

比亚迪智慧开放平台 API 说明书









目录



版本: V1	.U	1.5
--------	----	-----

2 API 接口说明······	9
2.1 调用流程 9	
2.2 注意事项 11	
3 键值说明	·· 11
4 版本说明	·· 11
5 其它说明	11
6 API 接口定义····································	·· 12
6.1 车身状态类 12	
6.1.1 方法概要	
6.1.2 获取车架号	13
6.1.3 获取车型名称	14
6.1.4 获取整车状态	
6.1.5 获取车门、前舱盖、后舱盖状态	
6.1.6 获取车窗状态	
6.1.7 获取天窗、遮阳帘位置百分比	17
6.1.8 获取蓄电池电压水平	
6.1.9 获取电源档位	19
6.1.10 获取方向盘角度信息	19
6.1.11 获取油量电量低信息	20
6.1.12 获取报警器状态	21
6.1.13 获取天窗遮阳帘配置	
6.1.14 监听方法	22
6.2 行驶数据类 24	
6.2.1 方法概要	24
6.2.2 获取行驶时间	25
6.2.3 获取电续驶里程	26
6.2.4 获取电量百分比	26
6.2.5 获取燃油续驶里程	27
6.2.6 获取燃油百分比	27
6.2.7 获取最近百公里电耗	
6.2.8 获取最近百公里油耗	
6.2.9 获取累计平均电耗	



	版本: V1.0.5
6.2.10 获取累计平均油耗	29
6.2.11 获取燃油消耗总量	29
6.2.12 获取电消耗总量	30
6.2.13 获取总里程	30
6.2.14 获取钥匙电量	31
6.2.15 获取 EV 里程	31
6.2.16 监听方法	32
6.3 车速类 32	
6.3.1 方法概要	32
6.3.2 获取油门深度	
6.3.3 获取制动深度	33
6.3.4 获取当前车速	
6.3.5 监听方法	1 10 20 7
6.4 能量、模式类 34	_
6.4.1 方法概要	3.1
6.4.2 获取整车工作模式	
6.4.3 获取整车运行模式	
6.4.4 获取原地踩油门发电状态	
6.4.5 获取原地踩油门发电功率	
6.4.6 获取路面模式命令	
6.4.7 监听方法	
6.5 全景、摄像头类 37	
6.5 全景、摄像头类 37 6.5.1 方法概要	37
6.5.2 获取全景视频模式	
6.5.3 获取全景打开状态	
6.5.4 获取 sec 倒车线配置状态	
6.5.5 获取影像输出状态	
6.5.6 获取影像的屏幕方向	
6.5.7 获取显示模式	
6.5.8 获取影像模块配置	
6.5.9 监听方法	
6.6 空调类 44	
6.6.1 方法概要	44
6.6.2 获取 A/C(压缩机) 状态	
6.6.3 获取 A/C(压缩机) 手动标志	46
6.6.4 获取风量手动标志	47
665 获取出风模式手动标志	47

	版本:	V1.0.5
6.6.6 获取空调开启状态		48
6.6.7 获取空调控制方式		48
6.6.8 获取空调循环方式		49
6.6.9 获取空调出风模式		49
6.6.10 获取空调除霜状态		50
6.6.11 获取空调风量档位		50
6.6.12 获取空调各区域温度		51
6.6.13 获取默认温度单位		52
6.6.14 获取主副驾驶温度分控方式		53
6.6.15 获取通风功能设置		53
6.6.16 获取后空调开启状态		
6.6.17 设置空调控制方式		
6.6.18 设置空调循环方式		
6.6.19 设置空调出风模式		
6.6.20 设置前后排空调除霜状态		
6.6.21 设置空调风量档位		58
6.6.22 设置空调温度		
6.6.23 设置主副驾驶温度分控方式		
6.6.24 设置驻车通风功能		
6.6.25 开启空调		
6.6.26 关闭空调		63
6.6.27 开启后排空调		
6.6.28 关闭后排空调		65
6.6.29 监听方法		66
6.7 空气质量类 68		
6.7.1 方法概要		68
6.7.2 获取 pm2.5 是否在线		
6.7.3 获取车内/车外检测状态		
6.7.4 获取车内/车外 PM2.5 等级		
6.7.5 获取车内/车外 PM2.5 数值		70
6.7.6 监听方法		71
6.8 发动机类 72		
6.8.1 方法概要		72
6.8.2 获取发动机排量		
6.8.3 获取发动机型号		
6.8.4 获取功率		
6.8.5 获取发动机转速		
6.8.6 获取冷却液位		
687 获取燃油油位信号		



	版本: V1.0
6.8.8 监听方法	76
6.9 变速箱、制动类 77	
6.9.1 方法概要	77
6.9.2 获取变速箱型号	
6.9.3 获取变速箱类型	79
6.9.4 获取自动变速箱档位	80
6.9.5 获取手动变速箱档位	80
6.9.6 获取制动液位信号	81
6.9.7 获取驻车制动开关状态	81
6.9.8 获取制动踏板状态	
6.9.9 监听方法	82
5.10 门锁类 83	(2): (2)
6.10.1 方法概要	83
6.10.2 获取各门锁状态	
6.10.3 监听方法	85
.11 车灯类 85	
6.11.1 方法概要	85
6.11.2 获取灯光 AUTO 档开关状态	86
6.11.3 获取各车灯状态	86
6.11.4 获取 AFS 开关状态	87
6.11.5 监听方法	88
5.12 安全带类 89	
6.12.1 方法概要	89
5.12.2 获取各位置安全带状态	89
5.12.3 获取各位置乘客状态	
5.12.4 监听方法	91
.13 雷达类 92	
6.13.1 方法概要	92
6.13.2 获取各位置探头状态	
6.13.3 获取所有位置的探头状态	94
6.13.4 获取倒车雷达开关状态	95
6.13.5 监听方法	96
5.14 充电类 97	
6.14.1 方法概要	97
6.14.2 获取车载充电器故障状态	98



J	饭本:	V1.0.5
6.14.3 获取车载充电器工作状态		99
6.14.4 获取本次累计总充电量		99
6.14.5 获取充电模式		100
6.14.6 获取当前充满电剩余时间(时,分)		100
6.14.7 获取充电口盖开关状态		101
6.14.8 获取交流充电口电锁执行反馈		101
6.14.9 获取放电请求状态		102
6.14.10 获取充电器状态		103
6.14.11 获取充电枪连接状态		.103
6.14.12 获取充电功率		104
6.14.13 获取动力电池管理器与外接充电设备当前状态		
6.14.14 获取定时充电功能状态		
6.14.15 获取预约充电状态		107
6.14.16 获取充电枪未插提醒状态		107
6.14.17 获取预约充电倒计时(时,分)		
6.14.18 监听方法		108
6.15 轮胎类 111		
6.15.1 方法概要		111
6.15.2 获取各轮胎漏气状态		
6.15.3 获取胎压系统电池电量状态		
6.15.4 获取各轮胎压力状态		
6.15.5 获取各轮胎压力值		
6.15.6 获取各轮胎信号状态		
6.15.7 获取胎压系统状态		
6.15.8 获取胎压系统温度状态		
6.15.9 监听方法		
6.16 仪表类 120		
6.16.1 方法概要		
6.16.2 获取故障提示信息		
6.16.3 获取蜂鸣器状态		
6.16.4 获取温度、气压、油耗距离、功率单位		
6.16.5 获取保养时间/里程		
6.16.6 设置温度、气压、油耗距离、功率单位		
6.16.7 设置保养时间/里程		
6.16.8 获取外接充电量		
6.16.9 监听方法		130
6.17 时间类 131		
6.17.1 方法概要		131



	版本:	V1.0.5
6.17.2	获取时间	132
6.17.3	获取时制	132
6.17.4	设置年月日、星期	133
	设置时分秒	
6.17.6	设置时制	135
6.17.7	监听方法	135
6.18 车	辆设置类 136	
6.18.1	方法概要	136
6.18.2	是否有某项功能配置	143
	获取后排空调在线状态	
6.18.4	监听方法	144
	感器类 147	
6.19.1	方法概要	147
6.19.2	获取光照强度等级	148
6.19.3	监听方法	148
	基体中心 类 149	
6.20.1	方法概要	149
6.20.2	获取播放类型	150
6.20.3	获取播放模式	152
6.20.4	获取播放状态	153
	获取当前播放的音视频信息	
	媒体中心控制	
6.20.7	监听方法	157



1概述

开放 API 内容包括车身、行驶数据、车速、能力模式、全景、空调、 PM2.5、雷达、充电设备、车辆设置等 18 类数据。各模块主要通过 get、set、监听三种方式开放数据。 get 用于获取车辆状态数据, set 用于控制车辆和更改车辆设置项, 监听可以实时获取各模块数据的变化。 其中 set 接口的返回值,仅表示命令下发是否成功,需要通过监听接口确定设置是否成功。

开发者需要使用比亚迪公开的 SDK 开发, SDK 是基于 Android 7.1.2 开发的。

2 API 接口说明

2.1 调用流程

以空调类说明:

1、AndroidManifest.xml 声明权限

<uses-permission android:name="android.permission.BYDAUTO_AC_COMMON" />

当调用 getXxx()接口时需要添加以下权限:

<uses-permission android:name="android.permission.BYDAUTO_AC_GET" />

当调用 setXxx()接口时需要添加以下权限:

<uses-permission android:name="android.permission.BYDAUTO AC SET" />

当调用媒体中心类接口 controlMedia 时需要另外加入以下权限:

<uses-permission

android:name="com.byd.mediacenter.STARTSERVER"

android:protectionLevel="signatureOrSystem"/>

其中 BYDAUTO AC COMMON 属于在代码中动态申请的权限。

目前只有空调类、车身类、门锁类、仪表类、全景影像类、设置类需要申请动态权限, 具体权限请参考 api 开发文档。

2、创建实例

在创建实例之前需要动态申请获得 BYDAUTO_AC_COMMON 权限,否则整个类的所有接口都不



能使用。

BYDAutoAcDevice bydAutoAcDevice = BYDAutoAcDevice.getInstance(mContext);

3、调用接口

bydAutoAcDevice.start(BYDAutoAcDevice.AC_CTRL_SOURCE_VOICE);

4、注册监听

bydAutoAcDevice.registerListener(absBYDAutoAcListener);

```
AbsBYDAutoAcListener absBYDAutoAcListener = new AbsBYDAutoAcListener() {
           @Override
           public void onAcStarted() {
               super.onAcStarted();
           @Override
           public void onAcStoped() {
               super.onAcStoped();
          }
           @Override
           public void onAcOnlineStateChanged(int state) {
               super.onAcOnlineStateChanged(state);
          }
           @Override
           public void onAcRearStarted() {
              super.onAcRearStarted();
           @Override
           public void onAcRearStoped() {
               super.onAcRearStoped();
           @Override
           public void onAcCtrlModeChanged(int mode) {
               super.onAcCtrlModeChanged(mode);
          }
           @Override
           public void onAcCycleModeChanged(int mode) {
               super.onAcCycleModeChanged(mode);
```

@Override



```
public void onAcVentilationStateChanged(int state) {
        super.onAcVentilationStateChanged(state);
}
@Override
public void onAcTemperatureControlModeChanged(int mode) {
        super.onAcTemperatureControlModeChanged(mode);
}
```

5、取消监听

bydAutoAcDevice.unregisterListener(absBYDAutoAcListener);

2.2 注意事项

- 1、由于各车型配置不同、电源档位不同, 某些接口不能正确返回。 set 接口建议在 ON 档电下操作。
- 2、接口实际的输入输出以公开的 SDK 为准。

3 键值说明

车内转向盘开关可以控制车载多媒体部分功能,开放以下三个健值供开发者使用。

键名	描述
KEYCODE_MEDIA_PREVIOUS	上一首
KEYCODE_MEDIA_NEXT	下一首
KEYCODE_AUTO_MEDIA_VOICE	激活语音功能

4 版本说明

此 API 版本为 V1.0.5, 对应的 SDK 版本为 V1.0.5。

5 其它说明

- 1、获取 GPS 信息请参考标准安卓接口。
- 2、获取音量信息请参考标准安卓接口。
- 3、APK 需要系统签名才能安装运行。
- 4、部分车型有摄像头配置,包括行车记录仪摄像头(车外影像)、顶灯摄像头(车内影像),开发者如



果需要获取摄像头影像信息,可调用安卓 Camera 标准接口。不同摄像头可用camerald 区分。

摄像头类型	id
行车记录仪摄像头	0
顶灯摄像头	1

6 API 接口定义

6.1 车身状态类

6.1.1 方法概要

类 BYDAutoBodyworkDevice			
public class BYDAutoBodyworkDevice			
	方法概要		
限定符和返回类型	方法	描述	
Static	/////		
BYDAutoBodyworkDevi		获取实例	
ce	getInstance(Context con)		
String	getAutoVIN()	获取车架号	
int	getAutoModelName()	获取车型名称	
int	getAutoSystemState()	获取整车状态	
		获取车门、前舱盖、后	
int	getDoorState(int area)	舱盖状态	
int	getWindowState(int area)	获取车窗状态	



int		获取天窗、遮阳帘位
	getWindowOpenPercent(int area)	置百分比
int	getBatteryVoltageLevel()	获取蓄电池电压水平
int	getPowerLevel()	获取电源档位
double	getSteeringWheelValue(int type)	获取方向盘角度信息
int	getFuelElecLowPower()	获取油量电量低信息
int	getAlarmState()	获取报警器状态
int	getMoonRoofConfig()	获取天窗遮阳帘配置
	registerListener(AbsBYDAutoBodywor	
void	kListener l)	监听注册接口
	unregisterListener(AbsBYDAutoBody	
void	workListenerl)	取消注册接口

6.1.2 获取车架号

API 名称	String getAutoVIN()		
接口描述	获取车架号		
输入参数	数		
	返回值		
数据类型	名称	值	描述
String	车架号		17 位字符串



6.1.3 获取车型名称

API 名称		int getAutoModelName()						
接口描述		获取车型名称						
输入参数		无						
	返回值							
数据类型	名称	值	描述					
		AUTO_MODEL_SONG_MAX_HE	宋 MAX 混动					
		AUTO_MODEL_NEW_QIN_HEV	秦 Pro 混动					
		AUTO_MODEL_NEW_QIN_EV	秦 Pro 纯电					
		AUTO_MODEL_NEW_QIN_FUEL	秦 Pro 燃油					
		AUTO_MODEL_NEW_TANG_HEV	全新一代唐混动					
int	车型名	AUTO_MODEL_NEW_TANG_EV	全新一代唐纯电					
	称	称	AUTO_MODEL_NEW_TANG_FUE	全新一代唐燃油				
W		AUTO_MODEL_SONG_18_HEV	宋 18 款混动					
V		AUTO_MODEL_SONG_18_EV	宋 18 款纯电					
		AUTO_MODEL_SONG_18_FUEL	宋 18 款燃油					
		AUTO_MODEL_NULL	无					
备注	请以实际	请以实际上市名称和配置为准						



6.1.4 获取整车状态

API 名称	int getAutoSystemState()						
接口描述	获取整车状态						
输入参数	无						
		返回值					
数据类型	名称	值	描述				
	整车状态	BODYWORK_AUTO_SYSTEM_STATE_ NORMAL	正常系统状态				
			BODYWORK_AUTO_SYSTEM_STATE_S ET_SECURE	设定安全系统状态			
int		BODYWORK_AUTO_SYSTEM_STATE_S TART_SECURE	启动安全系统状态				
		BODYWORK_AUTO_SYSTEM_STATE_ UNDEFINED	未定义				

6.1.5 获取车门、前舱盖、后舱盖状态

API 名称	int getDoorState(int area)					
接口描述	获取车门、前舱盖、后舱盖、油箱盖状态					
	输入参数					
数据类型	数据类型 字段名 名称 值 描述					
int	area	门区域	BODYWORK_CMD_DOOR_LEFT_F	左前门		



			RONT	
			BODYWORK_CMD_DOOR_RIGHT	
			_FRONT	右前门
			BODYWORK_CMD_DOOR_LEFT_	
			REAR	左后门
			BODYWORK_CMD_DOOR_RIGHT	
			_REAR	右后门
			BODYWORK_CMD_DOOR_HOOD	发动机前舱盖
			BODYWORK_CMD_DOOR_LUGG AGE_DOOR	后舱盖
			返回值	
数据类型	名	称	值	描述
		-1/	BODYWORK_STATE_OPEN	开启
	1.697		BODYWORK_STATE_CLOSED	关闭
int	יכו	门状态	BODYWORK_STATE_UNDEFINED	未定义
N	37/		BODYWORK_COMMAND_INVALI	tv) this
7			D_VALUE	输入错误

6.1.6 获取车窗状态

API 名称	int getWindowState (int area)
接口描述	获取车窗状态



输入参数					
数据类型	字段名	名称	值	描述	
			BODYWORK_CMD_WINDOW_LE		
			FT_FRONT	左前窗	
			BODYWORK_CMD_WINDOW		
			_RIGHT_FRONT	右前窗	
int	area	车窗区域	BODYWORK_CMD_WINDOW	1.50	
			_LEFT_REAR	左后窗	
			BODYWORK_CMD_WINDOW_RI	ナ ビ卒	
			GHT_REAR	右后窗	
		4	返回值		
数据类型	名	3称	值	描述	
		7/7	BODYWORK_STATE_OPEN	开启	
			BODYWORK_STATE_CLOSED	关闭	
int	车置	窗状态	BODYWORK_STATE_UNDEFINED	未定义	
			BODYWORK_COMMAND_INVALI	<i>t</i> △ <i>\ </i>	
1			D_VALUE	输入错误	

6.1.7 获取天窗、遮阳帘位置百分比

API 名称	int getWindowOpenPercent(int area)
接口描述	获取天窗、遮阳帘位置百分比



输入参数					
数据类型	字段名 名称		值	描述	
			BODYWORK_CMD_MOON_ROOF	天窗	
int	area	位置	BODYWORK_CMD_SUNSHADE_P	\ <u></u>	
			ANEL	遮阳帘	
			返回值		
数据类型 名称			值	描述	
int	百分比		[WINDOW_OPEN_PERCENT_MIN, WINDOW_OPEN_PERCENT_MAX]	[0;100]%	
备注 适用于有天窗、遮阳帘配置的车型。0%表示关闭,100%表示全部打开					

6.1.8 获取蓄电池电压水平

API 名称	int getBatteryVoltageLevel()							
接口描述		获取蓄电池电压水平						
输入参数		无						
	返回值							
数据类型	名称	值	描述					
A		BODYWORK_BATTERY_VOLTAGE_						
int		LEVEL_NORMAL	电压正常					
	电压水平	BODYWORK_BATTERY_VOLTAGE_	<i>(</i>					
		LEVEL_LOW	低电压					



В	ODYWORK_BATTERY_VOLTAGE_	
	LEVEL_INVALID	无效

6.1.9 获取电源档位

API 名称	int getPowerLevel()						
接口描述		获取电源档位					
输入参数		无					
		返回值					
数据类型	名称	值	描述				
	电源档位	BODYWORK_POWER_LEVEL_ACC	0x1: ACC 档				
int		BODYWORK_POWER_LEVEL_OFF	0x0: OFF档				
		BODYWORK_POWER_LEVEL_ON	0x2: ON 档				

