uint8	迎宾踏板当前位置，单位：%，255 代表未知

3.13 BCM_Window(车窗服务)

提供车窗的控制和状态服务，每一个车窗一个原子服务。

BCM_Window(车窗服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
BCM_ReturnCode	lock()	闭锁车窗的窗锁。
BCM_ReturnCode	unlock()	解锁车窗的窗锁。
BCM_LockStatus_Window	getLockStatus()	返回车窗的窗锁状态。
BCM_LockStatus_Window	notifyLockStatus()	上报车窗的锁状态。
BCM_ReturnCode	adjustPosition(uint8 position)	调整车窗的车窗开度。
uint8	getPosition()	返回车窗的车窗开度。
BCM_ReturnCode	open()	打开车窗。
BCM_ReturnCode	close()	关闭车窗。
BCM_ReturnCode	stop()	停止开启或关闭车窗。
BCM_LearnedStatus	getLearnedStatus()	返回车窗的学习状态。
BCM_AntiPinchStatus	notifyAntiPinch()	上报防夹状态。
BCM_ActuateStatus	notifyActuateStatus()	上报车窗执行状态。
BCM_IceBreakingStatus	notifyIceBreaking()	上报破冰模式。
uint8	notifyPosition()	上报车窗开度变化。

3.13.1 lock

功能说明

闭锁车窗的窗锁。

函数原型

BCM_ReturnCode lock()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.13.2 unlock

功能说明

解锁车窗的窗锁。

函数原型

BCM_ReturnCode unlock()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.13.3 getLockStatus

功能说明

返回车窗的窗锁状态。

函数原型

BCM_LockStatus_Window getLockStatus()

参数说明

无。

返回值说明

类型	描述
BCM_LockStatus_Window	当前车窗的上锁状态

3.13.4 notifyLockStatus

功能说明

上报车窗的锁状态。

函数原型

BCM_LockStatus_Window notifyLockStatus()

参数说明

无。

返回值说明

类型	描述
BCM_LockStatus_Window	当前车窗的锁状态

3.13.5 adjustPosition

功能说明

调整车窗的车窗开度。

函数原型

BCM_ReturnCode adjustPosition(uint8 position)

参数说明

参数名称	类型	描述
position	uint8	车窗玻璃开闭的目标开度，单位：最大开度的 1%

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.13.6 getPosition

功能说明

返回车窗的车窗开度。

函数原型

```
uint8 getPosition()
```

参数说明

无。

返回值说明

类型	描述
uint8	当前天窗的开度，单位：%，255 代表未知

3.13.7 open

功能说明

打开车窗。

函数原型

```
BCM_ReturnCode open()
```

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.13.8 close

功能说明

关闭车窗。

函数原型

BCM_ReturnCode close()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.13.9 stop

功能说明

停止开启或关闭车窗。

函数原型

BCM_ReturnCode stop()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.13.10 getLearnedStatus

功能说明

返回车窗的学习状态。

函数原型

BCM_LearnedStatus getLearnedStatus()

参数说明

无。

返回值说明

类型	描述
BCM_LearnedStatus	当前车窗的学习状态

3.13.11 notifyAntiPinch

功能说明

上报防夹状态。

函数原型

BCM_AntiPinchStatus notifyAntiPinch()

参数说明

无。

返回值说明

类型	描述
BCM_AntiPinchStatus	当前的防夹状态

3.13.12 notifyActuateStatus

功能说明

上报车窗执行状态。

函数原型

BCM_ActuateStatus notifyActuateStatus()

参数说明

无。

返回值说明

类型	描述
BCM_ActuateStatus	当前车窗开关的执行状态

3.13.13 notifyIceBreaking

功能说明

上报破冰模式。

函数原型

```
BCM_IceBreakingStatus notifyIceBreaking()
```

参数说明

无。

返回值说明

类型	描述
BCM_IceBreakingStatus	当前的破冰模式状态

3.13.14 notifyPosition

功能说明

上报车窗开度变化。

函数原型

```
uint8 notifyPosition()
```

参数说明

无。

返回值说明

类型	描述
uint8	当前车窗的开度，单位：%，255 代表未知

3.14 BCM_Sunshade(遮阳服务)

提供玻璃遮阳服务，每一个遮阳车窗一个原子服务。

BCM_Sunshade(遮阳服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
BCM_ReturnCode	adjustShade(uint8 shade)	调整车窗的遮阳程度。
uint8	getShade()	返回车窗的遮阳程度。
uint8	notifyShade()	返回遮阳的开启程度。
BCM_ShadeType	getShadeType()	返回车窗的遮阳类型。
BCM_ReturnCode	open()	打开遮阳。
BCM_ReturnCode	close()	关闭遮阳。
BCM_ReturnCode	stop()	停止开启或关闭遮阳。
BCM_AntiPinchStatus	notifyAntiPinch()	上报防夹状态。
BCM_ActuateStatus	notifyActuateStatus()	上报车窗遮阳执行状态。

3.14.1 adjustShade

功能说明

调整车窗的遮阳程度。

函数原型

```
BCM_ReturnCode adjustShade(uint8 shade)
```

参数说明

参数名称	类型	描述
shade	uint8	车窗遮阳的程度，单位：最大遮阳能力的1%

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.14.2 getShade

功能说明

返回车窗的遮阳程度。

函数原型

```
uint8 getShade()
```

参数说明

无。

返回值说明

类型	描述
uint8	车窗当前的遮阳程度，单位：%，255 代表未知

3.14.3 notifyShade

功能说明

返回遮阳的开启程度。

函数原型

```
uint8 notifyShade()
```

参数说明

无。

返回值说明

类型	描述
uint8	当前遮阳的程度，单位：%，255 代表未知

3.14.4 getShadeType

功能说明

返回车窗的遮阳类型。

函数原型

```
BCM_ShadeType getShadeType()
```

参数说明

无。

返回值说明

类型	描述
BCM_ShadeType	窗口遮阳帘的类型

3.14.5 open

功能说明

打开遮阳。

函数原型

```
BCM_ReturnCode open()
```

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.14.6 close

功能说明

关闭遮阳。

函数原型

BCM_ReturnCode close()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.14.7 stop

功能说明

停止开启或关闭遮阳。

函数原型

BCM_ReturnCode stop()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.14.8 notifyAntiPinch

功能说明

上报防夹状态。

函数原型

```
BCM_AntiPinchStatus notifyAntiPinch()
```

参数说明

无。

返回值说明

类型	描述
BCM_AntiPinchStatus	当前的防夹状态

3.14.9 notifyActuateStatus

功能说明

上报车窗遮阳执行状态。

函数原型

```
BCM_ActuateStatus notifyActuateStatus()
```

参数说明

无。

返回值说明

类型	描述
BCM_ActuateStatus	当前车窗遮阳开关的执行状态

3.15 BCM_WiperWash(雨刷清洗服务)

提供雨刷与喷淋的控制和状态服务，每一个雨刷或清洗一个原子服务。

BCM_WiperWash(雨刷清洗服务)服务包括的 API 如下表所示:

类型	API 函数	功能说明
BCM_ReturnCode	stopWiping()	停止雨刮。
BCM_ReturnCode	startWiping()	启动雨刮。
BCM_ReturnCode	setMaintenanceMode(BCM_MaintenanceMode maintenanceMode)	设置维修模式。
BCM_ReturnCode	setWipingLevel(BCM_WipingLevel wipingLevel)	设置雨刮的速度等级。
BCM_ReturnCode	setWipingLevelImme(BCM_WipingLevel wipingLevel)	设置雨刮的速度等级且立即执行。
BCM_ReturnCode	setWipingInterval(uint16 wipingInterval)	设置雨刮的两个刮窗周期间的停止时长。
boolean	isWiping()	返回雨刮是否正在运行。
BCM_WipingLevel	getWipingLevel()	返回雨刮的刮窗速度等级。
uint16	getWipingInterval()	返回雨刮周期间的停止时长。
BCM_ReturnCode	stopSprayWashing()	停止清洗。
BCM_ReturnCode	startSprayWashing()	启动清洗。
uint8	getWaterLevel()	返回清洗水箱水位高度。
uint8	notifyWaterLevelStatus()	上报清洗水箱水位高度。
boolean	isSprayWashing()	返回清洗是否正在运行。
boolean	notifySprayingStatus()	上报清洗的开启状态。
boolean	notifyWipingStatus()	上报雨刮的开启状态。

3.15.1 stopWiping

功能说明

停止雨刮。

函数原型

BCM_ReturnCode stopWiping()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.15.2 startWiping

功能说明

启动雨刮。

函数原型

BCM_ReturnCode startWiping()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.15.3 setMaintenanceMode

功能说明

设置维修模式。

函数原型

BCM_ReturnCode setMaintenanceMode(BCM_MaintenanceMode maintenanceMode)

参数说明

参数名称	类型	描述
maintenance Mode	BCM_MaintenanceMode	维修模式的开启与关闭

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.15.4 setWipingLevel

功能说明

设置雨刮的速度等级。

函数原型

BCM_ReturnCode setWipingLevel(BCM_WipingLevel wipingLevel)

参数说明

参数名称	类型	描述
wipingLevel	BCM_WipingLevel	雨刮的速度等级

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.15.5 setWipingLevelImme

功能说明

设置雨刮的速度等级且立即执行。

函数原型

BCM_ReturnCode setWipingLevelImme(BCM_WipingLevel wipingLevel)

参数说明

参数名称	类型	描述
wipingLevel	BCM_WipingLevel	雨刮的速度等级

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.15.6 setWipingInterval

功能说明

设置雨刮的两个刮窗周期期间的停止时长。

函数原型

```
BCM_ReturnCode setWipingInterval(uint16 wipingInterval)
```

参数说明

参数名称	类型	描述
wipingInterval	uint16	两个刮窗周期之间的时间间隔，单位：毫秒

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.15.7 isWiping

功能说明

返回雨刮是否正在运行。

函数原型

```
boolean isWiping()
```

参数说明

无。

返回值说明

类型	描述
boolean	雨刮正在运行返回 TRUE, 否则返回 FALSE

3.15.8 getWipingLevel

功能说明

返回雨刮的刮窗速度等级。

函数原型

BCM_WipingLevel getWipingLevel()

参数说明

无。

返回值说明

类型	描述
BCM_WipingLevel	当前雨刮的刮窗速度等级

3.15.9 getWipingInterval

功能说明

返回雨刮周期期间的停止时长。

函数原型

uint16 getWipingInterval()

参数说明

无。

返回值说明

类型	描述
uint16	当前雨刮周期期间的停止时长，单位：毫秒

3.15.10 stopSprayWashing

功能说明

停止清洗。

函数原型

BCM_ReturnCode stopSprayWashing()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.15.11 startSprayWashing

功能说明

启动清洗。

函数原型

BCM_ReturnCode startSprayWashing()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.15.12 getWaterLevel

功能说明

返回清洗水箱水位高度。

函数原型

```
uint8 getWaterLevel()
```

参数说明

无。

返回值说明

类型	描述
uint8	当前冲洗水箱的水位，单位：%，255 代表未知

3.15.13 notifyWaterLevelStatus

功能说明

上报清洗水箱水位高度。

函数原型

```
uint8 notifyWaterLevelStatus()
```

参数说明

无。

返回值说明

类型	描述
uint8	当前清洗水箱水位高度，单位：%，255 代表未知

3.15.14 isSprayWashing

功能说明

返回清洗是否正在运行。

函数原型

boolean isSprayWashing()

参数说明

无。

返回值说明

类型	描述
boolean	正在喷水清洗返回 TRUE， 否则返回 FALSE

3.15.15 notifySprayingStatus

功能说明

上报清洗的开启状态。

函数原型

boolean notifySprayingStatus()

参数说明

无。

返回值说明

类型	描述
boolean	当前清洗的启动状态，正在喷淋为 TRUE， 否则为 FALSE

3.15.16 notifyWipingStatus

功能说明

上报雨刮的开启状态。

函数原型

boolean notifyWipingStatus()

参数说明

无。

返回值说明

类型	描述
boolean	当前雨刮的启动状态，正在刮刷为 TRUE，否则为 FALSE

3.16 BCM_BrakeLight(刹车灯服务)

提供刹车灯的控制和状态服务，同时控制多个物理刹车灯的亮和灭,同时控制多个物理灯的亮和灭。如果需要单独控制某个物理灯，请使用 BCM_CommonLight。

BCM_BrakeLight(刹车灯服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
BCM_ReturnCode	turnOff()	关闭刹车灯。
BCM_ReturnCode	turnOn()	打开刹车灯。
BCM_ReturnCode	emergencyOn()	打开紧急刹车灯指示。
BCM_LightStatus	getStatus()	返回刹车灯的当前状态。
BCM_LightStatus	notifyStatus()	上报刹车灯的当前状态。

3.16.1 turnOff

功能说明

关闭刹车灯。

函数原型

```
BCM_ReturnCode turnOff()
```

参数说明

无。

返回值说明

类型	描述
----	----

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.16.2 turnOn

功能说明

打开刹车灯。

函数原型

BCM_ReturnCode turnOn()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.16.3 emergencyOn

功能说明

打开紧急刹车灯指示。

函数原型

BCM_ReturnCode emergencyOn()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.16.4 getStatus

功能说明

返回刹车灯当前状态。

函数原型

```
BCM_LightStatus getStatus()
```

参数说明

无。

返回值说明

类型	描述
BCM_LightStatus	当前刹车灯的状态

3.16.5 notifyStatus

功能说明

上报刹车灯当前状态。

函数原型

```
BCM_LightStatus notifyStatus()
```

参数说明

无。

返回值说明

类型	描述
BCM_LightStatus	当前刹车灯的开启状态

3.17 BCM_SteerLight(转向灯服务)

提供转向灯的控制和状态服务，同时控制多个物理灯的亮和灭。如果需要单独控制某个物理灯，请使用 `BCM_CommonLight`。

BCM_SteerLight(转向灯服务)服务包括的 API 如下表所示:

类型	API 函数	功能说明
BCM_ReturnCode	turnOff(BCM_SteerLightId steerLightId)	关闭转向灯。
BCM_ReturnCode	turnOn(BCM_SteerLightId steerLightId)	打开转向灯。
BCM_LightStatus	getStatus(BCM_SteerLightId steerLightId)	返回转向灯的当前状态。
BCM_SteerLightingStatus	notifyStatus()	上报转向灯的当前状态。

3.17.1 turnOff

功能说明

关闭转向灯。

函数原型

BCM_ReturnCode turnOff(BCM_SteerLightId steerLightId)

参数说明

参数名称	类型	描述
steerLightId	BCM_SteerLightId	转向灯的标识

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS, 否则返回 FAIL

3.17.2 turnOn

功能说明

打开转向灯。

函数原型

BCM_ReturnCode turnOn(BCM_SteerLightId steerLightId)

参数说明

参数名称	类型	描述
steerLightId	BCM_SteerLightId	转向灯的标识

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.17.3 getStatus

功能说明

返回转向灯的当前状态。

函数原型

```
BCM_LightStatus getStatus(BCM_SteerLightId steerLightId)
```

参数说明

参数名称	类型	描述
steerLightId	BCM_SteerLightId	转向灯的标识

返回值说明

类型	描述
BCM_LightStatus	当前转向灯的状态

3.17.4 notifyStatus

功能说明

上报转向灯的当前状态。

函数原型

```
BCM_SteerLightingStatus notifyStatus()
```

参数说明

无。

返回值说明

类型	描述
BCM_SteerLightingStatus	当前转向灯的开启状态

3.18 BCM_HazardLight(报警灯服务)

提供报警灯的控制和状态服务,同时控制多个物理灯的亮和灭。如果需要单独控制某个物理灯, 请使用 BCM_CommonLight。

BCM_HazardLight(报警灯服务)服务包括的 API 如下表所示:

类型	API 函数	功能说明
BCM_ReturnCode	turnOff()	关闭报警灯。
BCM_ReturnCode	turnOn()	打开报警灯。
BCM_ReturnCode	setFlashMode(BCM_FlashMode flashMode)	设置灯的闪烁方式。
BCM_LightStatus	getStatus()	返回报警灯的当前状态。
BCM_LightStatus	notifyStatus()	上报报警灯的当前状态。

3.18.1 turnOff

功能说明

关闭报警灯。

函数原型

```
BCM_ReturnCode turnOff()
```

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.18.2 turnOn

功能说明

打开报警灯。

函数原型

```
BCM_ReturnCode turnOn()
```

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.18.3 setFlashMode

功能说明

设置灯的闪烁方式。

函数原型

```
BCM_ReturnCode setFlashMode(BCM_FlashMode flashMode)
```

参数说明

参数名称	类型	描述
flashMode	BCM_FlashMode	灯的闪烁周期

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.18.4 getStatus

功能说明

返回报警灯的当前状态。

函数原型

```
BCM_LightStatus getStatus()
```

参数说明

无。

返回值说明

类型	描述
BCM_LightStatus	当前报警灯的状态

3.18.5 notifyStatus

功能说明

上报报警灯的当前状态。

函数原型

```
BCM_LightStatus notifyStatus()
```

参数说明

无。

返回值说明

类型	描述
BCM_LightStatus	当前报警灯的开启状态

3.19 BCM_DaytimeLight(日行灯服务)

提供日间行车灯的控制和状态服务,同时控制多个物理灯的亮和灭。如果需要单独控制某个物理灯, 请使用 BCM_CommonLight。

BCM_DaytimeLight(日行灯服务)服务包括的 API 如下表所示:

类型	API 函数	功能说明
BCM_ReturnCode	turnOff()	关闭日间行车灯。
BCM_ReturnCode	turnOn()	打开日间行车灯。
BCM_LightStatus	getStatus()	返回日间行车灯的当前状态。
BCM_LightStatus	notifyStatus()	上报日行灯的当前状态。

3.19.1 turnOff

功能说明

关闭日间行车灯。

函数原型

```
BCM_ReturnCode turnOff()
```

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS, 否则返回 FAIL

3.19.2 turnOn

功能说明

打开日间行车灯。

函数原型

BCM_ReturnCode turnOn()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.19.3 getStatus

功能说明

返回日间行车灯的当前状态。

函数原型

BCM_LightStatus getStatus()

参数说明

无。

返回值说明

类型	描述
BCM_LightStatus	当前日间行车灯的状态

3.19.4 notifyStatus

功能说明

上报日行灯的当前状态。

函数原型

BCM_LightStatus notifyStatus()

参数说明

无。

返回值说明

类型	描述
BCM_LightStatus	当前日行灯的开启状态

3.20 BCM_FogLight(雾灯服务)

提供雾灯的控制和状态服务,同时控制多个物理灯的亮和灭。如果需要单独控制某个物理灯,请使用 BCM_CommonLight。

BCM_FogLight(雾灯服务)服务包括的 API 如下表所示:

类型	API 函数	功能说明
BCM_ReturnCode	turnOff(BCM_FogLightId fogLightId)	关闭雾灯。
BCM_ReturnCode	turnOn(BCM_FogLightId fogLightId)	打开雾灯。
BCM_LightStatus	getStatus(BCM_FogLightId fogLightId)	返回雾灯的当前状态。
BCM_FogLightingStatus	notifyStatus()	上报雾灯的当前状态。

3.20.1 turnOff

功能说明

关闭雾灯。

函数原型

```
BCM_ReturnCode turnOff(BCM_FogLightId fogLightId)
```

参数说明

参数名称	类型	描述
fogLightId	BCM_FogLightId	雾灯的标识

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.20.2 turnOn

功能说明

打开雾灯。

函数原型

BCM_ReturnCode turnOn(BCM_FogLightId fogLightId)

参数说明

参数名称	类型	描述
fogLightId	BCM_FogLightId	雾灯的标识

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.20.3 getStatus

功能说明

返回雾灯的当前状态。

函数原型

BCM_LightStatus getStatus(BCM_FogLightId fogLightId)

参数说明

参数名称	类型	描述
fogLightId	BCM_FogLightId	雾灯的标识

返回值说明

类型	描述
BCM_LightStatus	当前雾灯的状态

3.20.4 notifyStatus

功能说明

上报雾灯的当前状态。

函数原型

`BCM_FogLightingStatus notifyStatus()`

参数说明

无。

返回值说明

类型	描述
BCM_FogLightingStatus	当前雾灯的状态

3.21 BCM_HighBeam(远光灯服务)

提供远光灯的控制和状态服务,同时控制多个物理灯的亮和灭。如果需要单独控制某个物理灯, 请使用 `BCM_CommonLight`。

`BCM_HighBeam`(远光灯服务)服务包括的 API 如下表所示:

类型	API 函数	功能说明
<code>BCM_ReturnCode</code>	turnOff()	关闭远光灯。
<code>BCM_ReturnCode</code>	turnOn()	打开远光灯。
<code>BCM_LightStatus</code>	getStatus()	返回远光灯的当前状态。
<code>BCM_LightStatus</code>	notifyStatus()	上报远光灯的当前状态。

3.21.1 turnOff

功能说明

关闭远光灯。

函数原型

BCM_ReturnCode turnOff()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS, 否则返回 FAIL

3.21.2 turnOn

功能说明

打开远光灯。

函数原型

BCM_ReturnCode turnOn()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS, 否则返回 FAIL

3.21.3 getStatus

功能说明

返回远光灯的当前状态。

函数原型

```
BCM_LightStatus getStatus()
```

参数说明

无。

返回值说明

类型	描述
BCM_LightStatus	当前远光灯的状态

3.21.4 notifyStatus

功能说明

上报远光灯的当前状态。

函数原型

```
BCM_LightStatus notifyStatus()
```

参数说明

无。

返回值说明

类型	描述
BCM_LightStatus	当前远光灯的开启状态

3.22 BCM_LowBeam(近光灯服务)

提供近光灯的控制和状态服务,同时控制多个物理灯的亮和灭。如果需要单独控制某个物理灯, 请使用 `BCM_CommonLight`。

BCM_LowBeam(近光灯服务)服务包括的 API 如下表所示:

类型	API 函数	功能说明
BCM_ReturnCode	turnOff()	关闭近光灯。
BCM_ReturnCode	turnOn()	打开近光灯。
BCM_LightStatus	getStatus()	返回近光灯的当前状态。
BCM_LightStatus	notifyStatus()	上报近光灯的当前状态。

3.22.1 turnOff

功能说明

关闭近光灯。

函数原型

BCM_ReturnCode turnOff()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS, 否则返回 FAIL

3.22.2 turnOn

功能说明

打开近光灯。

函数原型

BCM_ReturnCode turnOn()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.22.3 getStatus

功能说明

返回近光灯的当前状态。

函数原型

```
BCM_LightStatus getStatus()
```

参数说明

无。

返回值说明

类型	描述
BCM_LightStatus	当前近光灯的状态

3.22.4 notifyStatus

功能说明

上报近光灯的当前状态。

函数原型

```
BCM_LightStatus notifyStatus()
```

参数说明

无。

返回值说明

类型	描述
BCM_LightStatus	当前近光灯的开启状态

3.23 BCM_OutlineLight(位置灯服务)

提供示廓灯的控制和状态服务,同时控制多个物理灯的亮和灭。如果需要单独控制某个物理灯,请使用 BCM_CommonLight。

BCM_OutlineLight(位置灯服务)服务包括的 API 如下表所示:

类型	API 函数	功能说明
BCM_ReturnCode	turnOff()	关闭示廓灯。
BCM_ReturnCode	turnOn()	打开示廓灯。
BCM_LightStatus	getStatus()	返回示廓灯的当前状态。
BCM_LightStatus	notifyStatus()	上报示廓灯的当前状态。

3.23.1 turnOff

功能说明

关闭示廓灯。

函数原型

BCM_ReturnCode turnOff()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS, 否则返回 FAIL

3.23.2 turnOn

功能说明

打开示廓灯。

函数原型

BCM_ReturnCode turnOn()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.23.3 getStatus

功能说明

返回示廓灯的当前状态。

函数原型

BCM_LightStatus getStatus()

参数说明

无。

返回值说明

类型	描述
BCM_LightStatus	当前示廓灯的状态

3.23.4 notifyStatus

功能说明

上报示廓灯的当前状态。

函数原型

```
BCM_LightStatus notifyStatus()
```

参数说明

无。

返回值说明

类型	描述
BCM_LightStatus	当前示廓灯的开启状态

3.24 BCM_ReverseLight(倒车灯服务)

提供倒车灯的控制和状态服务,同时控制多个物理灯的亮和灭。如果需要单独控制某个物理灯, 请使用 `BCM_CommonLight`。

BCM_ReverseLight(倒车灯服务)服务包括的 API 如下表所示:

类型	API 函数	功能说明
BCM_ReturnCode	turnOff()	关闭倒车灯。
BCM_ReturnCode	turnOn()	打开倒车灯。
BCM_LightStatus	getStatus()	返回倒车灯的当前状态。
BCM_LightStatus	notifyStatus()	返回倒车灯的当前状态。

3.24.1 turnOff

功能说明

关闭倒车灯。

函数原型

```
BCM_ReturnCode turnOff()
```

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.24.2 turnOn

功能说明

打开倒车灯。

函数原型

```
BCM_ReturnCode turnOn()
```

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.24.3 getStatus

功能说明

返回倒车灯的当前状态。

函数原型

```
BCM_LightStatus getStatus()
```

参数说明

无。

返回值说明

类型	描述
BCM_LightStatus	当前倒车灯的状态

3.24.4 notifyStatus

功能说明

返回倒车灯的当前状态。

函数原型

BCM_LightStatus notifyStatus()

