版本：V1.0.5

比亚迪智慧开放平台 API 说明书

版本：V1.0.5

版本：V1.0.5

目录

版本：V1.0.5

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cameraId 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 BYDAutoBodyworkDevice	getInstance(Context con)	获取实例
String	getAutoVIN()	获取车架号
int	getAutoModelName()	获取车型名称
int	getAutoSystemState()	获取整车状态
int	getDoorState(int area)	获取车门、前舱盖、后舱盖状态
int	getWindowState(int area)	获取车窗状态

int	getWindowOpenPercent(int area)	获取天窗、遮阳帘位置百分比
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
void	registerListener(AbsBYDAutoBodyworkListener l)	监听注册接口
void	unregisterListener(AbsBYDAutoBodyworkListenerl)	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	无		
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
int	车型名称	AUTO_MODEL_SONG_MAX_HEV	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
		AUTO_MODEL_NEW_TANG_EV	全新一代唐纯电
		AUTO_MODEL_NEW_TANG_FUEL	全新一代唐燃油
		AUTO_MODEL_SONG_18_HEV	宋 18 款混动
		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
int	整车状态	BODYWORK_AUTO_SYSTEM_STATE_NORMAL	正常系统状态
		BODYWORK_AUTO_SYSTEM_STATE_SET_SECURE	设定安全系统状态
		BODYWORK_AUTO_SYSTEM_STATE_START_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	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_LUGGAGE_DOOR	后舱盖
返回值				
数据类型	名称		值	描述
int	门状态		BODYWORK_STATE_OPEN	开启
			BODYWORK_STATE_CLOSED	关闭
			BODYWORK_STATE_UNDEFINED	未定义
			BODYWORK_COMMAND_INVALID_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
int	area	车窗区域	BODYWORK_CMD_WINDOW_LEFT_FRONT	左前窗
			BODYWORK_CMD_WINDOW_RIGHT_FRONT	右前窗
			BODYWORK_CMD_WINDOW_LEFT_REAR	左后窗
			BODYWORK_CMD_WINDOW_RIGHT_REAR	右后窗
返回值				
数据类型	名称		值	描述
int	车窗状态		BODYWORK_STATE_OPEN	开启
			BODYWORK_STATE_CLOSED	关闭
			BODYWORK_STATE_UNDEFINED	未定义
			BODYWORK_COMMAND_INVALID_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
int	area	位置	BODYWORK_CMD_MOON_ROOF	天窗
			BODYWORK_CMD_SUNSHADE_PANEL	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
int	电压水平	BODYWORK_BATTERY_VOLTAGE_LEVEL_NORMAL	电压正常
		BODYWORK_BATTERY_VOLTAGE_LEVEL_LOW	低电压

		BODYWORK_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
int	电源档位	BODYWORK_POWER_LEVEL_ACC	0x1: ACC 档
		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			
输入参数				
数据类型	字段名	名称	值	描述
int	type	需要获取的类型	BODYWORK_CMD_STEERING_WHEEL_ANGEL	角度
			BODYWORK_CMD_STEERING_WHEEL_SPEED	速度

返回值			
数据类型	名称	值	描述
double	角度/速度	[BODYWORK_STEERING_WHEEL_SPEED_MIN, BODYWORK_STEERING_WHEEL_SPEED_MAX]°/s	[0,1016]°/s 方向盘旋转速度
		[BODYWORK_STEERING_WHEEL_ANGEL_MIN, BODYWORK_STEERING_WHEEL_ANGEL_MAX] °	[-780.0°,+779.9°] 方向盘角度
		BODYWORK_COMMAND_INVALID_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		
返回值			
数据类型	名称	值	描述
int	低能量类型	BODYWORK_LOW_POWER_NORMAL	油量、电量都正常

		BODYWORK_LOW_POWER_FUEL	油量低
		BODYWORK_LOW_POWER_ELEC	电量低
		BODYWORK_LOW_POWER_BOTH	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
int	报警器状态	BODYWORK_ALARM_STATE_ON	报警
		BODYWORK_ALARM_STATE_OFF	不报警
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

int	天窗遮阳帘配置	CONFIG_NONE	无
		CONFIG_MOON_ROOF_SUNSHADE_PANEL	全景天窗遮阳帘
		CONFIG_SUNSHADE_PANEL	全景遮阳帘(天窗无法打开)
		CONFIG_ANTI_PINCH_MOON_ROOF	防夹小天窗