6.1.10 获取方向盘角度信息

API 名称	double getSteeringWheelValue(int type)					
接口描述	获取方向盘角度信息					
	27/	输入参数				
数据类型	字段名	描述				
			BODYWORK_CMD_STEERING_W	<i>(</i> 2, () 2		
		需要获取的	HEEL_ANGEL	角度		
int	type	类型	BODYWORK_CMD_STEERING_W	\		
			HEEL_SPEED	速度		



	返回值				
数据类型	名称	值	描述		
		[BODYWORK_STEERING_WHEEL			
		_SPEED_MIN,	[0,1016]°/s 方向		
		BODYWORK_STEERING_WHEEL_	盘旋转速度		
		SPEED_MAX]°/s			
		[BODYWORK_STEERING_WHEEL	1>		
double	角度/速度	_ANGEL_MIN,	1-		
		BODYWORK_STEERING_WHEEL_	780.0°,+779.9°]		
		DOD I WOUNT_STEEL NITE_TITLE_	方向盘角度		
		ANGEL_MAX] °			
	,	BODYWORK_COMMAND_INVALI	/A > /#>=		
	_ / \	D_VALUE	输入错误		

6.1.11 获取油量电量低信息

API 名称	int getFuelElecLowPower()			
接口描述		获取油量电量低信息		
输入参数)/	无		
N	返回值			
数据类型	型 名称 值 描述		描述	
	/r/사용 사제	BODYWORK_LOW_POWER_NOR	,+8 +8+0+4	
int	低能量类型	MAL	油量、电量都正常	



	BODYWORK_LOW_POWER_FUEL	油量低
	BODYWORK_LOW_POWER_ELEC	电量低
	BODYWORK_LOW_POWER_BOT	N = +=+=
	н	油量、电量都低

6.1.12 获取报警器状态

API 名称	int getAlarmState()			
接口描述	获取报警器状态			
输入参数		无		
	返回值			
数据类型	名称 值		描述	
	15#fc5 15 to 1	BODYWORK_ALARM_STATE_ON	报警	
int	报警器状态	BODYWORK_ALARM_STATE_OFF	不报警	
备注	备注 钥匙远离探测范围,且门开着时报警,闭锁不报警。			

6.1.13 获取天窗遮阳帘配置

API 名称	int getMoonRoofConfig()		
接口描述	获取天窗遮阳帘配置		
输入参数	无		
返回值			
数据类型	名称	值	描述



		CONFIG_NONE	无
		CONFIG_MOON_ROOF_SUNSHA	人目于恋嫔们会
		DE_PANEL	全景天窗遮阳帘
int	天窗遮阳帘		全景遮阳帘(天窗无
	配置	CONFIG_SUNSHADE_PANEL	法打开)
		CONFIG_ANTI_PINCH_MOON_R	
		OOF	防夹小天窗

6.1.14 监听方法

	lywork Listener	
文 / losb i D / latoboo	iy work Listerici	
tract class AbsBYDAutoBodywork	Listener	
 象数值发生变化时,推送给用户		
方法概	要	
方法	描述	输入
on Auto System State Changed (int state)	监听整车状态变化	0:正常系统状态 1:设定安全系统 状态 2:启动安全系统 状态
		1: 电压正常 2: 低电压
	方法概: 方法概: 方法 onAutoSystemStateChanged(int	方法 描述 onAutoSystemStateChanged(int state) which is a state with a s



			//X/ + 1. • 1.0.5
void	onDoorStateChanged(int area, int	监听车门变化	开启
	state)		关闭
void	onWindowStateChanged(int area,	监听车窗变化	开启
Void	int state)	<u></u> 一	关闭
			area:
			天窗
void	onWindowOpenPercent(int area,		遮阳帘
	int value)	分比的变化	value:
		(KS)	位置百分比
			0x1: ACC 档
void	onPowerLevelChanged(int level)	监听电源档位变化	0x0: OFF档
Volu			0x2: ON 档
	7-1-7		0x3: OK/READY
			type 为角度时,
void	onSteeringWheelValueChanged(i	监听方向盘角度变化	输入方向盘角度;
10,0	nt type, double value)		type 为速度时,
V			输入方向盘速度
		监听油量电量低提醒	油量、电量都正常
void	onFuelElecLowPowerChanged(int		油量低
VOIG	state)		电量低
			油量电量都低



void	onAlarmStateChanged(int state)	监听报警器状态	报警
Void			不报警
备注	输入值的取值 参见 get 函数的具体描述		

6.2 行驶数据类

6.2.1 方法概要

类BYDAutoStatisticDevice			
public classBYD	AutoStatisticDevice		
	方法概要		
限定符和返回类	方法	描述	
型			
staticBYDAuto	getInstance(Context con)	获取实例	
StatisticDevice	-Y/_ X		
double	getDrivingTimeValue()	获取行驶时间	
int	getElecDrivingRangeValue()	获取电续驶里程	
double	getElecPercentageValue()	获取电量百分比	
int	getFuelDrivingRangeValue()	获取燃油续驶里程	
int	getFuelPercentageValue()	获取燃油百分比	
double	getLastElecConPHMValue()	获取最近百公里电耗	
double	getLastFuelConPHMValue()	获取最近百公里油耗	
double	getTotalElecConPHMValue()	获取累计平均电耗	



double	getTotalFuelConPHMValue()	获取累计平均油耗
double	getTotalFuelConValue()	获取燃油消耗总量
double	getTotalElecConValue()	获取电消耗总量
int	getTotalMileageValue()	获取总里程
int	getKeyBatteryLevel()	获取钥匙电量
int	getEVMileageValue()	获取 EV 里程
void	registerListener(AbsBYDAutoStatisticLi stener I)	监听注册接口
void	unregisterListener(AbsBYDAutoStatisti cListener I)	取消注册接口
备注	与电池电量、电耗相关的接口适用于混动、	· 纯电动车型

6.2.2 获取行驶时间

API 名称	doublegetDrivingTimeValue()		
接口描述	获取行驶时间		
输入参数	无		
	返回值		
数据类型	类型 名称 值		描述
	n 12-3	[STATISTIC_DRIVING_TIME_MIN,	(0.0000.0).
double	时间	STATISTIC_DRIVING_TIME_MAX]	{0,9999.9} h



6.2.3 获取电续驶里程

API 名称		int getElecDrivingRangeValue()		
接口描述		获取电续驶里程		
输入参数		无		
返回值				
数据类型	名称	值	描述	
int	时间	[STATISTIC_ELEC_DRIVING_RANG E_MIN, STATISTIC_ELEC_DRIVING_RANG E_MAX]	[0,511]KM	

6.2.4 获取电量百分比

, 1X/				
API 名称	Tiller.	double getElecPercentageValue()		
接口描述	_>\\	获取电量百分比		
输入参数		无		
· VI	返回值			
数据类型	名称	值	描述	
4		[STATISTIC_ELEC_PERCENTAGE_M		
daulala	电量百分比	IN,	[0,100]%	
double		STATISTIC_ELEC_PERCENTAGE_M	[0,100] //	
		AX]		



6.2.5 获取燃油续驶里程

API 名称		int getFuelDrivingRangeValue()		
接口描述	获取燃油续驶里程			
输入参数		无		
	返回值			
数据类型	名称	值	描述	
int	燃油续驶里程	[STATISTIC_FUEL_DRIVING_RANG E_MIN, STATISTIC_FUEL_DRIVING_RANG E_MAX]	{0,4095}KM	

6.2.6 获取燃油百分比

API 名称	Int getFuelPercentageValue()			
接口描述	_ >\)	获取燃油百分比		
输入参数		无		
返回值				
数据类型	名称	值	描述	
A		[STATISTIC_FUEL_PERCENTAGE_		
	燃油百分比	MIN,	{0,100}%	
int		STATISTIC_FUEL_PERCENTAGE_M	{∪,1∪∪}∞	
		AX]		



6.2.7 获取最近百公里电耗

API 名称	double getLastElecConPHMValue()			
接口描述		获取最近百公里电耗		
输入参数		无		
返回值				
数据类型	名称	值	描述	
		[STATISTIC_LAST_ELEC_CON_PH	1/1/2	
double	最近百公里	M_MIN,	{-99.9;99.9}KWH/100KM	
	电耗	STATISTIC_LAST_ELEC_CON_PHM	(-33.3,33.3)KWIII/ TOOKIVI	
		_MIN]		

6.2.8获取最近百公里油耗

API 名称	William	double getLastFuelConPHMValue()		
接口描述	_XX	获取最近百公里油耗		
输入参数		无		
. VÍ.	返回值			
数据类型	名称	值	描述	
A		[STATISTIC_LAST_FULE_CON_PH		
double	最近百公里	M_MIN,	(0.54.4) (4.00)(0.4	
	油耗	STATISTIC_LAST_FULE_CON_PHM	{0;51.1} L/100KM	
		_MIN]		



6.2.9 获取累计平均电耗

	1			
API 名称		double getTotalElecConPHMValue()		
接口描述		获取累计平均电耗		
输入参数		无		
返回值				
数据类型	名称	值	描述	
		[TATISTIC_TOTAL_ELEC_CON_PHM_MI		
double	男计亚拉中托	N,	{-99.9;99.9}KWH/100KM	
	累计平均电耗	STATISTIC_TOTAL_ELEC_CON_PHM_M		
		AX]	1//	

6.2.10 获取累计平均油耗

API 名称	double getTotalFuelConPHMValue()			
接口描述		获取累计平均油耗		
输入参数	无			
	返回值			
数据类型	名称	值	描述	
	\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	[STATISTIC_TOTAL_FUEL_CON_PH		
	累计平均油	M_MIN,	(0.51.1) 1. (1.00)///4	
double	耗	STATISTIC_TOTAL_FUEL_CON_PH	{0;51.1} L/100KM	
N	3	M_MAX]		

6.2.11 获取燃油消耗总量

API 名称	double getTotalFuelConValue()
接口描述	获取燃油消耗量
输入参数	无



返回值			
数据类型	名称	值	描述
		[STATISTIC_TOTAL_FUEL_CONSU	
	燃油消耗总	MPTION_MIN,	
double			{0; 104857.4} L
	量	STATISTIC_TOTAL_FUEL_CONSUM	
		PTION_MAX]	112

6.2.12 获取电消耗总量

	T		
API 名称	double getTotalElecConValue()		
接口描述		获取电消耗总量	
输入参数		无	
返回值	Total Control of the	//XX	
数据类型	名称	值	描述
		[STATISTIC_TOTAL_ELEC_CONSU	
double	电消耗总量	MPTION_MIN, STATISTIC_TOTAL_	{-1000, 1676721.4} kW•h
	7-	ELEC_CONSUMPTION_MAX]	(HP.h)
Ø:+	电池放电为正		
备注 	车辆下坡、制	动、发动机发电时动力电池可得电,	得电为负。

6.2.13 获取总里程

API 名称	int getTotalMileageValue()



接口描述	获取总里程			
输入参数		无		
		返回值		
数据类型	名称	值	描述	
		[STATISTIC_TOTAL_MILEAGE_MIN		
		,		
int	总里程		{0,999999}km	
		STATISTIC_TOTAL_MILEAGE_MAX	Kr.	
		1 (1)	YK.	

6.2.14 获取钥匙电量

API 名称	int getKeyBatteryLevel()			
接口描述	获取钥匙电量			
输入参数	无			
	返回值			
数据类型	名称 值	描述		
int	STATISTIC_KEY_BATTERY_LEVEL_LOW 钥匙电量 STATISTIC_KEY_BATTERY_LEVEL_NORM AL	1: 电量不足 2: 电量正常		

6.2.15 获取 EV 里程

API 名称	int getEVMileageValue()			
接口描述		获取 EV 里程		
输入参数		无		
	返回值			
数据类型	名称	值	描述	
int	EV 里程	[STATISTIC_MILEAGE_MIN, STATISTIC_MILEAGE_MAX]	{0,999999}km	



6.2.16 监听方法

类AbsBYDAutoStatisticListener			
public ab	estract classAbsBYDAutoStatisticListener		
i .	村象数值发生变化时,推送给用户		
	方法概要		
	7372/90		
返回类型	方法	描述	输入
void	onDrivingTimeChanged(double value)	监听行驶时间变化	行驶时间
void	onElecDrivingRangeChanged(int value)	监听电续驶里程变化	电续驶里程
void	onElecPercentageChanged(double value)	监听电量百分比变化	电量百分比
void	onFuelDrivingRangeChanged(int value)	监听燃油续驶里程变化	燃油续驶里程
void	onFuelPercentageChanged(int value)	监听燃油百分比变化	燃油百分比
void	onLastElecConPHMChanged(double value)	监听最近百公里电耗变化	最近百公里电耗
void	onLastFuelConPHMChanged(double value)	监听最近百公里油耗变化	最近百公里油耗
void	onTotalElecConPHMChanged(double value)	监听累计平均电耗变化	累计平均电耗
void	onTotalFuelConChanged(double value)	监听燃油消耗量变化	燃油消耗总量
void	onTotalElecConChanged(double value)	监听电消耗量的变化	电消耗总量
void	onTotalFuelConPHMChanged(double value)	监听累计平均油耗变化	累计平均油耗
void	onTotalMileageValueChanged(int value)	监听总里程变化	总里程
void	ontKeyBatteryLevelChanged(int value)	监听钥匙电池电量变化	电池电量不足、正 常
void	onEVMileageValueChanged(int value)	监听 EV 里程的变化	EV 里程
备注	备注 输入值的取值 参见 get 函数的具体描述		

6.3 车速类

6.3.1 方法概要

	类 BYDAutoSpeedDevice				
public classBYDAutoSpeed	public classBYDAutoSpeedDevice				
	方法概要				
限定符和返回类型	方法	描述			
staticBYDAutoSpeedDevic e	getInstance(Context con)	获取实例			
int	getAccelerateDeepness()	获取油门深度			
int	getBrakeDeepness ()	获取制动深度			



double	getCurrentSpeed()	获取当前车速
	registerListener(AbsBYDAutoSpeedListen er I)	监听注册接口
void	unregisterListener(AbsBYDAutoSpeedList enerl)	取消注册接口

6.3.2 获取油门深度

API 名称	int getAccelerateDeepness ()		
接口描述	————————————————————— 获取油门深度		
输入参数			. '/^\
		返回值	, - 01
数据类型	名称	值	描述
int	油门深度	[DEEP_PERSENT_MIN, DEEP_PERSENT_MAX]	[0,100]%

6.3.3 获取制动深度

API 名称	int getBrakeDeepness ()			
接口描述	获取制动深度			
输入参数	无			
	返回值			
数据类型	名称	值	描述	
int	制动深度	[DEEP_PERSENT_MIN, DEEP_PERSENT_MAX]	[0,100]%	

6.3.4 获取当前车速

API 名称	doublegetCurrentSpeed())			
接口描述		获取当前车速		
输入参数	无			
M	返回值			
数据类型	名称	值	描述	
double	车速	[SPEED_MIN,SPEED_MAX]	[0,282.0] km/h	

6.3.5 监听方法

类AbsBYDAutoSpeedListener
public classAbsBYDAutoSpeedListener



当监听的对象数值发生变化时,推送给用户				
	方法概要			
限定符和返回类型	方法	描述	输入	
void	onAccelerateDeepnessChanged(int value)	监听油门深度变化	油门深度	
void	onBrakeDeepnessChanged(int value)	监听制动深度变化	制动深度	
void	onSpeedChanged(double value)	监听车速变化	车速值	
备注	输入值的取值范围参见 get 函数的具体描述			

6.4 能量、 模式类

6.4.1 方法概要

类 BYDAutoEnergyDevice				
public class BYDAutoEnergyDevice				
	方法概要			
限定符和返回类型	方法	描述		
static BYDAutoEnergyDevice	getInstance(Context con)	获取实例		
int	getEnergyMode()	获取整车工作模式		
int	getOperationMode()	获取整车运行模式		
int	getPowerGenerationState()	获取原地踩油门发电状 态		
int	getPowerGenerationValue()	获取原地踩油门发电功 率		
int	getRoadSurfaceMode()	获取路面模式命令		
void	registerListener(AbsBYDAutoEnergyListener I)	监听注册接口		
void	unregisterListener(AbsBYDAutoEnergyListener	取消注册接口		
备注	该类适用于混动车型			

6.4.2 获取整车工作模式

API 名称	intgetEnergyMode()		
接口描述	获取整车工作模式		
输入参数	无		
返回值			
数据类型	名称	值	描述



int	整车工作模式	ENERGY_MODE_STOP	停止
		ENERGY_MODE_EV	EV
		ENERGY_MODE_FUEL	燃油模式
		ENERGY_MODE_FORCE_EV	强制 EV
		ENERGY_MODE_HEV	HEV
备注	燃油模式暂不支持		

6.4.3 获取整车运行模式

API 名称		int getOperationMode()	
接口描述		整车运行模式	
输入参数	无		
		返回值	
数据类型	名称	值	描述
int	整车运行模式	ENERGY_OPERATION_ECONO	OMY 经济模式
		ENERGY_OPERATION_SPOR	RT 运动模式

6.4.4 获取原地踩油门发电状态

API 名称	intgetPowerGenerationState()			
接口描述	获取原地踩油门发电状态			
输入参数	无			
返回值				
数据类型	名称	值	描述	
	-1/	ENERGY_POWER_GENERATION_INVALID	无效	
int	原地踩油门发电状态	ENERGY_POWER_GENERATING	发电中	
		ENERGY_POWER_GENERATION_END	发电结束	
		ENERGY_POWER_GENERATION_ERRO R	无法进入	

6.4.5 获取原地踩油门发电功率

int getPowerGenerationValue()			
获取原地踩油门发电功率			
无			
返回值			
名称	值	描述	
原地踩油门发电功率		[1,31]KW	
		获取原地踩油门发电功率 无 返回值 名称 值 [ENERGY_POWER_GENERATION_VALUE] E MIN.	