参数说明

无。

返回值说明

类型	描述
BCM_LightStatus	当前倒车灯的开启状态

3.25 BCM_CommonLight(通用灯服务)

提供通用灯的控制和状态服务，每一个普通物理灯一个原子服务。

BCM_CommonLight(通用灯服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
BCM_ReturnCode	turnOff()	关闭物理灯。
BCM_ReturnCode	turnOn()	打开物理灯。
BCM_ReturnCode	flash(BCM_PeriodMode periodMode)	设置灯的闪烁方式。
BCM_ReturnCode	setBrightness(uint8 brightness)	设置物理灯的亮度。
BCM_LightStatus	getStatus()	返回物理灯的当前状态。

类型	API 函数	功能说明
uint8	getBrightness()	返回物理灯的点亮亮度。
BCM_LightStatus	notifyStatus()	上报物理灯的当前状态。

3.25.1 turnOff

功能说明

关闭物理灯。

函数原型

BCM_ReturnCode turnOff()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.25.2 turnOn

功能说明

打开物理灯。

函数原型

BCM_ReturnCode turnOn()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.25.3 flash

功能说明

设置灯的闪烁方式。

函数原型

BCM_ReturnCode flash(BCM_PeriodMode periodMode)

参数说明

参数名称	类型	描述
periodMode	BCM_PeriodMode	灯的闪烁周期模式

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.25.4 setBrightness

功能说明

设置物理灯的亮度。

函数原型

BCM_ReturnCode setBrightness(uint8 brightness)

参数说明

参数名称	类型	描述
brightness	uint8	灯的亮度

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.25.5 getStatus

功能说明

返回物理灯的当前状态。

函数原型

```
BCM_LightStatus getStatus()
```

参数说明

无。

返回值说明

类型	描述
BCM_LightStatus	当前通用灯的状态

3.25.6 getBrightness

功能说明

返回物理灯的点亮亮度。

函数原型

```
uint8 getBrightness()
```

参数说明

无。

返回值说明

类型	描述
uint8	当前通用灯的亮度，单位：%，255 代表未知

3.25.7 notifyStatus

功能说明

上报物理灯的当前状态。

函数原型

BCM_LightStatus notifyStatus()

参数说明

无。

返回值说明

类型	描述
BCM_LightStatus	当前通用灯的开启状态

3.26 BCM_BeamDirection(光束方向调节服务)

提供光束方向调节控制和状态服务，每一个可角度调节的物理灯一个原子服务。

BCM_BeamDirection(光束方向调节服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
BCM_ReturnCode	adjustZAngle(uint8 beamZAngle)	设置灯的上下仰角的角度。
uint8	getZAngle()	返回灯的上下仰角的角度。
BCM_ReturnCode	startAdjustZAngle(BCM_StepActionType actionType)	启动灯的上下仰角角度的步进式调节。
BCM_ReturnCode	stopAdjustZAngle()	停止灯的上下仰角角度调节。
BCM_ReturnCode	adjustYAngle(uint8 beamYAngle)	设置灯的左右偏角的角度。
uint8	getYAngle()	返回灯的左右偏角的角度。
BCM_ReturnCode	startAdjustYAngle(BCM_StepActionType actionType)	启动灯的左右偏角角度的步进式调节。
BCM_ReturnCode	stopAdjustYAngle()	停止灯的左右偏角角度调节。

类型	API 函数	功能说明
ode		

3.26.1 adjustZAngle

功能说明

设置灯的上下仰角的角度。

函数原型

BCM_ReturnCode adjustZAngle(uint8 beamZAngle)

参数说明

参数名称	类型	描述
beamZAngle	uint8	光束的目标倾斜角度，单位：最大倾斜度的 1%

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.26.2 getZAngle

功能说明

返回灯的上下仰角的角度。

函数原型

uint8 getZAngle()

参数说明

无。

返回值说明

类型	描述
----	----

类型	描述
uint8	当前光束的仰角，单位：%，255 代表未知

3.26.3 startAdjustZAngle

功能说明

启动灯的上下仰角角度的步进式调节。

函数原型

```
BCM_ReturnCode startAdjustZAngle(BCM_StepActionType actionType)
```

参数说明

参数名称	类型	描述
actionType	BCM_StepActionType	步进调节类型

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.26.4 stopAdjustZAngle

功能说明

停止灯的上下仰角角度调节。

函数原型

```
BCM_ReturnCode stopAdjustZAngle()
```

参数说明

无。

返回值说明

类型	描述
----	----

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.26.5 adjustYAngle

功能说明

设置灯的左右偏角的角度。

函数原型

```
BCM_ReturnCode adjustYAngle(uint8 beamYAngle)
```

参数说明

参数名称	类型	描述
beamYAngle	uint8	光束的目标倾斜角度，单位：最大倾斜度的 1%

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.26.6 getYAngle

功能说明

返回灯的左右偏角的角度。

函数原型

```
uint8 getYAngle()
```

参数说明

无。

返回值说明

类型	描述
----	----

类型	描述
uint8	当前光束的偏角，单位：%，255 代表未知

3.26.7 startAdjustYAngle

功能说明

启动灯的左右偏角角度的步进式调节。

函数原型

```
BCM_ReturnCode startAdjustYAngle(BCM_StepActionType actionType)
```

参数说明

参数名称	类型	描述
actionType	BCM_StepActionType	步进调节类型

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.26.8 stopAdjustYAngle

功能说明

停止灯的左右偏角角度调节。

函数原型

```
BCM_ReturnCode stopAdjustYAngle()
```

参数说明

无。

返回值说明

类型	描述
----	----

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.27 BCM_Buzzer(翁鸣器服务)

提供翁鸣器的声音调节控制和状态服务。

BCM_Buzzer(翁鸣器服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
BCM_ReturnCode	start()	启动警示音。
BCM_ReturnCode	stop()	停止警示音。
BCM_ReturnCode	alert(BCM_PeriodMode periodMode)	启动警示音。
BCM_ReturnCode	setAlarmMode(BCM_AlarmMode alarmMode)	设置警示音模式。
BCM_AlarmMode	getAlarmMode()	返回警示音服务模式。
BCM_BuzzerStatus	notifyStatus()	上报警示音服务的当前状态。
BCM_BuzzerStatus	getStatus()	返回警示音服务的当前状态。

3.27.1 start

功能说明

启动警示音。

函数原型

```
BCM_ReturnCode start()
```

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.27.2 stop

功能说明

停止警示音。

函数原型

BCM_ReturnCode stop()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.27.3 alert

功能说明

启动警示音。

函数原型

BCM_ReturnCode alert(BCM_PeriodMode periodMode)

参数说明

参数名称	类型	描述
periodMode	BCM_PeriodMode	嗡鸣周期模式

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.27.4 setAlarmMode

功能说明

设置警示音模式。

函数原型

```
BCM_ReturnCode setAlarmMode(BCM_AlarmMode alarmMode)
```

参数说明

参数名称	类型	描述
alarmMode	BCM_AlarmMode	警示模式

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.27.5 getAlarmMode

功能说明

返回警示音服务模式。

函数原型

```
BCM_AlarmMode getAlarmMode()
```

参数说明

无。

返回值说明

类型	描述
BCM_AlarmMode	当前警示音服务模式

3.27.6 notifyStatus

功能说明

上报警示音服务的当前状态。

函数原型

```
BCM_BuzzerStatus notifyStatus()
```

参数说明

无。

返回值说明

类型	描述
BCM_BuzzerStatus	当前警示音服务的状态

3.27.7 getStatus

功能说明

返回警示音服务的当前状态。

函数原型

```
BCM_BuzzerStatus getStatus()
```

参数说明

无。

返回值说明

类型	描述
BCM_BuzzerStatus	当前警示音服务的状态

3.28 BCM_Avas(行人警示器服务)

提供行人警示器的声音调节控制和状态服务。

BCM_Avas(行人警示器服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
BCM_ReturnCode	start(BCM_AlarmMode alarmMode)	启动行人警示器。
BCM_ReturnCode	stop()	停止行人警示器。
BCM_AvasStatus	getStatus()	返回行人警示器服务的当前状态。

3.28.1 start

功能说明

启动行人警示器。

函数原型

BCM_ReturnCode start(BCM_AlarmMode alarmMode)

参数说明

参数名称	类型	描述
alarmMode	BCM_AlarmMode	警示模式

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.28.2 stop

功能说明

停止行人警示器。

函数原型

BCM_ReturnCode stop()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS, 否则返回 FAIL

3.28.3 getStatus

功能说明

返回行人警示器服务的当前状态。

函数原型

BCM_AvasStatus getStatus()

参数说明

无。

返回值说明

类型	描述
BCM_AvasStatus	当前行人警示器服务的状态

3.29 BCM_RearSpoiler(尾翼服务)

提供尾翼的控制和状态服务。

BCM_RearSpoiler(尾翼服务)服务包括的 API 如下表所示:

类型	API 函数	功能说明
BCM_ReturnCode	fold()	收起尾翼。
BCM_ReturnCode	unfold()	展开尾翼。

类型	API 函数	功能说明
ode		
BCM_ReturnCode	stop()	停止收起或展开尾翼。
BCM_ActuateStatus	notifyActuateStatus()	上报尾翼执行状态。
BCM_ReturnCode	adjustPosition(uint8 position)	调整尾翼开度大小。
uint8	getPosition()	返回尾翼开度信息。
uint8	notifyPosition()	上报尾翼开度变化。
boolean	isFolded()	返回尾翼是否处于折叠状态。
BCM_LearnedStatus	getLearnedStatus()	返回尾翼的学习状态。
BCM_AntiPinchStatus	notifyAntiPinch()	上报防夹状态。
BCM_IceBreakingStatus	notifyIceBreaking()	上报破冰模式。

3.29.1 fold

功能说明

收起尾翼。

函数原型

BCM_ReturnCode fold()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.29.2 unfold

功能说明

展开尾翼。

函数原型

BCM_ReturnCode unfold()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.29.3 stop

功能说明

停止收起或展开尾翼。

函数原型

BCM_ReturnCode stop()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.29.4 notifyActuateStatus

功能说明

上报尾翼执行状态。

函数原型

```
BCM_ActuateStatus notifyActuateStatus()
```

参数说明

无。

返回值说明

类型	描述
BCM_ActuateStatus	当前尾翼的执行状态

3.29.5 adjustPosition

功能说明

调整尾翼开度大小。

函数原型

```
BCM_ReturnCode adjustPosition(uint8 position)
```

参数说明

参数名称	类型	描述
position	uint8	尾翼开度大小，单位：最大开度的 1%

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.29.6 getPosition

功能说明

返回尾翼开度信息。

函数原型

```
uint8 getPosition()
```

参数说明

无。

返回值说明

类型	描述
uint8	当前尾翼开度信息，单位：%，255 代表未知

3.29.7 notifyPosition

功能说明

上报尾翼开度变化。

函数原型

uint8 notifyPosition()

参数说明

无。

返回值说明

类型	描述
uint8	当前尾翼的开度，单位：%，255 代表未知

3.29.8 isFolded

功能说明

返回尾翼是否处于折叠状态。

函数原型

boolean isFolded()

参数说明

无。

返回值说明

类型	描述
boolean	尾翼已折叠返回 TRUE, 否则返回 FALSE

3.29.9 getLearnedStatus

功能说明

返回尾翼的学习状态。

函数原型

BCM_LearnedStatus getLearnedStatus()

参数说明

无。

返回值说明

类型	描述
BCM_LearnedStatus	当前尾翼的学习状态

3.29.10 notifyAntiPinch

功能说明

上报防夹状态。

函数原型

BCM_AntiPinchStatus notifyAntiPinch()

参数说明

无。

返回值说明

类型	描述
BCM_AntiPinchStatus	当前的防夹状态

3.29.11 notifyIceBreaking

功能说明

上报破冰模式。

函数原型

BCM_IceBreakingStatus notifyIceBreaking()

参数说明

无。

返回值说明

类型	描述
BCM_IceBreakingStatus	当前的破冰模式状态

3.30 BCM_SafetyBelt(安全带服务)

提供安全带卡扣控制和状态服务。

BCM_SafetyBelt(安全带服务)服务包括的 API 如下表所示:

类型	API 函数	功能说明
BCM_BuckleStatus	getBuckleStatus()	返回安全带卡扣的当前状态。
BCM_BuckleStatus	notifyBuckleStatus()	上报安全带卡扣的当前状态。

3.30.1 getBuckleStatus

功能说明

返回安全带卡扣的当前状态。

函数原型

BCM_BuckleStatus getBuckleStatus()

参数说明

无。

返回值说明

类型	描述
BCM_BuckleStatus	当前安全带卡扣的状态

3.30.2 notifyBuckleStatus

功能说明

上报安全带卡扣的当前状态。

函数原型

BCM_BuckleStatus notifyBuckleStatus()

参数说明

无。

返回值说明

类型	描述
BCM_BuckleStatus	当前安全带卡扣的状态

3.31 BCM_ScreenAdjust(屏位置调节服务)

屏位置控制和状态服务。

BCM_ScreenAdjust(屏位置调节服务)服务包括的 API 如下表所示:

类型	API 函数	功能说明
BCM_ReturnCode	fold()	收起。
BCM_ReturnCode	unfold()	展开。
BCM_ReturnCode	stop()	停止收起或展开。

类型	API 函数	功能说明
BCM_ActuateStatus	notifyActuateStatus()	上报执行状态。
BCM_ReturnCode	adjustPosition(uint8 position)	调整开度大小。
uint8	getPosition()	返回开度信息。
uint8	notifyPosition()	上报开度变化。
BCM_FoldedStatus	getFoldedStatus()	返回是否处于折叠状态。
BCM_LearnedStatus	getLearnedStatus()	返回学习状态。
BCM_AntiPinchStatus	notifyAntiPinch()	上报防夹状态。

3.31.1 fold

功能说明

收起。

函数原型

BCM_ReturnCode fold()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.31.2 unfold

功能说明

展开。

函数原型

BCM_ReturnCode unfold()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.31.3 stop

功能说明

停止收起或展开。

函数原型

BCM_ReturnCode stop()

参数说明

无。

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

3.31.4 notifyActuateStatus

功能说明

上报执行状态。

函数原型

BCM_ActuateStatus notifyActuateStatus()

参数说明

无。

返回值说明

类型	描述
BCM_ActuateStatus	当前的执行状态

3.31.5 adjustPosition

功能说明

调整开度大小。

函数原型

```
BCM_ReturnCode adjustPosition(uint8 position)
```

参数说明

参数名称	类型	描述
position	uint8	开度大小，单位：最大开度的 1%，255 代表未知

返回值说明

类型	描述
BCM_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

3.31.6 getPosition

功能说明

返回开度信息。

函数原型

```
uint8 getPosition()
```

参数说明

无。

返回值说明

类型	描述
uint8	当前的开度信息，单位：%，255 代表未知

3.31.7 notifyPosition

功能说明

上报开度变化。

函数原型

```
uint8 notifyPosition()
```

参数说明

无。

返回值说明

类型	描述
uint8	当前的开度，单位：%，255 代表未知

3.31.8 getFoldedStatus

功能说明

返回是否处于折叠状态。

函数原型

```
BCM_FoldedStatus getFoldedStatus()
```

参数说明

无。

返回值说明

类型	描述
BCM_FoldedStatus	当前的折叠状态

3.31.9 getLearnedStatus

功能说明

返回学习状态。

函数原型

```
BCM_LearnedStatus getLearnedStatus()
```

参数说明

无。

返回值说明

类型	描述
BCM_LearnedStatus	当前的学习状态

3.31.10 notifyAntiPinch

功能说明

上报防夹状态。

函数原型

```
BCM_AntiPinchStatus notifyAntiPinch()
```

参数说明

无。

返回值说明

类型	描述
BCM_AntiPinchStatus	当前的防夹状态

3.32 数据类型定义

3.32.1 BCM_ReturnCode

说明

车身原子服务 API 调用返回信息。

定义

```
typedef enum {
    SUCCESS=0,
    FAIL_WORKING_CONDITION_UNFULFILLED=1,
    FAIL_HIGHER_PRIORITY_TASK_ONGOING=2,
    FAIL_FUNCTION_UNAVAILABLE=3,
    FAIL_INVALID_SOURCE_ID=4,
    FAIL_DEVICE_ABNORMAL=5,
    FAIL_POWER_SHORTAGE=6,
    FAIL_TIMEOUT=7,
    FAIL_OVERHEATING=8,
    FAIL_OUT_OF_RANGE=9,
    FAIL_ANTIPLAY=10,
    FAIL_DRAG=11,
    FAIL_OVERCURRENT=12,
    FAIL_NOTLEARNED=13,
    FAIL_OTHERS=255
} BCM_ReturnCode;
```

成员

成员名称	描述
SUCCESS	成功
FAIL_WORKING_CONDITION_UNFULFILLED	执行失败，原因为当前模式不支持
FAIL_HIGHER_PRIORITY_TASK_ONGOING	执行失败，原因为高优先级任务忙
FAIL_FUNCTION_UNAVAILABLE	执行失败，原因为功能不可用
FAIL_INVALID_SOURCE_ID	执行失败，原因为无效的资源
FAIL_DEVICE_ABNORMAL	执行失败，原因为设备异常
FAIL_POWER_SHORTAGE	执行失败，原因为电量不足
FAIL_TIMEOUT	执行失败，原因为超时

成员名称	描述
FAIL_OVERHEATING	执行失败，原因为过热保护
FAIL_OUT_OF_RANGE	执行失败，原因为参数超过范围
FAIL_ANTIPLAY	执行失败，原因为触发防玩
FAIL_DRAG	执行失败，原因为阻转
FAIL_OVERCURRENT	执行失败，原因为过流
FAIL_NOTLEARNED	执行失败，原因为未完成学习
FAIL_OTHERS	执行失败，原因请参考部件健康信息

3.32.2 BCM_LockStatus

说明

锁的状态。

定义

```
typedef enum {
    LOCKED=0,
    UNLOCKED=1,
    SUPERLOCKED=2,
    UNKNOWN=255
} BCM_LockStatus;
```

成员

成员名称	描述
LOCKED	已上锁
UNLOCKED	未上锁
SUPERLOCKED	已上超级锁
UNKNOWN	未知

3.32.3 BCM_ActuateStatus

说明

执行状态。

定义

```
typedef enum {
    IDLE=0,
    CLOSING_BACKWARD_FOLDING_DOWNWARD_DECREASING=1,
    OPENING_FORWARD_UNFOLDING_UPWARD_INCREASING=2,
    UNKNOWN=255
} BCM ActuateStatus;
```

成员

成员名称	描述
IDLE	空闲
CLOSING_BACKWARD_FOLDING_DOWNWARD_DECREASING	正在关/后退/折叠/上升/减少
OPENING_FORWARD_UNFOLDING_UPWARD_INCREASING	正在开/前移/展开/下降/增加
UNKNOWN	未知

3.32.4 BCM_LockStatus_Window

说明

车窗锁的状态。

定义

```
typedef enum {
    LOCKED=0,
    UNLOCKED=1,
    UNKNOWN=255
} BCM LockStatus Window;
```

成员

成员名称	描述
LOCKED	已上锁
UNLOCKED	未上锁
UNKNOWN	未知

3.32.5 BCM_AuthStatus

说明

认证状态。

定义

```
typedef enum {
    UNAUTHENTICATED=0,
    AUTHENTICATED=1,
    UNKNOWN=255
} BCM_AuthStatus;
```

成员

成员名称	描述
UNAUTHENTICATED	未认证
AUTHENTICATED	已认证
UNKNOWN	未知

3.32.6 BCM_KeyLocation

说明

钥匙位置。

定义

```
typedef enum {
    INSIDE=0,
    LEFT_WELCOME_AREA=1,
    RIGHT_WELCOME_AREA=2,
    LEFT_UNLOCK_AREA=3,
    RIGHT_UNLOCK_AREA=4,
    NEARBY=5,
    FARAWAY=6,
    TAILGATE_WELCOME_AREA=7,
    TAILGATE_UNLOCK_AREA=8,
    IN_TRUNK=9,
    UNKNOWN=255
} BCM_KeyLocation;
```

成员

成员名称	描述
INSIDE	钥匙在车内
LEFT_WELCOME_AREA	车外左迎宾区
RIGHT_WELCOME_AREA	车外右迎宾区
LEFT_UNLOCK_AREA	车外左解锁区
RIGHT_UNLOCK_AREA	车外右解锁区
NEARBY	车的附件区域
FARAWAY	远离车的区域
TAILGATE_WELCOME_AREA	车外尾门迎宾区
TAILGATE_UNLOCK_AREA	车外尾门解锁区
IN_TRUNK	后备箱
UNKNOWN	未知，无法检测到钥匙

3.32.7 BCM_ShadeType

说明

遮阳帘类型。

定义

```
typedef enum {
    FABRIC=0,
    GLASS=1
} BCM_ShadeType;
```

成员

成员名称	描述
FABRIC	遮阳帘
GLASS	智能玻璃

3.32.8 BCM_FogLightId

说明

雾灯标识。

定义

```
typedef enum {  
    FRONT_FOG_LIGHT=0,  
    REAR_FOG_LIGHT=1  
} BCM_FogLightId;
```

成员

成员名称	描述
FRONT_FOG_LIGHT	前雾灯
REAR_FOG_LIGHT	后雾灯

3.32.9 BCM_SteerLightId

说明

转向灯标识。

定义

```
typedef enum {  
    TURN_LEFT_LIGHT=2,  
    TURN_RIGHT_LIGHT=3  
} BCM_SteerLightId;
```

成员

成员名称	描述
TURN_LEFT_LIGHT	左转向指示灯
TURN_RIGHT_LIGHT	右转向指示灯

3.32.10 BCM_WipingLevel

说明

雨刮频度等级。

定义

```
typedef enum {  
    LOW=0,  
    MEDIUM=1,  
    HIGH=2  
} BCM_WipingLevel;
```

成员

成员名称	描述
LOW	低档运行
MEDIUM	中档运行
HIGH	高档运行

3.32.11 BCM_LightStatus

说明

灯的状态。

定义

```
typedef enum {  
    LIGHT_ON=0,  
    LIGHT_OFF=1,  
    UNKNOWN=255  
} BCM_LightStatus;
```

成员

成员名称	描述
LIGHT_ON	开启
LIGHT_OFF	关闭
UNKNOWN	未知

3.32.12 BCM_KeyContent

说明

钥匙信息。

定义

```
typedef struct {
    BCM AuthStatus bodyAuthStatus;
    BCM AuthStatus engineAuthStatus;
    BCM keyLocation keyLocation;
    uint32 keySerialNo;
    uint8 keyId;
} BCM KeyContent;
```

成员

成员名称	描述
bodyAuthStatus	钥匙对象的钥匙与车身验证状态
engineAuthStatus	钥匙对象的钥匙与动力验证状态
keyLocation	钥匙对象的钥匙位置
keySerialNo	钥匙的电子串号
keyId	钥匙对象标识

3.32.13 BCM_FlashMode

说明

灯光闪烁模式。

定义

```
typedef struct {
    uint16 onDuration;
    uint16 offDuration;
} BCM FlashMode;
```

成员

成员名称	描述
onDuration	灯的闪烁周期内亮的时长，单位：毫秒

成员名称	描述
offDuration	灯的闪烁周期内灭的时长，单位：毫秒

3.32.14 BCM_SteerLightingStatus

说明

转向灯的开启状态。

定义

```
typedef enum {
    RIGHT_ON=0,
    LEFT_ON=1,
    LIGHT_OFF=2,
    UNKNOWN=255
} BCM_SteerLightingStatus;
```

成员

成员名称	描述
RIGHT_ON	右转开启
LEFT_ON	左转开启
LIGHT_OFF	关闭
UNKNOWN	未知

3.32.15 BCM_FogLightingStatus

说明

雾灯的状态。

定义

```
typedef enum {
    FRONT_ON=0,
    REAR_ON=1,
    LIGHT_ON=2,
    LIGHT_OFF=3,
    UNKNOWN=255
} BCM_FogLightingStatus;
```

成员

成员名称	描述
FRONT_ON	前雾灯开启
REAR_ON	后雾灯开启
LIGHT_ON	全部雾灯开启
LIGHT_OFF	全部雾灯关闭
UNKNOWN	未知

3.32.16 BCM_LearnedStatus

说明

学习状态。

定义

```
typedef enum {
    UNFINISHED=0,
    FINISHED=1,
    UNKNOWN=255
} BCM_LearnedStatus;
```

成员

成员名称	描述
UNFINISHED	未完成学习
FINISHED	完成学习
UNKNOWN	未知

3.32.17 BCM_StepActionType

说明

操作类型。

定义

```
typedef enum {
    CLOSING_BACKWARD_FOLDING_DOWNWARD_DECREASING=0,
    OPENING_FORWARD_UNFOLDING_UPWARD_INCREASING=1
}
```

```
} BCM_StepActionType;
```

成员

成员名称	描述
CLOSING_BACKWARD_FOLDING_DOWNWARD_DECREASING	关闭_后退_折叠_下降_减少
OPENING_FORWARD_UNFOLDING_UPWARD_INCREASING	开启_前移_展开_上升_增加

3.32.18 BCM_AlarmMode

说明

警示音模式。

定义

```
typedef struct {
    uint16 tone;
    uint8 volume;
} BCM_AlarmMode;
```

成员

成员名称	描述
tone	警示声的频率，单位：赫兹
volume	警示声的音量，单位：%

3.32.19 BCM_AntiGlareMode

说明

防眩模式。

定义

```
typedef struct {
    uint8 antiGlareLevel;
    BCM_AntiGlareActMode antiGlareActMode;
```

```
} BCM_AntiGlareMode;
```

成员

成员名称	描述
antiGlareLevel	防眩的程度，单位：%
antiGlareActMode	防眩的执行模式

3.32.20 BCM_AntiGlareActMode

说明

防眩执行模式。

定义

```
typedef enum {
    NORMAL=0,
    IMMEDIATE=1
} BCM_AntiGlareActMode;
```

成员

成员名称	描述
NORMAL	通用模式
IMMEDIATE	立即模式

3.32.21 BCM_PeriodMode

说明

周期模式。

定义

```
typedef struct {
    uint32 onDuration;
    uint32 offDuration;
    uint16 times;
} BCM_PeriodMode;
```

成员

成员名称	描述
onDuration	间歇嗡鸣周期中的嗡鸣时长
offDuration	间歇嗡鸣周期中的停歇时长，0 代表无停歇
times	间歇嗡鸣周期的次数，65535 代表不限制次数

3.32.22 BCM_AvasStatus

说明

警示音状态。

定义

```
typedef enum {
    ON=0,
    OFF=1,
    UNKNOWN=255
} BCM_AvasStatus;
```

成员

成员名称	描述
ON	启动警示
OFF	停止警示
UNKNOWN	未知

3.32.23 BCM_BuzzerStatus

说明

警示音状态。

定义

```
typedef enum {
    ON=0,
    OFF=1,
    UNKNOWN=255
} BCM_BuzzerStatus;
```

成员

成员名称	描述
ON	启动警示
OFF	停止警示
UNKNOWN	未知

3.32.24 BCM_MessageMode

说明

按摩模式。

定义

```
typedef struct {
    BCM MessageType type;
    BCM MessageStrength strength;
} BCM MessageMode;
```

成员

成员名称	描述
type	按摩类型
strength	按摩强度

3.32.25 BCM_MessageStrength

说明

按摩强度。

定义

```
typedef enum {
    GENTLE=0,
    MIDDLE=1,
    HEAVY=2
} BCM_MessageStrength;
```

成员

成员名称	描述
GENTLE	轻
MIDDLE	中
HEAVY	重

3.32.26 BCM_MessageType

说明

按摩类型。

定义

```
typedef enum {
    KNEAD=0,
    ACUPRESSURE=1,
    BEAT=2,
    TUINA=3
} BCM_MessageType;
```

成员

成员名称	描述
KNEAD	揉捏
ACUPRESSURE	指压
BEAT	捶打
TUINA	推拿

3.32.27 BCM_AntiPinchStatus

说明

防夹状态。

定义

```
typedef enum {
    NOTOCCUR=0,
    OCCUR=1,
    UNKNOWN=255
}
```

```
} BCM_AntiPinchStatus;
```

成员

成员名称	描述
NOTOCCUR	未发生
OCCUR	发生
UNKNOWN	未知

3.32.28 BCM_IceBreakingStatus

说明

破冰状态。

定义

```
typedef enum {
    NOICEBREAKING=0,
    ICEBREAKING=1,
    UNKNOWN=255
} BCM_IceBreakingStatus;
```

成员

成员名称	描述
NOICEBREAKING	无破冰
ICEBREAKING	破冰中
UNKNOWN	未知

3.32.29 BCM_FoldedStatus

说明

折叠状态。

定义

```
typedef enum {
    UNFOLDED=0,
    FOLDED=1,
```



```
UNKNOWN=255
} BCM_FoldedStatus;
```

成员

成员名称	描述
UNFOLDED	展开
FOLDED	折叠
UNKNOWN	未知

3.32.30 BCM_OpenStatus

说明

开启状态。

定义

```
typedef enum {
    CLOSE=0,
    AJAR=1,
    OPEN=2,
    UNKNOWN=255
} BCM_OpenStatus;
```

成员

成员名称	描述
CLOSE	关闭
AJAR	微开
OPEN	打开
UNKNOWN	未知

3.32.31 BCM_BlowStatus

说明

鸣笛状态。

定义

```
typedef enum {
    IDLE=0,
    BLOWING=1,
    UNKNOWN=255
} BCM_BlowStatus;
```

成员

成员名称	描述
IDLE	空闲
BLOWING	正在鸣笛
UNKNOWN	未知

3.32.32 BCM_BuckleStatus

说明

卡扣状态。

定义

```
typedef enum {
    UNFASTEN=0,
    FASTEN=1,
    UNKNOWN=255
} BCM_BuckleStatus;
```

成员

成员名称	描述
UNFASTEN	未系
FASTEN	已系
UNKNOWN	未知

3.32.33 BCM_MaintenanceMode

说明

维修模式。

定义

```
typedef enum {  
    OFF=0,  
    ON=1  
} BCM MaintenanceMode;
```

成员

成员名称	描述
OFF	关闭
ON	打开

3.32.34 BCM_OccupiedStatus

说明

占用状态。

定义

```
typedef enum {  
    UNOCCUPIED=0,  
    OCCUPIED=1,  
    UNKNOWN=255  
} BCM_OccupiedStatus;
```

成员

成员名称	描述
UNOCCUPIED	未占用
OCCUPIED	占用
UNKNOWN	未知

4 TMS(热管理)原子服务 API

4.1 TMS_Battery(电池包温控)

提供动力电池包的温控服务。包括控制电池包工作温度、控制电池包冷却回路冷却液最小流量等。

TMS_Battery(电池包温控)服务包括的 API 如下表所示：

类型	API 函数	功能说明
TMS_ReturnCode	setTargetTemp(sint16 targetTemp)	设置电池包温控目标温度。TMS 通过调节温控回路水温及流量等，使电池包工作在指定的目标温度。
sint16	getTargetTemp()	返回电池包目标温控温度。
uint16	getCurrentFlow()	返回当前电池包温控回路流量。
TMS_ReturnCode	setMinFlow(uint16 minFlow)	设置电池包温控回路最小流量(实现电池包内部均温等功能)。
TMS_ReturnCode	setTargetWorkMode(TMS_BatteryMode mode)	设置电池包温控回路目标工作模式。
TMS_BatteryMode	getCurrentWorkMode()	返回电池包温控回路当前工作模式。
TMS_BatteryMode	notifyCurrentWorkMode()	上报电池包温控回路当前工作模式。

4.1.1 setTargetTemp

功能说明

设置电池包温控目标温度。TMS 通过调节温控回路水温及流量等，使电池包工作在指定的目标温度。

函数原型

TMS_ReturnCode setTargetTemp(sint16 targetTemp)

参数说明

参数名称	类型	描述
targetTemp	sint16	电池包目标工作温度，单位：℃，精度：1/10

返回值说明

类型	描述
TMS_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

4.1.2 getTargetTemp

功能说明

返回电池包目标温控温度。

函数原型

sint16 getTargetTemp()

参数说明

无。

返回值说明

类型	描述
sint16	当前设置的电池目标工作温度，单位：℃，精度：1/10

4.1.3 getCurrentFlow

功能说明

返回当前电池包温控回路流量。

函数原型

```
uint16 getCurrentFlow()
```

参数说明

无。

返回值说明

类型	描述
uint16	当前电池包温控回路冷却液流量，单位：L/min

4.1.4 setMinFlow

功能说明

设置电池包温控回路最小流量(实现电池包内部均温等功能)。

函数原型

```
TMS_ReturnCode setMinFlow(uint16 minFlow)
```

参数说明

参数名称	类型	描述
minFlow	uint16	电池包温控回路最小流量需求，单位：L/min。0xFFFF 为取消最小流量限制。

返回值说明

类型	描述
TMS_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

4.1.5 setTargetWorkMode

功能说明

设置电池包温控回路目标工作模式。

函数原型

```
TMS_ReturnCode setTargetWorkMode(TMS_BatteryMode mode)
```

参数说明

参数名称	类型	描述
mode	TMS_BatteryMode	电池包温控回路工作模式

返回值说明

类型	描述
TMS_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

4.1.6 getCurrentWorkMode

功能说明

返回电池包温控回路当前工作模式。

函数原型

```
TMS_BatteryMode getCurrentWorkMode()
```

参数说明

无。

返回值说明

类型	描述
TMS_BatteryMode	电池包温控回路当前工作模式

4.1.7 notifyCurrentWorkMode

功能说明

上报电池包温控回路当前工作模式。

函数原型

```
TMS_BatteryMode notifyCurrentWorkMode()
```

参数说明

无。

返回值说明

类型	描述
TMS_BatteryMode	电池包温控回路当前工作模式

4.2 TMS_Device(设备散热控制)

提供电驱等部件的散热温控服务。包括控制指定部件的工作温度、控制指定部件温控回路冷却液的最小流量等。

TMS_Device(设备散热控制)服务包括的 API 如下表所示：

类型	API 函数	功能说明
TMS_ReturnCode	setTargetTemp(uint16 devId,sint16 targetTemp)	设置指定部件散热目标温度。
sint16	getTargetTemp(uint16 devId)	返回指定部件散热目标温度。
sint16	getCurrentTemp(uint16 devId)	返回指定部件当前温度。
TMS_ReturnCode	setMinFlow(uint16 devId,uint16 minFlow)	设置指定部件散热的最小流量需求。
uint16	getCurrentFlow(uint16 devId)	返回流经指定部件温控回路流量。
TMS_DeviceFlowAry	notifyCurrentFlow()	上报部件温控回路的当前流量。
sint16	getCurrentWaterTemp(uint16 devId,TMS_LoopPort portId)	返回指定部件温控回路的当前水温。
TMS_DeviceWaterTempAry	notifyCurrentWaterTemp()	上报部件温控回路的当前水温。

4.2.1 setTargetTemp

功能说明

设置指定部件散热目标温度。

函数原型

```
TMS_ReturnCode setTargetTemp(uint16 devId, sint16 targetTemp)
```

参数说明

参数名称	类型	描述
devId	uint16	部件编号
targetTemp	sint16	部件散热的目标工作温度，单位：℃，精度：1/10

返回值说明

类型	描述
TMS_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

4.2.2 getTargetTemp

功能说明

返回指定部件散热目标温度。

函数原型

```
sint16 getTargetTemp(uint16 devId)
```

参数说明

参数名称	类型	描述
devId	uint16	部件编号

返回值说明

类型	描述
----	----

类型	描述
sint16	指定部件当前设置的目标温度，单位：℃，精度：1/10

4.2.3 getCurrentTemp

功能说明

返回指定部件当前温度。

函数原型

```
sint16 getCurrentTemp(uint16 devId)
```

参数说明

参数名称	类型	描述
devId	uint16	部件编号

返回值说明

类型	描述
sint16	指定部件当前的温度，单位：℃，精度：1/10

4.2.4 setMinFlow

功能说明

设置指定部件散热的最小流量需求。

函数原型

```
TMS_ReturnCode setMinFlow(uint16 devId, uint16 minFlow)
```

参数说明

参数名称	类型	描述
devId	uint16	部件编号
minFlow	uint16	部件散热的最小流量需求，单位：L/min。 0xFFFF 为取消最小流量限制

返回值说明

类型	描述
TMS_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

4.2.5 getCurrentFlow

功能说明

返回流经指定部件温控回路流量。

函数原型

```
uint16 getCurrentFlow(uint16 devId)
```

参数说明

参数名称	类型	描述
devId	uint16	部件编号

返回值说明

类型	描述
uint16	该部件温控回路当前的流量，单位：L/min

4.2.6 notifyCurrentFlow

功能说明

上报部件温控回路的当前流量。

函数原型

```
TMS_DeviceFlowAry notifyCurrentFlow()
```

参数说明

无。

返回值说明

类型	描述
TMS_DeviceFlowAr y	温控回路的当前流量，单位：L/min，精度：1/10

4.2.7 getCurrentWaterTemp

功能说明

返回指定部件温控回路的当前水温。

函数原型

```
sint16 getCurrentWaterTemp(uint16 devId, TMS_LoopPort portId)
```

参数说明

参数名称	类型	描述
devId	uint16	部件编号
portId	TMS_LoopPort	指定部件温控回路的回路端口

返回值说明

类型	描述
sint16	指定部件温控回路的当前水温，单位：℃，精度：1/10

4.2.8 notifyCurrentWaterTemp

功能说明

上报部件温控回路的当前水温。

函数原型

```
TMS_DeviceWaterTempAry notifyCurrentWaterTemp()
```

参数说明

无。

返回值说明

类型	描述
TMS_DeviceWaterTempAry	部件温控回路的当前水温

4.3 TMS_AC(座舱温控服务)

提供座舱温控服务。包括空调开关、温区目标温度设置、出风口温度获取和光照强度查询等。

TMS_AC(座舱温控服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
TMS_ReturnCode	turnOn()	启动乘员舱温控系统。
TMS_ReturnCode	turnOff()	关闭乘员舱温控系统。
boolean	isRunning()	返回乘员舱温控系统状态。
TMS_ReturnCode	setTargetTemp(uint8 zoneId,sint16 targetTemp)	设置指定温区目标温度。
sint16	getTargetTemp(uint8 zoneId)	返回指定温区目标温度。
sint16	getAirOutletTemp(uint8 outletId)	返回指定出风口当前温度。

4.3.1 turnOn

功能说明

启动乘员舱温控系统。

函数原型

TMS_ReturnCode turnOn()

参数说明

无。

返回值说明

类型	描述
TMS_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

4.3.2 turnOff

功能说明

关闭乘员舱温控系统。

函数原型

TMS_ReturnCode turnOff()

参数说明

无。

返回值说明

类型	描述
TMS_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

4.3.3 isRunning

功能说明

返回乘员舱温控系统状态。

函数原型

boolean isRunning()

参数说明

无。

返回值说明

类型	描述
boolean	乘员舱温控系统运行时返回 TRUE， 处于关闭状态时返回 FALSE

4.3.4 setTargetTemp

功能说明

设置指定温区目标温度。

函数原型

TMS_ReturnCode setTargetTemp(uint8 zoneId, sint16 targetTemp)

参数说明

参数名称	类型	描述
zoneId	uint8	温区编号
targetTemp	sint16	该温区目标温度，单位：℃，精度：1/10

返回值说明

类型	描述
TMS_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

4.3.5 getTargetTemp

功能说明

返回指定温区目标温度。

函数原型

sint16 getTargetTemp(uint8 zoneId)

参数说明

参数名称	类型	描述
zoneId	uint8	温区编号

返回值说明

类型	描述
----	----

类型	描述
sint16	指定温区的目标温度，单位：℃，精度：1/10

4.3.6 getAirOutletTemp

功能说明

返回指定出风口当前温度。

函数原型

```
sint16 getAirOutletTemp(uint8 outletId)
```

参数说明

参数名称	类型	描述
outletId	uint8	出风口编号

返回值说明

类型	描述
sint16	指定出风口的当前温度，单位：℃，精度：1/10

4.4 TMS_Ventilation(通风服务)

提供通风服务。包括出风口风速、出风口风向及内外循环模式的控制等。

TMS_Ventilation(通风服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
TMS_ReturnCode	turnOn(uint8 outletId)	启动指定出风口通风功能。
TMS_ReturnCode	turnOff(uint8 outletId)	关闭指定出风口通风功能。
boolean	isRunning(uint8 outletId)	返回指定出风口通风功能启动状态。
TMS_ReturnCode	setDirection(uint8 outletId, TMS_AirDirection direction)	设置指定出风口吹风风向。

类型	API 函数	功能说明
TMS_AirDirection	<code>getDirection(uint8 outletId)</code>	返回指定出风口吹风风向。
TMS_VentilationDirectionAry	<code>notifyDirection()</code>	上报出风口吹风风向。
TMS_ReturnCode	<code>setLevel(uint8 zoneId,uint8 level)</code>	设置指定温区风速档位。
uint8	<code>getLevel(uint8 zoneId)</code>	返回指定温区风速档位。
TMS_VentilationLevelAry	<code>notifyLevel()</code>	上报温区的风速档位。
TMS_ReturnCode	<code>setMode(uint8 zoneId,TMS_AirMode mode)</code>	设置指定温区的吹风模式。
TMS_AirMode	<code>getMode(uint8 zoneId)</code>	返回指定温区的吹风模式。
TMS_VentilationModeAry	<code>notifyMode()</code>	上报温区的吹风模式。
TMS_ReturnCode	<code>setCyclingMode(TMS_CyclingMode mode)</code>	设置通风循环工作模式。
TMS_CyclingMode	<code>getCyclingMode()</code>	返回通风循环工作模式。
TMS_CyclingMode	<code>notifyCyclingMode()</code>	上报通风循环工作模式。

4.4.1 turnOn

功能说明

启动指定出风口通风功能。

函数原型

TMS_ReturnCode turnOn(uint8 outletId)

参数说明

参数名称	类型	描述
outletId	uint8	出风口编号

返回值说明

类型	描述
TMS_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

4.4.2 turnOff

功能说明

关闭指定出风口通风功能。

函数原型

```
TMS_ReturnCode turnOff(uint8 outletId)
```

参数说明

参数名称	类型	描述
outletId	uint8	出风口编号

返回值说明

类型	描述
TMS_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

4.4.3 isRunning

功能说明

返回指定出风口通风功能启动状态。

函数原型

```
boolean isRunning(uint8 outletId)
```

参数说明

参数名称	类型	描述
outletId	uint8	出风口编号

返回值说明

类型	描述
boolean	通风功能运行时返回 TRUE，否则返回 FALSE

4.4.4 setDirection

功能说明

设置指定出风口吹风风向。

函数原型

TMS_ReturnCode setDirection(uint8 outletId, TMS_AirDirection direction)

参数说明

参数名称	类型	描述
outletId	uint8	出风口编号
direction	TMS_AirDirection	吹风风向

返回值说明

类型	描述
TMS_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

4.4.5 getDirection

功能说明

返回指定出风口吹风风向。

函数原型

TMS_AirDirection getDirection(uint8 outletId)

参数说明

参数名称	类型	描述
outletId	uint8	出风口编号

返回值说明

类型	描述
TMS_AirDirection	吹风风向

4.4.6 notifyDirection

功能说明

上报出风口吹风风向。

函数原型

TMS_VentilationDirectionAry notifyDirection()

参数说明

无。

返回值说明

类型	描述
TMS_VentilationDirectionAry	出风口吹风风向

4.4.7 setLevel

功能说明

设置指定温区风速档位。

函数原型

TMS_ReturnCode setLevel(uint8 zoneId, uint8 level)

参数说明

参数名称	类型	描述
zoneId	uint8	温区编号
level	uint8	风速档位

返回值说明

类型	描述
TMS_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

4.4.8 getLevel

功能说明

返回指定温区风速档位。

函数原型

```
uint8 getLevel(uint8 zoneId)
```

参数说明

参数名称	类型	描述
zoneId	uint8	温区编号

返回值说明

类型	描述
uint8	指定温区风速档位

4.4.9 notifyLevel

功能说明

上报温区的风速档位。

函数原型

```
TMS_VentilationLevelAry notifyLevel()
```

参数说明

无。

返回值说明

类型	描述
TMS_VentilationLevelAry	温区风速档位

4.4.10 setMode

功能说明

设置指定温区的吹风模式。

函数原型

TMS_ReturnCode setMode(uint8 zoneId, TMS_AirMode mode)

参数说明

参数名称	类型	描述
zoneId	uint8	温区编号
mode	TMS_AirMode	吹风模式

返回值说明

类型	描述
TMS_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

4.4.11 getMode

功能说明

返回指定温区的吹风模式。

函数原型

TMS_AirMode getMode(uint8 zoneId)

参数说明

参数名称	类型	描述
------	----	----

参数名称	类型	描述
zoneId	uint8	温区编号

返回值说明

类型	描述
TMS_AirMode	吹风模式

4.4.12 notifyMode

功能说明

上报温区的吹风模式。

函数原型

TMS_VentilationModeAry notifyMode()

参数说明

无。

返回值说明

类型	描述
TMS_VentilationModeAry	温区的吹风模式

4.4.13 setCyclingMode

功能说明

设置通风循环工作模式。

函数原型

TMS_ReturnCode setCyclingMode(TMS_CyclingMode mode)

参数说明

参数名称	类型	描述
------	----	----

参数名称	类型	描述
mode	TMS_CyclingMode	通风循环模式

返回值说明

类型	描述
TMS_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

4.4.14 getCyclingMode

功能说明

返回通风循环工作模式。

函数原型

```
TMS_CyclingMode getCyclingMode()
```

参数说明

无。

返回值说明

类型	描述
TMS_CyclingMode	通风循环模式

4.4.15 notifyCyclingMode

功能说明

上报通风循环工作模式。

函数原型

```
TMS_CyclingMode notifyCyclingMode()
```

参数说明

无。

返回值说明

类型	描述
TMS_CyclingMode	通风循环模式

4.5 TMS_Purifier(空气净化服务)

提供空气净化相关服务。包括空气净化、颗粒物检测、紫外线杀菌及活物检测等功能。

TMS_Purifier(空气净化服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
TMS_ReturnCode	turnOn()	启动空气净化。
TMS_ReturnCode	turnOff()	停止空气净化。
boolean	isRunning()	返回空气净化工作状态。
TMS_ReturnCode	startDetecting()	启动颗粒物检测功能。
TMS_ReturnCode	stopDetecting()	停止颗粒物检测功能。
boolean	isDetecting()	返回颗粒物检测功能工作状态。
TMS_ReturnCode	setDetectMode(TMS_ParticleDetectMode mode)	设置颗粒物检测工作模式。
TMS_ParticleDetectMode	getDetectMode()	返回颗粒物检测工作模式。
uint16	getParticle(TMS_ParticleType type)	返回指定类型颗粒物的浓度。
TMS_PMConcentrationAry	notifyParticle()	上报颗粒物的浓度。
TMS_ReturnCode	startSterilization()	开启紫外线杀菌。
TMS_ReturnCode	stopSterilization()	关闭紫外线杀菌。
boolean	isSterilization()	返回紫外线杀菌功能工作状态。
uint16	getUvLiveDetection()	返回车内活物检测结果。
TMS_ReturnCode	startAQSDetecting()	启动 AQS 空气质量检测功能。

类型	API 函数	功能说明
de		
TMS_ReturnCode	stopAQSDetecting()	停止 AQS 空气质量检测功能。
boolean	isAQSDetecting()	返回 AQS 空气质量检测功能工作状态。
uint16	getAQI()	返回乘员舱空气质量等级。
uint16	notifyAQI()	上报乘员舱空气质量等级。
uint16	getCO2()	返回乘员舱 CO2 浓度。
uint16	notifyCO2()	上报乘员舱 CO2 浓度。

4.5.1 turnOn

功能说明

启动空气净化。

函数原型

TMS_ReturnCode turnOn()

参数说明

无。

返回值说明

类型	描述
TMS_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

4.5.2 turnOff

功能说明

停止空气净化。

函数原型

TMS_ReturnCode turnOff()

参数说明

无。

返回值说明

类型	描述
TMS_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

4.5.3 isRunning

功能说明

返回空气净化工作状态。

函数原型

```
boolean isRunning()
```

参数说明

无。

返回值说明

类型	描述
boolean	空气净化运行时返回 TRUE，否则返回 FALSE

4.5.4 startDetecting

功能说明

启动颗粒物检测功能。

函数原型

```
TMS_ReturnCode startDetecting()
```

参数说明

无。

返回值说明

类型	描述
TMS_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

4.5.5 stopDetecting

功能说明

停止颗粒物检测功能。

函数原型

TMS_ReturnCode stopDetecting()

参数说明

无。

返回值说明

类型	描述
TMS_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

4.5.6 isDetecting

功能说明

返回颗粒物检测功能工作状态。

函数原型

boolean isDetecting()

参数说明

无。

返回值说明

类型	描述
boolean	颗粒物检测运行时返回 TRUE， 否则返回 FALSE

4.5.7 setDetectMode

功能说明

设置颗粒物检测工作模式。

函数原型

TMS_ReturnCode setDetectMode(TMS_ParticleDetectMode mode)

参数说明

参数名称	类型	描述
mode	TMS_ParticleDetectMode	颗粒物检测工作模式

返回值说明

类型	描述
TMS_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

4.5.8 getDetectMode

功能说明

返回颗粒物检测工作模式。

函数原型

TMS_ParticleDetectMode getDetectMode()

参数说明

无。

返回值说明

类型	描述
TMS_ParticleDetect Mode	当前颗粒物检测工作模式

4.5.9 getParticle

功能说明

返回指定类型颗粒物的浓度。

函数原型

```
uint16 getParticle(TMS_ParticleType type)
```

参数说明

参数名称	类型	描述
type	TMS_ParticleType	颗粒物类型

返回值说明

类型	描述
uint16	颗粒物浓度, 单位:ug/m3

4.5.10 notifyParticle

功能说明

上报颗粒物的浓度。

函数原型

```
TMS_PMConcentrationAry notifyParticle()
```

参数说明

无。

返回值说明

类型	描述
TMS_PMConcentrationAry	颗粒物浓度

4.5.11 startSterilization

功能说明

开启紫外线杀菌。

函数原型

TMS_ReturnCode startSterilization()

参数说明

无。

返回值说明

类型	描述
TMS_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

4.5.12 stopSterilization

功能说明

关闭紫外线杀菌。

函数原型

TMS_ReturnCode stopSterilization()

参数说明

无。

返回值说明

类型	描述
TMS_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

4.5.13 isSterilization

功能说明

返回紫外线杀菌功能工作状态。

函数原型

boolean isSterilization()

参数说明

无。

返回值说明

类型	描述
boolean	紫外线杀菌运行时返回 TRUE，否则返回 FALSE

4.5.14 getUvLiveDetection

功能说明

返回车内活物检测结果。

函数原型

uint16 getUvLiveDetection()

参数说明

无。

返回值说明

类型	描述
uint16	活物检测结果

4.5.15 startAQSDetecting

功能说明

启动 AQS 空气质量检测功能。

函数原型

TMS_ReturnCode startAQSDetecting()