6.4.6 获取路面模式命令

API 名称	int getRoadSurfaceMode()			
接口描述	获取路面模式命令			
输入参数		无		
	返回值			
数据类型	名称	描述		
		ENERGY_ROAD_SURFACE_KEEP	保持在线	
		ENERGY_ROAD_SURFACE_COMMON	普通模式	
int	路面模式	ENERGY_ROAD_SURFACE_SNOW	雪地/砂砾地 /草地	
		ENERGY_ROAD_SURFACE_MUDDY	泥泞地面 /车辙地	
		ENERGY_ROAD_SURFACE_SAND	沙地	

6.4.7 监听方法

	监听类 AbsBYDAutoEnergyListener			
public class	AbsBYDAuto Energy Listener	1,1,		
当监听的对象	数值发生变化时,推送给用户			
	方法概要			
限定符和返 回类型	方法	描述	輸入	
void	onEnergyModeChanged(int mode)	监听整车工作模式	EV/强制 EV/HEV	
void	onOperationModeChanged(int mode)	心心脉整在法分模式	经济模式;运动 模式	
void	onRoadSurfaceChanged(int type)	监听路面模式	保持在线 普通模式 草地 /砂砾地 / 雪地 泥泞地面 /车辙 地 沙地	
void	onPowerGenerationStateChanged(int mode)	监听原地踩油门发电状态	发电状态	
void	onPowerGenerationValueChanged(int value)	监听原地踩油门发电功率	发电功率	
备注	输入值的取值范围参见 get 函数的具体描述			



6.5全景、 摄像头类

6.5.1 方法概要

类BYDAutoPanoramaDevice					
public classBYDAutoPano	public classBYDAutoPanoramaDevice				
	方法概要				
限定符和返回类型	方法	描述			
staticBYDAutoPanorama		하나			
Device	getInstance(Context con)	获取实例			
int	getPanoOutputSignal()	获取全景视频模式			
int	getPanoWorkState()	获取全景打开状态			
		获取 sec 倒车线配置			
int	getBackLineConfig()	状态			
int	getPanoOutputState()	获取影像输出状态			
int	getPanoRotation()	获取影像的屏幕方向			
int	getDisplayMode()	获取显示模式			
int	getPanoramaOnlineState()	获取影像模块配置			
	registerListener(AbsBYDAutoPanora				
void	maListener I)	监听注册接口			
	unregisterListener(AbsBYDAutPanor				
void	amaListener I)	取消注册接口			



6.5.2 获取全景视频模式

API 名称	intgetPanoOutputSignal()			
接口描述	获取全景视频模式			
输入参数	无			
返回值				
数据类型	名称	值	描述	
		PANORAMA_OUTPUT_SIGNAL_C VBS	CVBS	
int	全景视频模式	PANORAMA_OUTPUT_SIGNAL_L VDS	LVDS	

6.5.3 获取全景打开状态

API 名称	intgetPanoWorkState()			
接口描述	获取全景打开状态			
输入参数		无		
W.	返回值			
数据类型	名称	值	描述	
		PANORAMA_WORK_OFF	关闭状态	
int	全景打开状态	PANORAMA_WORK_ON	工作状态	

6.5.4 获取 sec 倒车线配置状态

API 名称	intgetBackLineConfig()
--------	------------------------



接口描述	获取 sec 倒车线配置状态		
输入参数	无		
返回值			
数据类型	名称	值	描述
		BACK_LINE_NOT_SUPPORT	不支持配置
int	sec 倒车线配置状	BACK_LINE_PAN_INTERNAL	配置为全景内部倒车线
	态	BACK_LINE_MULTIMEDIA	配置为多媒体倒车线

6.5.5 获取影像输出状态

U.J.J 3/14/5	沙冰部山火心		
API 名称	intgetPanoOutputState()		
接口描述	获取全景输出状态		
输入参数	X	无	
	-\/\	返回值	
数据类型	名称	值	描述
		PANORAMA_OUTPUT_OFF	关闭显示
11		PANORAMA_OUTPUT_FRON	前视
		Т	
int	输出状态	PANORAMA_OUTPUT_REAR	后视
		PANORAMA_OUTPUT_LEFT	左视
		PANORAMA_OUTPUT_RIGHT	右视
		PANORAMA_OUTPUT_COMP	左右合成



			加火牛、V1.0.3
		OSE	
		PANORAMA_OUTPUT_MATC	开始匹配
		HING	
		PANORAMA_OUTPUT_FRON	前视+左视
		T_LEFT	
		PANORAMA_OUTPUT_FRON	前视+右视
		T_RIGHT	1>
		PANORAMA_OUTPUT_REAR	后视+左视
		_LEFT	
		PANORAMA_OUTPUT_REAR_	后视+右视
		RIGHT	
	全景模式、大图模式	大下、小窗口模式下都能实现 前被	见,后视、左
	视,右视、前+左、前+右、后+左、后+右。		
A7 \}	前+左、前+右、后+左、后+右,这个当前仅针对横屏状态。		
备注	全景模式,实际显示为:全景+(前+左、前+右、后+左、后+		
N	右)。		
	没有全景配置的车	(右前影像配置),仅返右视、后袖	Д.

6.5.6 获取影像的屏幕方向

API 名称	intgetPanoRotation()
接口描述	获取影像的屏幕方向



输入参数	无				
	返回值				
数据类型	数据类型 名称 值 描述				
		PANORAMA_ROTATION_HOR			
		IZONTAL	横屏		
int	影像的屏幕方向	PANORAMA_ROTATION_VERT			
		ICAL	竖屏		

6.5.7 获取显示模式

API 名称	int getDisplayMode()			
接口描述		获取显示模式		
输入参数		无		
	返回值			
数据类型	名称	值	描述	
		DISPLAY_MODE_PANORAMA	全景模式	
M.		DISPLAY_MODE_FULL_SCREE	大图模式(全屏)	
		N		
int	显示模式	DISPLAY_MODE_WIDGET	小窗口模式(widget)	
		DISPLAY_MODE_BACK_RIGHT	倒车右前模式	
		DISPLAY_MODE_BACK	倒车模式	



6.5.8 获取影像模块配置

API 名称	int getPanoramaOnlineState()			
接口描述	获取影像模块配置			
输入参数		无		
	返回值			
数据类型	名称	值	描述	
		PANORAMA_RF_REVERSE	倒车+右前影像配置	
	#7 PP	PANORAMA_ONLINE	全景影像配置	
int	配置	PANORAMA_REVERSE	倒车影像配置	
		PANORAMA_OFFLINE	无影像配置	

6.5.9 监听方法

	监听类 AbsBYDAutoPanoramaListener			
public class Al	public class AbsBYDAutPanoramaListener			
当监听的对象数	当监听的对象数值发生变化时,推送给用户			
	方法概要			
限定符和返回	方法	描述	输入	
void	onPanoOutputStateChanged(int		关闭显示 前视	
	mode)		后视	



			//////////////////////////////////////
			左视
			右视
			左右合成
			开始匹配
			前+左
			前+后
			后+左
			后+右
void	onPanoWorkStateChanged(int	监听全景打开状态	关闭状态
Void	mode)	血明王泉打开伙心	工作状态
			不支持配置
	onBackLineConfigChanged(int	监听 sec 倒车线配	配置为全景内部倒
void	mode)		车线
,	illioue)	置状态	配置为多媒体倒车
			线
	onPanoRotationChanged(int	监听影像的屏幕方	横屏
void	value)	向的变化	竖屏
			全景模式
اد:مار	on Display Mode Changed (int	监听显示模式的变	大图模式
void	mode)	化	小窗口模式
			倒车右前模式



			倒车模式
备注	输入值的取值范围参见 get 函数的具	体描述	

6.6 空调类

6.6.1 方法概要

	类 BYDAutoAcDevice	
public class BYDAu	toAcDevice	1-01
	方法概要	
限定符和返回类型	方法	描述
Static BYDAutoAcDevice	getInstance(Context con)	获取实例
int	getAcCompressorMode()	获取 A/C (压缩机)状态
	-\/\\X	获取 A/C (压缩机)手动
int	getAcCompressorManualSign()	标志
int	getAcWindLevelManualSign()	获取风量手动标志
int	getAcWindModeManualSign()	获取出风模式手动标志
int	getAcStartState()	获取空调开启状态
int	getAcControlMode()	获取空调控制方式
int	getAcCycleMode()	获取空调循环方式
int	getAcWindMode()	获取空调出风模式
int	getAcDefrostState(int area)	获取空调除霜状态



		八八年、 1.0.3
int	getAcWindLevel()	获取空调风量档位
int	getTemprature(int area)	获取空调各区域温度
int	getTemperatureUnit()	获取默认温度单位
		获取主副驾驶温度分控方
int	getAcTemperatureControlMode()	式
int	getAcVentilationState()	获取通风功能设置
int	getRearAcStartState()	获取后空调开启状态
	setAcControlMode(int setSource,int	
int	mode)	设置空调控制方式
	setAcCycleMode(int setSource,int	设置空调循环方式
int	mode)	以直工则相外刀以
	setAcWindMode(int setSource,int	设置空调出风模式
int	mode)	以巨工列山//√/大小
int	setAcDefrostState(int setSource,int	设置前后排空调除霜状态
	area,int state)	<u>以自由加州工业</u> 则的相似的
int	setAcWindLevel(int setSource ,int level)	设置空调风量档位
int	setAcTemperature(int type,int value,int	设置空调温度
int	tempSource, int unit)	以且工例/四次
int	setAcTemperatureControlMode(int	设置主副驾驶温度分控方
int	setSource,int mode)	式
int	setAcVentilationState(int setSource,int	设置驻车通风功能



	state)	
int	start(int setSource)	开启空调
int	startRearAc(int setSource)	开启后排空调
int	stop(int setSource)	关闭空调
int	stopRearAc(int setSource)	关闭后排空调
	registerListener(AbsBYDAutoAcListener	
void	l)	监听注册接口
	unregisterListener(AbsBYDAutoAcListe	
void	ner l)	取消注册接口

6.6.2 获取 A/C (压缩机)状态

API 名称		int getAcCompressorMc	ode()	
接口描述	1	获取 A/C (压缩机)状态		
输入参数		无		
返回值				
数据类型	名称 值 描述			
	A/C (压缩	AC_COMPRESSOR_OFF	压缩机关	
int	机)状态	AC_COMPRESSOR_ON	压缩机开	

6.6.3获取 A/C (压缩机)手动标志

API 名称	int getAcCompressorManualSign()
接口描述	获取 A/C (压缩机)手动标志



输入参数	无			
	返回值			
数据类型	名称	值	描述	
	A/C (压缩	AC_COMPRESSOR_MANUAL_SIG	自动控制	
		N_OFF	日初江市	
int	机)手动标	AC_COMPRESSOR_MANUAL_SIG		
	志	N_ON	手动控制	

6.6.4 获取风量手动标志

API 名称	int getAcWindLevelManualSign()				
接口描述		获取风量手动标志			
输入参数		无			
	返回值				
数据类型	名称	值	描述		
		AC_WINDLEVEL_MANUAL_SIGN_	-		
N.	风量手动标	OFF	自动控制		
int	志	AC_WINDLEVEL_MANUAL_SIGN_	工工社会出		
-		ON	手动控制		

6.6.5 获取出风模式手动标志

API 名称	int getAcWindModeManualSign()
接口描述	获取出风模式手动标志



输入参数	无						
	返回值						
数据类型	名称	名称 值 描述					
		AC_WINDMODE_MANUAL_SIGN_	▞ ▔▀▙▞▞▞▍				
int	出风模式手动标志	OFF	自动控制				
		AC_WINDMODE_MANUAL_SIGN_					
		ON	手动控制				

6.6.6 获取空调开启状态

API 名称	int getAcStartState()				
接口描述		获取空调开启状态			
输入参数		无			
	Ting.	返回值			
数据类型	名称	值	描述		
. /	空调开启状	AC_POWER_ON	开启		
int	态	AC_POWER_OFF	关闭		

6.6.7 获取空调控制方式

API 名称	int getAcControlMode()			
接口描述	获取空调控制方式			
输入参数	无			
返回值				



数据类型	名称	值	描述
	空调控制方	AC_CTRLMODE_MANUAL	手动
int	式	AC_CTRLMODE_AUTO	自动

6.6.8 获取空调循环方式

API 名称	int getAcCycleMode()				
接口描述		获取空调循环方式			
输入参数		无			
	返回值				
数据类型	名称	值	描述		
	空调循环方	AC_CYCLEMODE_OUTLOOP	外循环		
int	式	AC_CYCLEMODE_INLOOP	内循环		

6.6.9 获取空调出风模式

API 名称	int getAcWindMode()				
接口描述	获取空调出风模式				
输入参数	2	无			
1	返回值				
数据类型	名称 值 描述				
	空调出风模	AC_WINDMODE_FACE	1: 吹面		
int	式	AC_WINDMODE_FACEFOOT	2: 吹面吹脚		



	AC_WINDMODE_FOOT	3: 吹脚
	AC_WINDMODE_FOOTDEFROST	4:吹脚除霜
	AC_WINDMODE_DEFROST	5: 除霜
	AC_WINDMODE_FACEFOOTDEFR	
	OST	6: 吹面+吹脚+除霜
	AC_WINDMODE_FACEDEFROST	7: 吹面+除霜

6.6.10 获取空调除霜状态

			4 7			
API 名称	int getAcDefrostState(int area)					
接口描述		获取空调除霜状态				
			输入参数	^		
数据类型	字段	名称	值	描述		
	area		AC_DEFROST_AREA_FRONT	前除霜开关		
int		除霜开关位置	AC_DEFROST_AREA_REAR	后除霜开关		
			返回值			
数据类型 描述						
	CCV4 44-0A = V 44-1N-1		AC_DEFROST_STATE_OFF	除霜状态关闭		
int	外选中	的除霜开关的状态	AC_DEFROST_STATE_ON	除霜状态开启		
备注	备注 后除霜仅在有后除霜功能配置的车型上适用					

6.6.11获取空调风量档位

API 名称	int getAcWindLevel()				
接口描述	获取风量档位				
输入参数	无				
	返回值				
数据类型	名称 值 描述				
int	风量档位	AC_WINDLEVEL_0	0~7,一共8个等级的风量		



	AC_WINDLEVEL_1	
	AC_WINDLEVEL_2	
	AC_WINDLEVEL_3	
	AC_WINDLEVEL_4	
	AC_WINDLEVEL_5	
	AC_WINDLEVEL_6	
	AC_WINDLEVEL_7	1/1/2°

6.6.12 获取空调各区域温度

API 名称		int getTemprature(int area)				
接口描述		获取空调各区域温度				
	输入参数					
数据类型	字段	字段 名称 值 描述				
	area	rea 区域	AC_TEMPERATURE_MAIN	主驾驶温度		
1			AC_TEMPERATURE_DEPUTY	副驾驶温度		
int			AC_TEMPERATURE_REAR	后空调		
1			AC_TEMPERATURE_OUT	车外温度		
	返回值					
数据类型	类型 名称 描述			描述		



			车内主副驾温度:
		[AC_TEMP_IN_CELSIUS_MIN,	1)当温度单位为摄氏温
		AC_TEMP_IN_CELSIUS_MAX] °C;	度℃时(默认):范围:
		[AC_TEMP_IN_FAHRENHEIT_MIN,	[17, 33];
		AC_TEMP_IN_FAHRENHEIT_MAX]	2)当温度单位为华氏温度
		°F	°F时:
	获取空调各 区域温度		范围[64, 91]。
int		[AC_TEMP_OUT_CELSIUS_MIN,	
		AC_TEMP_OUT_CELSIUS_MAX]	车外温度:
		°C;	1)摄氏单位:温度范围[-
		[AC_TEMP_OUT_FAHRENHEIT_MI	40°C, 50°C];
		N,	2)华氏单位: [-40℉,122
	100	AC_TEMP_OUT_FAHRENHEIT_MA	°F]。
		X] °F	

6.6.13 获取默认温度单位

	Jan 10 700.					
API 名称	int getTemperatureUnit()					
接口描述		获取默认温度单位				
输入参数	无					
	返回值					
数据类型	名称 值 描述					



	默认温度单	AC_TEMPERATURE_UNIT_OF	华氏℉
int	位	AC_TEMPERATURE_UNIT_OC	摄氏℃

6.6.14 获取主副驾驶温度分控方式

API 名称	int getAcTemperatureControlMode()					
接口描述		获取主副驾驶温度分控方式				
输入参数		无				
		返回值				
数据类型	名称	值	描述			
	主副驾驶温	AC_TEMPCTRL_SEPARATE_OFF	不分控			
int	度分控方式	AC_TEMPCTRL_SEPARATE_ON	分控			

6.6.15 获取通风功能设置

API 名称	int getAcVentilationState()					
接口描述	2:38	获取通风功能设置				
输入参数		无				
\mathcal{V}	返回值					
数据类型	名称	值	描述			
	通风功能设	AC_VENTILATION_STATE_ON	打开			
int	置	AC_VENTILATION_STATE_OFF	关闭			



6.6.16 获取后空调开启状态

API 名称	int getRearAcStartState()					
接口描述	获取后空调开启状态					
输入参数		无				
	返回值					
数据类型	名称	描述				
	后空调开启	AC_POWER_ON	开启			
int	状态 AC_POWER_OFF 关闭					
备注	适用于有后排空调配置的车型					

6.6.17设置空调控制方式

API 名称		int setAcControlMode(int setSource,int mode)					
接口描		-1///					
述			设置空调控制方式				
	11		输入参数				
数据类							
型	字段	名称	值	描述			
	setSourc		AC_CTRL_SOURCE_VOICE	声控			
int	е	控制来源	AC_CTRL_SOURCE_UI_KEY	触屏/按键			
		空调控制	AC_CTRLMODE_MANUAL	手动			
int	mode	方式	AC_CTRLMODE_AUTO	自动			



返回值					
数据类					
型	名称	值	描述		
int	设置状态后的结果	AC_COMMAND_SUCCESS	成功		
		AC_COMMAND_INVALID_VA	T->6/#		
		LUE	无效值		
		AC_COMMAND_TIMEOUT	超时		
		AC_COMMAND_BUSY	忙碌		
		AC_COMMAND_FAILED	失败		

6.6.18 设置空调循环方式

API 名称		int setAcCycleMode(int setSource,int mode)					
接口描述		设置空调循环方式					
	Till State		輸入参数				
数据类型	字段	名称	值	描述			
M.	setSo	控制来源	AC_CTRL_SOURCE_VOICE	声控			
int	urce		AC_CTRL_SOURCE_UI_KEY	触屏/按键			
		空调循环	AC_CYCLEMODE_INLOOP	内循环			
int	mode	方式	AC_CYCLEMODE_OUTLOOP	外循环			
返回值							
数据类型	:	名称	值	描述			



		AC_COMMAND_SUCCESS	成功
		AC_COMMAND_INVALID_VA	
	设置状态后的结	LUE	无效值
int	果	AC_COMMAND_TIMEOUT	超时
		AC_COMMAND_BUSY	忙碌
		AC_COMMAND_FAILED	失败

6.6.19 设置空调出风模式

API 名称		int setAcWindMode(int setSource,int mode)						
接口描述		设置空调出风模式						
			输入参数					
数据类型	字段	名称	值	描述				
	setSo	控制来	AC_CTRL_SOURCE_VOICE	声控				
int	urce	源	AC_CTRL_SOURCE_UI_KEY	触屏/按键				
			AC_WINDMODE_FACE	1: 吹面				
M.	mode	mode 模式	AC_WINDMODE_FACEFOOT	2: 吹面吹脚				
			AC_WINDMODE_FOOT	3: 吹脚				
int			AC_WINDMODE_FOOTDEFROST	4:吹脚除霜				
			AC_WINDMODE_DEFROST	5: 除霜				
			AC_WINDMODE_FACEFOOTDEF	6: 吹面+吹脚+除霜				
			ROST	7: 吹面+除霜				



		AC_WINDMODE_FACEDEFROST					
返回值							
数据类型	名称	值	描述				
		AC_COMMAND_SUCCESS	成功				
		AC_COMMAND_INVALID_VALU	731.00				
	设置状态后的	E	无效值				
int	结果	AC_COMMAND_TIMEOUT	超时				
		AC_COMMAND_BUSY	忙碌				
		AC_COMMAND_FAILED	失败				

6.6.20 设置前后排空调除霜状态

API 名称		int setAcDefrostState(int setSource,int area, int state)					
接口描述		设置前后排空调除霜状态					
	100	XX ⁻	输入参数				
数据类型	字段	名称	值	描述			
W.	setSo	控制来	AC_CTRL_SOURCE_VOICE	声控			
int	urce	源	AC_CTRL_SOURCE_UI_KEY	触屏/按键			
		4-1-2	AC_DEFROST_AREA_FRONT	前除霜开关			
int	area	区域	AC_DEFROST_AREA_REAR	后除霜开关			
int	state		AC_DEFROST_STATE_ON	除霜状态开启			
		状态	AC_DEFROST_STATE_OFF	除霜状态关闭			



返回值					
数据类型	名称	值	描述		
		AC_COMMAND_SUCCESS	成功		
	AC_COMMAND_INVALID_VALU				
	设置状态后的	E	无效值		
int	结果	AC_COMMAND_TIMEOUT	超时		
		AC_COMMAND_BUSY	忙碌		
		AC_COMMAND_FAILED	失败		
备注	后除霜仅在有后除霜功能配置的车型上适用				

6.6.21 设置空调风量档位

	1						
API 名称		int setAcWindLevel(int setSource ,int level)					
接口描述		-//	设置空调风量档位				
	i i i i i	XX_	输入参数				
数据类型	字段	名称	值	描述			
M.	setSo	D-	AC_CTRL_SOURCE_VOICE	声控			
int	urce	控制来源	AC_CTRL_SOURCE_UI_KEY	触屏/按键			
			AC_WINDLEVEL_1	 1 ~7,一共 7 个等级的风量 			
			AC_WINDLEVEL_2				
int	level	风量档位	AC_WINDLEVEL_3				
			AC_WINDLEVEL_4				



		AC_WINDLEVEL_5	
		AC_WINDLEVEL_6	
		AC_WINDLEVEL_7	
		返回值	
数据类型	名称	值	描述
		AC_COMMAND_SUCCESS	成功
		AC_COMMAND_INVALID_VA	T 並/古
	设置状态后的结	LUE	无效值
int	果	AC_COMMAND_TIMEOUT	超时
		AC_COMMAND_BUSY	忙碌
		AC_COMMAND_FAILED	失败

6.6.22 设置空调温度

API 名称	int setAcTemperature(int type,int value,int tempSource,int unit)						
接口描述	$\langle 1 \rangle$	设置空调温度					
M.		输入参数					
数据类型	字段	名称	值	描述			
			AC_TEMPERATURE_MAIN	主驾驶空调			
int	type	空调位置	AC_TEMPERATURE_DEPUT Y	副驾驶空调			
			AC_TEMPERATURE_MAIN_	主副驾 (不分控时有效)			



			DEPUTY		
			AC_TEMPERATURE_REAR	后空调	
			[AC_TEMP_IN_FAHRENHEIT		
			_MIN,		
			AC_TEMP_IN_FAHRENHEIT		
			_MAX]°F		
int	value	温度值	或	华氏温度范围[64, 91]	
			[AC_TEMP_IN_CELSIUS_MI	摄氏温度范围[17, 33]	
			N,	16,4	
			AC_TEMP_IN_CELSIUS_MA		
			X]℃		
	tempSo	1240047	AC_CTRL_SOURCE_UI_KEY	触屏/按键	
int	urce	控制途径	AC_CTRL_SOURCE_VOICE	语音控制	
	~	1.02	AC_TEMPERATURE_UNIT_O		
			F	温度单位:华氏温度	
int	unit	温度单位	AC_TEMPERATURE_UNIT_O		
			С	温度单位:摄氏度	
返回值					
数据类型	名称		值	描述	
			AC_COMMAND_SUCCESS	成功	
int	设置状态	后的结果	AC_COMMAND_INVALID_	无效值	



	VALUE	
	AC_COMMAND_TIMEOUT	超时
	AC_COMMAND_BUSY	忙碌
	AC_COMMAND_FAILED	失败

6.6.23设置主副驾驶温度分控方式

API 名称		int setAcTemperatureControlMode(int setSource,int mode)				
接口描述			设置主副驾驶温度分控方	式		
			输入参数	DV		
数据类型	字段	名称	值	描述		
	setSou	12 40 -1 27	AC_CTRL_SOURCE_VOICE	声控		
int	rce	控制来源	AC_CTRL_SOURCE_UI_KEY	触屏/按键		
int	mode	node 分控方式	AC_TEMPCTRL_SEPARATE_OF	不分控		
	17		AC_TEMPCTRL_SEPARATE_ON	分控		
11.			返回值			
数据类型	名	名称	值	描述		
4			AC_COMMAND_SUCCESS	成功		
int		态后的结 果	AC_COMMAND_INVALID_VA	无效值		
	朱		AC_COMMAND_TIMEOUT	超时		



AC_COMMAND_BUSY	忙碌
AC_COMMAND_FAILED	失败

6.6.24设置驻车通风功能

API 名称		int setAcVentilationState(int setSource,int state)					
接口描述		设置驻车通风功能					
			输入参数				
数据类型	字段	名称	值	描述			
	setSou		AC_CTRL_SOURCE_VOICE	声控			
int	rce	控制来源	AC_CTRL_SOURCE_UI_KEY	触屏/按键			
) -	AC_VENTILATION_STATE_ON	打开			
int	state	通风功能	AC_VENTILATION_STATE_OFF	关闭			
		-//	返回值				
数据类型	ź	3称	值	描述			
			AC_COMMAND_SUCCESS	成功			
11.			AC_COMMAND_INVALID_VA	T-24/t+			
	设置状态后的结 int 果		LUE	无效值			
int			AC_COMMAND_TIMEOUT	超时			
			AC_COMMAND_BUSY	忙碌			
			AC_COMMAND_FAILED	失败			



6.6.25开启空调

API 名称		int start(int setSource)					
接口描述		开启空调					
			输入参数				
数据类型	字段	名称	值	描述			
_	setSou		AC_CTRL_SOURCE_VOICE	声控			
int	控制来源 rce		AC_CTRL_SOURCE_UI_KEY	触屏/按键			
			返回值	Kr			
数据类型	ء	3称	值	描述			
			AC_COMMAND_SUCCESS	成功			
			AC_COMMAND_INVALID_VA	无效值			
int	开启的	≌调结果	AC_COMMAND_TIMEOUT	超时			
4			AC_COMMAND_BUSY	忙碌			
			AC_COMMAND_FAILED	失败			

6.6.26 关闭空调

API 名称		int stop(int setSource)				
接口描述		关闭空调				
	输入参数					
数据类型	字段	名称	值	描述		



	setSou	控制来源	AC_CTRL_SOURCE_VOICE	声控	
int	rce		AC_CTRL_SOURCE_UI_KEY	触屏/按键	
			返回值		
数据类型	名称		值	描述	
		关闭空调结果	AC_COMMAND_SUCCESS	成功	
	关闭空			AC_COMMAND_INVALID_VA	
int			LUE	无效值	
			天闭空调结果	AC_COMMAND_TIMEOUT	超时
			AC_COMMAND_BUSY	忙碌	
			AC_COMMAND_FAILED	失败	

6.6.27开启后排空调

API 名称		int startRearAc(int setSource)		
接口描述		开启后排空调		
	输入参数			
数据类型	字段	名称	值	描述
	setSou		AC_CTRL_SOURCE_VOICE	声控
int	rce	控制来源	AC_CTRL_SOURCE_UI_KEY	触屏/按键
返回值				
数据类型	名称		值	描述
int	开启后排空调结		AC_COMMAND_SUCCESS	成功



	果	AC_COMMAND_INVALID_VA	
		LUE	无效值
		AC_COMMAND_TIMEOUT	超时
		AC_COMMAND_BUSY	忙碌
		AC_COMMAND_FAILED	失败
备注	适用于有后排空调配置的车型		

6.6.28关闭后排空调

API 名称	int stopRearAc(int setSource)			
接口描述		关闭后排空调		
			输入参数	
数据类型	字段	名称	值	描述
	setSou	12414	AC_CTRL_SOURCE_VOICE	声控
int	rce	控制来源	AC_CTRL_SOURCE_UI_KEY	触屏/按键
		V	返回值	
数据类型	数据类型 名称		值	描述
)°		AC_COMMAND_SUCCESS	成功
			AC_COMMAND_INVALID_VA	- 1/4
int	关闭后排空调结		LUE	无效值
	اِ	果	AC_COMMAND_TIMEOUT	超时
			AC_COMMAND_BUSY	忙碌



		AC_COMMAND_FAILED	失败
备注	适用于有后排空调配置的车型		

6.6.29 监听方法

	类 AbsBYDAutoAcLis	stener	_
public abstrac	t class AbsBYDAutoAcListener		
	方法概要		()
限定符和返回			7
类型	方法	描述	输入
	on Ac Compressor Manual Sign Chan	监听 A/C(压缩机)手	
void	ged(int sign)	动标志的变化	手动标志
	on Ac Compressor Mode Changed (i	监听 A/C(压缩机)状	
void	nt mode)	态的变化	压缩机状态
void	onAcCtrlModeChanged(int mode)	监听控制方式的变化	控制方式
void	onAcCycleModeChanged(int mode)	监听循环方式的变化	循环方式
7			区域 (前除霜
		监听空调除霜方式的	开关, 后除霜
	onAcDefrostStateChanged(int	变化	开关)
void	area,int state)		开关的状态
void	onAcRearStarted()	监听后空调开启状态	无



	T	
onAcRearStoped()	监听后空调关闭状态	无
onAcStarted()	监听空调开启状态	无
onAcStoped()	监听空调关闭状态	无
on Ac Ventilation State Changed (int	监听通风功能设置的	
state)	变化	状态
onAcWindLevelChanged(int level)	监听风量档位的变化	风量档位
onAcWindLevelManualSignChang	监听风量手动标志的	
ed(int sign)	变化	手动标志
on Ac Wind Mode Changed (int	 佐元山図井子65赤/レ	Д Д Д Д Д Д Д Д Д Д Д Д Д Д Д Д Д Д Д
mode)	监听正从候式的变化	
on Ac Wind Mode Manual Sign Chan	监听出风模式手动标	手动标志
ged(int sign)	志的变化	」 4/J/初小心
on Ac Wind Mode Shown State Chan	监听空调出风模式显	\ \\ \
ged(int state)	示状态	状态
onTemperatureChanged(int	监听各区域温度的变	空调区域
area,int value)	化	温度值
onTemperatureUnitChanged(int		泪 在
unit)	西则·迪皮牛121119党16	画
输入值的取值范围参见 get 函数的具体	本描述	
	onAcStarted() onAcStoped() onAcVentilationStateChanged(int state) onAcWindLevelChanged(int level) onAcWindLevelManualSignChang ed(int sign) onAcWindModeChanged(int mode) onAcWindModeManualSignChan ged(int sign) onAcWindModeShownStateChan ged(int state) onTemperatureChanged(int area,int value) onTemperatureUnitChanged(int unit)	onAcStarted() 监听空调开启状态 onAcStoped() 监听空调关闭状态 onAcVentilationStateChanged(int 监听通风功能设置的 state) 变化 onAcWindLevelChanged(int level) 监听风量档位的变化 onAcWindLevelManualSignChang 监听风量手动标志的 ed(int sign) 变化 onAcWindModeChanged(int 监听出风模式的变化 mode) onAcWindModeManualSignChan 监听出风模式手动标 ged(int sign) 志的变化 onAcWindModeShownStateChan 监听空调出风模式显 ged(int state) 示状态 onTemperatureChanged(int 监听各区域温度的变 area,int value) 化 onTemperatureUnitChanged(int 监听温度单位的变化



6.7 空气质量类

6.7.1 方法概要

类 BYDAutoPM2p5Device			
public class BYDAutoPN	M2p5Device		
	方法概要		
限定符和返回类型	方法	描述	
static BYDAutoPM2p5	^	112	
Device	getInstance(Context con)	获取实例	
int	getPM2p5OnlineState()	获取 PM2.5 是否在线	
int[]	getPM2p5CheckState()	获取车内/车外检测状	
		态	
int[]	getPM2p5Level()	获取车内/车外 PM2.5	
		等级	
int[]	getPM2p5Value()	获取车内/车外 PM2.5	
		数值	
void	registerListener(AbsBYDAutoPM2p5Li		
1	stener I)	监听注册接口	
void	unregisterListener(AbsBYDAutPM2p5		
	Listener I)	取消注册接口	
备注	适用于有 PM2.5 功能配置的车型		



6.7.2 获取 pm2.5 是否在线

API 名称	int getPM2p5OnlineState()				
接口描述		获取 PM2.5 是否在线			
输入参数		无			
	返回值				
数据类型	名称	值	描述		
int	状态	PM2P5_ONLINE_STATE_NULL	不在线(未检测到/没有该模 块)		
		PM2P5_ONLINE_STATE_ON	在线(检测到/有该模块)		
备注	注 ON 档电用来判断 pm2.5 模块是否检测到。				

6.7.3 获取车内/车外检测状态

API 名称	int[] getPM2p5CheckState()			
接口描述	获取车内/车外检测状态			
输入参数		无		
返回值				
数据类型	名称	值	描述	
4		PM2P5_STATE_ON/PM2P5_ST	开启/关闭	
:417	int[0]车内检测状态	ATE_OFF		
int[]		PM2P5_STATE_ON/PM2P5_ST	开启/关闭	
	int[1]车外检测状态	ATE_OFF		



6.7.4 获取车内/车外 PM2.5 等级

API 名称	int[] getPM2p5Level()				
接口描述		获取车内/车外 PM2.5 等级			
输入参数		无			
		返回值			
数据类型	名称	值	描述		
	int[0]车内 PM2.5	PM2P5_LEVEL_INVALID	0x0: 无效		
		PM2P5_LEVEL_EXCELLENT	0x1:优		
		PM2P5_LEVEL_GOOD	0x2:良		
		PM2P5_LEVEL_LOW_GRADE	0x3: 轻度		
int[]	等级	PM2P5_LEVEL_MIDDLE	0x4: 中度		
	-4/	PM2P5_LEVEL_HEAVY	0x5: 重度		
		PM2P5_LEVEL_SERIOUS	0x6: 严重		
	int[1]车外 PM2.5	同上	同上		
	等级	PLE.			

6.7.5 获取车内/车外 PM2.5 数值

API 名称	int[] getPM2p5Value()	
接口描述	获取车内/车外 PM2.5 数值	
输入参数	无	
返回值		



数据类型	名称	值	描述
	int[0]车内 PM2.5	[PM2P5_VALUE_MIN ,PM2P5_	{0 3000} ug/m³
	数值	VALUE_MAX]	
int[]	int[1]车外 PM2.5		同上
	数值	同上	

6.7.6 监听方法

监听类 AbsBYDAutoPM2p5Listener			
public class AbsBYDAutoPM2p5Listener			
当监听的对象数值发生变化时,推送给用户			
方法概要			
返回类		III.) IS	
型	方法	描述	输入
void	onPM2p5CheckStateChanged(int state_in,in	监听车内/车外 PM2.5	
	state_out))	检测状态	
void	on PM2 p5 Level Changed (int level_in, in	监听车内/车外 PM2.5	
	level_out)	等级	
void	onPM2p5ValueChanged(int value_in,in	:监听车内/车外 PM2.5	
	value_out)	数值	
备注	输入值的取值范围参见 get 函数的具体描述		



6.8 发动机类

6.8.1 方法概要

类BYDAutoEngineDevice			
public class BYDAutoEngineDevice			
	方法概要		
限定符和返回类型	方法	描述	
static BYDAutoEngine			
Device	getInstance(Context con)	获取实例	
double	getEngineDisplacement()	获取发动机排量	
String	getEngineCode()	获取发动机型号	
		获取功率(发动机和电	
int	getEnginePower()	机)	
int	getEngineSpeed()	获取发动机转速	
int	getEngineCoolantLevel()	获取冷却液位	
int	getOilLevel()	获取燃油油位信号	
void	registerListener(AbsBYDAutoEngineLi		
	stener l)	监听注册接口	
void	unregisterListener(AbsBYDAutEngineL	邢兴:十四·李二	
	istener I)	取消注册接口	



6.8.2 获取发动机排量

API 名称	double getEngineDisplacement()		
接口描述	·····································		
输入参数	无		
	返回值		
数据类型	名称 值 描述		描述
	[ENGINE_DISPLACEMENT_MI		112
double	排量	N, ENGINE_DISPLACEMENT_MA X]	[0.0,25.5]L

6.8.3获取发动机型号

API 名称 String getEngineCode()				
接口描述	获取发动机型号			
输入参数		无		
V	返回值			
数据类型	名称 值		描述	
- 1	发动机型号	0x00: 371QA		
String		0x01: 473QB	刑日欠物	
		0x02: 473QC	型号名称	
		0x03: 473QD		



	///×+·· • 1.0.
0x04: 476ZQA	
0x05: 483QA	
0x06: 483QB	
0x07: 483QB CNG	
0x08: 487ZQA	
0x09: 488QA	(6)
0x0A:4G15	112
0x0B:4G18	ŠT.
0x0C:4G69	3/4
0x0D:473QE	
0x0E:471ZQA	

6.8.4 获取功率

API 名称	int getEnginePower()		
接口描述	获取功率		
输入参数	无		
	返回值		
数据类型	名称 值		描述
	功率 (发动机和发	[ENGINE_POWER_MIN,	I 400 2001
int	电机)	ENGINE_POWER_MAX]	[-100,300]kw



6.8.5 获取发动机转速

API 名称	int getEngineSpeed()		
接口描述	获取发动机转速		
输入参数	无		
	返回值		
数据类型	名称	值	描述
	4454	[ENGINE_POWER_MIN,	10 00001
int	转速	ENGINE_POWER_MAX]	[0,8000]r/min

6.8.6 获取冷却液位

API 名称	int getEngineCoolantLevel()			
接口描述	获取冷却液位			
输入参数	无			
	返回值			
数据类型	名称			
1		ENGINE_COOLANT_LEVEL_LO		
		W	过低	
int	冷却液位	ENGINE_COOLANT_LEVEL_N	— N4	
		ORMAL	正常	

6.8.7 获取燃油油位信号

API 名称	PI 名称 int getOilLevel()
--------	-------------------------



接口描述	获取燃油油位信 号		
输入参数	无		
返回值			
数据类型	名称 值		描述
	油位	[ENGINE_OIL_MIN,	
int		ENGINE_OIL_MAX]	[0,254]

6.8.8 监听方法

	监听类 AbsBYDAutoEngineListener			
public	class AbsBYDAutoEngineListener	1116	*	
当监听的	り对象数值发生变化时,推送给用户			
	方法相	既要		
返回类 方法 描述 输 型				
void	onEngineSpeedChanged(int value)	监听发动机转速变 化	发动机转速	
void	on Engine Coolant Level Changed (int state)	监听冷却液位变化	过低 正常	
void	onOilLevelChanged(int value)	监听油位信 号 变化	油位值	
备注	注 输入值的取值范围参见 get 函数的具体描述			



6.9 变速箱、 制动类

6.9.1 方法概要

类BYDAutoGearboxDevice				
public classBYDAutoG	public classBYDAutoGearboxDevice			
	方法概要	\wedge		
限定符和返回类型	方法	描述		
staticBYDAutoGearbo				
xDevice	getInstance(Context con)	获取实例		
String	getGearboxCode()	获取变速箱型号		
int	getGearboxType()	获取变速箱类型		
int		获取自 动变 速箱档		
getGearboxAutoModeType()		位		
int	int			
	getGearboxManualModeLevel()	位		
int	getBrakeFluidLevel()	获取制动液位信号		
int	getParkBrakeSwitch()	获 取驻 车制 动开 关		
		状态		
int	getBrakePedalState()	获取制动踏板状态		
void	registerListener(AbsBYDAutoGearboxList			
	ener l)	监听注册接口		
void	unregisterListener(AbsBYDAutGearboxLi	取消注册接口		