参数说明

无。

返回值说明

类型	描述
TMS_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

4.5.16 stopAQSDetecting

功能说明

停止 AQS 空气质量检测功能。

函数原型

```
TMS_ReturnCode stopAQSDetecting()
```

参数说明

无。

返回值说明

类型	描述
TMS_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

4.5.17 isAQSDetecting

功能说明

返回 AQS 空气质量检测功能工作状态。

函数原型

```
boolean isAQSDetecting()
```

参数说明

无。

返回值说明

类型	描述
boolean	AQS 检测功能运行时返回 TRUE，否则返回 FALSE

4.5.18 getAQI

功能说明

返回乘员舱空气质量等级。

函数原型

```
uint16 getAQI()
```

参数说明

无。

返回值说明

类型	描述
uint16	乘员舱空气质量等级

4.5.19 notifyAQI

功能说明

上报乘员舱空气质量等级。

函数原型

```
uint16 notifyAQI()
```

参数说明

无。

返回值说明

类型	描述
uint16	乘员舱空气质量等级

4.5.20 getCO2

功能说明

返回乘员舱 CO2 浓度。

函数原型

```
uint16 getCO2()
```

参数说明

无。

返回值说明

类型	描述
uint16	乘员舱二氧化碳浓度，单位：ppm

4.5.21 notifyCO2

功能说明

上报乘员舱 CO2 浓度。

函数原型

```
uint16 notifyCO2()
```

参数说明

无。

返回值说明

类型	描述
uint16	乘员舱二氧化碳浓度，单位：ppm

4.6 TMS_Fragrance(香氛服务)

提供香氛服务。包括香氛香型和浓度的控制等。

TMS_Fragrance(香氛服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
TMS_ReturnCode	turnOn()	打开香氛。
TMS_ReturnCode	turnOff()	关闭香氛。

类型	API 函数	功能说明
de		
boolean	isRunning()	返回香氛工作状态。
TMS_ReturnCode	setConcentration(uint16 concentration)	设置香氛浓度。
uint16	getConcentration()	返回香氛浓度设置。
TMS_ReturnCode	setType(TMS_FragranceType type)	设置香氛香型。
TMS_FragranceType	getType()	返回香氛香型设置。

4.6.1 turnOn

功能说明

打开香氛。

函数原型

TMS_ReturnCode turnOn()

参数说明

无。

返回值说明

类型	描述
TMS_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

4.6.2 turnOff

功能说明

关闭香氛。

函数原型

TMS_ReturnCode turnOff()

参数说明

无。

返回值说明

类型	描述
TMS_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

4.6.3 isRunning

功能说明

返回香氛工作状态。

函数原型

boolean isRunning()

参数说明

无。

返回值说明

类型	描述
boolean	香氛功能运行时返回 TRUE，否则返回 FALSE

4.6.4 setConcentration

功能说明

设置香氛浓度。

函数原型

TMS_ReturnCode setConcentration(uint16 concentration)

参数说明

参数名称	类型	描述
concentration	uint16	香氛浓度

返回值说明

类型	描述
TMS_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

4.6.5 getConcentration

功能说明

返回香氛浓度设置。

函数原型

```
uint16 getConcentration()
```

参数说明

无。

返回值说明

类型	描述
uint16	当前设置的香氛浓度

4.6.6 setType

功能说明

设置香氛香型。

函数原型

```
TMS_ReturnCode setType(TMS_FragranceType type)
```

参数说明

参数名称	类型	描述
type	TMS_FragranceType	香氛类型

返回值说明

类型	描述
TMS_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

4.6.7 getType

功能说明

返回香氛香型设置。

函数原型

TMS_FragranceType getType()

参数说明

无。

返回值说明

类型	描述
TMS_FragranceType	当前设置的香氛类型

4.7 TMS_Demister(除雾除霜服务)

提供除雾除霜服务。包括除雾除霜强度控制等。

TMS_Demister(除雾除霜服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
TMS_ReturnCode	start(TMS_DemistZoneId zoneId)	启动指定挡风玻璃除雾除霜功能。
TMS_ReturnCode	stop(TMS_DemistZoneId zoneId)	停止指定挡风玻璃除雾除霜功能。
boolean	isRunning(TMS_DemistZoneId zoneId)	返回指定挡风玻璃除雾除霜功能工作状态。
TMS_ReturnCode	setLevel(TMS_DemistZoneId zoneId,uint8 level)	设置指定挡风玻璃除雾除霜等级。
uint8	getLevel(TMS_DemistZoneId zoneId)	返回指定挡风玻璃除雾除霜等级。

4.7.1 start

功能说明

启动指定挡风玻璃除雾除霜功能。

函数原型

TMS_ReturnCode start(TMS_DemistZoneId zoneId)

参数说明

参数名称	类型	描述
zoneId	TMS_DemistZoneId	挡风玻璃编号

返回值说明

类型	描述
TMS_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

4.7.2 stop

功能说明

停止指定挡风玻璃除雾除霜功能。

函数原型

TMS_ReturnCode stop(TMS_DemistZoneId zoneId)

参数说明

参数名称	类型	描述
zoneId	TMS_DemistZoneId	挡风玻璃编号

返回值说明

类型	描述
TMS_ReturnCode	成功返回 SUCCESS，否则返回 FAIL

4.7.3 isRunning

功能说明

返回指定挡风玻璃除雾除霜功能工作状态。

函数原型

```
boolean isRunning(TMS_DemistZoneId zoneId)
```

参数说明

参数名称	类型	描述
zoneId	TMS_DemistZoneId	挡风玻璃编号

返回值说明

类型	描述
boolean	除雾除霜功能运行时返回 TRUE，否则返回 FALSE

4.7.4 setLevel

功能说明

设置指定挡风玻璃除雾除霜等级。

函数原型

```
TMS_ReturnCode setLevel(TMS_DemistZoneId zoneId, uint8 level)
```

参数说明

参数名称	类型	描述
zoneId	TMS_DemistZoneId	挡风玻璃编号
level	uint8	除霜除雾等级

返回值说明

类型	描述
----	----

类型	描述
TMS_ReturnCode	成功返回 SUCCESS， 否则返回 FAIL

4.7.5 getLevel

功能说明

返回指定挡风玻璃除雾除霜等级。

函数原型

```
uint8 getLevel(TMS_DemistZoneId zoneId)
```

参数说明

参数名称	类型	描述
zoneId	TMS_DemistZoneId	挡风玻璃编号

返回值说明

类型	描述
uint8	除雾除霜等级

4.8 TMS_EnvMonitor(环境服务)

提供环境监控、环境感知服务。包括车内外温度查询、车内湿度查询、前档阳光雨量以及乘员舱空气质量相关查询等。

TMS_EnvMonitor(环境服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
sint16	getTemp(TMS_EnvZone zoneId)	返回指定区域的温度。
TMS_EnvZone TempAry	notifyTemp()	上报区域温度。
uint8	getHumidity(TMS_EnvZone zoneId)	返回指定区域湿度。
TMS_EnvZone	notifyHumidity()	上报区域湿度。

类型	API 函数	功能说明
HumidityAry		
uint16	getLightIntensity(TMS_EnvZone zoneId)	返回指定区域光照强度。
TMS_EnvZoneLightIntensityAry	notifyLightIntensity()	上报区域光照强度。

4.8.1 getTemp

功能说明

返回指定区域的温度。

函数原型

```
sint16 getTemp(TMS_EnvZone zoneId)
```

参数说明

参数名称	类型	描述
zoneId	TMS_EnvZone	区域编号

返回值说明

类型	描述
sint16	指定区域的温度，单位：℃，精度：1/10

4.8.2 notifyTemp

功能说明

上报区域温度。

函数原型

```
TMS_EnvZoneTempAry notifyTemp()
```

参数说明

无。

返回值说明

类型	描述
TMS_EnvZoneTemp Ary	区域温度

4.8.3 getHumidity

功能说明

返回指定区域湿度。

函数原型

```
uint8 getHumidity(TMS_EnvZone zoneId)
```

参数说明

参数名称	类型	描述
zoneId	TMS_EnvZone	区域编号

返回值说明

类型	描述
uint8	指定区域的湿度，单位：%

4.8.4 notifyHumidity

功能说明

上报区域湿度。

函数原型

```
TMS_EnvZoneHumidityAry notifyHumidity()
```

参数说明

无。

返回值说明

类型	描述
TMS_EnvZoneHumidityAry	区域湿度

4.8.5 getLightIntensity

功能说明

返回指定区域光照强度。

函数原型

```
uint16 getLightIntensity(TMS_EnvZone zoneId)
```

参数说明

参数名称	类型	描述
zoneId	TMS_EnvZone	区域编号

返回值说明

类型	描述
uint16	指定区域的光照强度，单位：w/m2

4.8.6 notifyLightIntensity

功能说明

上报区域光照强度。

函数原型

```
TMS_EnvZoneLightIntensityAry notifyLightIntensity()
```

参数说明

无。

返回值说明

类型	描述
TMS_EnvZoneLightIntensityAry	区域光照强度

4.9 数据类型定义

4.9.1 TMS_ReturnCode

说明

API 返回值。

定义

```
typedef enum {
    SUCCESS=0,
    FAIL=1
} TMS_ReturnCode;
```

成员

成员名称	描述
SUCCESS	API 返回成功
FAIL	API 返回失败

4.9.2 TMS_ParticleDetectMode

说明

颗粒物检测工作模式。

定义

```
typedef enum {
    SILENT MODE=0,
    NORMAL MODE=1
} TMS_ParticleDetectMode;
```

成员

成员名称	描述
SILENT_MODE	静音模式
NORMAL_MODE	普通模式

4.9.3 TMS_ParticleType

说明

颗粒物类型。

定义

```
typedef enum {
    PM2D5=0,
    PM10=1
} TMS_ParticleType;
```

成员

成员名称	描述
PM2D5	PM2.5
PM10	PM10

4.9.4 TMS_AirMode

说明

通风功能吹风模式。

定义

```
typedef enum {
    FACE=0,
    FACE_FOOT=1,
    FOOT=2,
    FOOT DEFROST=3,
    DEFROST=4,
    FACE DEFROST=5,
    FACE FOOT WINDOW=6
} TMS AirMode;
```

成员

成员名称	描述
FACE	吹脸
FACE_FOOT	吹脸和吹脚
FOOT	吹脚
FOOT_DEFROST	吹脚和除霜
DEFROST	除霜
FACE_DEFROST	吹面和除霜
FACE_FOOT_WINDO W	吹面吹脚和吹窗

4.9.5 TMS_AirDirection

说明

出风口吹风风向。

定义

```
typedef struct {
    sint16 xDir;
    sint16 yDir;
} TMS AirDirection;
```

成员

成员名称	描述
xDir	X 轴风向角度(左右)
yDir	Y 轴风向角度(上下)

4.9.6 TMS_CyclingMode

说明

通风循环模式。

定义

```
typedef enum {
    INTERIOR_CYCLE=0,
    EXTERIOR_CYCLE=1,
    AUTO_CYCLE=2
} TMS CyclingMode;
```

成员

成员名称	描述
INTERIOR_CYCLE	内循环
EXTERIOR_CYCLE	外循环
AUTO_CYCLE	自动循环

4.9.7 TMS_FragranceType

说明

香氛香型。

定义

```
typedef enum {
    GULONG=0,
    OCEAN=1,
    APPLE=2,
    LEMON=3,
    ROSE=4
} TMS FragranceType;
```

成员

成员名称	描述
GULONG	古龙香型
OCEAN	海洋香型
APPLE	苹果香型
LEMON	柠檬香型
ROSE	玫瑰香型

4.9.8 TMS_DemistZoneId

说明

挡风玻璃 ID。

定义

```
typedef enum {  
    FRONT WINDSHIELD=0,  
    REAR WINDSHIELD=1  
} TMS_DemistZoneId;
```

成员

成员名称	描述
FRONT_WINDSHIELD	前挡风玻璃
REAR_WINDSHIELD	后挡风玻璃

4.9.9 TMS_EnvZone

说明

环境区域。

定义

```
typedef enum {  
    INTERIOR_ZONE=0,  
    EXTERIOR_ZONE=1  
} TMS_EnvZone;
```

成员

成员名称	描述
INTERIOR_ZONE	内部
EXTERIOR_ZONE	外部

4.9.10 TMS_VentilationDirectionAry

说明

出风口吹风风向数组。

定义

```
typedef TMS WindDirectionSt TMS VentilationDirectionAry[20];
```

成员

成员名称	描述
TMS_VentilationDirectionAry	出风口吹风风向数组，数组长度 20

4.9.11 TMS_WindDirectionSt

说明

出风口吹风风向信息。

定义

```
typedef struct {
    uint8 outletId;
    TMS AirDirection direction;
} TMS WindDirectionSt;
```

成员

成员名称	描述
outletId	出风口编号
direction	出风口吹风风向

4.9.12 TMS_VentilationLevelAry

说明

温区风速档位数组。

定义

```
typedef TMS WindLevelSt TMS VentilationLevelAry[6];
```

成员

成员名称	描述
TMS_VentilationLevelAry	温区风速档位数组，数组长度 6

4.9.13 TMS_WindLevelSt

说明

温区风速档位信息。

定义

```
typedef struct {
    uint8 zoneId;
    uint8 level;
} TMS WindLevelSt;
```

成员

成员名称	描述
zoneId	温区编号
level	温区风速档位

4.9.14 TMS_VentilationModeAry

说明

温区吹风模式数组。

定义

```
typedef TMS WindModeSt TMS VentilationModeAry[6];
```

成员

成员名称	描述
------	----

成员名称	描述
TMS_VentilationModeArray	温区吹风模式数组，数组长度 6

4.9.15 TMS_WindModeSt

说明

温区吹风模式信息。

定义

```
typedef struct {
    uint8 zoneId;
    TMS AirMode mode;
} TMS WindModeSt;
```

成员

成员名称	描述
zoneId	温区编号
mode	温区吹风模式

4.9.16 TMS_PMConcentrationAry

说明

颗粒物浓度数组。

定义

```
typedef TMS_PMConcentrationSt TMS_PMConcentrationAry[2];
```

成员

成员名称	描述
TMS_PMConcentrationAry	颗粒物浓度数组，数组长度 2

4.9.17 TMS_PMConcentrationSt

说明

颗粒物浓度信息。

定义

```
typedef struct {  
    TMS ParticleType particleType;  
    uint16 concentration;  
} TMS PMConcentrationSt;
```

成员

成员名称	描述
particleType	颗粒物类型
concentration	颗粒物浓度，单位:ug/m3

4.9.18 TMS_EnvZoneTempAry

说明

环境区域温度数组。

定义

```
typedef TMS EnvZoneTempSt TMS EnvZoneTempAry[2];
```

成员

成员名称	描述
TMS_EnvZoneTempAry	环境区域温度数组，数组长度 2

4.9.19 TMS_EnvZoneTempSt

说明

环境区域温度信息。

定义

```
typedef struct {
    TMS_EnvZone zoneId;
    sint16 temp;
} TMS_EnvZoneTempSt;
```

成员

成员名称	描述
zoneId	环境区域
temp	环境区域温度，单位：℃，精度：1/10

4.9.20 TMS_EnvZoneHumidityAry

说明

环境区域湿度数组。

定义

```
typedef TMS_EnvZoneHumiditySt TMS_EnvZoneHumidityAry[2];
```

成员

成员名称	描述
TMS_EnvZoneHumidityAry	环境区域湿度数组，数组长度 2

4.9.21 TMS_EnvZoneHumiditySt

说明

环境区域湿度信息。

定义

```
typedef struct {
    TMS_EnvZone zoneId;
    uint8 humidity;
} TMS_EnvZoneHumiditySt;
```

成员

成员名称	描述
zoneId	环境区域
humidity	环境区域湿度，单位：%

4.9.22 TMS_EnvZoneLightIntensityAry

说明

环境区域光照强度数组。

定义

```
typedef TMS_EnvZoneLightIntensitySt TMS_EnvZoneLightIntensityAry[2];
```

成员

成员名称	描述
TMS_EnvZoneLightIntensityAry	环境区域光照强度数组，数组长度 2

4.9.23 TMS_EnvZoneLightIntensitySt

说明

环境区域光照强度信息。

定义

```
typedef struct {
    TMS_EnvZone zoneId;
    uint16 lightIntensity;
} TMS_EnvZoneLightIntensitySt;
```

成员

成员名称	描述
zoneId	环境区域
lightIntensity	环境区域的光照强度，单位：w/m2

4.9.24 TMS_BatteryMode

说明

电池包温控回路工作模式。

定义

```
typedef enum {
    OFF=0,
    COOLING=1,
    HEATING=2,
    EQUALIZING=3
} TMS_BatteryMode;
```

成员

成员名称	描述
OFF	关闭
COOLING	冷却模式
HEATING	加热模式
EQUALIZING	均温模式

4.9.25 TMS_LoopPort

说明

回路端口。

定义

```
typedef enum {
    INLET=0,
    OUTLET=1
} TMS_LoopPort;
```

成员

成员名称	描述
INLET	进口
OUTLET	出口

4.9.26 TMS_DeviceFlowAry

说明

设备散热温控回路流量数组。

定义

```
typedef TMS_DeviceFlowSt TMS_DeviceFlowAry[6];
```

成员

成员名称	描述
TMS_DeviceFlowAry	设备散热温控回路流量数组，数组长度 6

4.9.27 TMS_DeviceFlowSt

说明

设备散热温控回路流量信息。

定义

```
typedef struct {  
    uint16 devId;  
    uint16 flow;  
} TMS_DeviceFlowSt;
```

成员

成员名称	描述
devId	部件编号
flow	部件散热温控回路的流量，单位：L/min，精度：1/10

4.9.28 TMS_DeviceWaterTempAry

说明

设备散热温控回路水温数组。

定义

```
typedef TMS_DeviceWaterTempSt TMS_DeviceWaterTempAry[12];
```

成员

成员名称	描述
TMS_DeviceWaterTempAry	设备散热温控回路水温数组，数组长度 12

4.9.29 TMS_DeviceWaterTempSt

说明

设备散热温控回路水温信息。

定义

```
typedef struct {
    uint16 devId;
    TMS_LoopPort portId;
    sint16 temp;
} TMS_DeviceWaterTempSt;
```

成员

成员名称	描述
devId	部件编号
portId	设备散热温控回路端口
temp	设备散热温控回路端口水温，单位：℃，精度：1/10

5 VCS(运动控制)原子服务 API

5.1 VCS_VehSpd(基础车速服务)

提供基础车速服务，服务提供基于 ESC 上报车速和电机转速综合后的车速信息。

VCS_VehSpd(基础车速服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
VCS_VehSpd	notifyStatus()	上报基础车速。

5.1.1 notifyStatus

功能说明

上报基础车速。

函数原型

VCS_VehSpd notifyStatus()

参数说明

无。

返回值说明

类型	描述
VCS_VehSpd	基础计算车速及有效位

5.2 VCS_BrakePedal(制动状态服务)

上报制动踏板的开关，制动行程和开度等相关信息。

VCS_BrakePedal(制动状态服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
VCS_BrakeSwitch	notifySwitch()	上报制动踏板开关切换状态。
VCS_BrakePosn	notifyStatus()	上报制动踏板行程和开度状态。

5.2.1 notifySwitch

功能说明

上报制动踏板开关切换状态。

函数原型

VCS_BrakeSwitch notifySwitch()

参数说明

无。

返回值说明

类型	描述
VCS_BrakeSwitch	制动踏板及有效位

5.2.2 notifyStatus

功能说明

上报制动踏板行程和开度状态。

函数原型

VCS_BrakePosn notifyStatus()

参数说明

无。

返回值说明

类型	描述
VCS_BrakePosn	制动开度及有效位

5.3 VCS_AccrPedal(加速踏板状态服务)

上报加速踏板的开度，kickdown 状态信息。

VCS_AccrPedal(加速踏板状态服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
VCS_AccrPedal	notifyStatus()	上报加速踏板状态。

5.3.1 notifyStatus

功能说明

上报加速踏板状态。

函数原型

VCS_AccrPedal notifyStatus()

参数说明

无。

返回值说明

类型	描述
VCS_AccrPedal	加速踏板状态及有效位

5.4 VCS_Gear(档位状态服务)

提供车辆档位识别接口，可以接受上层档位的请求；提供实际的档位状态服务。

VCS_Gear(档位状态服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
boolean	setTarget(VCS_Gear targetGear)	设置目标档位。
VCS_Gear	notifyStatus()	上报当前档位。

5.4.1 setTarget

功能说明

设置目标档位。

函数原型

```
boolean setTarget(VCS_Gear targetGear)
```

参数说明

参数名称	类型	描述
targetGear	VCS_Gear	目标档位请求

返回值说明

类型	描述
boolean	成功返回 SUCCESS，否则返回 FAIL

5.4.2 notifyStatus

功能说明

上报当前档位。

函数原型

```
VCS_Gear notifyStatus()
```

参数说明

无。

返回值说明

类型	描述
VCS_Gear	当前档位及有效位

5.5 VCS_TqDmd(扭矩计算服务)

实现爬行扭矩计算、巡航扭矩计算、滑行扭矩计算、驱动扭矩计算、ADS 扭矩计算、制动扭矩计算（主要为电机的制动回收扭矩和制动的压力请求）。

VCS_TqDmd(扭矩计算服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
sint32	notifyCreepTq()	上报爬行的扭矩信息。
boolean	setCreepSpd(sint32 creepSpd)	设置目标爬行车速。
sint32	notifyOnePedalTq()	上报单踏板扭矩信息。
sint32	notifyCrsTq()	上报巡航扭矩信息。
boolean	setCrsSpd(sint32 crsSpd)	设置目标巡航速度。
sint32	notifyCoastRgnTq()	上报滑行能量回收计算扭矩。
void	setCoastRgnTqCoeff(unit8 coastRgnTqCoeff)	设置滑行能量回收系数。
sint32	notifyDriveTq()	上报驱动计算扭矩。
sint32	notifyADSTq()	上报 ADS 计算扭矩。
void	setADSTarget(VCS_ADSTarget ADSTarget)	设置 ADS 目标请求。
void	setDrivePedalPosn(unit16 drivePedalPosn)	设置加速踏板开度。
VCS_TotalBrakeTq	notifyTotalBrakeTq()	上报制动总扭矩，包括制动和电机产生的叠加负扭矩。

5.5.1 notifyCreepTq

功能说明

上报爬行的扭矩信息。

函数原型

sint32 notifyCreepTq()

参数说明

无。

返回值说明

类型	描述
sint32	Creep 扭矩需求, 单位: Nm, 精度: 0.0625

5.5.2 setCreepSpd

功能说明

设置目标爬行车速。

函数原型

boolean setCreepSpd(sint32 creepSpd)

参数说明

参数名称	类型	描述
creepSpd	sint32	目标爬行车速, 单位: km/h, 精度: 0.0625

返回值说明

类型	描述
boolean	成功返回 SUCCESS, 否则返回 FAIL

5.5.3 notifyOnePedalTq

功能说明

上报单踏板扭矩信息。

函数原型

sint32 notifyOnePedalTq()

参数说明

无。

返回值说明

类型	描述
sint32	单踏板 Pedalmap 查表扭矩，单位：Nm，精度：0.0625

5.5.4 notifyCrsTq

功能说明

上报巡航扭矩信息。

函数原型

sint32 notifyCrsTq()

参数说明

无。

返回值说明

类型	描述
sint32	巡航扭矩需求，单位：Nm，精度：0.0625

5.5.5 setCrsSpd

功能说明

设置目标巡航速度。

函数原型

boolean setCrsSpd(sint32 crsSpd)

参数说明

参数名称	类型	描述
crsSpd	sint32	目标巡航车速，单位：km/h，精度：0.0625

返回值说明

类型	描述
boolean	成功返回 SUCCESS，否则返回 FAIL

5.5.6 notifyCoastRgnTq

功能说明

上报滑行能量回收计算扭矩。

函数原型

```
sint32 notifyCoastRgnTq()
```

参数说明

无。

返回值说明

类型	描述
sint32	滑行扭矩需求，单位：Nm，精度：0.0625

5.5.7 setCoastRgnTqCoeff

功能说明

设置滑行能量回收系数。

函数原型

```
void setCoastRgnTqCoeff(unit8 coastRgnTqCoeff)
```

参数说明

参数名称	类型	描述
coastRgnTq Coeff	unit8	滑行回收扭矩系数，取值范围：0~1，精度：0.1

返回值说明

类型	描述
void	返回值为空

5.5.8 notifyDriveTq

功能说明

上报驱动计算扭矩。

函数原型

```
sint32 notifyDriveTq()
```

参数说明

无。

返回值说明

类型	描述
sint32	驱动扭矩需求，单位：Nm，精度：0.0625

5.5.9 notifyADSTq

功能说明

上报 ADS 计算扭矩。

函数原型

```
sint32 notifyADSTq()
```

参数说明

无。

返回值说明

类型	描述
sint32	ADS 扭矩计算, 单位: Nm, 精度: 0.0625

5.5.10 setADSTarget

功能说明

设置 ADS 目标请求。

函数原型

```
void setADSTarget(VCS_ADSTarget ADSTarget)
```

参数说明

参数名称	类型	描述
ADSTarget	VCS_ADSTarget	ADS 输入请求

返回值说明

类型	描述
void	返回值为空

5.5.11 setDrivePedalPosn

功能说明

设置加速踏板开度。

函数原型

```
void setDrivePedalPosn(unit16 drivePedalPosn)
```

参数说明

参数名称	类型	描述
drivePedalPosn	unit16	驱动加速踏板开度值，单位：% ，精度 0.0625

返回值说明

类型	描述
void	返回值为空

5.5.12 notifyTotalBrakeTq

功能说明

上报制动总扭矩，包括制动和电机产生的叠加负扭矩。

函数原型

VCS_TotalBrakeTq notifyTotalBrakeTq()