-		Τ
	stener I)	
	sterier ij	

6.9.2 获取变速箱型号

0.5.2			
API 名称	String getGearboxCode()		
接口描述	获取变速箱型号		
输入参数		无	
		返回值	1200
数据类型	名称	值	描述
String	变速箱型号	0x00: DABS15-41 0x01: F4A4 0x02: F4A4B 0x03: VT2-04O 0x04: SSG 0x05: 5T09 0x06: 5T14 0x07: 5T19 0x08: 5T19-1	型号名称
		0x09: 5RT10 0x0A: 5RT14	
		0x0B: 6T25 0x0C: 6DT25	



0x0D: 6DT33 0x0E: 6T18	
Ox0F: 6DT35	
X	1/2

6.9.3 获取变速箱类型

API 名称	int getGearboxType()			
接口描述		变速箱类型		
输入参数	7-17	无		
		返回值		
数据类型	名称	值	描述	
W	3	GEARBOX_TYPE_MT	MT,手动档	
V		GEARBOX_TYPE_AMT	AMT,半自动变速箱	
int	变速箱类型	GEARBOX_TYPE_AT	AT ,自动变速	
		GEARBOX_TYPE_CVT	CVT,无极变速	
		GEARBOX_TYPE_DCT	DCT,双离合变速器	



6.9.4 获取自动变速箱档位

API 名称	intgetGearboxAutoModeType()			
接口描述	获取自动变速箱档位			
输入参数	无			
	返回值			
数据类型	名称 值		描述	
	自动变速箱档位	GEARBOX_AUTO_MODE_P	P - 自动档驻车	
		GEARBOX_AUTO_MODE_R	R - 自动档倒车	
		GEARBOX_AUTO_MODE_N	N - 自动档空档	
int		GEARBOX_AUTO_MODE_D	D - 自动档前进	
		GEARBOX_AUTO_MODE_S	S	
		GEARBOX_AUTO_MODE_M	М	

6.9.5 获取手动变速箱档位

API 名称	intgetGearboxManualModeLevel()			
接口描述	获取手动变速箱档位			
输入参数	无			
	返回值			
数据类型	名称 值 描述			
		GEARBOX_REAL_LEVEL_D	前进挡	
int	手动变速箱档位	GEARBOX_REAL_LEVEL_R	倒档	



		GEARBOX_REAL_LEVEL_N	空档
备注	目前只返回 R、N 档	, D 档预留	

6.9.6 获取制动液位信号

API 名称	int getBrakeFluidLevel()		
接口描述	获取制动液位信号		
输入参数	无		
返回值			
数据类型	名称	值	描述
		GEARBOX_BRAKE_FLUID_LEV EL_LOW	过低
int	制动液位	GEARBOX_BRAKE_FLUID_LEV EL_NORMAL	正常

6.9.7 获取驻车制动开关状态

API 名称		int getParkBrakeSwitch()		
接口描述	获取驻车制动开关状态			
输入参数		无		
	返回值			
数据类型	名称	值	描述	
	╗╾ ╱╸ ╇╜═ ╸ ┰╸ ╱ ╶╏	GEARBOX_PARK_BREAK_SWIT	11+ /- /- /- /-	
int	驻车制动开关状态	CH_VALID	驻车有效	



GEARBOX_PARK_BREAK_SWIT	
CH_INVALID	宝车无效 。 。

6.9.8 获取制动踏板状态

API 名称	int getBrakePedalState()			
接口描述	获取制动踏板状态			
输入参数		无		
	返回值			
数据类型	名称	值	描述	
		GEARBOX_BREAK_PADAL_PRE	踩下	
int	制动踏板状态	GEARBOX_BREAK_PADAL_NO T_PRESS	未踩下	

6.9.9 监听方法

	监听类 AbsBYDAutoGearboxListener			
public	public class AbsBYDAutoGearboxListener			
当监听的	当监听的对象数值发生变化时,推送给用户			
方法概要				
返回类				
対 方法 描述 型 (1) (2) (2) (3) (3) (4) (4) (4) (4) (5) (4) (6) (4) (7) (4) (7) (4) (8) (4) (9) (4) (9) (4) (10) (4) (10) (4) (10) (4) (10) (4) (10) (4) (10) (4) (10) (4) (10) (4) <tr< td=""></tr<>				
void	onGearboxAutoModeTypeChanged(int level)	监听自动变速箱档位变化		



	on Gearbox Manual Mode Level Changed (int	
void	level)	监听手动变速箱档位变化
void	onBrakeFluidLevelChanged (int level)	监听制动液位状态变化
void	onParkBrakeSwitchChanged (int level)	监听驻车制动开关状态变化
void	onBrakePedalStateChanged (int level)	监听制动踏板状态变化
备注	输入值的取值范围参见 get 函数的具体描述	

6.10 门锁类

6.10.1 方法概要

类 BYDAutoDoorLockDevice			
public class BYDAutoDo	oorLockDevice		
	方法概要		
限定符和返回类型	方法	描述	
Static			
BYDAutoDoorLockDevi		获取实例	
ce	getInstance(Context con)		
int	getDoorLockStatus(int area)	获取各门锁状态	
	registerListener(AbsBYDAutoDoorL		
void	ockListener I)	监听注册接口	
	unregister Listener (Abs BYDAuto Do		
void	orLockListener I)	取消注册接口	



6.10.2 获取各门锁状态

API 名称	int getDoorLockStatus(int area)					
接口描述	获取各门锁状态					
			输入参数			
数据类型	字段名	名称	值	描述		
			DOOR_LOCK_AREA_LEFT_FRONT	左前门门锁状态		
			DOOR_LOCK_AREA_LEFT_REAR	左后门门锁状态		
			DOOR_LOCK_AREA_RIGHT_FRON	右前门门锁状态		
	area 门锁区域	area 门锁区域	DOOR_LOCK_AREA_RIGHT_REAR	右后门门锁状态		
int			DOOR_LOCK_AREA_BACK	后背门门锁状态		
			DOOR_LOCK_AREA_CHILDLOCK_	+ II		
			LEFT	左儿童锁位置信息		
			DOOR_LOCK_AREA_CHILDLOCK_	<i>十</i> 李炒/>		
		RIGHT	右儿童锁位置信息			
	返回值					
数据类型	名	名称	值	描述		
int	门锁状态		DOOR_LOCK_STATE_INVALID	无效		
			DOOR_LOCK_STATE_UNLOCK	解锁		
			DOOR_LOCK_STATE_LOCK	闭锁		



6.10.3 监听方法

类 AbsBYDAutoDoorLockListener						
public ab	public abstract class AbsBYDAutoDoorLockListener					
	方法概要	ī				
限定符和返回类型	方法	描述	输入			
void	onDoorLockStatusChanged(int area, int state)	状态的变化	区域 (左前门门锁, 左后门门锁, 右前门门锁, 右前门门锁, 后背门门锁, 后背门门锁, 左儿童锁, 右儿童锁, 右儿童锁)			
备注	输入值的取值范围参见 get 函数的具体	描述				

6.11 车灯类

6.11.1 方法概要

类 BYDAutoLightDevice				
public class BYDAutoLightDevice				
方法概要				
限定符和返回类型方法描述				



Static BYDAutoLightDevice	getInstance(Context con)	获取实例
int	getLightAutoStatus()	获取灯光 AUTO 档开关状态
int	getLightStatus(int type)	获取各车灯状态
int	getAFSSwitch()	获取 AFS 开关状态
void	registerListener(AbsBYDAutoLightListener I)	监听注册接口
void	unregisterListener(AbsBYDAutoLig htListener I)	取消注册接口

6.11.2 获取灯光 AUTO 档开关状态

API 名称	int getLightAutoStatus()			
接口描述	获取灯光 AUTO 档开关状态			
输入参数	无			
返回值				
数据类型	名称	值	描述	
	灯光 AUTO 档开关状	LIGHT_ON	打开	
int	态	LIGHT_OFF	关闭	

6.11.3 获取各车灯状态



接口描述	获取各车灯状态				
	输入参数				
数据类型	字段名	名称	值	描述	
			LIGHT_SIDE	小灯	
			LIGHT_LOW_BEAM	近光灯	
		type 车灯类型	LIGHT_HIGH_BEAM	远光灯	
	type		LIGHT_LEFT_TURN_SIGNAL	左转向灯	
int			LIGHT_RIGHT_TURN_SIGNAL	右转向灯	
			LIGHT_FRONT_FOG	前雾灯	
			LIGHT_REAR_FOG	后雾灯	
			UCUT FOOT	照脚灯(外后视镜	
		X	LIGHT_FOOT	下的灯)	
返回值					
数据类型	ź	3称	值	描述	
int			LIGHT_ON	打开	
	车灯状态		LIGHT_OFF	关闭	

6.11.4 获取 AFS 开关状态

API 名称	int getAFSSwitch()
接口描述	获取光照强度等级
输入参数	无



返回值				
数据类型	类型 名称 值 描述			
	AFS 开关状态	LIGHT_ON	打开	
int	(自适应转向大灯系	LIGHT OFF	· ·	
	统)	LIGHT_OFF	关闭	

6.11.5 监听方法

			1		
	监听类 AbsBYDAutoLightListener				
public	class AbsBYDAutoLightListe	ener	V		
当监听的	的对象数值发生变化时,推送	给用户			
	,	方法概要			
返回类	X	,)/			
型	方法	描述	输入		
void	onLightAutoSwitchOff()	监听灯光 AUTO 档开关关闭			
void	onLightAutoSwitchOn()	监听灯光 AUTO 档开关打开			
void	onLightOff(int type)	监听各车灯关闭	车灯类型		
void	onLightOn(int type)	监听各车灯打开	车灯类型		
void	onAFSSwitchStateChange	监听 AFS 开关状态变化			
	(int state)		打开/关闭		
备注	音注 输入值的取值范围参见 get 函数的具体描述				



6.12 安全带类

6.12.1 方法概要

类 BYDAutoSafetyBeltDevice			
public class BYDAutoSa	fetyBeltDevice		
	方法概要		
限定符和返回类型	方法	描述	
Static		1/>	
BYDAutoSafetyBeltDevi		获取实例	
ce	getInstance(Context con)		
		获取各位置安全带状	
int	getSafetyBeltStatus(int area)	态	
		获取各位置的乘客状	
int	getPassengerStatus(int area)	态	
	registerListener(AbsBYDAutoSafetyBel	佐元汁皿 按口	
void	tListener I)	监听注册接口	
live d	unregisterListener(AbsBYDAutoSafety	取光汁皿按口	
void	BeltListener I)	取消注册接口	

6.12.2 获取各位置安全带状态

API 名称	int getSafetyBeltStatus(int area)
接口描述	获取各位置安全带状态



	IIXT VIIO.			
	输入参数			
数据类型	字段名	名称	值	描述
			SAFETY_BELT_AREA_MAIN	驾驶员安全带
			SAFETY_BELT_AREA_DEPUTY	副驾驶安全带
			SAFETY_BELT_AREA_SECOND_RO	+W-> A ##
		ch ^ ## 6514*	W_SEAT_LEFT	左后二排安全带
int	area	安全带区域	SAFETY_BELT_AREA_SECOND_RO	* W - A ##
			W_SEAT_RIGHT	右后二排安全带
			SAFETY_BELT_AREA_SECOND_RO	1W-A#
			W_SEATMID	中后二排安全带
			返回值	
数据类型	往	3称	值	描述
		-1/	SAFETY_BELT_STATE_INVALID	无效
int	安全	带状态	SAFETY_BELT_STATE_UNLOCK	解开
		,	SAFETY_BELT_STATE_LOCK	扣上

6.12.3 获取各位置乘客状态

- ///				
API 名称		int getPassengerStatus(int area)		
接口描述	获取各位置乘客状态			
输入参数				
数据类型	字段名	名称	值	描述



				///×+*: 11.0.5
			SAFETY_BELT_PASSENGER_DEPUTY	副驾驶位座椅
			SAFETY_BELT_PASSENGER_SECOND	<u></u>
			_ROW_SEAT_LEFT	左后二排座椅
int	area	座椅区	SAFETY_BELT_PASSENGER_SECOND	
		域	_ROW_SEAT_RIGHT	右后二排座椅
			SAFETY_BELT_PASSENGER_SECOND	
			_ROW_SEAT_MID	中后二排座椅
			返回值	
数据类型	名和	弥	值	描述
			SAFETY_BELT_PASSENGER_STATE_IN VALID	无效
int	int 乘客状态		SAFETY_BELT_PASSENGER_STATE_S OMEBODY	有人
	,-X	17	SAFETY_BELT_PASSENGER_STATE_N	
			OBODY	无人
	Telegraph (1997)			

6.12.4 监听方法

类 AbsBYDAutoSafetyBeltListener					
public abstract class AbsBYDAutoSafetyBeltListener					
	方法概要				
限定符和返回	方法	描述	输入		



类型			
			区域 (驾驶员安全带,
	onSafetyBeltStatusChanged(in		副驾驶安全带,左后二
		监听各 位置安全	排安全带,右后二排安
Void	t area, int state)	带状态的变化	全带,中后二排安全带)
	int state)		安全带的状态(扣上,
		\triangle	解开)
			座椅区域(副驾驶位座
	on Passenger Status Changed (in	A RAIL VL P	椅, 左后二排座椅, 右
void	t area,		后二排座椅,中后二排
	int state)	状态	座椅)
	χ_{χ}		乘客状态(有人,无人)
备注	输入值的取值范围参见 get 函数的具体描述		

6.13雷达类

6.13.1 方法概要

类 BYDAutoRadarDevice				
public class BYDAutoRadarDevice				
方法概要				
限定符和返回类型	方法	描述		
Static	getInstance(Context con)	获取实例		



BYDAutoRadarDevice		
int	getRadarProbeState(int area)	获取各位置探头状态
		获取所有位置的探头状
int[]	getAllRadarProbeStates()	态
int	getReverseRadarSwitchState()	获取倒车雷达开关状态
	registerListener(AbsBYDAutoRadarL	
void	istener I)	监听注册接口
	unregisterListener(AbsBYDAutoRad	
void	arListener I)	取消注册接口

6.13.2 获取各位置探头状态

API 名称	int getRadarProbeState(int area)			
接口描述	获取各位置探头状态			
	ZX	7	输入参数	
数据类型	字段名	名称	值	描述
N.	area	探头位置	RADAR_AREA_LEFT	左探头
int			RADAR_AREA_LEFT_FRONT	左前探头
4	area 探头位	area 探头位置	RADAR_AREA_FRONT_LFET_MID	前左中位置探头
			RADAR_AREA_LEFT_REAR	左后探头
int			RADAR_AREA_RIGHT	右探头
			RADAR_AREA_RIGHT_FRONT	右前探头



		RADAR_AREA_FRONT_RIGHT_MI	前右中位置探头
		RADAR_AREA_RIGHT_REAR	右后探头
		返回值	
数据类型	名称	值	描述
		RADAR_COMMAND_INVALID_VA	
		LUE	无效
	探头状态	RADAR_PROBE_STATE_ABNORM AL	传感器异常
int		RADAR_PROBE_STATE_SAFE	安全
		RADAR_PROBE_STATE_GREEN	绿色警告
		RADAR_PROBE_STATE_YELLOW	黄色警告
		RADAR_PROBE_STATE_RED	红色警告

6.13.3 获取所有位置的探头状态

API 名称	int[] getAllRadarProbeStates()			
接口描述	<i>></i>	获取所有位置的探头状态		
输入参数		无		
		返回值		
数据类型	名称 值 描述			
int	探头状态	RADAR_COMMAND_INVALID_V	state[0]:左前探头状态	



ALUE	state[1] 右前探头状
RADAR_PROBE_STATE_ABNOR	态
MAL	state[2] 左后探头状
RADAR_PROBE_STATE_SAFE	态
RADAR_PROBE_STATE_GREEN	state[3] 右后探头状
RADAR_PROBE_STATE_YELLOW	态
RADAR_PROBE_STATE_RED	state[4] 左探头状态
	state[5] 右探头状态
	state[6] 前左中探头
RADAR_COMMAND_FAILED	状态
	state[7] 前右中探头
	状态

6.13.4 获取倒车雷达开关状态

API 名称	int getReverseRadarSwitchState()				
接口描述	获取倒车雷达开关状态				
输入参数	无				
	返回值				
数据类型	名称	描述			
	 	RADAR_REVERSE_SWITCH_ON	打开		
int	倒车雷达开关状态	RADAR_REVERSE_SWITCH_OFF	关闭		



6.13.5 监听方法

	类 AbsBYDAutoRadarListener				
public abs	public abstract class AbsBYDAutoRadarListener				
	方法概要	5			
限定符和返回类型	方法	描述	输入		
void	onRadarProbeStateChanged(int area,int state)	监听各 位置探头 状态的变化	area:探头位置(左探头,左前探头,前左中位置探头,左后探头,右前探头,前右中位置探头,右后探头,右后探头); state:探头状态(传感器异常,安全,绿色警告,黄,红)		
void	onReverseRadarSwitchStateChang ed(int state)		打开/关闭		
备注	输入值的取值范围参见 get 函数的具体	本描述			



6.14 充电类

6.14.1 方法概要

类BYDAutoChargingDevice					
public classBYDAu	public classBYDAutoChargingDevice				
	方法概要				
限定符和返回类型	方法	描述			
Static		2/12			
BYDAutoCharging	getInstance(Context con)	获取实例			
Device		7,			
int	getChargerFaultState()	获取车载充电器故障状态			
int	getChargerWorkState()	获取车载充电器工作状态			
double	getChargingCapacity()	获取本次累计总充电量			
int	getChargingType()	获取充电模式			
int[]	getChargingRestTime()	获取当前充满电剩余时间			
11,00	geteriargingicstrinie()	小时,分			
int	getChargingCapState(int type)	获取充电口盖开关状态			
int	getChargingPortLockRebackState()	获取交流充电口电锁执行			
int	geteriargingi ortzockiebackstate()	反馈			
int	getDischargeRequestState()	获取放电请求状态			
int	getChargerState()	获取充电器状态			
int	getChargingGunState()	获取充电枪连接状态			



double	getChargingPower()	获取充电功率
	gotPatton/ManagementDeviceState()	获取动力电池管理器与外
int	getBatteryManagementDeviceState()	接充电设备当前状态
int	getChargingScheduleEnableState()	获取定时充电功能状态
int	getChargingScheduleState()	获取预约充电状态
int	getChargingGunNotInsertedState()	获取充电枪未插提醒状态
in a m	a at Chausius a Cahadu la Tima a ()	获取预约充电倒计时(时,
int[]	getChargingScheduleTime()	分)
	registerListener(AbsBYDAutoCharging	
void	Listener I)	监听注册接口
	unregisterListener(AbsBYDAutoChargi	
void	ngListener I)	取消注册接口

6.14.2 获取车载充电器故障状态

API 名称	int getChargerFaultState()				
接口描述	获取车载充电器故障状态				
输入参数)"	无			
All		返回值			
数据类型	名称 值 描述				
	车载充电器故障状	CHARGING_FAULT_STATE_NORMAL	正常		
int	态	CHARGING_FAULT_STATE_MINOR	一般故障		



	CHARGING_FAULT_STATE_MAJOR	严重故障
--	----------------------------	------

6.14.3 获取车载充电器工作状态

API 名称	int getChargerWorkState()					
接口描述		获取车载充电器工作状态				
输入参数		无				
	返回值					
数据类型	名称	描述				
	车载充电器工作状	CHARGING_WORK_STATE_READY	准备就绪			
		CHARGING_WORK_STATE_START	充电开始			
int		CHARGING_WORK_STATE_FINISH	充电结束			
	态	CHARGING_WORK_STATE_TERMIN				
	-(/)	ATE	充电终止			

6.14.4 获取本次累计总充电量

API 名称	기 名称 double getChargingCapacity()			
接口描述	获取本次累计总充电量			
输入参数	无			
		返回值		
数据类型	名称	值	描述	
double	本次累计总充电量	[CHARGING_CAPACITY_MIN,CHAR	[0, 65.534]KWH	
double	一个八系川 心儿 心里	GING_CAPACITY_MAX]KWH	[0, 03.334][[0]	



6.14.5 获取充电模式

API 名称	int getChargingType()			
接口描述	获取充电模式			
输入参数		无		
	返回值			
数据类型	名称	描述		
		CHARGING_TYPE_DEFAULT	默认	
		CHARGING_TYPE_AC	交流充电	
int	充电模式	CHARGING_TYPE_VTOG	VTOG 充电	
		CHARGING_TYPE_GB_DC	国标直流充电	
	1	CHARGING_TYPE_GB_NON_DC	国标非直流充电	

6.14.6 获取当前充满电剩余时间(时,分)

API 名称	int[] getChargingRestTime()					
接口描述		获取当前充满电剩余时间				
输入参数	3/	无				
N		返回值				
数据类型	名称	描述				
	当前充满电剩余时	[REST_HOUR_MIN,REST_HOUR_MAX				
int[]	间: 小时]h	time[0]: [0,254]h			



当前充满电剩		time[1]:
间: 分钟	[MINUTE_MIN,MINUTE_MAX]min	[0,59]min

6.14.7 获取充电口盖开关状态

API 名称	int getChargingCapState(int type)			
接口描述	获取充电口盖开关状态			
			输入参数	-(>)
数据类型	字段名	名称	值	描述
			CHARGING_CAP_AC	交流充电口盖
int	type	/pe 充电口盖	CHARGING_CAP_DC	直流充电口盖
			返回值	
数据类型	名	3称	值	描述
	-1/2	CHARGING_CAP_STATE_ON	打开	
int	int 充电口盖开关状态		CHARGING_CAP_STATE_OFF	关闭
备注 直流充电口盖适用于纯电动车型				

6.14.8 获取交流充电口电锁执行反馈

API 名称	int getChargingPortLockRebackState()		
接口描述	获取交流充电口电锁执行反馈		
输入参数	无		
返回值			



数据类型	名称	值	描述	
	_	CHARGING_PORT_STATE_LOCK_FINI SH	开锁完成	
		CHARGING_PORT_STATE_LOCK_IN\ 充电口电锁执行状 ALID	开锁失效	
int	态反馈	CHARGING_PORT_STATE_UNLOCK_	闭锁完成	
			FINISH CHARGING_PORT_STATE_UNLOCK_I NVALID	闭锁失效

6.14.9 获取放电请求状态

API 名称	int getDischargeRequestState()		
接口描述	获取放电请求状态		
输入参数	-XX=(无	
		返回值	
数据类型	名称	值	描述
	放电请求状态	DISCHARGE_STATE_NON	无放电请求
A		DISCHARGE_STATE_HOUSEHOLD_A	
int		PPLIANCE	家用设备
		DISCHARGE_STATE_THREE_PHASE_E	
		QUIPMENT	三相设备



	DISCHARGE_STATE_THREE_PHASE_	
	VEHICLE	三相车辆
	DISCHARGE_STATE_POWER_SYSTE	
	M	电网
	DISCHARGE_STATE_SOCKET	车内插座
	DISCHARGE_STATE_SINGLE_PHASE_	
	VEHICLE	单相车辆

6.14.10获取充电器状态

API 名称	int getChargerState()				
接口描述	获取充电器状态				
输入参数	无				
	返回值				
数据类型	名称	值	描述		
		CHARGING_CHARGER_STATE_NOT_			
N.		CONNECTED	未连接		
int	充电器状态	CHARGING_CHARGER_STATE_CON			
V		NECTED	连接		

6.14.11 获取充电枪连接状态

API 名称	int getChargingGunState()
接口描述	获取充电枪连接状态



输入参数	无				
	返回值				
数据类型	名称	值	描述		
		CHARGING_GUN_STATE_CONN	<u> </u>		
		ECTED_NONE	未连接		
		CHARGING_GUN_STATE_CONN			
		ECTED_AC	交流充电枪连接		
		CHARGING_GUN_STATE_CONN			
int	充电枪连接状态	ECTED_DC	直流充电枪连接		
		CHARGING_GUN_STATE_CONN	交流与直流充电枪同		
		ECTED_AC_DC	时连接		
		CHARGING_GUN_STATE_CONN			
		ECTED_VTOL	VTOL 放电枪连接		

6.14.12 获取充电功率

API 名称		double getChargingPower()		
接口描述)"	获取充电功率		
输入参数	无			
	返回值			
数据类型	名称	值	描述	
double	充电功率(电池组当	[CHARGING_POWER_MIN,	[0,500]kW	



前总电压和总电流	CHARGING_POWER_MAX]	
相乘)		

6.14.13 获取动力电池管理器与外接充电设备当前状态

ADI ATA		t matDattamiNama manantDavisaCtata	0		
API 名称	int getBatteryManagementDeviceState()				
接口描述	获取动力电池管理器与外接充电设备当前状态				
输入参数	无				
	返回值				
数据类型	名称	值	描述		
		CHARGING_BATTERY_STATE_READY	准备就绪		
	动力电池管理器与 外接充电设备当前 状态	CHARGING_BATTERY_STATE_CHAR			
		GING	正在充电		
		CHARGING_BATTERY_STATE_CHAR			
		G_FINISH	充电结束		
		CHARGING_BATTERY_STATE_DISCH			
int		ARG	正在放电		
1		CHARGING_BATTERY_STATE_CHAR			
		G_TERMINATE	充电终止		
		CHARGING_BATTERY_STATE_BREAK			
		DOWN_C10	C10 充电柜故障		
		CHARGING_BATTERY_STATE_BREAK	充电枪连接故障		



DOWN_CHARGING_GUN	
CHARGING_BATTERY_STATE_BREAK	
DOWN_CHARGER	车载充电器故障
CHARGING_BATTERY_STATE_BREAK	
DOWN_AC	交流外充设备故障
CHARGING_BATTERY_STATE_SCHED	
ULE	预约等待
CHARGING_BATTERY_STATE_DISCH	
ARG_CBU	正在给整车放电
CHARGING_BATTERY_STATE_TIMEO	┡╾ ╱ ┍┪ ╵ ┎╵ ӡ╵ ╖╶┼╖╻┵
UT	与充电柜通讯超时
CHARGING_BATTERY_STATE_DISCH	<i>></i> /_+-/-
ARG_FINISH	放电结束
	CHARGING_BATTERY_STATE_BREAK DOWN_CHARGER CHARGING_BATTERY_STATE_BREAK DOWN_AC CHARGING_BATTERY_STATE_SCHED ULE CHARGING_BATTERY_STATE_DISCH ARG_CBU CHARGING_BATTERY_STATE_TIMEO UT CHARGING_BATTERY_STATE_DISCH

6.14.14 获取定时充电功能状态

API 名称	int getChargingScheduleEnableState()		
接口描述	获取定时充电功能状态		
输入参数	无		
数据类型	名称	值	描述
	定时充电功能状	CHARGING_STATE_ENABLE	定时充电允许
int	态	CHARGING_STATE_DISABLE	定时充电不允许



6.14.15 获取预约充电状态

API 名称	int getChargingScheduleState()			
接口描述	获取预约充电状态			
输入参数	无			
返回值				
数据类型	名称	值	描述	
int	预约充电状态	CHARGING_SCHEDULE_STATE_INVALI	无效态	
		CHARGING_SCHEDULE_STATE_CANCE L	取消预约充电	
		CHARGING_SCHEDULE_STATE_NONE	无预约充电	
		CHARGING_SCHEDULE_STATE_LOCAL	本地预约充电	
		CHARGING_SCHEDULE_STATE_REMOT	远程预约充电	

6.14.16 获取充电枪未插提醒状态

API 名称	int getChargingGunNotInsertedState()				
接口描述	获取充电枪未插提醒状态				
输入参数	无				
返回值					
数据类型	名称	值	描述		



	充电枪未插提醒状	CHARGING_GUN_STATE_ON	提醒
int	态	CHARGING_GUN_STATE_OFF	不提醒

6.14.17 获取预约充电倒计时(时,分)

API 名称	int[] getChargingScheduleTime()			
接口描述	获取预约充电倒计时			
输入参数	无			
	返回值			
数据类型	名称	值	描述	
	预约充电倒计时	[HOUR_MIN, HOUR_MAX]h	time[0]: 0-23 时	
int[]	预约充电倒计时 (分数)	[MINUTE_MIN, MINUTE_MAX]min	time[1]: 0-59 分	

6.14.18 监听方法

	类 AbsBYDAutoChargingListener				
public a	public abstract class AbsBYDAutoChargingListener				
	方法概要				
限定符					
和返回	方法	描述	输入		
类型					



			版本: V1.0.5
void	onChargerFaultStateChanged(int state)	监听车载充电 器故障状态的 变化	状态(正常,一般故障, 严重故障)
void	onChargerWorkStateChanged(int state)		状态 (准备就绪,充电 开始, 充电结束,充电终止)
void	onChargingCapacityChanged(double value)	监听本次累计 总充电量的变 化	电 量 ({0,131.07}KWH)
void	onChargingTypeChanged(int type)	监听充电模式的变化	充电模式 (默认,交流充电, VTOG 充电,国标直流充电, 国标非直流充电)
void	onChargingRestTimeChanged(int hour,int min)	监听当前充满 电剩余时间的 变化	hour :([0,254]h) min: ([0,59]min)
void	onChargingCapStateChanged(int type,int state)	监听充电口盖 开关状态的变 化	type(交流充电口盖,直 流充电口盖) state(打开,关闭)
void	on Charging Port Lock Reback State Chang	监听充电口电	state(开锁完成,开锁



			从4:V1.U.5
	ed(int state)	锁执行反馈的	失效,
		变化	闭锁完成,闭锁失效)
void	on Discharge Request State Changed (int state)	监听放电请求状态的变化	state(无放电请求,家用设备 三相设备,三相车辆,电 网 车内插座,单相车辆)
void	onChargerStateChanged(int state)	监听充电器状 态的变化	state(未连接,连接)
void	onChargingGunStateChanged(int state)	监听充电枪连	state(未连接,交流充电 枪连接,直流充电枪连 接,交流与直流充电枪 同时连接,VTOL 放电枪 连接)
void	on Charging Power Changed (double value)	监听充电功率的变化	功率
void	onBatteryManagementDeviceStateCha nged(int state)	管理器与外接 充电设备当前	state(准备就绪,正在充电,充电结束,正在放电,充电结束,正在放电,充电终止,C10 充电柜
		状态的变化	故障,充电枪连接故障, 车载充电器故障,交流



			外充设备故障,预约等
			待,正在给整车放电,与
			充电柜通讯超时,放电
			结束)
	onChargingScheduleEnableStateChang	监听定时充电	state(充电允许,充电
void	ed(int state)	功能状态的变	不允许)
		化	717611)
			state(无效态,取消预约
void	on Charging Schedule State Changed (int	监听预约充电	充电,
70.0	state)	状态的变化	无预约充电,本地预约
			充电,远程预约充电)
		」 监听充电枪未	
void		插提醒状态的	state(提醒,不提醒)
	ed(int state)	变化	
void	onChargingScheduleTimeChanged(int	监听预约充电	hour: 0-23 时
void	hour,int min)	倒计时的变化	min: 0-59 分
备注	输入值的取值范围参见 get 函数的具体描述		

6.15 轮胎类

6.15.1 方法概要



public class BYDAutoTy	vreDevice				
	方法概要				
限定符和返回类型	方法	描述			
Static BYDAutoTyreDevice	getInstance(Context con)	获取实例			
int	getTyreAirLeakState(int area)	获取各轮胎漏气状态			
		获取胎压系统电池电量状			
int	getTyreBatteryState()	态			
int	getTyrePressureState(int area)	获取各轮胎压力状态			
int	getTyrePressureValue(int area)	获取各轮胎压力值			
int	getTyreSignalState(int area)	获取各轮胎信号状态			
int	getTyreSystemState()	获取胎压系统状态			
int	getTyreTemperatureState()	获取胎压系统温度状态			
	registerListener(AbsBYDAutoTyreLi	监听注册接口			
void	stener I)				
1//2	unregister Listener (Abs BYDAuto Tyr	取消注册接口			
void	eListener I)				

6.15.2 获取各轮胎漏气状态

API 名称	int getTyreAirLeakState(int area)
接口描述	获取各轮胎漏气状态



输入参数				
数据类型	字段名	名称	值	描述
			TYRE_COMMAND_AREA_LEFT_FR	左前轮胎
			ONT	
			TYRE_COMMAND_AREA_RIGHT_	右前轮胎
		*^~\^\	FRONT	1 <u></u> 1 8 9 7 6 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
int	area	轮胎位置	TYRE_COMMAND_AREA_LEFT_RE	+=t\0
			AR	左后轮胎
			TYRE_COMMAND_AREA_RIGHT_	≠ ⊑₩₩
			REAR	右后轮胎
			返回值	
数据类型	名	3称	值	描述
		7/7	TYRE_AIR_LEAK_STATE_NORMAL	正常
			TYRE_AIR_LEAK_STATE_QUICK	快速漏气
int	探头	-状态	TYRE_AIR_LEAK_STATE_SLOW	慢速漏气
			TYRE_COMMAND_INVALID_VAL	T->L
1			UE	无效

6.15.3 获取胎压系统电池电量状态

API 名称	int getTyreBatteryState()
接口描述	获取胎压系统电池电量状态



输入参数	无			
返回值				
数据类型	名称 值 描述			
	胎压系统电池电量	TYRE_BATTERY_STATE_NORMAL	正常	
int	状态	TYRE_BATTERY_STATE_LOW	低电量	

6.15.4 获取各轮胎压力状态

API 名称	int getTyrePressureState(int area)						
接口描述		获取各轮胎压力状态					
			输入参数				
数据类型	字段名	字段名 名称 值					
	area 轮胎位置	X	TYRE_COMMAND_AREA_LEFT_FR ONT	左前轮胎			
		轮胎位置	TYRE_COMMAND_AREA_RIGHT_ FRONT	右前轮胎			
int			TYRE_COMMAND_AREA_LEFT_RE AR	左后轮胎			
4		TYRE_COMMAND_AREA_RIGHT_ REAR	右后轮胎				
	返回值						
数据类型	型 名称 描述			描述			



		TYRE_PRESSURE_STATE_NORMAL	正常
		TYRE_PRESSURE_STATE_OVERPR	
		ESSURE	过压
int	轮胎压力状态	TYRE_PRESSURE_STATE_UNDERP	
		RESSURE	欠压
		TYRE_COMMAND_INVALID_VAL	
		UE	无效

6.15.5 获取各轮胎压力值

API 名称	int getTyrePressureValue(int area)					
接口描述	获取各轮胎压力值					
		X	输入参数			
数据类型	字段名 名称 值			描述		
			TYRE_COMMAND_AREA_LEFT_FR	1 251 6004		
			ONT	左前轮胎		
M.			TYRE_COMMAND_AREA_RIGHT_	/ \		
			FRONT	右前轮胎		
int	area	轮胎位置 	TYRE_COMMAND_AREA_LEFT_RE	+=±\0\		
			AR	左后轮胎		
			TYRE_COMMAND_AREA_RIGHT_	≠ ⊏ + ∧ □ ∧		
		REAR	右后轮胎			



返回值				
数据类型	数据类型 名称 值 描述			
:4	松松压力店	[TYRE_PRESSURE_VALUE_MIN,TY	[0 4094]kpa	
int	轮胎压力值	RE_PRESSURE_VALUE_MAX]kpa	[0,4094]kpa	

6.15.6 获取各轮胎信号状态

API 名称	int getTyreSignalState(int area)				
接口描述		获取各轮胎信号状态			
			输入参数		
数据类型	字段名	描述			
	area 轮胎位置	. *	TYRE_COMMAND_AREA_LEFT_FR ONT	左前轮胎	
		a 轮胎位置	TYRE_COMMAND_AREA_RIGHT_ FRONT	右前轮胎	
int			TYRE_COMMAND_AREA_LEFT_RE AR	左后轮胎	
V		TYRE_COMMAND_AREA_RIGHT_ REAR	右后轮胎		
返回值					
数据类型		3称	值	描述	
int	轮胎信号状态		TYRE_SIGNAL_STATE_NORMAL	正常	



TYRE_SIGNAL_STATE_ERROR	异常
TYRE_COMMAND_INVALID_VAL	T > L
UE	无效

6.15.7 获取胎压系统状态

API 名称	int getTyreSystemState()				
接口描述	获取胎压系统状态				
输入参数	无				
		返回值			
数据类型	名称	值	描述		
	胎压系统状态	TYRE_SYSTEM_STATE_NORMAL	正常		
		TYRE_SYSTEM_STATE_SELF_CHEC KING	自检		
int		TYRE_SYSTEM_STATE_SIGNAL_A NOMAL	信号异常		
N.		TYRE_SYSTEM_STATE_BREAKDO WN	故障		
		TYRE_SYSTEM_STATE_MASKED	屏蔽状态		
备注	整车处于静止状态,胎压控制器会发送"系统屏蔽状态"				



6.15.8 获取胎压系统温度状态

API 名称	int getTyreTemperatureState()				
接口描述		获取胎压系统温度状态			
输入参数		无			
		返回值			
数据类型	名称 值		描述		
int	胎压系统温度状态	TYRE_TEMPERATURE_STATE_NORMA	正常		
		TYRE_TEMPERATURE_STATE_SUPER_	超高温		
		TYRE_TEMPERATURE_STATE_HIGH	高温		
		TYRE_TEMPERATURE_STATE_SLEEP	胎温显示休眠		

6.15.9 监听方法

	类 AbsBYDAutoTyreListener				
public al	bstract class AbsBYDAutoTyreListene	r			
	方法概要				
限定符					
和返回	方法	描述	输入		
类型					
void	onTyreAirLeakStateChanged(int	监听各轮胎漏气状	轮胎位置(左前轮		



			//////////////////////////////////////
	area,int state)	态的变化	胎,右前轮胎,左后
			轮胎,右后轮胎)
			状态(正常, 快速漏
			气,慢速漏气)
المنصد	onTyreBatteryStateChanged(int	监听胎压系统电池	
void	state)	电量状态的变化	状态(正常,低电量)
			轮胎位置(左前轮
	onTyroProssuroStatoChangod(int	医原复数形压 力快	胎,右前轮胎,左后
void		态的变化	轮胎,右后轮胎)
			压力状态(正常,过
			压,欠压)
			轮胎位置(左前轮
	1.00	监听 各轮 胎压 力值 的变化	胎,右前轮胎,左后
void			轮胎,右后轮胎)
4			压 力 值
			([0,4094]kpa)
			轮胎位置(左前轮
void	on Tyre Signal State Changed (int	监听 各轮 胎信 号状态的变化	胎,右前轮胎,左后
			轮胎,右后轮胎)
	area,int state)		信号状态(正常,异
			常)
	<u> </u>	l	



	and Trump Country and State Change and Single		系统状态(正常,自
void	onTyreSystemStateChanged(int	监听胎压系统状态	检,信号异常,故障,
	state)	的变化	屏蔽状态)
	onTyreTemperatureStateChanged(in		温度状态(正常,超
void			高温, 高温, 胎温显
	t state)	状态的变化	示休眠)
备注	输入值的取值范围参见 get 函数的具体	描述	17