参数说明

无。

返回值说明

类型	描述
VCS_TotalBrakeTq	制动力协调控制扭矩

5.6 VCS_SteerWhl(方向盘状态和控制)

提供方向盘角度，角速度，扭矩信息和方向盘转向角度和扭矩控制服务。

VCS_SteerWhl(方向盘状态和控制)服务包括的 API 如下表所示：

类型	API 函数	功能说明
VCS_SteerAngle	notifyAngle()	上报当前的方向盘角度。

类型	API 函数	功能说明
void	setAngle(VCS_SteerAngle steerAngle)	设置目标方向盘角度。
VCS_SteerAngleSpd	notifyAngleSpd()	上报当前的方向盘角速度。
VCS_SteerTq	notifyTq()	上报当前的方向盘转向扭矩。
void	setOverlayTq(VCS_SteerTq steerTq)	设置目标方向盘转向叠加力矩。

5.6.1 notifyAngle

功能说明

上报当前的方向盘角度。

函数原型

VCS_SteerAngle notifyAngle()

参数说明

无。

返回值说明

类型	描述
VCS_SteerAngle	当前的方向盘角度

5.6.2 setAngle

功能说明

设置目标方向盘角度。

函数原型

void setAngle(VCS_SteerAngle steerAngle)

参数说明

参数名称	类型	描述
steerAngle	VCS_SteerAngle	目标方向盘角度

返回值说明

类型	描述
void	返回值为空

5.6.3 notifyAngleSpd

功能说明

上报当前的方向盘角速度。

函数原型

```
VCS_SteerAngleSpd notifyAngleSpd()
```

参数说明

无。

返回值说明

类型	描述
VCS_SteerAngleSpd	当前的方向盘角速度

5.6.4 notifyTq

功能说明

上报当前的方向盘转向扭矩。

函数原型

```
VCS_SteerTq notifyTq()
```

参数说明

无。

返回值说明

类型	描述
----	----

类型	描述
VCS_SteerTq	当前的方向盘转向扭矩

5.6.5 setOverlayTq

功能说明

设置目标方向盘转向叠加力矩。

函数原型

```
void setOverlayTq(VCS_SteerTq steerTq)
```

参数说明

参数名称	类型	描述
steerTq	VCS_SteerTq	目标方向盘转向叠加力矩

返回值说明

类型	描述
void	返回值为空

5.7 VCS_Steer(转向状态和控制)

提供前轮转角及后轮转角控制服务。

VCS_Steer(转向状态和控制)服务包括的 API 如下表所示:

类型	API 函数	功能说明
VCS_WhlAngle	notifyFrntWhlAngle()	上报当前的前轮转向角度（前轴两轮转角平均值）。
void	setFrntWhlAngle(VCS_WhlAngle whlAngle)	设置目标前轮转向角度。
VCS_WhlAngle	notifyReWhlAngle()	上报当前的后轮转向角度（后轴两轮转角平均值）。
void	setReWhlAngle(VCS_WhlAngle whlAngle)	设置目标后轮转向角度。

5.7.1 notifyFrntWhlAngle

功能说明

上报当前的前轮转向角度（前轴两轮转角平均值）。

函数原型

```
VCS_WhlAngle notifyFrntWhlAngle()
```

参数说明

无。

返回值说明

类型	描述
VCS_WhlAngle	当前的前轮转向角度

5.7.2 setFrntWhlAngle

功能说明

设置目标前轮转向角度。

函数原型

```
void setFrntWhlAngle(VCS_WhlAngle whlAngle)
```

参数说明

参数名称	类型	描述
whlAngle	VCS_WhlAngle	目标前轮转向角度

返回值说明

类型	描述
void	返回值为空

5.7.3 notifyReWhlAngle

功能说明

上报当前的后轮转向角度（后轴两轮转角平均值）。

函数原型

```
VCS_WhlAngle notifyReWhlAngle()
```

参数说明

无。

返回值说明

类型	描述
VCS_WhlAngle	当前的后轮转向角度

5.7.4 setReWhlAngle

功能说明

设置目标后轮转向角度。

函数原型

```
void setReWhlAngle(VCS_WhlAngle whlAngle)
```

参数说明

参数名称	类型	描述
whlAngle	VCS_WhlAngle	目标后轮转向角度

返回值说明

类型	描述
void	返回值为空

5.8 VCS_Motion(车辆运动状态)

提供车辆运动状态服务，包括侧向加速度查询、纵向加速度查询以及横摆角速度查询。

VCS_Motion(车辆运动状态)服务包括的 API 如下表所示：

类型	API 函数	功能说明
VCS_Acceleration	notifyAx()	上报纵向加速度。
VCS_Acceleration	notifyAy()	上报侧向加速度。
VCS_YawRate	notifyYawRate()	上报横摆角速度。

5.8.1 notifyAx

功能说明

上报纵向加速度。

函数原型

VCS_Acceleration notifyAx()

参数说明

无。

返回值说明

类型	描述
VCS_Acceleration	当前纵向加速度及有效位

5.8.2 notifyAy

功能说明

上报侧向加速度。

函数原型

VCS_Acceleration notifyAy()

参数说明

无。

返回值说明

类型	描述
VCS_Acceleration	当前侧向加速度及有效位

5.8.3 notifyYawRate

功能说明

上报横摆角速度。

函数原型

VCS_YawRate notifyYawRate()

参数说明

无。

返回值说明

类型	描述
VCS_YawRate	横摆角速度

5.9 VCS_Brake(制动状态和控制)

提供制动状态和控制服务。包括制动主缸压力查询、目标纵向加速度查询、总纵向制动力请求以及扭矩请求等。

VCS_Brake(制动状态和控制)服务包括的 API 如下表所示：

类型	API 函数	功能说明
VCS_MCPPrs	notifyMCPPrs()	上报制动主缸压力。
void	setTargetAx(VCS_Acceleration ax)	设置目标纵向减速度。
void	setTotalForce(VCS_LgtBrkFTotal lgtBrkFTotal)	设置纵向总制动力。

类型	API 函数	功能说明
VCS_TqCtrl	notifyTq()	上报制动系统的执行扭矩。

5.9.1 notifyMCPrs

功能说明

上报制动主缸压力。

函数原型

VCS_MCPrs notifyMCPrs()

参数说明

无。

返回值说明

类型	描述
VCS_MCPrs	制动主缸压力

5.9.2 setTargetAx

功能说明

设置目标纵向减速度。

函数原型

void setTargetAx(VCS_Acceleration ax)

参数说明

参数名称	类型	描述
ax	VCS_Acceleration	目标纵向减速度

返回值说明

类型	描述
----	----

类型	描述
void	返回值为空

5.9.3 setTotalForce

功能说明

设置纵向总制动力。

函数原型

```
void setTotalForce(VCS_LgtBrkFTotal lgtBrkFTotal)
```

参数说明

参数名称	类型	描述
lgtBrkFTotal	VCS_LgtBrkFTotal	纵向制动力

返回值说明

类型	描述
void	返回值为空

5.9.4 notifyTq

功能说明

上报制动系统的执行扭矩。

函数原型

```
VCS_TqCtrl notifyTq()
```

参数说明

无。

返回值说明

类型	描述
----	----

类型	描述
VCS_TqCtrl	制动控制扭矩请求

5.10 VCS_Whl(车轮状态服务)

提供车轮状态服务。包括轮速查询、轮速脉冲查询以及轮胎胎压查询等。

VCS_Whl(车轮状态服务)服务包括的 API 如下表所示:

类型	API 函数	功能说明
VCS_WhlSpd	notifySpd()	上报轮速信息。
VCS_WhlSpdPls	notifySpdPls()	上报轮速脉冲。
VCS_TirePrs	notifyTirePrs()	上报轮胎胎压。

5.10.1 notifySpd

功能说明

上报轮速信息。

函数原型

VCS_WhlSpd notifySpd()

参数说明

无。

返回值说明

类型	描述
VCS_WhlSpd	轮速信息

5.10.2 notifySpdPls

功能说明

上报轮速脉冲。

函数原型

VCS_WhlSpdPls notifySpdPls()

参数说明

无。

返回值说明

类型	描述
VCS_WhlSpdPls	轮速脉冲

5.10.3 notifyTirePrs

功能说明

上报轮胎胎压。

函数原型

VCS_TirePrs notifyTirePrs()

参数说明

无。

返回值说明

类型	描述
VCS_TirePrs	轮胎胎压

5.11 VCS_EPB(电子驻车服务)

提供电子驻车服务。包括设置 EPB 开关状态以及 EPB 执行状态查询。

VCS_EPB(电子驻车服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
boolean	setSwitchStatus(VCS_EPBSwitch EPBSwitchStatus)	设置 EPB 状态。

类型	API 函数	功能说明
VCS_EPB	notifyStatus()	上报当前的 EPB 状态。

5.11.1 setSwitchStatus

功能说明

设置 EPB 状态。

函数原型

```
boolean setSwitchStatus(VCS_EPBSwitch EPBSwitchStatus)
```

参数说明

参数名称	类型	描述
EPBSwitchStatus	VCS_EPBSwitch	设置 EPB 状态

返回值说明

类型	描述
boolean	成功返回 SUCCESS，否则返回 FAIL

5.11.2 notifyStatus

功能说明

上报当前的 EPB 状态。

函数原型

```
VCS_EPB notifyStatus()
```

参数说明

无。

返回值说明

类型	描述
----	----

类型	描述
VCS_EPB	当前的 EPB 状态

5.12 VCS_Slope(坡度服务)

提供车辆的静态坡度服务。

VCS_Slope(坡度服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
VCS_Slope	notifystaticSlope()	上报车辆的静态坡度。

5.12.1 notifystaticSlope

功能说明

上报车辆的静态坡度。

函数原型

VCS_Slope notifystaticSlope()

参数说明

无。

返回值说明

类型	描述
VCS_Slope	当前的车辆坡度信息

5.13 VCS_MotCtrl(驱动电机控制服务)

提供驱动电机的状态信息如：工作模式，输出扭矩，电机转速，主动阻尼状态；同时提供电机控制接口服务，包括电机的请求工作模式，请求扭矩，请求转速和请求主动阻尼模式。

VCS_MotCtrl(驱动电机控制服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
VCS_MotStatus	notifyStatus()	上报电机控制状态。
void	setMotTq(sint16 motTq)	设置电机请求驱动扭矩。
void	setMotSpd(sint16 motSpd)	设置电机请求转速。
void	setMotMode(VCS_MotMode motMode)	设置电机工作模式。
void	setDampingMode(uint8 dampingMode)	设置电机主动阻尼模式。

5.13.1 notifyStatus

功能说明

上报电机控制状态。

函数原型

```
VCS_MotStatus notifyStatus()
```

参数说明

无。

返回值说明

类型	描述
VCS_MotStatus	当前驱动电机的工作模式，输出扭矩，电机转速状态，主动阻尼状态

5.13.2 setMotTq

功能说明

设置电机请求驱动扭矩。

函数原型

```
void setMotTq(sint16 motTq)
```

参数说明

参数名称	类型	描述
motTq	sint16	电机的请求驱动扭矩

返回值说明

类型	描述
void	返回值为空

5.13.3 setMotSpd

功能说明

设置电机请求转速。

函数原型

```
void setMotSpd(sint16 motSpd)
```

参数说明

参数名称	类型	描述
motSpd	sint16	电机的请求转速

返回值说明

类型	描述
void	返回值为空

5.13.4 setMotMode

功能说明

设置电机工作模式。

函数原型

```
void setMotMode(VCS_MotMode motMode)
```

参数说明

参数名称	类型	描述
motMode	VCS_MotMode	电机的工作模式请求

返回值说明

类型	描述
void	返回值为空

5.13.5 setDampingMode

功能说明

设置电机主动阻尼模式。

函数原型

```
void setDampingMode(uint8 dampingMode)
```

参数说明

参数名称	类型	描述
dampingMode	uint8	电机的主动阻尼模式请求

返回值说明

类型	描述
void	返回值为空

5.14 VCS_Crash(碰撞服务)

提供碰撞状态信息服务，用于表征车辆是否发生碰撞。

VCS_Crash(碰撞服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
----	--------	------

类型	API 函数	功能说明
VCS_Crash	notifyStatus()	上报碰撞状态。

5.14.1 notifyStatus

功能说明

上报碰撞状态。

函数原型

VCS_Crash notifyStatus()

参数说明

无。

返回值说明

类型	描述
VCS_Crash	碰撞状态及有效位

5.15 数据类型定义

5.15.1 VCS_VehSpd

说明

基础车速计算。

定义

```
typedef struct {
    sint32 absoluteValue;
    sint32 actualValue;
    uint8 direction;
    boolean directionValid;
    boolean valid;
} VCS_VehSpd;
```

成员

成员名称	描述
absoluteValue	车速绝对值，单位：km/h，精度：0.0625
actualValue	车速实际值，单位：km/h，精度：0.0625
direction	车速方向： 0x0: Forward 0x1: Backward 0x2: Unknown
directionValid	车速方向标志位： False: 无效 True: 有效
valid	车速有效标志位： False: 无效 True: 有效

5.15.2 VCS_BrakeSwitch

说明

制动踏板计算。

定义

```
typedef struct {
    uint8 switchStatus;
    boolean switchValid;
} VCS BrakeSwitch;
```

成员

成员名称	描述
switchStatus	制动踏板踩下状态 0x0: 未踩下 0x1: 踩下
switchValid	制动踏板开关有效标志位： False: 无效 True: 有效

5.15.3 VCS_AccrPedal

说明

加速踏板计算。

定义

```
typedef struct {
    uint16 position;
    boolean valid;
    uint8 switchStatus;
    uint8 kickdownStatus;
    uint8 faultStatus;
    sint32 positonRate;
} VCS AccrPedal;
```

成员

成员名称	描述
position	加速踏板开度值，单位：%，精度：0.0625
valid	加速踏板有效标志位： False：无效 True：有效
switchStatus	加速踏板踩下状态 0x0：未踩下 0x1：踩下
kickdownStatus	加速踏板 kick down 状态： 0x0：未激活 0x1：激活
faultStatus	加速踏板故障状态： 0x0：无故障 0x1：第一路故障 0x2：第二路故障 0x3：双路都故障 0x4：双路差异故障
positonRate	加速踏板开度变化率，单位：%/s，精度：0.0625

5.15.4 VCS_Gear

说明

档位状态。

定义

```
typedef struct {
    uint8 status;
    boolean valid;
} VCS Gear;
```

成员

成员名称	描述
status	当前档位的具体信息： 0x0: 缺省值 0x1: P 档 0x2: R 档 0x3: N 档 0x4: D 档
valid	当前档位有效标志位： False: 无效 True: 有效

5.15.5 VCS_ADSTarget

说明

ADS 请求。

定义

```
typedef struct {
    sint16 accValue;
    boolean accValueValid;
    sint32 torque;
    boolean torqueValid;
    sint32 spdValue;
    boolean spdValueValid;
} VCS ADSTarget;
```

成员

成员名称	描述
accValue	ADS 请求加速度, 单位: m/s ² , 精度: 0.0625
accValueValid	ADS 请求加速度标志位: False: No Request True: Request
torque	ADS 扭矩请求, 单位: Nm, 精度: 0.0625
torqueValid	ADS 请求扭矩标志位: False: 无效 True: 有效
spdValue	ADS 请求车速, 单位: km/h, 精度: 0.0625
spdValueValid	ADS 请求车速有效标志位: False: 无效 True: 有效

5.15.6 VCS_TotalBrakeTq

说明

制动力协调控制。

定义

```
typedef struct {
    sint32 value;
    sint32 frontAxle;
    sint32 rearAxle;
    uint16 axleTargetPressure;
} VCS_TotalBrakeTq;
```

成员

成员名称	描述
value	电机回收总扭矩, 单位: Nm, 精度: 0.0625
frontAxle	前轴电机回收扭矩, 单位: Nm, 精度: 0.0625
rearAxle	后轴电机回收扭矩, 单位: Nm, 精度: 0.0625
axleTargetPressure	总目标压力, 单位: Bar, 精度: 0.0078125

5.15.7 VCS_SteerAngle

说明

转向角度。

定义

```
typedef struct {
    sint16 value;
    boolean valid;
} VCS SteerAngle;
```

成员

成员名称	描述
value	转向角度(向左为正, 向右为负), 单位: deg, 精度: 0.03125
valid	转向角度标志位: False: 无效 True: 有效

5.15.8 VCS_SteerAngleSpd

说明

转向角速度。

定义

```
typedef struct {
    sint16 value;
    boolean valid;
} VCS SteerAngleSpd;
```

成员

成员名称	描述
value	转向角速度(向左为正, 向右为负), 单位: deg/s, 精度: 0.0625
valid	转向角速度标志位:

成员名称	描述
	False: 无效 True: 有效

5.15.9 VCS_SteerTq

说明

转向力矩。

定义

```
typedef struct {
    sint16 value;
    boolean valid;
} VCS_SteerTq;
```

成员

成员名称	描述
value	转向扭矩(向左为正, 向右为负), 单位: Nm, 精度: 0.003906
valid	转向扭矩标志位: False: 无效 True: 有效

5.15.10 VCS_WhlAngle

说明

轮角度。

定义

```
typedef struct {
    sint16 value;
    boolean valid;
} VCS_WhlAngle;
```

成员

成员名称	描述
------	----

成员名称	描述
value	轮角度(向左为正, 向右为负), 单位: deg, 精度: 0.003906
valid	轮角度标志位: False: 无效 True: 有效

5.15.11 VCS_YawRate

说明

横摆角速度。

定义

```
typedef struct {
    sint32 value;
    boolean valid;
} VCS YawRate;
```

成员

成员名称	描述
value	横摆角速度(向左为正, 向右为负), 单位: deg/s, 精度: 0.003906
valid	横摆角速度标志位: False: 无效 True: 有效

5.15.12 VCS_Acceleration

说明

加速度。

定义

```
typedef struct {
    sint16 value;
    boolean valid;
} VCS_Acceleration;
```

成员

成员名称	描述
value	加速度, 单位: m/s ² , 精度: 0.0625
valid	加速度标志位: False: 无效 True: 有效

5.15.13 VCS_MCPrs

说明

主缸压力。

定义

```
typedef struct {
    sint16 value;
    boolean valid;
} VCS_MCPrs;
```

成员

成员名称	描述
value	主缸压力, 单位: bar, 精度: 0.0625
valid	主缸压力标志位: False: 无效 True: 有效

5.15.14 VCS_BrakePosn

说明

制动踏板行程。

定义

```
typedef struct {
    sint16 stroke;
    boolean strokeValid;
    uint16 position;
    boolean positionValid;
```

```

    sint16 strokeRate;
    boolean strokeRateValid;
} VCS_BrakePosn;

```

成员

成员名称	描述
stroke	制动踏板行程，单位：mm，精度：0.0625
strokeValid	制动踏板行程标志位： False: 无效 True: 有效
position	制动踏板开度，单位：%，精度：0.0625
positionValid	制动踏板开度标志位： False: 无效 True: 有效
strokeRate	制动踏板开度变化率，单位：%/s，精度：0.25
strokeRateValid	制动踏板开度变化率标志位： False: 无效 True: 有效

5.15.15 VCS_WhlSpd

说明

轮速。

定义

```

typedef struct {
    sint16 FL;
    boolean FLValid;
    uint8 directionFL;
    boolean directionFLValid;
    sint16 FR;
    boolean FRValid;
    uint8 directionFR;
    boolean directionFRValid;
    sint16 RL;
    boolean RLValid;
    uint8 directionRL;
    boolean directionRLValid;
    sint16 RR;
    boolean RRValid;
}

```



```

uint8 directionRR;
boolean directionRRValid;
} VCS_WhlSpd;

```

成员

成员名称	描述
FL	左前轮速, 单位: km/h, 精度: 0.0078125
FLValid	左前轮速标志位: False: 无效 True: 有效
directionFL	左前轮速方向: 0x0: Forward (前进) 0x1: Backward (后退) 0x2: Standstill (静止) 0x3: Unknown (未知)
directionFLValid	左前轮速方向标志位: False: 无效 True: 有效
FR	右前轮速, 单位: km/h, 精度: 0.0078125
FRValid	右前轮速标志位: False: 无效 True: 有效
directionFR	右前轮速方向: 0x0: Forward (前进) 0x1: Backward (后退) 0x2: Standstill (静止) 0x3: Unknown (未知)
directionFRValid	右前轮速方向标志位: False: 无效 True: 有效
RL	左后轮速, 单位: km/h, 精度: 0.0078125
RLValid	左后轮速标志位: False: 无效 True: 有效
directionRL	左后轮速方向:

成员名称	描述
	0x0: Forward (前进) 0x1: Backward (后退) 0x2: Standstill (静止) 0x3: Unknown (未知)
directionRLValid	左后轮速方向标志位: False: 无效 True: 有效
RR	右后轮速, 单位: km/h, 精度: 0.0078125
RRValid	右后轮速标志位: False: 无效 True: 有效
directionRR	右后轮速方向: 0x0: Forward (前进) 0x1: Backward (后退) 0x2: Standstill (静止) 0x3: Unknown (未知)
directionRRValid	右后轮速方向标志位: False: 无效 True: 有效

5.15.16 VCS_EPBSwitch

说明

驻车开关状态。

定义

```
typedef struct {
    uint8 status;
} VCS_EPBSwitch;
```

成员

成员名称	描述
status	EPB 执行状态: 0x0: Neutral (无动作)

成员名称	描述
	0x1: Release (释放动作) 0x2: Apply (夹紧动作) 0x3: Reserved (预留)

5.15.17 VCS_EPB

说明

驻车执行状态。

定义

```
typedef struct {
    uint8 status;
    boolean Valid;
} VCS_EPB;
```

成员

成员名称	描述
status	驻车执行状态： 0x0: Undefined (未知状态) 0x1: Closed (卡钳处于夹紧状态) 0x2: Closing (卡钳正处于夹紧过程中) 0x3: Released (卡钳正处释放状态) 0x4: Releasing (卡钳正处于释放过程中) 0xFF: Invalid (无效)
Valid	驻车执行状态标志位： False: 无效 True: 有效

5.15.18 VCS_LgtBrkFTotal

说明

纵向总制动力。

定义

```
typedef struct {
    uint32 value;
    boolean valid;
} VCS LgtBrkFTotal;
```

成员

成员名称	描述
value	纵向总制动力，单位：N，精度：0.0625
valid	纵向总制动力标志位： False：无效 True：有效

5.15.19 VCS_WhlSpdPls

说明

轮速脉冲。

定义

```
typedef struct {
    uint16 FL;
    boolean FLValid;
    uint16 FR;
    boolean FRValid;
    uint16 RL;
    boolean RLValid;
    uint16 RR;
    boolean RRValid;
} VCS WhlSpdPls;
```

成员

成员名称	描述
FL	左前轮速脉冲数
FLValid	左前轮速脉冲数标志位： False：无效 True：有效
FR	右前轮速脉冲数

成员名称	描述
FRValid	右前轮速脉冲数标志位: False: 无效 True: 有效
RL	左后轮速脉冲数
RLValid	左后轮速脉冲数标志位: False: 无效 True: 有效
RR	右后轮速脉冲数
RRValid	右后轮速脉冲数标志位: False: 无效 True: 有效

5.15.20 VCS_TirePrs

说明

轮胎胎压。

定义

```
typedef struct {
    sint16 FL;
    boolean FLValid;
    sint16 FR;
    boolean FRValid;
    sint16 RL;
    boolean RLValid;
    sint16 RR;
    boolean RRValid;
} VCS TirePrs;
```

成员

成员名称	描述
FL	左前轮胎压, 单位: bar, 精度: 0.0625
FLValid	左前轮胎压标志位: False: 无效 True: 有效

成员名称	描述
FR	右前轮胎压, 单位: bar, 精度: 0.0625
FRValid	右前轮胎压标志位: False: 无效 True: 有效
RL	左后轮胎压, 单位: bar, 精度: 0.0625
RLValid	左后轮胎压标志位: False: 无效 True: 有效
RR	右后轮胎压, 单位: bar, 精度: 0.0625
RRValid	右后轮胎压标志位: False: 无效 True: 有效

5.15.21 VCS_TqCtrl

说明

制动控制扭矩请求。

定义

```
typedef struct {
    sint32 value;
    boolean valid;
} VCS_TqCtrl;
```

成员

成员名称	描述
value	制动总扭矩请求, 单位: Nm, 精度: 0.0625
valid	制动总扭矩请求标志位: False: 无效 True: 有效

5.15.22 VCS_Slope

说明

静态坡度。

定义

```
typedef struct {
    uint16 staticValue;
    boolean staticValid;
} VCS Slope;
```

成员

成员名称	描述
staticValue	静态坡度，单位：%，精度：0.0625
staticValid	静态坡度有效标志位： False: 无效 True: 有效

5.15.23 VCS_MotBaseStatus

说明

电机的工作基本状态。

定义

```
typedef struct {
    uint8 motMode;
    sint16 motTq;
    boolean motTqValid;
    sint16 motSpd;
    boolean motSpdValid;
    uint8 motDampingMode;
    boolean motDampingModeValid;
    sint16 motMaxTq;
    boolean motMaxTqValid;
    sint16 motMinTq;
    boolean motMinTqValid;
} VCS MotBaseStatus;
```

成员

成员名称	描述
motMode	电机工作模式： 0x0: Not initialized（未初始化） 0x1: Precharge（预充） 0x2: Standby（低压待机） 0x3: HVStandby（高压待机） 0x4: Torque Control（扭矩控制） 0x5: Speed Control（速度控制） 0x6: Shut Down（高压下电） 0x7: Power Down（低压下电） 0x8: Calibration（标定态） 0x9: Fault（故障态） 0xA: ASC（主动短路） 0xB: Discharge（主动放电） 0xC: UdcCtl（恒压控制） 0xD: alOffsetCal（偏置角自检） 0xE: ChrgBuck（降压模式） 0xF: PreChrgBoost（升压预充） 0x10: SupChrgBoost（升压快充） 0x11: ChrgReady（充电 Ready） 0x12: ChrgDischarge（充电放电） 0x13: ChrgFailure（充电失败） 0x14: AuxHeat（辅助加热）
motTq	驱动电机工作扭矩，单位：Nm，精度：0.0625
motTqValid	驱动电机工作扭矩有效标志位： False: 无效 True: 有效
motSpd	驱动电机工作转速，单位：rpm，精度：1
motSpdValid	驱动电机工作转速有效标志位： False: 无效 True: 有效
motDampingMode	主动阻尼工作模式： 0x0: Disable（禁用主动阻尼功能） 0x1: Enable（未禁用主动阻尼功能）