6.16 仪表类

6.16.1 方法概要

类BYDAutoInstrumentDevice					
public classBYDAu	utoInstrumentDevice				
	方法概要				
限定符和返回类型	方法	描述			
staticBYDAutoIns trumentDevice	getInstance(Context con)	获取实例			
int	getMalfunctionInfo(int typeName)	获取故障提示信息			
int	getAlarmBuzzleState()	获取蜂鸣器状态			
int	getUnit(int unitName)	获取温度、气压、 油耗距 离、功率单位			
int	getMaintenanceInfo(int typeName)	获取保养时间或里程			



int	setUnit(int unitName, int unitValue)	设置温度、气压、油耗距
	,	离、功率单位
int	setMaintenanceInfo(int typeName, int	
	infoValue)	设置保养时间或里程
double	getExternalChargingPower()	获取外接充电量
	registerListener(AbsBYDAutoInstrumen	
		监听注册接口
void	tListener I)	1112
	unregister Listener (Abs BYD AutoInstrum	&LV
void	XA	取消注册接口
	entListener I)	

6.16.2 获取故障提示信息

	1					
API 名称		int getMalfunctionInfo(int typeName)				
接口描述	1		获取故障提示信息			
	ZX	7.	输入参数			
数据类型	字段名	名称	值	描述		
. \ '			MALFUNCTION_MACHINE_OIL_L			
	7/		OW_PRESSURE	机油压力低		
- N	4	N/ Til	MALFUNCTION_PARKING_BRAKE	驻车制动故障		
int	type	type	类型	MALFUNCTION_CHARGING_SYS		
			TEM	充电系统故障		
			MALFUNCTION_ENGINE	发动机故障		



				//////////////////////////////////////
			MALFUNCTION_ABS_SYSTEM	ABS 系统故障
			MALFUNCTION_ESP	ESP 故障
			MALFUNCTION_QUICK_AIR_LEA	
			K	快速漏气
			MALFUNCTION_HIGH_WATER_TE	
			MPERATURE	水温高
			MALFUNCTION_ELECTRIC_PARKI	1>
			NG_BRAKE	电子驻车故障
			MALFUNCTION_SRS	SRS 故障
			MALFUNCTION_EPS	EPS 故障
			MALFUNCTION_TYRE_PRESSURE	胎压故障报警
			MALFUNCTION_SVS	SVS
		7/	MALFUNCTION_HIGH_MOTOR_T	
	250		EMPERATURE	电机过高
			MALFUNCTION_BATTERY	电池故障
N	2)//		MALFUNCTION_HIGH_BATTERY_	
V			TEMPERATURE	电池过高
			MALFUNCTION_POWER_SYSTEM	
				动力系统故障
			MALFUNCTION_OK	OK 指示灯
			MALFUNCTION_EV	纯电模式指示



			MALFUNCTION_HEV	混合动力模式指示
			智能钥匙系统告警	
			MALFUNCTION_SMART_KEY	灯
			MALFUNCTION_FRONT_BELT	前排座椅安全带未
			WALFONCTION_FRONT_BELT	系提示
			返回值	
数据类型	名	称	值	描述
			DEVICE_HAS_THE_MALFUNCTIO	有故障/提示
int 故障		状态	DEVICE_NOT_HAS_THE_MALFU NCTION	无故障/提示

6.16.3 获取蜂鸣器状态

API 名称	int getAlarmBuzzleState()				
接口描述		获取蜂鸣器状态			
输入参数		无			
	返回值				
数据类型	名称	值	描述		
		INSTRUMENT_ALARM_BUZZL	nta nta		
int	蜂鸣器状态	E_ATATE_ON	鸣响		
		INSTRUMENT_ALARM_BUZZL	停止		



	_	
	E ATATE OFF	

6.16.4 获取温度、气压、 油耗距离、功率单位

API 名称	int getUnit(int unitName)						
接口描述		获取温度、气压、油耗距离、功率的单位					
			输入参数				
数据类型	字段名	名称	值	描述			
			TEMPERATURE_UNIT	温度			
			PRESSURE_UNIT	气压			
int	unitNam e	类型	FUEL_CONSUMPTION_AND_DIST ANCE_UNIT	油耗距离			
		X	POWER_UNIT	功率			
	1	-1/\	返回值				
数据类型	名	称	值	描述			
		(),	DEGREE_CENTIGRADE/DEGREE_F	单位(温度)			
M.		p. C.	AHRENHEIT	摄氏度/华氏温度			
)°			单位(气压)			
int	È	单位	BAR/PSI/KPA	0x1:bar			
			27.1.4.1.34.1.1.1	0x2:psi			
				0x3:kpa			
			L_P_100KM_AND_KM/	单位(油耗距离)			



		T
	KM_P_L_AND_KM/	0x1:L/100km,km
	MPG_GB_AND_MILE/	0x2:km/L,km
	MPG_US_AND_MILE/	0x3:mpg(GB),mil
	KWH_P_100KM_AND_KM/	e
	KWH_P_100MI_AND_MILE	0x4:mpg(US),mil
		e
		0x5:
		KWH/100km,
		km
		0x6:KWH/100Mi
		, mile
. X	\times , \times	单位(功率)
-1/_	KW/HP	0x1:kW
1755-		0x2:HP
毛距离单位目前仅支拉	持1、3、4	
	毛距离单位目前仅支	MPG_GB_AND_MILE/ MPG_US_AND_MILE/ KWH_P_100KM_AND_KM/ KWH_P_100MI_AND_MILE

6.16.5 获取保养时间/里程

API 名称	int getMaintenanceInfo(int typeName)				
接口描述	获取保养时间/里程				
	输入参数				
数据类型	字段名	名称	值	描述	



	typeNam	New	MAINTENANCE_TIME	保养时间
int	e	类型	MAINTENANCE_MILEAGE	保养里程
			返回值	
数据类型	名	称	值	描述
	保养时间		[MAINTENANCE_TIME_DAY_MIN,	10.7201T
			MAINTENANCE_TIME_DAY_MAX]	[0,720]天
			[MAINTENANCE_MILEAGE_KILO	\>`
int	保养里程		METER_MIN,	[0,20000]km
			MAINTENANCE_MILEAGE_KILOM	
			ETER_MAX]	

6.16.6 设置温度、气压、 油耗距离、功率单位

API 名称	int setUnit(int unitName,int unitValue)					
接口描述		设置温度、气压、油耗距离、功率单位				
		?)\	输入参数			
数据类型	字段名	名称	值	描述		
)"		TEMPERATURE_UNIT	温度		
A			PRESSURE_UNIT	气压		
int	unitNam	类型	FUEL_CONSUMPTION_AND_DISTA	∖ - h+T□⊏ - ⇒-		
	е		NCE_UNIT	油耗距离		
			POWER_UNIT	功率		



				/////////////////////////////////////
			DEGREE_CENTIGRADE/DEGREE_FAH	单位(温度)
			RENHEIT	摄氏度/华氏温度
				单位(气压)
			BAR/PSI/KPA	0x1:bar
			2, 4 . 3 4	0x2:psi
				0x3:kpa
				单位(油耗距离)
				0x1:L/100km,km
				0x2:km/L,km
			L_P_100KM_AND_KM/	0x3:mpg(GB),mil
int	unitValue	值	KM_P_L_AND_KM/	e
			MPG_GB_AND_MILE/	0x4:mpg(US),mil
		-1/	MPG_US_AND_MILE/	e
	,-X	1.6	KWH_P_100KM_AND_KM/	0x5:
			KWH_P_100MI_AND_MILE	KWH/100km,
				km
				0x6:KWH/100Mi
				, mile
				单位(功率)
			KW/HP	0x1:kW
				0x2:HP



返回值				
数据类型	名称	值	描述	
		INSTRUMENT_COMMAND_SUCCES S	成功	
		INSTRUMENT_COMMAND_FAILED;	失败	
int	设置结果	INSTRUMENT_COMMAND_INVALID ;	无效输入	
		INSTRUMENT_COMMAND_TIMEOU T;	超时	
备注	油耗距离单位目前仅支持 1、3、4			

6.16.7 设置保养时间/里程

API 名称	int setMaintenanceInfo(int typeName,int infoValue)					
接口描述	_X	设置保养时间/里程				
			输入参数			
数据类型	字段名	名称	值	描述		
	typeNam	类型	MAINTENANCE_TIME	保养时间		
int	e		MAINTENANCE_MILEAGE	保养里程		
	infoValue 值			[MAINTENANCE_TIME_DAY_MIN,	[0.720]T	
int		MAINTENANCE_TIME_DAY_MAX]	[0,720]天			
		[MAINTENANCE_MILEAGE_KILOME	[0,20000]km			



			TER_MIN,	
			MAINTENANCE_MILEAGE_KILOMET	
			ER_MAX]	
			返回值	
数据类型	名和	尔	值	描述
	设置结果	INSTRUMENT_COMMAND_SUCCES		
		S	成功	
		INSTRUMENT_COMMAND_FAILED;	失败	
int		INSTRUMENT_COMMAND_INVALID		
			无效输入	
		INSTRUMENT_COMMAND_TIMEOU	+72n-4	
		T;	超时	

6.16.8 获取外接充电量

API 名称	宮称 double getExternalChargingPower()				
接口描述		获取外接充电量			
输入参数	7/	无			
W.	返回值				
数据类型	名称	描述			
		[EXTERNAL_CHARGING_POWER_	(0.0.40000.0)		
double	外接充电量	MIN,	{0.0, 10000.0} kw.h		



		EVERNAL GUARGING ROMER	
		EXTERNAL_CHARGING_POWER_ MAX]	
备注	车辆通过充电设备累计获得的总电量		

6.16.9 监听方法

类AbsBYDAutoInstrumentListener					
public abstract	: classAbsBYDAutoInstrumentl	Listener			
	方法概要				
限定符和返回类 方法 描述 類					
void	onMalfunctionInfoChanged(i nt typeName, int hasMalfunction)	监听故障提示信息	类型和是否故障		
void	onAlarmBuzzleStateChange(i	监听蜂鸣器状态变 化	蜂鸣器状态		
void	onMaintenanceInfoChanged(int typeName, int infoValue)	监听保养里程、时间 变化	类型,里程或时间		
void	onExternalChargingPowerCh anged(double value)	监听外接充电量的 变化	外接充电量		
备注	备注 输入值的取值范围参见 get 函数的具体描述				



6.17 时间类

6.17.1 方法概要

类 BYDAutoTimeDevice						
public classBYI	public classBYDAutoTimeDevice					
	方法概要					
限定符和返回类						
型型	方法	描述				
staticBYDAuto	another to a co(Combout com)					
TimeDevice	getInstance(Context con)	获取实例				
int[]	getTime()	获取时间				
int	getTimeFormat()	获取时制				
int	setDate(int year,int month,int day,int	设置年月日、星期				
	weekday)					
int	setTime(int hour,int minute,int second)	设置时分秒				
int	setTimeFormat(int value)	设置时制				
	registerListener(AbsBYDAutoTimeListener					
void	1)	监听注册接口				
	unregisterListener(AbsBYDAutoTimeListe					
void	ner l)	取消注册接口				



6.17.2 获取时间

API 名称	int[] getTime()					
接口描述		获取时间				
输入参数		无				
		返回值				
数据类型	名称	值	描述			
	int[0] 年	[TIME_YEAR_MIN,TIME_YEAR _MAX]	[2001,2255]			
	int[1] 月	[TIME_MONTH_MIN,TIME_M ONTH_MAX]	[1,12]			
	int[2] 日	[TIME_DAY_MIN,TIME_DAY_M	[1,31]			
int[]	int[3] 时	[TIME_HOUR_MIN,TIME_HOU R_MAX]	[0,23]			
	int[4] 分	[TIME_MINUTE_MIN,TIME_MI	[0,59]			
	int[5] 秒	[TIME_SECOND_MIN,TIME_SE	[0,59]			

6.17.3 获取时制

API 名称	int getTimeFormat()
接口描述	获取时制



输入参数	无				
	返回值				
数据类型	名称 值 描述				
int	时制	TIME_FORMAT_24H_ON	24H		
		TIME_FORMAT_24H_OFF	12H		

6.17.4 设置年月日、星期

ADI 夕称	API 名称 int setDate(int year,int month,int day,int weekday)					
AFI		mre sets ate (me year) me memany me acontacy)				
接口描述			设置年月日、星期			
			输入参数			
数据类型	字段名	名称	值	描述		
		X	[TIME_YEAR_MIN,			
	year	年	TIME VEAD MAY	[2001,2255]		
	1	-1/^	TIME_YEAR_MAX]			
	month	月	[TIME_MONTH_MIN,			
			TIME_MONTH_MAX]	[1,12]		
int		[TIME_DAY_MIN,				
	day	日	TIME_DAY_MAX]	[1,31]		
4			[TIME_WEEKDAY_MIN,			
	weekday 星期	星期	TIME_WEEKDAY_MAX]	[1,7]		
	返回值					
数据类型 名称 值				描述		



		TIME_COMMAND_SUCCESS	成功
		TIME_COMMAND_FAILED;	失败
int	设置结果	TIME_COMMAND_INVALID;	无效输入
		TIME_COMMAND_TIMEOUT;	超时
		TIME_COMMAND_BUSY.	系统忙

6.17.5 设置时分秒

API 名称	int setTime(int hour,int minute,int second)				
接口描述	设置时分秒				
			输入参数		
数据类型	字段名	名称	值	描述	
	hour	时	[TIME_HOUR_MIN,TIME_HOUR_	[0,23]	
int	minute	分	[TIME_MINUTE_MIN,TIME_ MINUTE_MAX]	[0,59]	
second		秒	[TIME_SECONG_MIN,TIME_ SECONG_MAX]	[0,59]	
返回值					
数据类型	名称		值	描述	
	уд 🛱		ENERGY_COMMAND_SUCCESS	成功	
int	设置	括果	ENERGY_COMMAND_FAILED; 失败		



	ENERGY_COMMAND_INVALID;	无效输入
	ENERGY_COMMAND_TIMEOUT;	超时
	ENERGY_COMMAND_BUSY.	系统忙

6.17.6 设置时制

API 名称	int setTimeFormat(int value)			
接口描述	设置时制			
			输入参数	
数据类型	字段名	名称	值	描述
		- 14 4	TIME_FORMAT_24H_ON	24H
int	value	时制	TIME_FORMAT_24H_OFF 12H	12H
		X	返回值	
数据类型	数据类型 名称 值 描述			描述
	/ X		ENERGY_COMMAND_SUCCESS	成功
		(X),	ENERGY_COMMAND_FAILED;	失败
int	设置结果		ENERGY_COMMAND_INVALID;	无效输入
)°		ENERGY_COMMAND_TIMEOUT;	超时
4			ENERGY_COMMAND_BUSY.	系统忙

6.17.7 监听方法



public abstract class AbsBYDAutoTimeListener				
	方法概要			
限定符				
和返回	方法	描述	输入	
类型				
	au Tiura Chan a a d'intil tiura)		year, month, day, hour, minute, se	
void	onTimeChanged(int[] time)	监听时间变化	cond 变化	
	onTimeFormatChanged(int			
void	value)	监听时制变化	时制 12H/24H	
备注	输入值的取值范围参见 get 函数	女的具体描述		

6.18 车辆设置类

6.18.1 方法概要

	类 BYDAutoSettingDevice				
public clas	public classBYDAutoSettingDevice				
. \ \ \ \	方法概要				
1/1/	>		值		
限定符和返	方法	 	(get 的返回值,	set 的	
回类型			输入值)		
staticBYD	and the state of (Control to an a)				
AutoSettin	getInstance(Context con)	狄 取买例 	返回		



		Т	版本: VI.U.:
gDevice			BYDAutoSettingDevic
			e
		空调相关	
int	getACBTWind()	蓝牙通话自动降风速(开 -	SET_ON
int	setACBTWind(int value)	或关),默认开	SET_OFF
int	getACTunnelCycle()		SET_ON
int	setACTunnelCycle(int	进隧道自动内循环模式	SET_OFF
	value)	(开/关,默认开)	(LV
int	getACPauseCycle()		SET_ON
int	setACPauseCycle(int	驻车自动内循环(开/关,	SET_OFF
	value)	默认关)	
int	getACAutoAir()		SET_AC_AUTO_AIR_EC
int	7-1/1	空调自动模式设置(经济	ONOMY
	setACAutoAir(int value)	/舒适,默认舒适)	SET_AC_AUTO_AIR_C
			OMFORT
int	getPM25Power()	PM2.5 上电检测(开/关,	SET_ON
int	setPM25Power(int value)	默认开)	SET_OFF
int	getPM25SwitchCheck()		SET_ON
int	setPM25SwitchCheck(int	PM2.5 开关门检测(开/	SET_OFF
	value)	关,默认关)	
int	getPM25TimeCheck()	PM2.5 30 分钟定时检测	SET_ON



			/////////////////////////////////////
int	setPM25TimeCheck(int	(开/关,默认关)	SET_OFF
	value)		
		高压	
int	getEnergyFeedback()		SET_DR_ENERGY_FB_S
int		 	TANDARD,
		/较大),默认标准	SET_DR_ENERGY_FB_L
	value)		ARGE
int	getSOCTarget()		[SET_DR_SOC_TARGET
int		SOC 目标点设置,可设	_MIN
	setSOCTarget(int value)	置范围 15%~70% , 默	,SET_DR_SOC_TARGET
		认 25%	_MAX]
	, XX	信息媒体	
int	getChargingPort()	充电枪电锁工作模式(启	SET_ON
	1605	用防盗、停用防盗),默认	SET_OFF
int	setChargingPort(int value)	启用防盗	
	getAutoExternalRearMirr		SET_ON
int	orFollowUpSwitch()		SET_OFF
	setAutoExternalRearMirro	外后视镜随动(开或关),	
int	rFollowUpSwitch(int	默认开	
	value)		
int	getLockOff()	开锁方式(四门、仅驾驶	SET_CAR_LOCK_OFF_4



			版本: V1.U.5
		员侧),默认四门	DOORS
int	setLockOff(int value)		SET_CAR_LOCK_OFF_
			DR_SIDE
	getLanguage()(仅支持简体		SET_LANGUAGE_SIMP
	中文)		LE_CHINESE
			SET_LANGUAGE_COM
		语言(简体中文, 繁体中	PLEX_ CHINESE
int		文,English,俄语,阿拉	SET_LANGUAGE_ENG
	setLanguage(int value)		LISH
	(仅支持简体中文)	伯语,默认简体中文)	SET_LANGUAGE_RUS
			SIAN
	, XX	, /	SET_LANGUAGE_ARA
	-1/_/	•	BIC
int	getOverspeedLock()		SET_OFF
int	setOverspeedLock(int	超速闭锁(开或关),默认	SET_ON
	value)	开	
	<i>></i>	获取安全警告标志位	SET_ON:警告
int	getSafeWarnState()	(警告/不警告, "请切换	SET_OFF:不警告
nic		到 P 挡域"满足时的提	
		示)	
int	getMaintainRemindState(获取保养提醒标志	SET_ON:提示