成员名称	描述
motDampingModeValid	主动阻尼工作模式有效标志位： False: 无效 True: 有效
motMaxTq	电机最大允许工作扭矩，单位：Nm，精度：0.0625
motMaxTqValid	电机最大允许工作扭矩有效标志位： False: 无效 True: 有效
motMinTq	电机最小允许工作扭矩，单位：Nm，精度：0.0625
motMinTqValid	电机最小允许工作扭矩有效标志位： False: 无效 True: 有效

5.15.24 VCS_MotStatus

说明

各电机的工作状态。

定义

```
typedef struct {
    Struct VCS MotBaseStatus FLMot;
    Struct VCS MotBaseStatus FRMot;
    Struct VCS MotBaseStatus RLMot;
    Struct VCS MotBaseStatus RRMot;
} VCS MotStatus;
```

成员

成员名称	描述
FLMot	左前（或前轴）电机的工作状态
FRMot	右前电机的工作状态
RLMot	左后（或后轴）电机的工作状态
RRMot	右后电机的工作状态

5.15.25 VCS_Crash

说明

碰撞状态。

定义

```
typedef struct {  
    uint8 status;  
    boolean valid;  
} VCS_Crash;
```

成员

成员名称	描述
status	碰撞状态： 0x0: No_Collision（未发生碰撞） 0x1: Collision（发生碰撞） 0x2: Fault（发生信号错误） 0x3: Reserved（预留）
valid	碰撞状态有效标志位： False: 无效 True: 有效

6 EMS(能量管理)原子服务 API

6.1 EMS_ChargePort(充电接口服务)

提供高压电池包的充电连接服务，包含充电的连接状态检测，充电连接的上锁和解锁。

EMS_ChargePort(充电接口服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
EMS_ChargePortStatus	getStatus(EMS_ChargePortId portId)	返回充电桩与充电口的物理连接状态。
EMS_ChargePortStatus	notifyStatus()	当充电枪与充电口的物理连接状态发生变化时，该方法被调用。
EMS_ChargePlugStatus	notifyACStatus()	上报交流充电口插枪状态。
EMS_ChargePlugStatus	notifyDCStatus()	上报直流充电口插枪状态。
EMS_ChargePortId	getActive()	返回充电枪与充电口的物理接口类型。
EMS_ChargePortTemp	notifyACTemp()	上报交流充电口温度。
EMS_ChargePortTemp	notifyDCTemp()	上报直流充电口温度。
EMS_ChargeLockStatus	getLockStatus(EMS_ChargePortId portId)	返回充电枪与充电口的物理连接的上锁状态。
EMS_ChargeLockStatus	notifyLockStatus()	上报充电枪与充电口的物理连接的上锁状态。
EMS_ReturnCode	lock(EMS_ChargePortId portId)	闭锁充电枪与充电接口物理连接。
EMS_ReturnCode	unlock(EMS_ChargePortId portId)	解锁充电枪与充电接口物理连接。

6.1.1 getStatus

功能说明

返回充电桩与充电口的物理连接状态。

函数原型

EMS_ChargePortStatus getStatus(EMS_ChargePortId portId)

参数说明

参数名称	类型	描述
portId	EMS_ChargePortId	充电枪与充电口的物理接口标识

返回值说明

类型	描述
EMS_ChargePortStatus	充电枪与充电口的物理连接状态

6.1.2 notifyStatus

功能说明

当充电枪与充电口的物理连接状态发生变化时，该方法被调用。

函数原型

EMS_ChargePortStatus notifyStatus()

参数说明

无。

返回值说明

类型	描述
EMS_ChargePortStatus	充电枪与充电口的物理连接状态

6.1.3 notifyACStatus

功能说明

上报交流充电口插枪状态。

函数原型

EMS_ChargePlugStatus notifyACStatus()

参数说明

无。

返回值说明

类型	描述
EMS_ChargePlugStatus	充电枪与交流充电口的物理连接状态

6.1.4 notifyDCStatus

功能说明

上报直流充电口插枪状态。

函数原型

EMS_ChargePlugStatus notifyDCStatus()

参数说明

无。

返回值说明

类型	描述
EMS_ChargePlugStatus	充电枪与直流充电口的物理连接状态

6.1.5 getActive

功能说明

返回充电枪与充电口的物理接口类型。

函数原型

```
EMS_ChargePortId getActive()
```

参数说明

无。

返回值说明

类型	描述
EMS_ChargePortId	充电枪与充电口的物理连接接口类型

6.1.6 notifyACTemp

功能说明

上报交流充电口温度。

函数原型

```
EMS_ChargePortTemp notifyACTemp()
```

参数说明

无。

返回值说明

类型	描述
EMS_ChargePortTemp	交流充电口温度

6.1.7 notifyDCTemp

功能说明

上报直流充电口温度。

函数原型

```
EMS_ChargePortTemp notifyDCTemp()
```

参数说明

无。

返回值说明

类型	描述
EMS_ChargePortTemp	直流充电口温度

6.1.8 getLockStatus

功能说明

返回充电枪与充电口的物理连接的上锁状态。

函数原型

```
EMS_ChargeLockStatus getLockStatus(EMS_ChargePortId portId)
```

参数说明

参数名称	类型	描述
portId	EMS_ChargePortId	充电枪与充电口的物理接口标识

返回值说明

类型	描述
EMS_ChargeLockStatus	充电枪与充电口的物理连接的上锁状态

6.1.9 notifyLockStatus

功能说明

上报充电枪与充电口的物理连接的上锁状态。

函数原型

```
EMS_ChargeLockStatus notifyLockStatus()
```

参数说明

无。

返回值说明

类型	描述
EMS_ChargeLockStatus	充电枪与充电口的物理连接的上锁状态

6.1.10 lock

功能说明

闭锁充电枪与充电接口物理连接。

函数原型

```
EMS_ReturnCode lock(EMS_ChargePortId portId)
```

参数说明

参数名称	类型	描述
portId	EMS_ChargePortId	充电枪与充电口的物理接口标识

返回值说明

类型	描述
EMS_ReturnCode	闭锁成功返回 SUCCESS，失败根据原因返回具体的返回码

6.1.11 unlock

功能说明

解锁充电枪与充电接口物理连接。

函数原型

```
EMS_ReturnCode unlock(EMS_ChargePortId portId)
```

参数说明

参数名称	类型	描述
portId	EMS_ChargePortId	充电桩与充电口的物理接口标识

返回值说明

类型	描述
EMS_ReturnCode	解锁成功返回 SUCCESS，失败根据原因返回具体的返回码

6.2 EMS_Charging(充电控制服务)

提供高压电池包的充电基础服务，包含充电桩和电池包的充电能力信息以及充电的启停控制及充电的实时状态监控服务。

EMS_Charging(充电控制服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
EMS_PowerInfo	getChgrCap()	返回充电装置的充电能力。
EMS_PowerInfo	getBattLimit()	返回电池的充电能力。
EMS_PowerInfo	notifyBattInput()	上报电池包的实时输入。
EMS_ChargerOutput	notifyChgrOutput()	上报充电供电端口的实时输出。
uint16	getProgress()	返回充电进度。
uint16	getChargedCp()	返回已充电容量。
uint16	getRemainCp()	返回剩余充电容量。
EMS_ReturnCode	setMaxCurrent(uint16 maxCurrent)	设置允许的最大充电电流。
EMS_ReturnCode	start()	启动充电。
EMS_ReturnCode	stop()	停止充电。

类型	API 函数	功能说明
de		
EMS_ChargeStatus	getStatus()	返回充电实时状态。
EMS_ChargeStatus	notifyStatus()	当充电状态发生变化时，该方法被调用。

6.2.1 getChrgrCap

功能说明

返回充电装置的充电能力。

函数原型

EMS_PowerInfo getChrgrCap()

参数说明

无。

返回值说明

类型	描述
EMS_PowerInfo	充电装置允许的最大电流，最大电压，最大功率

6.2.2 getBattLimit

功能说明

返回电池的充电能力。

函数原型

EMS_PowerInfo getBattLimit()

参数说明

无。

返回值说明

类型	描述
----	----

类型	描述
EMS_PowerInfo	电池允许的最大充电电流，最大电压，最大功率

6.2.3 notifyBattInput

功能说明

上报电池包的实时输入。

函数原型

EMS_PowerInfo notifyBattInput()

参数说明

无。

返回值说明

类型	描述
EMS_PowerInfo	指示充入电池包的实时功率，电流，电压

6.2.4 notifyChrgrOutp

功能说明

上报充电供电端口的实时输出。

函数原型

EMS_ChargerOutput notifyChrgrOutp()

参数说明

无。

返回值说明

类型	描述
EMS_ChargerOutput	指示充电源端口输出的实时功率，电流，电压

6.2.5 getProgress

功能说明

返回充电进度。

函数原型

```
uint16 getProgress()
```

参数说明

无。

返回值说明

类型	描述
uint16	充电进度，单位：%，精度：1/100

6.2.6 getChargedCp

功能说明

返回已充电容量。

函数原型

```
uint16 getChargedCp()
```

参数说明

无。

返回值说明

类型	描述
uint16	当前电量与本次充电开始时的电量的差值，单位：%，精度：1/100

6.2.7 getRemainCp

功能说明

返回剩余充电容量。

函数原型

```
uint16 getRemainCp()
```

参数说明

无。

返回值说明

类型	描述
uint16	充电目标电量与当前电量的差值，单位：%，精度：1/100

6.2.8 setMaxCurrent

功能说明

设置允许的最大充电电流。

函数原型

```
EMS_ReturnCode setMaxCurrent(uint16 maxCurrent)
```

参数说明

参数名称	类型	描述
maxCurrent	uint16	允许的最大充电电流，单位：A，精度1/10

返回值说明

类型	描述
EMS_ReturnCode	设置充电允许的最大充电电流，成功返回 SUCCESS，失败根据原因返回具体的返回码

6.2.9 start

功能说明

启动充电。

函数原型

EMS_ReturnCode start()

参数说明

无。

返回值说明

类型	描述
EMS_ReturnCode	启动充电，成功返回 SUCCESS，失败根据原因返回具体的返回码

6.2.10 stop

功能说明

停止充电。

函数原型

EMS_ReturnCode stop()

参数说明

无。

返回值说明

类型	描述
EMS_ReturnCode	停止充电，成功返回 SUCCESS，失败根据原因返回具体的返回码

6.2.11 getStatus

功能说明

返回充电实时状态。

函数原型

EMS_ChargeStatus getStatus()

参数说明

无。

返回值说明

类型	描述
EMS_ChargeStatus	充电实时状态

6.2.12 notifyStatus

功能说明

当充电状态发生变化时，该方法被调用。

函数原型

EMS_ChargeStatus notifyStatus()

参数说明

无。

返回值说明

类型	描述
EMS_ChargeStatus	充电实时状态

6.3 EMS_ChargeIndicator(充电指示服务)

提供充电指示服务，如是否正在充电以及当前 SOC 等信息。

EMS_ChargeIndicator(充电指示服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
EMS_ReturnCode	setIndicator(EMS_ChargingIndicator indicator, uint16 SoC)	设置充电状态指示及 SOC 指示。

6.3.1 setIndicator

功能说明

设置充电状态指示及 SOC 指示。

函数原型

EMS_ReturnCode setIndicator(EMS_ChargingIndicator indicator, uint16 SoC)

参数说明

参数名称	类型	描述
indicator	EMS_ChargingIndicator	充电指示
SoC	uint16	电量指示，单位：%，精度：1/100

返回值说明

类型	描述
EMS_ReturnCode	设置充电指示，返回成功或失败

6.4 EMS_HVBatt(高压电池状态服务)

提供高压电池包状态的基础服务，包含电池包的当前状态信息，电池包的累计充放电信息，以及电池包中的电芯的温度，电压等基础信息。

EMS_HVBatt(高压电池状态服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
uint16	getSOC()	返回高压电池的当前电量。
uint32	getSOE()	返回高压电池的当前能量。
uint16	getSOH()	返回电池的健康状态。
uint16	notifySOC()	上报高压电池的 SOC。
uint32	notifySOE()	上报高压电池的 SOE。
EMS_HvBattPowerLimit	getSOP()	返回高压电池当前放电功率最大限值。
EMS_HvBattPowerLimit	notifySOP()	上报高压电池当前放电功率最大限值。

类型	API 函数	功能说明
EMS_HvBattPowerLimit	getRegenerate()	返回高压电池允许的回馈功率。
EMS_HvBattPowerLimit	notifyRegenerate()	上报高压电池允许的回馈功率。
EMS_HvBattAccuStat	getAccuStat()	返回高压电池的累计统计信息。
uint32	getAccuChrgCp()	返回高压电池的累计充电容量。
uint32	getAccuDischrgCp()	返回高压电池的累计放电容量。
uint32	getAccuRgnCp()	返回高压电池的累计回收容量。
uint32	getAccuBalCp()	返回高压电池的累计均衡容量。
uint32	getAccuChrgEgy()	返回高压电池的累计充电能量。
uint32	getAccuDischrgEgy()	返回高压电池的累计放电能量。
uint32	getAccuRgnEgy()	返回高压电池的累计回收能量。
uint32	getAccuBalEgy()	返回高压电池的累计均衡能量。

6.4.1 getSOC

功能说明

返回高压电池的当前电量。

函数原型

```
uint16 getSOC()
```

参数说明

无。

返回值说明

类型	描述
uint16	当前电池包的电量，单位：%，精度：1/100

6.4.2 getSOE

功能说明

返回高压电池的当前能量。

函数原型

```
uint32 getSOE()
```

参数说明

无。

返回值说明

类型	描述
uint32	当前电池包的能量，单位：Wh

6.4.3 getSOH

功能说明

返回电池的健康状态。

函数原型

```
uint16 getSOH()
```

参数说明

无。

返回值说明

类型	描述
uint16	电池的健康状态，单位：%，精度：1/100

6.4.4 notifySOC

功能说明

上报高压电池的 SOC。

函数原型

uint16 notifySOC()

参数说明

无。

返回值说明

类型	描述
uint16	当前电池包的电量，单位：%，精度：1/100

6.4.5 notifySOE

功能说明

上报高压电池的 SOE。

函数原型

uint32 notifySOE()

参数说明

无。

返回值说明

类型	描述
uint32	当前电池包的能量，单位：Wh

6.4.6 getSOP

功能说明

返回高压电池当前放电功率最大限值。

函数原型

EMS_HvBattPowerLimit getSOP()

参数说明

无。

返回值说明

类型	描述
EMS_HvBattPowerLimit	高压电池当前放电功率及电流最大限值

6.4.7 notifySOP

功能说明

上报高压电池当前放电功率最大限值。

函数原型

EMS_HvBattPowerLimit notifySOP()

参数说明

无。

返回值说明

类型	描述
EMS_HvBattPowerLimit	高压电池当前放电功率及电流最大限值

6.4.8 getRegenerate

功能说明

返回高压电池允许的回馈功率。

函数原型

EMS_HvBattPowerLimit getRegenerate()

参数说明

无。

返回值说明

类型	描述
EMS_HvBattPowerLimit	高压电池允许的回馈功率及电流

类型	描述
imit	

6.4.9 notifyRegenerate

功能说明

上报高压电池允许的回馈功率。

函数原型

EMS_HvBattPowerLimit notifyRegenerate()

参数说明

无。

返回值说明

类型	描述
EMS_HvBattPowerLimit	高压电池允许的回馈功率及电流

6.4.10 getAccuStat

功能说明

返回高压电池的累计统计信息。

函数原型

EMS_HvBattAccuStat getAccuStat()

参数说明

无。

返回值说明

类型	描述
EMS_HvBattAccuStat	电池包的累计统计信息

6.4.11 getAccuChrgCp

功能说明

返回高压电池的累计充电容量。

函数原型

```
uint32 getAccuChrgCp()
```

参数说明

无。

返回值说明

类型	描述
uint32	电池包的累计充电容量，单位：Ah，精度：1/1000

6.4.12 getAccuDischrgCp

功能说明

返回高压电池的累计放电容量。

函数原型

```
uint32 getAccuDischrgCp()
```

参数说明

无。

返回值说明

类型	描述
uint32	电池包的累计放电容量，单位：Ah，精度：1/1000

6.4.13 getAccuRgnCp

功能说明

返回高压电池的累计回收容量。

函数原型

uint32 getAccuRgnCp()

参数说明

无。

返回值说明

类型	描述
uint32	电池包的累计回收容量，单位：Ah，精度：1/1000

6.4.14 getAccuBalCp

功能说明

返回高压电池的累计均衡容量。

函数原型

uint32 getAccuBalCp()

参数说明

无。

返回值说明

类型	描述
uint32	电池包的累计均衡容量，单位：Ah，精度：1/1000

6.4.15 getAccuChrgEgy

功能说明

返回高压电池的累计充电能量。

函数原型

uint32 getAccuChrgEgy()

参数说明

无。

返回值说明

类型	描述
uint32	电池包的累计充电能量，单位：kWh

6.4.16 getAccuDischrgEgy

功能说明

返回高压电池的累计放电能量。

函数原型

```
uint32 getAccuDischrgEgy()
```

参数说明

无。

返回值说明

类型	描述
uint32	电池包的累计放电能量，单位：kWh

6.4.17 getAccuRgnEgy

功能说明

返回高压电池的累计回收能量。

函数原型

```
uint32 getAccuRgnEgy()
```

参数说明

无。

返回值说明

类型	描述
uint32	电池包的累计回收能量，单位：kWh

6.4.18 getAccuBalEgy

功能说明

返回高压电池的累计均衡能量。

函数原型

```
uint32 getAccuBalEgy()
```

参数说明

无。

返回值说明

类型	描述
uint32	电池包的累计均衡能量，单位：kWh

6.5 EMS_LVBatt(低压电池状态服务)

提供低压蓄电池状态的基础服务，如果有多个低压电池，可以通过将本服务实例化多个支持。

EMS_LVBatt(低压电池状态服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
uint16	getSOC()	返回低压蓄电池的当前电量。
uint32	getSOE()	返回低压蓄电池的当前能量。
uint16	getSOH()	返回低压蓄电池的健康状态。
uint16	notifySOC()	上报低压蓄电池的当前电量。
uint32	notifySOE()	上报低压蓄电池的当前能量。
uint16	getVolt()	返回低压蓄电池的电压。
uint16	notifyVolt()	上报低压蓄电池的电压。
EMS_LvBattAccuStat	getAccuStat()	返回低压蓄电池的累计统计信息。
uint32	getAccuChrgCp()	返回低压蓄电池的累计充电容量。
uint32	getAccuDischrgCp()	返回低压蓄电池的累计放电容量。

6.5.1 getSOC

功能说明

返回低压蓄电池的当前电量。

函数原型

```
uint16 getSOC()
```

参数说明

无。

返回值说明

类型	描述
uint16	低压蓄电池的当前电量，单位：%，精度：1/100

6.5.2 getSOE

功能说明

返回低压蓄电池的当前能量。

函数原型

```
uint32 getSOE()
```

参数说明

无。

返回值说明

类型	描述
uint32	低压蓄电池的当前能量，单位：Wh

6.5.3 getSOH

功能说明

返回低压蓄电池的健康状态。

函数原型

```
uint16 getSOH()
```

参数说明

无。

返回值说明

类型	描述
uint16	低压蓄电池的健康状态，单位：%，精度：1/100

6.5.4 notifySOC

功能说明

上报低压蓄电池的当前电量。

函数原型

```
uint16 notifySOC()
```

参数说明

无。

返回值说明

类型	描述
uint16	低压蓄电池的当前电量，单位：%，精度：1/100

6.5.5 notifySOE

功能说明

上报低压蓄电池的当前能量。

函数原型

uint32 notifySOE()

参数说明

无。

返回值说明

类型	描述
uint32	低压蓄电池的当前能量，单位：Wh

6.5.6 getVolt

功能说明

返回低压蓄电池的电压。

函数原型

uint16 getVolt()

参数说明

无。

返回值说明

类型	描述
uint16	低压蓄电池的当前电压，单位：V，精度：1/1000

6.5.7 notifyVolt

功能说明

上报低压蓄电池的电压。

函数原型

uint16 notifyVolt()

参数说明

无。

返回值说明

类型	描述
uint16	低压蓄电池的当前电压，单位：V，精度：1/1000

6.5.8 getAccuStat

功能说明

返回低压蓄电池的累计统计信息。

函数原型

EMS_LvBattAccuStat getAccuStat()

参数说明

无。

返回值说明

类型	描述
EMS_LvBattAccuStat	低压蓄电池的累计统计信息

6.5.9 getAccuChrgCp

功能说明

返回低压蓄电池的累计充电容量。

函数原型

uint32 getAccuChrgCp()

参数说明

无。

返回值说明

类型	描述
uint32	电池包的累计充电容量，单位：Ah，精度：1/1000

6.5.10 getAccuDischrgCp

功能说明

返回低压蓄电池的累计放电容量。

函数原型

```
uint32 getAccuDischrgCp()
```

参数说明

无。

返回值说明

类型	描述
uint32	电池包的累计放电容量，单位：Ah，精度：1/1000

6.6 EMS_HvBattBalance(电池均衡服务)

提供高压电池包的电池均衡服务，包含单体电池均衡状态的管理，均衡启停控制以及均衡过程的基础信息服务。

EMS_HvBattBalance(电池均衡服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
EMS_ReturnCode	start()	启动均衡。
EMS_ReturnCode	stop()	停止均衡。
EMS_BalanceStatus	getStatus()	返回当前状态。
uint16	getCp()	返回本次均衡过程已均衡的容量。
uint32	getRemainTime()	返回本次均衡过程预估均衡剩余时间。
uint16	getRemainCp()	返回本次均衡过程预估剩余均衡的容量。

6.6.1 start

功能说明

启动均衡。

函数原型

EMS_ReturnCode start()

参数说明

无。

返回值说明

类型	描述
EMS_ReturnCode	对电池包的单体电芯设置均衡标识，完成返回成功，否则返回失败

6.6.2 stop

功能说明

停止均衡。

函数原型

EMS_ReturnCode stop()

参数说明

无。

返回值说明

类型	描述
EMS_ReturnCode	如果电池均衡被停止，返回成功，否则返回失败

6.6.3 getStatus

功能说明

返回当前状态。

函数原型

EMS_BalanceStatus getStatus()

参数说明

无。

返回值说明

类型	描述
EMS_BalanceStatus	电池包是否处于均衡的状态

6.6.4 getCp

功能说明

返回本次均衡过程已均衡的容量。

函数原型

uint16 getCp()

参数说明

无。

返回值说明

类型	描述
uint16	电池包本次均衡过程的已均衡的容量，单位：Ah，精度：1/1000

6.6.5 getRemainTime

功能说明

返回本次均衡过程预估均衡剩余时间。

函数原型

uint32 getRemainTime()

参数说明

无。

返回值说明

类型	描述
uint32	本次均衡过程预估均衡剩余时间，单位：秒

6.6.6 getRemainCp

功能说明

返回本次均衡过程预估剩余均衡的容量。

函数原型

uint16 getRemainCp()

参数说明

无。

返回值说明

类型	描述
uint16	电池包本次均衡过程的剩余均衡的容量，单位：Ah，精度：1/1000

6.7 EMS_PowerSupplier(电源管理服务)

提供电源管理服务，包括高压电池上下电等功能。

EMS_PowerSupplier(电源管理服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
EMS_HvReady	notifyHVReady()	上报是否可进行高压操作。
无	startHV()	高压电池包高压上电。
无	stopHV()	高压电池包高压下电。
EMS_HvResult	notifyHVStatus()	上报高压电池包的高压状态。

类型	API 函数	功能说明
EMS_PowerInfo	notifyPower()	上报高压电池的实时功率。
EMS_PowerSupplyType	getStatus()	返回当前状态。
EMS_PowerStat	getAccuStat()	返回高压电源的统计信息。
uint32	getDischrgCp()	返回本次上电过程的已放电容量。
uint32	getRegenerateCp()	返回本次上电过程的已回馈容量。

6.7.1 notifyHVReady

功能说明

上报是否可进行高压操作。

函数原型

EMS_HvReady notifyHVReady()

参数说明

无。

返回值说明

类型	描述
EMS_HvReady	是否可进行高压操作

6.7.2 startHV

功能说明

高压电池包高压上电。

函数原型

startHV()

参数说明

无。

返回值说明

无。

6.7.3 stopHV

功能说明

高压电池包高压下电。

函数原型

```
stopHV()
```

参数说明

无。

返回值说明

无。

6.7.4 notifyHVStatus

功能说明

上报高压电池包的高压状态。

函数原型

```
EMS_HvResult notifyHVStatus()
```

参数说明

无。

返回值说明

类型	描述
EMS_HvResult	高压电池包的高压状态

6.7.5 notifyPower

功能说明

上报高压电池的实时功率。

函数原型

EMS_PowerInfo notifyPower()

参数说明

无。

返回值说明

类型	描述
EMS_PowerInfo	高压电池包的放电功率，电压，电流

6.7.6 getStatus

功能说明

返回当前状态。

函数原型

EMS_PowerSupplyType getStatus()

参数说明

无。

返回值说明

类型	描述
EMS_PowerSupplyType	当前的电源放电状态

6.7.7 getAccuStat

功能说明

返回高压电源的统计信息。

函数原型

EMS_PowerStat getAccuStat()

参数说明

无。

返回值说明

类型	描述
EMS_PowerStat	高压电源的统计信息

6.7.8 getDischrgCp

功能说明

返回本次上电过程的已放电容量。

函数原型

```
uint32 getDischrgCp()
```

参数说明

无。

返回值说明

类型	描述
uint32	放电容量，单位：Ah，精度：1/1000

6.7.9 getRegenerateCp

功能说明

返回本次上电过程的已回馈容量。

函数原型

```
uint32 getRegenerateCp()
```

参数说明

无。

返回值说明

类型	描述
uint32	回馈容量, 单位: Ah, 精度: 1/1000

6.8 EMS_Circuit(电路配电服务)

提供高压供电电路系统的配电服务, 通过实例化支持 DCDC, TMS, 驱动, V2L 等高压电路子系统的配电功能。

EMS_Circuit(电路配电服务)服务包括的 API 如下表所示:

类型	API 函数	功能说明
EMS_HvReady	notifyReady()	上报是否可进行高压操作。
无	turnOn()	开启电路系统的配电。
无	turnOff()	关闭电路系统的配电。
EMS_HvResult	notifyStatus()	上报电路系统的配电状态。
EMS_PowerInfo	notifyPower()	上报电路系统的实际功率。
无	setVolt(uint16 volt)	设置电路系统的供电电压。
uint16	getVolt()	返回电路系统的供电电压。
无	setPowerLimit(uint32 power)	设置电路系统的可用功率。

6.8.1 notifyReady

功能说明

上报是否可进行高压操作。

函数原型

```
EMS_HvReady notifyReady()
```

参数说明

无。

返回值说明

类型	描述
EMS_HvReady	是否可进行高压操作

6.8.2 turnOn

功能说明

开启电路系统的配电。

函数原型

```
turnOn()
```

参数说明

无。

返回值说明

无。

6.8.3 turnOff

功能说明

关闭电路系统的配电。

函数原型

```
turnOff()
```

参数说明

无。

返回值说明

无。

6.8.4 notifyStatus

功能说明

上报电路系统的配电状态。

函数原型

EMS_HvResult notifyStatus()

参数说明

无。

返回值说明

类型	描述
EMS_HvResult	电路系统的配电状态

6.8.5 notifyPower

功能说明

上报电路系统的实际功率。

函数原型

EMS_PowerInfo notifyPower()

参数说明

无。

返回值说明

类型	描述
EMS_PowerInfo	功率，电压及电流

6.8.6 setVolt

功能说明

设置电路系统的供电电压。

函数原型

setVolt(uint16 volt)

参数说明

参数名称	类型	描述
volt	uint16	电路系统的目标供电电压，单位：V，精度：1/10

返回值说明

无。

6.8.7 getVolt

功能说明

返回电路系统的供电电压。

函数原型

```
uint16 getVolt()
```

参数说明

无。

返回值说明

类型	描述
uint16	电路系统的供电电压，单位：V，精度：1/10

6.8.8 setPowerLimit

功能说明

设置电路系统的可用功率。

函数原型

```
setPowerLimit(uint32 power)
```

参数说明

参数名称	类型	描述
power	uint32	电路系统的可用功率

返回值说明

无。

6.9 数据类型定义

6.9.1 EMS_PortConnectStatus

说明

充电口的物理连接状态。

定义

```
typedef enum {  
    INIT=0,  
    PLUGIN=1,  
    UNPLUG=2,  
    SEMILINK=3,  
    INVALID=255  
} EMS_PortConnectStatus;
```

成员

成员名称	描述
INIT	初始值
PLUGIN	检测到充电枪已经插入充电口并正确连接
UNPLUG	未检测到充电枪插入充电口
SEMILINK	检测到充电枪已经插入充电口但并未正确连接
INVALID	无效值

6.9.2 EMS_ChargePortId

说明

充电口的物理接口标识。

定义

```
typedef enum {  
    DCPOR=0,
```

```

    ACPORT=1
} EMS_ChargePortId;

```

成员

成员名称	描述
DCPORT	直流充电口
ACPORT	交流充电口

6.9.3 EMS_PlugType

说明

充电口插枪类型。

定义

```

typedef enum {
    CHARGING=0,
    DISCHARGING=1,
    UNKNOWN=2,
    INVALID=255
} EMS_PlugType;

```

成员

成员名称	描述
CHARGING	充电枪
DISCHARGING	放电枪
UNKNOWN	充电枪或放电枪（部分情况下无法识别）
INVALID	无效值

6.9.4 EMS_LockStatus

说明

电子锁的状态。

定义

```
typedef enum {
    LOCK=0,
    UNLOCK=1,
    INVALID=255
} EMS LockStatus;
```

成员

成员名称	描述
LOCK	闭锁
UNLOCK	解锁
INVALID	无效值

6.9.5 EMS_PlugCapability

说明

充电枪的线缆能力。

定义

```
typedef enum {
    PLUG_10A=0,
    PLUG_16A=1,
    PLUG_32A=2,
    PLUG_63A=3,
    INVALID=255
} EMS PlugCapability;
```

成员

成员名称	描述
PLUG_10A	10A
PLUG_16A	16A
PLUG_32A	32A
PLUG_63A	63A
INVALID	无效值

6.9.6 EMS_ChargePortStatus

说明

充电口的物理接口状态发生变化的事件信息。

定义

```
typedef struct {  
    EMS ChargePortId portId;  
    EMS PlugType plugType;  
    EMS PlugCapability plugCapability;  
    EMS PortConnectStatus status;  
} EMS ChargePortStatus;
```

成员

成员名称	描述
portId	充电口的物理接口标识
plugType	充电口的插枪类型
plugCapability	充电枪的线缆能力
status	充电口的物理连接状态

6.9.7 EMS_ChargePlugStatus

说明

充电枪状态信息。

定义

```
typedef struct {  
    EMS_PlugType plugType;  
    EMS_PlugCapability plugCapability;  
    EMS_PortConnectStatus status;  
} EMS_ChargePlugStatus;
```

成员

成员名称	描述
plugType	充电口的插枪类型
plugCapability	充电枪的线缆能力

成员名称	描述
status	充电口的物理连接状态

6.9.8 EMS_ReturnCode

说明

调用 EMS 服务 API 的统一返回码。

定义

```
typedef enum {
    SUCCESS=0,
    FAIL=1,
    FAIL_UNAVAILABLE=2,
    FAIL_INVALID_PARAM=3
} EMS_ReturnCode;
```

成员

成员名称	描述
SUCCESS	成功
FAIL	失败，未原因
FAIL_UNAVAILABLE	不可用
FAIL_INVALID_PARAM	无效的参数值

6.9.9 EMS_PowerInfo

说明

基础电学物理量信息单元。

定义

```
typedef struct {
    uint32 power;
    uint16 voltage;
    sint16 current;
} EMS_PowerInfo;
```

成员

成员名称	描述
power	电功率，单位：W，无效值 0xFFFFFFFF
voltage	电压，单位：V，精度 1/10，无效值 0xFFFF
current	电流，单位：A，精度 1/10，无效值 0x7FFF

6.9.10 EMS_SupplyMode

说明

电源供电方式。

定义

```
typedef enum {
    DC=0,
    AC_SINGLEPHASE=1,
    AC_THREEPHASE=2
} EMS_SupplyMode;
```

成员

成员名称	描述
DC	直流供电
AC_SINGLEPHASE	单相交流供电
AC_THREEPHASE	三相交流供电

6.9.11 EMS_ChargerOutput

说明

充电端输出。

定义

```
typedef struct {
    EMS_SupplyMode supplyMode;
    EMS_PowerInfo powerInfo;
} EMS_ChargerOutput;
```

成员

成员名称	描述
supplyMode	充电源端的供电方式
powerInfo	充电源端的功率输出

6.9.12 EMS_ChargeStatus

说明

充电状态。

定义

```
typedef enum {
    STANDBY=0,
    CHARGING=1,
    HEATING=2
} EMS_ChargeStatus;
```

成员

成员名称	描述
STANDBY	未充电
CHARGING	正在充电
HEATING	正在电池包加热

6.9.13 EMS_BalanceStatus

说明

电池的均衡状态。

定义

```
typedef enum {
    BALANCED=0,
    UNBALANCED=1
} EMS_BalanceStatus;
```


成员

成员名称	描述
BALANCED	已均衡
UNBALANCED	未均衡

6.9.14 EMS_PowerSupplyType

说明

电源放电状态。

定义

```
typedef enum {
    HV_DISCHARGING=0,
    LV_DISCHARGING=1
} EMS_PowerSupplyType;
```

成员

成员名称	描述
HV_DISCHARGING	高压放电
LV_DISCHARGING	低压放电

6.9.15 EMS_ChargeLockStatus

说明

充电口物理连接的上锁状态。

定义

```
typedef struct {
    EMS ChargePortId portId;
    EMS LockStatus status;
} EMS_ChargeLockStatus;
```

成员

成员名称	描述
------	----

成员名称	描述
portId	充电口的物理接口标识
status	充电口的物理连接状态

6.9.16 EMS_ChargingIndicator

说明

充电指示。

定义

```
typedef enum {
    STOP=0,
    NO_CONNECT=1,
    SEMI_CONNECT=2,
    FULL_CONNECT=3,
    PREPARE=4,
    SLOW_CHARGING=5,
    FAST_CHARGING=6,
    CHARGE_END=7,
    ENDING=8,
    FAILURE=9,
    INVALID=255
} EMS_ChargingIndicator;
```

成员

成员名称	描述
STOP	停止指示
NO_CONNECT	充电口盖已开启，未插入充电枪
SEMI_CONNECT	插入充电枪半连接
FULL_CONNECT	插入充电枪已连接
PREPARE	即将充电（正在建立联系或电池预加热等）
SLOW_CHARGING	慢速充电中
FAST_CHARGING	快速充电中
CHARGE_END	充电完成，停止充电
ENDING	充电完成后处理（电池保温等）
FAILURE	充电故障，停止充电

成员名称	描述
INVALID	无效

6.9.17 EMS_HvReady

说明

高压操作是否允许。

定义

```
typedef struct {
    uint8 startHvPermit;
    uint8 stopHvPermit;
} EMS_HvReady;
```

成员

成员名称	描述
startHvPermit	0:不能上高压 1:能上高压 255:无效值
stopHvPermit	0:不能下高压 1:能下高压 255:无效值

6.9.18 EMS_HvBattPowerLimit

说明

高压电池回馈/放电功率限值。

定义

```
typedef struct {
    uin32 instantPower;
    uin32 normalPower;
    uin32 longPower;
    sint16 instantCurrent;
    sint16 normalCurrent;
    sint16 longCurrent;
} EMS HvBattPowerLimit;
```

成员

成员名称	描述
instantPower	瞬时功率，单位：W，无效值 0xFFFFFFFF
normalPower	普通功率，单位：W，无效值 0xFFFFFFFF
longPower	长时功率，单位：W，无效值 0xFFFFFFFF
instantCurrent	瞬时电流，单位：A，精度：1/10，无效值 0x7FFF
normalCurrent	普通电流，单位：A，精度：1/10，无效值 0x7FFF
longCurrent	长时电流，单位：A，精度：1/10，无效值 0x7FFF

6.9.19 EMS_HvBattAccuStat

说明

高压电池累计统计。

定义

```
typedef struct {
    uin32 accuChrgCp;
    uin32 accuDischrgCp;
    uin32 accuRgnCp;
    uin32 accuBalCp;
    uin32 accuChrgEgy;
    uin32 accuDischrgEgy;
    uin32 accuRgnEgy;
    uin32 accuBalEgy;
} EMS_HvBattAccuStat;
```

成员

成员名称	描述
accuChrgCp	累计充电容量，单位：Ah，精度：1/1000
accuDischrgCp	累计放电容量，单位：Ah，精度：1/1000
accuRgnCp	累计回馈容量，单位：Ah，精度：1/1000
accuBalCp	累计均衡容量，单位：Ah，精度：1/1000
accuChrgEgy	累计充电能量，单位：kWh
accuDischrgEgy	累计放电能量，单位：kWh
accuRgnEgy	累计回馈能量，单位：kWh

成员名称	描述
accuBalEgy	累计均衡能量，单位：kWh

6.9.20 EMS_LvBattAccuStat

说明

低压蓄电池累计统计。

定义

```
typedef struct {
    uin32 accuChrgCp;
    uin32 accuDischrgCp;
} EMS_LvBattAccuStat;
```

成员

成员名称	描述
accuChrgCp	累计充电容量，单位：Ah，精度：1/1000
accuDischrgCp	累计放电容量，单位：Ah，精度：1/1000

6.9.21 EMS_PowerStat

说明

本次高压电源上电过程的统计容量。

定义

```
typedef struct {
    uin32 dischrgCp;
    uin32 regenerateCp;
    uin32 chrgCp;
    uin32 dischrgEgy;
    uin32 regenerateEgy;
    uin32 chrgEgy;
} EMS_PowerStat;
```

成员

成员名称	描述
------	----

成员名称	描述
dischrgCp	已放电容量, 单位: Ah, 精度: 1/1000
regenerateCp	已回馈容量, 单位: Ah, 精度: 1/1000
chrgCp	已充电容量, 单位: Ah, 精度: 1/1000
dischrgEgy	已放电能量, 单位: kWh
regenerateEgy	已回馈能量, 单位: kWh
chrgEgy	已充电能量, 单位: kWh

6.9.22 EMS_HvResult

说明

高压系统上下电结果及异常原因。

定义

```
typedef struct {
    EMS HvStatus status;
    EMS HvAbnormalCode code;
} EMS HvResult;
```

成员

成员名称	描述
status	高压状态
code	高压异常原因

6.9.23 EMS_HvStatus

说明

高压状态。

定义

```
typedef enum {
    INIT=0,
    OFF=1,
    ON=2,
    SAFE_OFF=3,
    PRE_ON=4,
```

```

PRE_OFF=5,
INVALID=255
} EMS_HvStatus;

```

成员

成员名称	描述
INIT	初始
OFF	下电
ON	上电
SAFE_OFF	下电安全
PRE_ON	预充
PRE_OFF	主动放电
INVALID	无效

6.9.24 EMS_HvAbnormalCode

说明

高压异常原因。

定义

```

typedef enum {
    NORMAL=0,
    ON_TIMEOUT=1,
    ON_FAULT=2,
    OFF_TIMEOUT=3,
    OFF_FAULT=4,
    ON_PRE_TIMEOUT=5,
    OFF_PRE_TIMEOUT=6,
    INVALID=65535
} EMS_HvAbnormalCode;

```

成员

成员名称	描述
NORMAL	无异常
ON_TIMEOUT	上高压超时
ON_FAULT	上高压过程器件故障

成员名称	描述
OFF_TIMEOUT	下高压超时
OFF_FAULT	下高压过程器件故障
ON_PRE_TIMEOUT	预充超时
OFF_PRE_TIMEOUT	主动放电超时
INVALID	无效

6.9.25 EMS_ChargePortTemp

说明

充电口温度。

定义

```
typedef struct {
    sint16 temp;
} EMS_ChargePortTemp;
```

成员

成员名称	描述
temp	温度，单位：°C，精度 1/10，无效值 0x7FFF

7 ADAS(智驾域)原子服务 API

7.1 ADAS_Radar(毫米波雷达感知服务)

毫米波雷达识别，上报目标物的信息，包括目标物类型、目标物速度、目标物加速度等信息。

ADAS_Radar(毫米波雷达感知服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
ADAS_RadarObjects	notifyObjects()	上报毫米波雷达识别的目标信息。

7.1.1 notifyObjects

功能说明

上报毫米波雷达识别的目标信息。

函数原型

ADAS_RadarObjects notifyObjects()

参数说明

无。

返回值说明

类型	描述
ADAS_RadarObjects	毫米波雷达识别的目标物信息

7.2 ADAS_USS(超声波雷达感知服务)

超声波雷达识别，上报目标物类型、目标物速度、目标物加速度、目标物距离等信息。

ADAS_USS(超声波雷达感知服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
ADAS_UssObjects	notifyObjects()	上报超声波雷达识别的目标信息。
ADAS_UssObjectsDistance	notifyObjectsDistance()	上报超声波雷达识别的距离信息。

7.2.1 notifyObjects

功能说明

上报超声波雷达识别的目标信息。

函数原型

ADAS_UssObjects notifyObjects()

参数说明

无。

返回值说明

类型	描述
ADAS_UssObjects	超声波雷达识别的目标物信息

7.2.2 notifyObjectsDistance

功能说明

上报超声波雷达识别的距离信息。

函数原型

ADAS_UssObjectsDistance notifyObjectsDistance()

参数说明

无。

返回值说明

类型	描述
ADAS_UssObjectsDistance	超声波雷达识别的目标距离信息

7.3 ADAS_Lidar(激光雷达感知服务)

激光雷达识别，上报目标物的信息，包括目标物类型、目标物速度、目标物加速度等信息。

ADAS_Lidar(激光雷达感知服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
ADAS_LidarObjects	notifyObjects()	上报激光雷达识别的目标信息。

7.3.1 notifyObjects

功能说明

上报激光雷达识别的目标信息。

函数原型

ADAS_LidarObjects notifyObjects()

参数说明

无。

返回值说明

类型	描述
ADAS_LidarObjects	激光雷达识别的目标物信息

7.4 ADAS_Perception(视觉感知)

视觉感知服务，获取目标识别信息、车道信息、路面标记信息、交通灯信息、交通标志信息、停车位信息、可行驶空间等信息。

ADAS_Perception(视觉感知)服务包括的 API 如下表所示：

类型	API 函数	功能说明
ADAS_TrackObjects	getTrackObjects()	获取目标识别和跟踪信息。
ADAS_LaneLineInfo	getLaneline()	获取车道信息。
ADAS_RoadMarkingInfo	getRoadMarking()	获取路面标记信息。
ADAS_TrafficLightInfo	getTrafficLight()	获取交通灯信息。
ADAS_TrafficSignInfo	getTrafficSign()	获取交通标志信息。
ADAS_ParkingInfo	getParkingSpace()	获取停车位信息。
ADAS_FreeSpaceInfo	getFreeSpace()	获取可行驶空间。
ADAS_PerceptionFusionObjects	getFusionTrackObjects()	目标识别融合接口。

7.4.1 getTrackObjects

功能说明

获取目标识别和跟踪信息。

函数原型

ADAS_TrackObjects getTrackObjects()

参数说明

无。

返回值说明

类型	描述

类型	描述
ADAS_TrackObjects	跟踪目标信息

7.4.2 getLaneline

功能说明

获取车道信息。

函数原型

ADAS_LaneLineInfo getLaneline()

参数说明

无。

返回值说明

类型	描述
ADAS_LaneLineInfo	车道线信息

7.4.3 getRoadMarking

功能说明

获取路面标记信息。

函数原型

ADAS_RoadMarkingInfo getRoadMarking()

参数说明

无。

返回值说明

类型	描述
ADAS_RoadMarkingInfo	路面标记信息

7.4.4 getTrafficLight

功能说明

获取交通灯信息。

函数原型

```
ADAS_TrafficLightInfo getTrafficLight()
```

参数说明

无。

返回值说明

类型	描述
ADAS_TrafficLightInfo	交通灯信息

7.4.5 getTrafficSign

功能说明

获取交通标志信息。

函数原型

```
ADAS_TrafficSignInfo getTrafficSign()
```

参数说明

无。

返回值说明

类型	描述
ADAS_TrafficSignInfo	交通标志信息

7.4.6 getParkingSpace

功能说明

获取泊车位信息。

函数原型

```
ADAS_ParkingInfo getParkingSpace()
```

参数说明

无。

返回值说明

类型	描述
ADAS_ParkingInfo	停车位信息

7.4.7 getFreeSpace

功能说明

获取可行驶空间。

函数原型

```
ADAS_FreeSpaceInfo getFreeSpace()
```

参数说明

无。

返回值说明

类型	描述
ADAS_FreeSpaceInfo	可行驶空间

7.4.8 getFusionTrackObjects

功能说明

目标识别融合接口。

函数原型

```
ADAS_PerceptionFusionObjects getFusionTrackObjects()
```

参数说明

无。

返回值说明

类型	描述
ADAS_PerceptionFusionObjects	感知汇总数据

7.5 数据类型定义

7.5.1 ADAS_RadarObjects

说明

毫米波雷达识别的目标物信息。

定义

```
typedef struct {
    ADAS_DetectionObjects radarDetectionObjects;
} ADAS_RadarObjects;
```

成员

成员名称	描述
radarDetectionObjects	毫米波雷达识别的目标物信息

7.5.2 ADAS_UssObjects

说明

超声波雷达识别的目标物信息。

定义

```
typedef struct {
    ADAS_DetectionObjects ussDetectionObjects;
```



```
} ADAS_UssObjects;
```

成员

成员名称	描述
ussDetectionObjects	超声波雷达识别的目标物信息

7.5.3 ADAS_LidarObjects

说明

激光雷达识别的目标物信息。

定义

```
typedef struct {
    ADAS_DetectionObjects lidarDetectionObjects;
} ADAS_LidarObjects;
```

成员

成员名称	描述
lidarDetectionObjects	激光雷达识别的目标物信息

7.5.4 ADAS_DetectionObjects

说明

雷达探测的目标物信息。

定义

```
typedef struct {
    uint16 objNumber;
    ADAS_DetectionObjAry detectionObjs;
} ADAS_DetectionObjects;
```

成员

成员名称	描述
objNumber	目标物个数

成员名称	描述
detectionObjs	雷达目标物数组

7.5.5 ADAS_DetectionObjAry

说明

雷达目标物数组。

定义

```
typedef ADAS_DetectionObj ADAS_DetectionObjAry[objNumber];
```

成员

成员名称	描述
ADAS_DetectionObjAry	雷达目标物数组，数组长度为 objNumber，即目标物个数

7.5.6 ADAS_DetectionObj

说明

雷达识别的单个目标物信息。

定义

```
typedef struct {
    uint32 objectId;
    float32 objectExstProb;
    uint8 objectType;
    float32 classConfidence;
    uint8 objectUpdateFlag;
    uint8 objectMeasureFlag;
    uint8 objectMotionPattern;
    ADAS TimeStamp timeStamp;
    uint8 coordinate;
    ADAS Pose referencePoint;
    ADAS Velocity objectAbsVelocity;
    ADAS Velocity objectRelVelocity;
    ADAS Acceleration objectAbsAccel;
    ADAS Acceleration objectRelAccel;
} ADAS_DetectionObj;
```

成员

成员名称	描述
objectId	目标 ID
objectExstProb	目标存在概率，单位：%，取值范围：0~100
objectType	目标对象类型 0x0: Unknown（未知） 0x1: 4 wheeler（4 轮车） 0x2: 2 wheeler（2 轮车） 0x3: Pedestrian（行人）
classConfidence	类型置信度，单位：%，取值范围：0~100
objectUpdateFlag	目标更新标志，是否是新创建，若调用方不使用该参数，可以任意填写 0x0: True 0x1: False
objectMeasureFlag	目标测量标志（实际测量还是推断），若调用方不使用该参数，可以任意填写 0x0: 实际测量 0x1: 理论推断
objectMotionPattern	目标运动类型，若无目标，填写默认值 0x05。若调用方不使用该参数，可以任意填写 0x0: Unknown（未知） 0x1: Stationary（静止） 0x2: Stopped（已停止） 0x3: Moving（移动） 0x4: Crossing（交叉） 0x5: Reserved（保留）
timeStamp	目标当前识别的时间戳
coordinate	坐标系类型 0x0: Unknow（未知） 0x1: Vehicle（车辆坐标系） 0x2: WGS84（全球地理坐标系） 0x3: UTM（局部坐标系）
referencePoint	相对自车坐标系，目标包围框中心（长方体质心）位置，位姿信息主要提供目标位姿信息，包括位置、姿态和协方差

成员名称	描述
objectAbsVelocity	目标绝对速度信息，包含误差
objectRelVelocity	目标相对速度信息，包含误差
objectAbsAccel	目标绝对加速度信息，包含误差
objectRelAccel	目标相对加速度信息，包含误差

7.5.7 ADAS_UssObjectsDistance

说明

超声波雷达识别的目标距离信息。

定义

```
typedef struct {
    uint16 objNumber;
    ADAS_UssObjDistanceAry listOfObjDistance;
} ADAS_UssObjectsDistance;
```

成员

成员名称	描述
objNumber	目标物个数
listOfObjDistance	目标物距离数组信息

7.5.8 ADAS_UssObjDistanceAry

说明

目标物距离数组。

定义

```
typedef ADAS_UssObjDistance ADAS_UssObjDistanceAry[objNumber];
```

成员

成员名称	描述
ADAS_UssObjDistanceAry	雷达目标距离数组，数组长度为 objNumber，即目标物个数

成员名称	描述
	数

7.5.9 ADAS_UssObjDistance

说明

目标物距离信息。

定义

```
typedef struct {
    uint32 objectId;
    float32 objectExstProb;
    ADAS_TimeStamp timeStamp;
    uint16 objectDistance;
} ADAS_UssObjDistance;
```

成员

成员名称	描述
objectId	目标 ID
objectExstProb	目标存在概率，单位：%，取值范围：0~100
timeStamp	时间戳
objectDistance	目标距离，单位：cm

7.5.10 ADAS_TrackObjects

说明

跟踪目标数据集。

定义

```
typedef struct {
    uint16 trackObjectsNum;
    ADAS_TrackObjectsAry trackObjectsInfo;
} ADAS_TrackObjects;
```