			////////////////////////////////////	
)		SET_OFF:不提示	
	电控			
int	getSteerAssis()	转向助力模式设置(舒适	SET_DR_ST_ASSIS_CO	
		(Comfort)、运动	MFORT	
int	setSteerAssis(int value)	(Sport)), 默认舒适, 交互需要区分是否有全	SET_DR_ST_ASSIS_SP	
		地形	ORT	
			ST.	
		辅电		
int	getRearViewMirrorFlip()			
int	setRearViewMirrorFlip (int	倒车外后视镜翻转(开或 关),默认开	SET_OFF、SET_ON	
	value)	人),為(以)]		
int	getDriverSeatAutoReturn(•		
) - () (驾驶员座椅自动回位(开 	SET_OFF、SET_ON	
int	setDriverSeatAutoReturn(i	或关),默认开		
	nt value)			
int	getSteerPositionAutoRetu			
	rn()	 转向盘位置自动恢复(开 	SET_OFF、SET_ON	
int	setSteerPositionAutoRetu	或关),默认开	_	
	rn(int value)			
int	getRemoteControlUpwin	遥控升窗(开或关),默认	SET_OFF、SET_ON	



			版本: V1.0.5
	dowState()	开,不带防夹时无此项	
int	setRemoteControlUpwind		
IIIC	owState(int value)		
int	getRemoteControlDownw		
IIIC	indowState()	遥控降窗(开或关),默认	SET_OFF、SET_ON
int	setRemoteControlDownw	关	
IIIC	indowState(int value)		(11>)
int	getLockCarRiseWindow()	锁车自动关窗(开或关),	St.
int	setLockCarRiseWindow(in	默认关,不带防夹时无此	SET_OFF、SET_ON
IIIC	t value)	项	
int	getMicroSwitchLockWind		
IIIC	owState()	长按微动开关闭锁升窗 (开或关),默认开,不带	SET OFF. SET ON
int	setMicroSwitchLockWind		_ , , , _ ,
IIIC	owState(int value)	防夹时无此项	
in i	getMicroSwitchUnlockWi		
int	ndowState()	长按微动开关解锁降窗	SET OFF、SET ON
int	setMicroSwitchUnlockWi	(开或关),默认关	
int	ndowState(int value)		
int	getBackHomeLightDelay	回家照明延时(10s 、	[SET_CAR_LIGHT_DEL
int	Value()	20s、30s、40s、 50s、	AY_VALUE_MIN,
int	setBackHomeLightDelayV	60s),默认 10s,0s 表示	SET_CAR_LIGHT_DELA



			702 1 1111	
	alue(int value)	关闭	Y_VALUE_MAX]	
	getLeftHomeLightDelayV	离家照明延时(10s 、	[SET_CAR_LIGHT_DEL	
int	alue()	20s、30s、40s、 50s、	AY_VALUE_MIN,	
14	setLeftHomeLightDelayVa	60s),默认 10s,0s 表示	SET_CAR_LIGHT_DELA	
int	lue(int value)	关闭	Y_VALUE_MAX]	
	getBackDoorElectricMode		(6)	
int	0	后背门电动功能设置状	SET_OFFmanual	
	setBackDoorElectricMode	态(电动或手动)	SET_ONelectric	
int	(int value)			
	registerListener(AbsBYDA			
void	utoSettingListener I)	监听注册接口	无	
	unregisterListener(AbsBY	, /		
void	DAutoSettingListenerl)	取消注册接口	无	
	set 函数的返回值包括:			
	成功=SETTING_COMMANI	D_SUCCESS		
	失败=SETTING_COMMAND_FAILED			
备注	无效输入=SETTING_COMMAND_INVALID			
	超时=SETTING_COMMAND_TIMEOUT			
	系统忙=SETTING_COMMAND_BUSY			
	get 函数的返回值如果是 SET	「_INVAID,表示无此设置」	页	



6.18.2 是否有某项功能配置

API 名称	int hasFeature(java.lang.String feature)							
接口描述	是否有某项功能配置							
输入参数								
数据类型	字段	名称	尔	值	描述			
String	featur e	功能		FEATURE_OVERSPEED_LOCKING	超速闭锁			
				FEATURE_BACK_DOOR	电动后背门			
			能	FEATURE_REARVIEW_MIRROR_FOL	外后视镜随动			
				LOW_UP	(整车闭锁后外后视			
					镜自动折叠)			
返回值								
数据类型	名称			值	描述			
int		_	V	DEVICE_HAS_THE_FEATURE	有			
	结果			DEVICE_NOT_HAS_THE_FEATURE	没有			

6.18.3 获取后排空调在线状态

API 名称	int getRearAcOnlineState()						
接口描述	获取后排空调在线状态						
输入参数	无						
返回值							
数据类型	名称	值	描述				



	是否在线	SET OFF	不在线(未检测到/没有该模
int		321_011	块)
		SET_ON	在线(检测到/有该模块)
备注	ON 档电用来判断后排空调模块是否检测到。		

6.18.4 监听方法

		/ /		
	类 AbsBYDAutoSettingListener			
public	abstract class AbsBYDAutoSettingListe	ener		
	方法概要			
返回类型	方法	描述		
void	onACBTWindSwitchChanged(int state)	监听蓝牙通话自动降风速开关变化		
void	onACTunnelCycleSwitchChanged(int state)	监听进隧道自动内循环模式开关变化		
void	onACPauseCycleSwitchChanged(int state)	监听驻车自动内循环开关变化		
void	onACAutoAirModeChanged(int state)	监听空调自动模式设置变化(经济/舒适)		
void	onPM25PowerSwitchChanged(int state)	监听 PM2.5 上电检测开关变化		
void	onPM25SwitchCheckChanged(int	监听 PM2.5 开关门检测开关变化		



		版本: V1.0.5
	state)	
void	onPM25TimeCheckChanged(int state)	监听 PM2.5 30 分钟定时检测开关变化
void	onEnergyFeedbackStrengthChanged(int level)	监听能量回馈强度变化 (标准/较大)
void	onSOCTargetRangeChanged(int state)	监听 SOC 目标点范围变化
	onChargingPortSwitchChanged(int	监听充电枪电锁工作模式变化 (启用防
void	state)	盗、停用防盗)
void	onAutoExternalRearMirrorFollowUpS witchChanged(int state)	监听外后视镜随动开关变化
void	onLockOffDoorChanged(int state)	监听开锁方式变化(四门、仅驾驶员)
	onLanguageChanged(int value) (仅	监听语言(简体中文,繁体中文,English,
void	支持简体中文)	俄语,阿拉伯语, 默认简体中文)变化
void	onOverspeedLockStateChanged(int state)	监听超速闭锁开关变化
void	onSafeWarnStateChanged(int state)	监听安全警告标志位变化(警告/不警告)
void	onMaintainRemindStateChanged(int state)	监听保养提醒标志变化
void	onSteerAssisModeChanged(int state)	监 听 转 向 助 力 模 式 变 化(舒 适 (Comfort)、运动(Sport))



		版本: V1.0.5	
void	onRearViewMirrorFlipSwitchChanged (int state)	监听倒车外后视镜翻转开关变化	
void	onDriverSeatAutoReturnSwitchChan ged(int state)	监听驾驶员座椅自动回位开关变化	
void	onSteerPositionAutoReturnSwitchCh anged(int state)	监听转向盘位置自动恢复开关变化	
void	onRemoteControlUpwindowState	监听遥控升窗开关变化,不带防夹时无此	
70.0	Changed(int state)	项	
void	onControlWindowSwitchChanged(int	监听遥控降窗开关变化	
void	state)	皿引迪比特图介入文化	
void	onLockCarRiseWindowChanged(int	监听锁车自动关窗开关变化,不带防夹时	
void	state)	无此项	
void	onMicroSwitchLockWindowStateCha	监听长按微动开关闭锁升窗开关变化,不	
void	nged(int state)	带防夹时无此项	
woid	onMicroSwitchUnlockWindowStateC	监听长按微动开关解锁降窗开关变化	
void	hanged(int state)	血引入技机划开大胜坝阵图开大文化	
ve:d	onBackHomeLightDelayValueChange	监听回家照明延时时间变化(10s、20s、	
void	d(int value)	30s、40s、 50s、60s), 0s 表示关闭	
ا- : - ،	onLeftHomeLightDelayValueChange	监听离家照明延时时间变化(10s、20s、	
void	d(int value)	30s、40s、 50s、60s), 0s 表示关闭	
void	onBackDoorElectricModeChanged(in	监听后背门电动功能设置状态变化(电动	



	t mode)	或手动)
备注	输入值的取值范围参见 get 函数的具体描述	

6.19传感器类

6.19.1 方法概要

	类 BYDAutoSensorDevice	(4)
	Z B1 D1 late ose il soi De vice	11-
public class BYI	DAutoSensorDevice	Qi V
	方法概要	
限定符和返回类		> `
型	方法	描述
static		
BYDAutoSenso	getInstance(Context con)	获取实例
rDevice	XI='\"	
/\		获取光照(环境光)强度等
int	getLightIntensity()	级
KD.	registerListener(AbsBYDAutoSensorListen	监听注册接口
void	er I)	
	unregisterListener(AbsBYDAutoSensorList	
void	ener l)	取消注册接口



6.19.2 获取光照强度等级

API 名称	int getLightIntensity()			
接口描述	获取光照(环境光)强度等级			
输入参数	无			
		返回值		
数据类型	名称	值	描述	
		LIGHT_ILLUM_LEVEL1	大于 230Lux	
		LIGHT_ILLUM_LEVEL2	大于 180Lux ,小	
			230Lux	
	/sts./471	LIGHT_ILLUM_LEVEL3	大于 130Lux ,小	
int	等级		180Lux	
		LIGHT_ILLUM_LEVEL4	大于 80Lux ,小	
	7-1/		130Lux	
		LIGHT_ILLUM_LEVEL5	小于 80Lux	

6.19.3 监听方法

类 AbsBYDAutoSensorListener					
public abstract class AbsBYDAutoSensorListener					
方法概要					
限定符					
和返回	方法	描述	输入		



类型			
void	onLightIntensityChanged(int value)		光照等级
备注	输入值的取值范围参见 get 函数	的具体描述	

6.20媒体中心类

6.20.1 方法概要

类 BYDAutoMultimediaDevice					
public class BYDAu	oublic class BYDAutoMultimediaDevice				
	方法概要	> `			
限定符和返回类型	方法	描述			
static BYDAutoMu ItimediaDevice	getInstance(Context con)	获取实例			
int	getMediaType()	获取播放类型			
int	getPlayMode()	获取播放模式			
int	getPlayState()	获取播放状态			
MediaInfo	getPlayMediaInfo()	获取当前播放的音视频信 息			
int	controlMedia(int mode, int action, MediaControlParam param)	媒体中心控制方法			
void	registerListener(AbsBYDAutoMultimed	监听注册接口			



	iaListener l)	
	unregister Listener (Abs BYD Auto Multim	
void	ediaListener l)	取消注册接口
	该类针对的是车载多媒体自带的媒体中心,	包括音视频播放、广播、蓝
备注	牙音乐等功能	

6.20.2 获取播放类型

API 名称	int getMediaType()			
接口描述	获取播放类型			
输入参数	无			
		返回值		
数据类型	名称 值 描述			
	播放类型	MULTIMEDIA_TYPE_AM	AM	
		MULTIMEDIA_TYPE_FM	FM	
		MULTIMEDIA_TYPE_CD	CD	
W		MULTIMEDIA_TYPE_VCD	VCD	
int		MULTIMEDIA_TYPE_DVD	DVD	
		MULTIMEDIA_TYPE_TV	TV	
		MULTIMEDIA_TYPE_AUDIO_O	Audio Off (音响关闭)	
		FF		
		MULTIMEDIA_TYPE_AUX	AUX	



		MULTIMEDIA_TYPE_USB_AUD	USB 音频	
		Ю		
		MULTIMEDIA_TYPE_USB_VIDE	USB 视频	
		0		
		MULTIMEDIA_TYPE_SD_AUDI	SD 音频	
		0		
		MULTIMEDIA_TYPE_SD_VIDE	SD 视频	
		0	Kr.	
		MULTIMEDIA_TYPE_HD_AUDI	HD (硬盘)音频	
		0		
		MULTIMEDIA_TYPE_HD_VIDE	HD (硬盘)视频	
		0		
		MULTIMEDIA_TYPE_LOCAL_A	本地音频	
	1.60	UDIO		
		MULTIMEDIA_TYPE_LOCAL_VI	本地视频	
N		DEO		
V		MULTIMEDIA_TYPE_BT	蓝牙音乐	
		MULTIMEDIA_TYPE_ROBOT	Robot	
		MULTIMEDIA_TYPE_INVAID	无效	
عدد بط	媒体中心未启动,或	诸被用户从近期任务清除,则返回	且	
备注 MULTIMEDIA_TYPE_INVAID				



否则返回媒体中心当前的播放类型。

播放类型随不同车型变化。

6.20.3 获取播放模式

0.20.0 9/4/38/3/(\$20							
API 名称	int getPlayMode()						
接口描述	获取播放模式						
输入参数		无					
		返回值	Kr.				
数据类型	名称	值	描述				
		MULTIMEDIA_PLAY_MODE_SI	单曲重复播放				
		NGLE_REPEAT	单曲重复播放R 随机播放D 预览播放S SCAN 状态				
		MULTIMEDIA_PLAY_MODE_R	随机播放				
	-1/	ANDOM	SI 单曲重复播放 R 随机播放 P 预览播放 S SCAN 状态 S 广播播放立体声频道				
	1.692	MULTIMEDIA_PLAY_MODE_P					
	1272451H-15	REVIEW					
int	播放模式	MULTIMEDIA_PLAY_MODE_S	SCAN 状态				
V		CAN					
		MULTIMEDIA_PLAY_MODE_S	广播播放立体声频道				
		TEREO					
		MULTIMEDIA_PLAY_MODE_A	全部重复播放				
		LL_REPEAT					



		MULTIMEDIA_PLAY_MODE_I	无效			
		NVAID				
	媒体中心未启动,或	诸被用户从近期任务清除,则返回	回			
	MULTIMEDIA_PLAY_MODE_INVAID,					
	否则返回媒体中心当前的播放模式。					
	FM: MULTIMEDIA_PLAY_MODE_STEREO					
备注	Music:					
	MULTIMEDIA_PLAY_MODE_SINGLE_REPEAT/MULTIMEDIA_PLAY_MODE_					
	RANDOM/MULTIMEDIA_PLAY_MODE_ALL_REPEAT					
	Video: MULTIMEDIA_PLAY_MODE_ALL_REPEAT					

6.20.4 获取播放状态

API 名称	API 名称 int getPlayState()						
接口描述							
输入参数		无					
W.		返回值					
数据类型							
		MULTIMEDIA_STATE_PLAY	播放				
int	播放状态	MULTIMEDIA_STATE_PAUSE	停止				
		MULTIMEDIA_STATE_STOP 暂停					
备注 媒体中心未启动,或者被用户近期任务清除,则返回停止,否则返回媒体中心							



当前的播放状态。

对于不同播放类型,返回结果有以下区别:

Music (本地,U 盘,蓝牙):播放,暂停。

Video(本地, U 盘): 播放, 暂停。

FM:播放,停止。

6.20.5 获取当前播放的音视频信息

API 名称		MediaInfo getPlayMediaInfo(
接口描述		获取当前播放的音视频信息					
输入参数		无					
		返回值					
	数据类型	值	描述				
	String fileName		文件名称				
	String		歌手名称				
MediaInf	artistName						
0	String		专辑名称				
V	albumName						
	媒体中心未启动,或者被用户从近期任务清除,返回 NULL。						
备注	媒体中心占据音频焦点时,根据当前的媒体中心类型返回相应信息:						
	FM: 返回空						
	Music: 返回音频名	称,艺术家名称(没有则为空),	专辑名称(没有则为空)。				



Video: 返回视频名称,艺术家和专辑为空。

其他应用占据音频焦点时,返回 NULL。

6.20.6 媒体中心控制

	0.20.0 殊件个约11的						
API 名称	int controlMedia(int mode,int action,MediaControlParam param)						
接口描述	控制音乐、视频、收音机						
	输入参数						
数据类型	名称	值	描述				
		MODE_RADIO	FM				
int	mode	MODE_MUSIC	音乐				
		MODE_VIDEO	视频				
	4	ACTION_ENTER	打开媒体中心				
	~	ACTION_PLAY	播放音乐/视频,收听 FM				
		ACTION_PAUSE	暂停音乐/视频,FM 无暂停				
			ACTION DIAV DDE	音乐/视频播放上一首, FM 搜			
	action	ACTION_PLAY_PRE	音乐/视频播放上一首,FM 搜索上一台				
int		ACTION PLAY NEXT	音乐/视频播放下一首, FM 搜				
		ACTION_PLAT_NEXT	索下一台				
		ACTION_SET_PLAY_PATTERN	音乐设置播放模式(循环 , 随				
			ACTION_SET_PLAY_PATTERN	机, 单曲), 视频设置全屏半屏			
			播放				



		I				
				ACTION_AU	TO_SEARCH	FM 自动搜索
			EL_RADIO_SEA	FM 停止搜台		
			SOURCE_LOC AL	本地音乐/视频		
		PARAM_SOUR	SOURCE_USB	U 盘音乐/视频		
		CE	SOURCE_SD	SD 音乐/视频		
			SOURCE_BTM USIC	蓝牙音乐		
		param		PATTERN_CYC	音乐全部	
MediaCo ntrolPara				param DOM	音乐随机	
m		PARAM_PATTE	PATTERN_SIN GLE	音乐单曲		
1			PATTERN_HAL F_SCREEN	视频半屏播放		
			PATTERN_FUL L_SCREEN	视频全屏播放		
		PARAM_RADI O_FREQ	FM 频道值,int 类型	FM 频道值		



		PARA	M_FILE_	歌	曲/视频名		歌曲/视频	而夕称
	PAF		AME	称,	string 类型		可人四/1989	火 石
			M_ARTIS	艺	术家名称,	_		D
		T_ N	NAME	st	ring 类型	歌曲所属艺术家名称		木家名称
返回值								
数据类型	名称		值描		描述			
	控制结果		MULTIM	EDIA	A_COMMAN	ND_SU	1/	>
			CCESS					
int			MULTIM	EDI	A_COMMAI_A	ND_FA	4	失败
			ILED					
,	MediaControlParam 是封装好的类型,里面的内容为键值对。 针对不同			针对不同的				
备注	mode+action, param 可以		am 可以填	充一	-项或者多项	į, 也可	以不填充。	

6.20.7监听方法

	类 AbsBYDAutoMultimediaListener				
pub	public abstract class AbsBYDAutoMultimediaListener				
当监	听的对象数值发生变化时,推送给	用户			
		方法概要			
返					
回					
类					



			710-4
型			
voi	on Media Type Changed (int		1524 V TU
d	type)	监听播放类型的变化	播放类型
voi	onPlayModeChanged (int		155 24 1++ 15
d	mode)	监听播放模式的变化	播放模式
voi			
d	onPlayStateChanged (int state)	监听播放状态的变化	播放状态
voi	on Play Media Info Changed	监听当前播放的音视频信息的变	当前播放的音视
d	(MediaInfo mediaInfo)	化	频信息
	输入值的取值 参见 get 函数的具体	本描述。	
备	当播放类型的变化引起播放模式、	播放状态、当前播放的音视频信息	即变化时,
注	也会同步通知变化。		