成员

成员名称	描述
------	----

成员名称	描述
trackObjectsNum	跟踪目标物个数，单位：个
trackObjectsInfo	跟踪目标数组

7.5.11 ADAS_TrackObjectsAry

说明

跟踪目标数组。

定义

```
typedef ADAS TrackObject ADAS TrackObjectsAry[trackObjectsNum];
```

成员

成员名称	描述
ADAS_TrackObjectsAry	跟踪目标物数组，数组长度为 trackObjectsNum，即目标物个数

7.5.12 ADAS_TrackObject

说明

跟踪目标元数据。

定义

```
typedef struct {
    uint32 objectId;
    float32 objectExstProb;
    uint8 objectType;
    float32 classConfidence;
    uint32 objectColour;
    uint8 isMovable;
    ADAS TimeStamp objectAge;
    float32 trackQuality;
    uint8 coordinate;
    ADAS Pose referencePoint;
    ADAS Dimension objectDimension;
    ADAS Velocity objectVelocity;
    ADAS Acceleration objectAccel;
    uint8 laneAssignment;
    float32 laneOffsetLeft;
    float32 laneOffsetRight;
}
```

```
uint8 lightStatus;
} ADAS_TrackObject;
```

成员

成员名称	描述
objectId	目标 ID
objectExstProb	目标存在概率，单位：%，取值范围：0~100
objectType	目标类型 0x00: Unknow（未知） 0x01: Pedestrian（行人） 0x02: Cyclist（自行车） 0x03: Car（汽车） 0x04: Truck（卡车） 0x05: Tram（有轨电车） 0x06: Motorcycle（摩托车） 0x07: Bicycle（自行车） 0x08: Tricycle（三轮车） 0x09: TrafficLight（交通灯） 0x0A: TrafficSign（交通标志） 0x0B: RoadSign（道路标志） 0x0C: TrafficCone（交通锥） 0x0D: TrafficStick（交通杆） 0x0E: BillBoard（广告牌） 0x0F: ParkingSlot（停车场） 0x10: FireHydrant（消防栓） 0x11: UnknowStatic（未知静态）
classConfidence	类型置信度，单位：%，取值范围：0~100
objectColour	目标物颜色，RGB 格式，占用 3 个字节
isMovable	静态、动态目标区分 0x0: 静态目标 0x1: 潜在动态目标
objectAge	目标当前被跟踪时间
trackQuality	跟踪质量，单位：%，取值范围：0~100
coordinate	坐标系类型 0x0: Unknow（未知）

成员名称	描述
	0x1: Vehicle (车辆坐标系) 0x2: WGS84 (全球地理坐标系) 0x3: UTM (局部坐标系)
referencePoint	相对自车坐标系, 目标包围框中心 (长方体质心) 位置, 位姿信息主要提供目标位姿信息, 包括位置、姿态和协方差
objectDimension	目标几何尺寸, 提供物体轮廓的长宽高。
objectVelocity	目标速度 (相对自车坐标系);
objectAccel	目标加速度 (相对自车坐标系);
laneAssignment	动态目标所属车道 取值范围: -31~31 本车行驶车道为 0, 左侧车道为负, 右侧车道为正。
laneOffsetLeft	距左侧车道偏移量, 单位: m, 取值范围: 0~10
laneOffsetRight	距右侧车道偏移量, 单位: m, 取值范围: 0~10
lightStatus	转向灯、刹车灯状态、紧急指示灯等灯光状态 0x0: Na (无) 0x1: Turnleft (左转) 0x2: Turnright (右转) 0x3: DoubleFlash (双闪) 0x4: Brake (刹车)

7.5.13 ADAS_LaneLineInfo

说明

车道线信息 (包含路沿)。

定义

```
typedef struct {
    uint8 laneLinesNum;
    ADAS LaneLinesAry listOfLaneLine;
} ADAS LaneLineInfo;
```

成员

成员名称	描述
------	----

成员名称	描述
laneLinesNum	车道线个数
listOfLaneLine	车道线集合数组

7.5.14 ADAS_LaneLinesAry

说明

车道线数组。

定义

```
typedef ADAS LaneLine ADAS LaneLinesAry[laneLinesNum];
```

成员

成员名称	描述
ADAS_LaneLinesAry	车道线数组，数组长度为 laneLinesNum，即车道线个数

7.5.15 ADAS_LaneLine

说明

车道线。

定义

```
typedef struct {
    uint32 lineId;
    float32 lineExstProb;
    ADAS_TimeStamp lineAge;
    uint8 lineType;
    float32 classConfidence;
    uint8 lineColor;
    uint8 lineWidth;
    uint8 coordinate;
    ADAS Points3D linePoints;
    ADAS Coe curve;
} ADAS LaneLine;
```

成员

成员名称	描述
------	----

成员名称	描述
lineId	车道线 ID
lineExstProb	车道线存在概率，单位：%，取值范围：0~100
lineAge	车道线被跟踪的时间
lineType	车道线类型：双线、单线、实线、虚线、路沿 0x0: Unknow（未知） 0x1: SingleSolid（单实线） 0x2: SingleDashed（单虚线） 0x3: DoubleDashedSolid（双_虚线_实线） 0x4: DoubleSolidDashed（双_实线_虚线） 0x5: DoubleDashedDashed（双_虚线_虚线） 0x6: DoubleSolidSolid（双_实线_实线） 0x7: LeftRoadEdge（左侧路沿） 0x8: RightRoadEdge（右侧路沿）
classConfidence	车道线类型置信度，单位：%，取值范围：0~100
lineColor	车道线颜色 0x0: Unknow（未知） 0x1: White（白色） 0x2: Yellow（黄色） 0x3: Blue（蓝色）
lineWidth	车道线宽度
coordinate	坐标系类型 0x0: Unknow（未知） 0x1: Vehicle（车辆坐标系） 0x2: WGS84（全球地理坐标系） 0x3: UTM（局部坐标系）
linePoints	车道线点列表（车道线取车道宽度的中心线；路沿取靠近马路的边界线）
curve	车道线拟合参数

7.5.16 ADAS_RoadMarkingInfo

说明

路面标记信息。

定义

```
typedef struct {
    uint8 roadMarkingNum;
    ADAS RoadMarkingAry listOfRoadMarking;
} ADAS RoadMarkingInfo;
```

成员

成员名称	描述
roadMarkingNum	路面标记个数
listOfRoadMarking	路面标记集合，单位：个

7.5.17 ADAS_RoadMarkingAry

说明

路面标记数组。

定义

```
typedef ADAS RoadMarking ADAS RoadMarkingAry[roadMarkingNum];
```

成员

成员名称	描述
ADAS_RoadMarkingAry	路面标记数组，数组长度为 roadMarkingNum，即路标线个数

7.5.18 ADAS_RoadMarking

说明

路面标记。

定义

```
typedef struct {
    uint8 roadMarkingType;
    float32 classConfidence;
    ADAS TimeStamp roadMarkingAge;
    uint8 coordinate;
    ADAS Points2D roadMarkingPoints;
    uint8 arrowType;
```

```
} ADAS_RoadMarking;
```

成员

成员名称	描述
roadMarkingType	地面标记类型 0x0: StopLine（停止线） 0x1: ZebraCrossing（斑马线）
classConfidence	地面标记置信度，单位：% ，取值范围：0~100
roadMarkingAge	路面标记被跟踪的时间
coordinate	坐标系类型 0x0: Unknow（未知） 0x1: Vehicle（车辆坐标系） 0x2: WGS84（全球地理坐标系） 0x3: UTM（局部坐标系）
roadMarkingPoints	点数据 停止线：需填写停止线左起点、右终点 斑马线：需填写斑马线右前点、斑马线左前点、斑马线右后点、斑马线左后点 其它：则按实际情况填写点信息
arrowType	路面箭头类型 0x0: NoArrow（无箭头） 0x1: Straight（直行） 0x2: Left（左转） 0x3: Right（右转） 0x4: StraightOrLeft（直行或左转） 0x5: StraightOrRight（直行或右转） 0x6: TurningPointLeft（左转点） 0x7: TurningPointRight（右转点） 0x8: 45DegLeft（45度左转） 0x9: 45DegRight（45度右转）

7.5.19 ADAS_TrafficLightInfo

说明

交通灯信息。

定义

```
typedef struct {
    uint8 trafficLightNum;
    ADAS TrafficLightAry listOfTrafficLight;
} ADAS TrafficLightInfo;
```

成员

成员名称	描述
trafficLightNum	交通灯个数
listOfTrafficLight	交通灯集合数组

7.5.20 ADAS_TrafficLightAry

说明

交通灯数组。

定义

```
typedef ADAS TrafficLight ADAS TrafficLightAry[trafficLightNum];
```

成员

成员名称	描述
ADAS_TrafficLightAry	交通灯数组，数组长度为 trafficLightNum，即交通灯个数

7.5.21 ADAS_TrafficLight

说明

交通灯结构。

定义

```
typedef struct {
    uint32 trafficLightId;
    float32 trafficLightExstProb;
    float32 trafficLightAge;
    uint8 trafficLightType;
    float32 classConfidence;
    uint8 trafficLightColour;
    float32 trafficLightDistance;
```

```

uint8 coordinate;
ADAS_Point3D trafficLightPosition;
ADAS_Points2D trafficLightBoxPoints;
sint32 laneAttached;
float32 remainingTime;
} ADAS_TrafficLight;

```

成员

成员名称	描述
trafficLightId	红绿灯标号
trafficLightExstProb	交通灯存在概率, 单位: %, 取值范围: 0~100
trafficLightAge	目标被跟踪的时间
trafficLightType	交通灯类型 0x0: Straight (直行) 0x1: Left (左转) 0x2: Right (右转) 0x3: Turn (掉头)
classConfidence	交通灯类型置信度, 单位: %, 取值范围: 0~100
trafficLightColour	交通灯颜色 0x0: Unknow (未知) 0x1: Red (红灯) 0x2: Yellow (黄灯) 0x3: Green (绿灯) 0x4: Black (熄灭)
trafficLightDistance	交通灯距离, 单位: m
coordinate	坐标系类型 0x0: Unknow (未知) 0x1: Vehicle (车辆坐标系) 0x2: WGS84 (全球地理坐标系) 0x3: UTM (局部坐标系)
trafficLightPosition	红绿灯位置, 地图坐标系, 灯框外轮廓左上角
trafficLightBoxPoints	灯框的外轮廓
laneAttached	对应车道, 取值范围: -31~31 (本车行驶车道为 0, 左侧车道为负, 右侧车道为正)
remainingTime	交通灯剩余时间, 单位: s

7.5.22 ADAS_TrafficSignInfo

说明

交通标志信息。

定义

```
typedef struct {
    uint8 trafficSignNum;
    ADAS_TrafficSignAry listOfTrafficSign;
} ADAS_TrafficSignInfo;
```

成员

成员名称	描述
trafficSignNum	交通标志集合数
listOfTrafficSign	交通标志集合数组，单位：个

7.5.23 ADAS_TrafficSignAry

说明

交通标志集合数组。

定义

```
typedef ADAS_TrafficSign ADAS_TrafficSignAry[trafficSignNum];
```

成员

成员名称	描述
ADAS_TrafficSignAry	交通标志数组，数组长度为 trafficSignNum，即交通标志个数

7.5.24 ADAS_TrafficSign

说明

交通标志结构。

定义

```
typedef struct {
    uint32 trafficSignId;
    float32 existProbability;
    float32 trafficSignAge;
    uint8 signSematic;
    float32 classConfidence;
    uint32 signValue;
    uint8 vlaueUnit;
    uint8 signState;
    uint8 coordinate;
    ADAS Point3D signPosition;
    ADAS Points2D signBoxPoints;
} ADAS TrafficSign;
```

成员

成员名称	描述
trafficSignId	交通标志标号
existProbability	存在概率，取值范围：0~1
trafficSignAge	目标被跟踪的时间，单位：s
signSematic	交通标志语义类型 0x0: SpeedLimit（限速） 0x1: HzardWarning（危险警示）
classConfidence	类型置信度，单位：% ，取值范围：0~100
signValue	交通标志对应的指示数值
vlaueUnit	数值单位 0x0: km/h 0x1: ton
signState	标志状态 0x0: Unkown（未知） 0x1: Dynamic（动态） 0x2: Static（静态） 0x3: Unavailable（不可用）
coordinate	坐标系类型 0x0: Unknow（未知） 0x1: Vehicle（车辆坐标系） 0x2: WGS84（全球地理坐标系） 0x3: UTM（局部坐标系）

成员名称	描述
signPosition	位置，参见车辆坐标系，标志外轮廓左上角
signBoxPoints	交通标志的外轮廓

7.5.25 ADAS_ParkingInfo

说明

停车位信息。

定义

```
typedef struct {
    uint8 parkingSpaceNum;
    ADAS ParkingSpaceAry parkingLots;
} ADAS ParkingInfo;
```

成员

成员名称	描述
parkingSpaceNum	停车位个数
parkingLots	停车位数据

7.5.26 ADAS_ParkingSpaceAry

说明

交通标志集合数组。

定义

```
typedef ADAS ParkingSpace ADAS ParkingSpaceAry[parkingSpaceNum];
```

成员

成员名称	描述
ADAS_ParkingSpaceAry	停车位数组，数组长度为 parkingSpaceNum，即停车位个数

7.5.27 ADAS_ParkingSpace

说明

停车位数据。

定义

```
typedef struct {
    uint32 parkingSpaceId;
    uint8 parkingStatus;
    uint8 parkingSpaceType;
    float32 classConfidence;
    float32 parkingSpaceWidth;
    float32 parkingSpaceLength;
    uint8 lineType;
    uint8 lineColor;
    uint8 lineWide;
    uint8 coordinate;
    ADAS Points3D parkingSpotPoints;
} ADAS ParkingSpace;
```

成员

成员名称	描述
parkingSpaceId	停车位 ID, 同一停车区域描述唯一
parkingStatus	车位状态 0x0: Unkown (未知) 0x1: Occupied (占用) 0x2: Vacant (空置)
parkingSpaceType	停车位类型 0x0: Unkown (未知) 0x1: Vertical (垂直) 0x2: Parallel (水平) 0x3: Inclined (倾斜)
classConfidence	类型置信度, 单位: %, 取值范围: 0~100
parkingSpaceWidth	停车位宽度, 单位: m
parkingSpaceLength	停车位长度, 单位: m
lineType	车位线类型 0x0: Unkown (未知) 0x1: UType (U 型) 0x2: TType (T 型)

成员名称	描述
	0x3: Rectangle (矩形)
lineColor	车位线颜色 0x0: Unkown (未知) 0x1: Red (红色) 0x2: Orange (橘色) 0x3: Yellow (黄色) 0x4: Green (绿色) 0x5: Blue (蓝色) 0x6: White (白色)
lineWide	车位线宽度
coordinate	坐标系类型 0x0: Unknow (未知) 0x1: Vehicle (车辆坐标系) 0x2: WGS84 (全球地理坐标系) 0x3: UTM (局部坐标系)
parkingSpotPoints	泊车位构成点列, 车辆坐标系下的 3D 点阵, 若是矩阵信息, 可以选四个角落的点, 单位: {m,m,m}

7.5.28 ADAS_FreeSpaceInfo

说明

可行驶空间信息。

定义

```
typedef struct {
    uint8 boundaryType;
    float32 classConfidence;
    uint8 coordinate;
    ADAS_Points3D freeSpacePoints;
    ADAS_Coe curve;
} ADAS_FreeSpaceInfo;
```

成员

成员名称	描述
boundaryType	可行驶空间边界类型

成员名称	描述
	0x0: Unknow (未知) 0x1: RoadEdge (路沿) 0x2: Line (道路线) 0x3: Vehicle (车辆) 0x4: Pedestrian (行人) 0x5: Cone (锥体)
classConfidence	类型置信度, 单位: %, 取值范围: 0~100
coordinate	坐标系类型 0x0: Unknow (未知) 0x1: Vehicle (车辆坐标系) 0x2: WGS84 (全球地理坐标系) 0x3: UTM (局部坐标系)
freeSpacePoints	可行驶区域边界点
curve	车道线拟合参数

7.5.29 ADAS_PerceptionFusionObjects

说明

感知信息汇总, 包含目标识别信息、车道信息、路面标记信息、交通灯信息、交通标志信息、停车位信息、可行驶空间等信息。

定义

```
typedef struct {
    ADAS TrackObjects trackObjects;
    ADAS LaneLineInfo laneLineInfo;
    ADAS RoadMarkingInfo roadMarkingInfo;
    ADAS TrafficLightInfo trafficLightInfo;
    ADAS TrafficSignInfo trafficSignInfo;
    ADAS ParkingInfo parkingInfo;
    ADAS FreeSpaceInfo freeSpaceInfo;
} ADAS PerceptionFusionObjects;
```

成员

成员名称	描述
trackObjects	跟踪目标信息
laneLineInfo	车道线信息

成员名称	描述
roadMarkingInfo	路面标记信息
trafficLightInfo	交通灯信息
trafficSignInfo	交通标志信息
parkingInfo	停车位信息
freeSpaceInfo	可行驶空间

7.5.30 ADAS_Pose

说明

位姿信息。

定义

```
typedef struct {
    ADAS Point3D position;
    ADAS Quaternion orientation;
    float32 covariance;
} ADAS Pose;
```

成员

成员名称	描述
position	位置信息，单位：{m, m, m}
orientation	姿态
covariance	协方差

7.5.31 ADAS_Dimension

说明

目标轮廓信息。

定义

```
typedef struct {
    float32 height;
    float32 width;
    float32 length;
```

```
} ADAS_Dimension;
```

成员

成员名称	描述
height	高度, 单位: m, 取值范围: 0~300
width	宽度, 单位: m, 取值范围: 0~300
length	长度, 单位: m, 取值范围: 0~300

7.5.32 ADAS_Velocity

说明

速度信息。

定义

```
typedef struct {
    ADAS_Point3D linear;
    ADAS_Point3D angular;
    float32 covariance;
} ADAS_Velocity;
```

成员

成员名称	描述
linear	线速度, 分 x、y、z 三个方向分量, 对应东向、北向、朝天方向 单位: {m/s, m/s, m/s}
angular	角速度, 分 x、y、z 三个方向分量, 对应横滚角、俯仰角、方位角。 x 分量: 车辆水平时, 横滚角为零, 左侧高于右侧时为正, 相反为负。 y 分量: 车辆水平时, 俯仰角为零, 上坡俯仰角为正, 下坡俯仰角为负。 z 分量: 车辆指向北方, 方位角为零, 车辆指向东方, 方位角为正 单位: {rad/s, rad/s, rad/s}
covariance	协方差

7.5.33 ADAS_Acceleration

说明

加速度信息。

定义

```
typedef struct {
    ADAS_Point3D linear;
    ADAS_Point3D angular;
    float32 covariance;
} ADAS_Acceleration;
```

成员

成员名称	描述
linear	线加速度, 单位: {m/s2, m/s2, m/s2}
angular	角加速度, 单位: {rad/s2, rad/s2, rad/s2}
covariance	协方差

7.5.34 ADAS_Points2D

说明

二维点阵信息, 由一系列 2D 点构成的 Array, 顺序相连。

定义

```
typedef struct {
    uint8 pointNum;
    ADAS_Point2DAry listOfPoint2D;
} ADAS_Points2D;
```

成员

成员名称	描述
pointNum	二维点个数
listOfPoint2D	二维坐标数组

7.5.35 ADAS_Point2DAry

说明

二维点数组。

定义

```
typedef ADAS_Point2D ADAS_Point2DAry[pointNum];
```

成员

成员名称	描述
ADAS_Point2DAry	二维坐标数组，数组长度为 pointNum，即点个数

7.5.36 ADAS_Points3D

说明

三维点阵信息，由一系列 3D 点构成的 Array，顺序相连。

定义

```
typedef struct {
    uint8 pointNum;
    ADAS_Point3DAry listOfPoint3D;
} ADAS_Points3D;
```

成员

成员名称	描述
pointNum	三维点个数形
listOfPoint3D	三维坐标数组

7.5.37 ADAS_Point3DAry

说明

三维点数组。

定义

```
typedef ADAS_Point3D ADAS_Point3DAry[pointNum];
```

成员

成员名称	描述
ADAS_Point3DAry	三维坐标数组，数组长度为 pointNum，即点个数

7.5.38 ADAS_Point2D

说明

二元组信息。

定义

```
typedef struct {
    float32 x;
    float32 y;
} ADAS_Point2D;
```

成员

成员名称	描述
x	X 轴坐标或向量
y	Y 轴坐标或向量

7.5.39 ADAS_Point3D

说明

三元组信息。

定义

```
typedef struct {
    float32 x;
    float32 y;
    float32 z;
} ADAS_Point3D;
```

成员

成员名称	描述
x	X 抽坐标或向量
y	Y 抽坐标或向量
z	Z 抽坐标或向量

7.5.40 ADAS_Quaternion

说明

四元组信息。

定义

```
typedef struct {
    float32 qx;
    float32 qy;
    float32 qz;
    float32 qw;
} ADAS Quaternion;
```

成员

成员名称	描述
qx	X 分量
qy	Y 分量
qz	Z 分量
qw	W 分量

7.5.41 ADAS_Coe

说明

轨迹曲线系数。

定义

```
typedef struct {
    float32 c0;
    float32 c1;
    float32 c2;
```

```
float32 c3;
float32 c4;
float32 c5;
} ADAS_Coe;
```

成员

成员名称	描述
c0	常数项
c1	一次项系数
c2	二次项系数
c3	三次项系数
c4	四次项系数
c5	五次项系数

7.5.42 ADAS_TimeStamp

说明

时间戳信息。

定义

```
typedef struct {
    uint32 timeStampS;
    uint32 timeStampNs;
} ADAS TimeStamp;
```

成员

成员名称	描述
timeStampS	时间戳整秒
timeStampNs	时间戳纳秒

8 HMI(人机交互)原子服务 API

8.1 HMI_ClickSwitch(通用点击型按钮操作服务)

提供通用点击型按钮原子服务，对用户点击按钮的行为进行抽象。点击型按钮按压一次有效：即按钮的一组 DOWN/UP 为一个原子操作，按钮的持续按压时长对该类型按钮无意义。

HMI_ClickSwitch(通用点击型按钮操作服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
HMI_ClickMode	notifyMode()	上报通用点击型按钮的点击模式。

8.1.1 notifyMode

功能说明

上报通用点击型按钮的点击模式。

函数原型

```
HMI_ClickMode notifyMode()
```

参数说明

无。

返回值说明

类型	描述
HMI_ClickMode	按钮的点击模式

8.2 HMI_PressSwitch(通用按压型按键按压服务)

提供通用按压型按键原子服务，对用户按压按键的行为进行抽象。按压型按键持续按下时状态一直有效。

HMI_PressSwitch(通用按压型按键按压服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
HMI_PressedStatus	notifyStatus()	上报通用按压型按键的按压状态。

8.2.1 notifyStatus

功能说明

上报通用按压型按键的按压状态。

函数原型

```
HMI_PressedStatus notifyStatus()
```

参数说明

无。

返回值说明

类型	描述
HMI_PressedStatus	按键的按压状态

8.3 HMI_RotarySwitch(通用旋钮型按键操作服务)

提供通用旋钮型按键原子服务，对用户操作旋钮的行为进行抽象，体现按键的顺时针和逆时针操作。

HMI_RotarySwitch(通用旋钮型按键操作服务)服务包括的 API 如下表所示：

类型	API 函数	功能说明
HMI_RotaryStatus	notifyStatus()	上报通用型旋转按钮的操作状态。

8.3.1 notifyStatus

功能说明

上报通用型旋转按钮的操作状态。

函数原型

HMI_RotaryStatus notifyStatus()

参数说明

无。

返回值说明

类型	描述
HMI_RotaryStatus	按钮的操作状态

8.4 数据类型定义

8.4.1 HMI_ClickMode

说明

按键的点击模式，如单击、双击、连击等。按键按下时，上报单击模式。当在连续识别两次单击模式，并且时间间隔小于门限（支持标定）时，识别为双击模式。

定义

```
typedef enum {
    NO_ACTION=0,
    SINGLE_CLICK=1,
    DOUBLE_CLICK=2,
    CONTINUOUS_CLICK=3
} HMI_ClickMode;
```

成员

成员名称	描述
NO_ACTION	无操作
SINGLE_CLICK	单击

成员名称	描述
DOUBLE_CLICK	双击
CONTINUOUS_CLICK	连击

8.4.2 HMI_PressedStatus

说明

指示按键被按压的状态，能识别出当前按键是否被按下。按键被按压的时长在原子服务中不体现，上层应用根据原子服务上报的按压状态，确定按压时长决定不同的处理逻辑。

定义

```
typedef enum {
    RELEASED=0,
    PRESSED=1
} HMI_PressedStatus;
```

成员

成员名称	描述
RELEASED	按键释放
PRESSED	按键按下

8.4.3 HMI_RotaryStatus

说明

旋转型按钮操作状态。

定义

```
typedef enum {
    NO_ACTION=0,
    CLOCKWISE_ROTARY=1,
    ANTICLOCKWISE_ROTARY=2
} HMI_RotaryStatus;
```

成员

成员名称	描述
------	----

成员名称	描述
NO_ACTION	无操作
CLOCKWISE_ROTARY	顺时针旋转
ANTICLOCKWISE_ROTARY	逆时针旋转

9 附录：基本数据类型说明

本章节描述原子服务所用到的基本数据类型。这些公共的基本数据类型，存放在数据类型.h 文件中，被应用于所有的原子服务数据类型定义中。

数据类型	说明	字节大小	取值范围
sint8	有符号的 8bit 整型	1	-128~127
uint8	无符号的 8bit 整型	1	0~255
sint16	有符号的 16bit 整型	2	-32768~32767
uint16	无符号的 16bit 整型	2	0~65535
sint32	有符号的 32bit 整型	4	-2147483648 ~2147483647
uint32	无符号的 32bit 整型	4	0~4294967295
float32	32bit 的浮点型	4	-3.4e38~3.4e38
boolean	布尔类型	-	1: True 0: False