6.1.14 监听方法

类 AbsBYDAutoBodyworkListener			
public abstract class AbsBYDAutoBodyworkListener			
当监听的对象数值发生变化时，推送给用户			
方法概要			
返回类型	方法	描述	输入
void	onAutoSystemStateChanged(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
void	onBatteryVoltageLevelChanged(int level)	监听蓄电池电压水平变化	1：电压正常 2：低电压

void	onDoorStateChanged(int area, int state)	监听车门变化	开启 关闭
void	onWindowStateChanged(int area, int state)	监听车窗变化	开启 关闭
void	onWindowOpenPercent(int area, int value)	监听天窗、遮阳帘位置百分比的变化	area: 天窗 遮阳帘 value: 位置百分比
void	onPowerLevelChanged(int level)	监听电源档位变化	0x1: ACC 档 0x0：OFF 档 0x2: ON 档 0x3：OK/READY
void	onSteeringWheelValueChanged(int type, double value)	监听方向盘角度变化	type 为角度时， 输入方向盘角度； type 为速度时， 输入方向盘速度
void	onFuelElecLowPowerChanged(int state)	监听油量电量低提醒	油量、电量都正常 油量低 电量低 油量电量都低

void	onAlarmStateChanged(int state)	监听报警器状态	报警 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 classBYDAutoStatisticDevice		
方法概要		
限定符和返回类型	方法	描述
staticBYDAutoStatisticDevice	getInstance(Context con)	获取实例
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(AbsBYDAutoStatisticListener l)	监听注册接口
void	unregisterListener(AbsBYDAutoStatisticListener l)	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	名称	值	描述
double	时间	[STATISTIC_DRIVING_TIME_MIN, 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_RANGE_MIN, STATISTIC_ELEC_DRIVING_RANGE_MAX]	[0,511]KM

6.2.4 获取电量百分比

API 名称	double getElecPercentageValue()		
接口描述	获取电量百分比		
输入参数	无		
返回值			
数据类型	名称	值	描述
double	电量百分比	[STATISTIC_ELEC_PERCENTAGE_MIN, STATISTIC_ELEC_PERCENTAGE_MAX]	[0,100]%

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_RANGE_MIN, STATISTIC_FUEL_DRIVING_RANGE_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
int	燃油百分比	[STATISTIC_FUEL_PERCENTAGE_MIN, STATISTIC_FUEL_PERCENTAGE_MAX]	{0,100}%

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
double	最近百公里电耗	[STATISTIC_LAST_ELEC_CON_PHM_MIN, STATISTIC_LAST_ELEC_CON_PHM_MIN]	{-99.9;99.9}KWH/100KM

6.2.8获取最近百公里油耗

API 名称	double getLastFuelConPHMValue()		
接口描述	获取最近百公里油耗		
输入参数	无		
返回值			
数据类型	名称	值	描述
double	最近百公里油耗	[STATISTIC_LAST_FULE_CON_PHM_MIN, STATISTIC_LAST_FULE_CON_PHM_MIN]	{0;51.1} L/100KM

6.2.9 获取累计平均电耗

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
double	累计平均电耗	[TATISTIC_TOTAL_ELEC_CON_PHM_MIN, STATISTIC_TOTAL_ELEC_CON_PHM_MAX]	{-99.9;99.9}KWH/100KM

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
double	累计平均油耗	[STATISTIC_TOTAL_FUEL_CON_PHM_MIN, STATISTIC_TOTAL_FUEL_CON_PHM__MAX]	{0;51.1} L/100KM

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
double	燃油消耗总量	[STATISTIC_TOTAL_FUEL_CONSUMPTION_MIN, STATISTIC_TOTAL_FUEL_CONSUMPTION_MAX]	{0; 104857.4} L

6.2.12 获取电消耗总量

API 名称	double getTotalElecConValue()		
接口描述	获取电消耗总量		
输入参数	无		
返回值			
数据类型	名称	值	描述
double	电消耗总量	[STATISTIC_TOTAL_ELEC_CONSUMPTION_MIN, STATISTIC_TOTAL_ELEC_CONSUMPTION_MAX]	{-1000，1676721.4} kW•h (HP.h)
备注	电池放电为正 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
int	总里程	[STATISTIC_TOTAL_MILEAGE_MIN , STATISTIC_TOTAL_MILEAGE_MAX]	{0,999999}km

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	钥匙电量	STATISTIC_KEY_BATTERY_LEVEL_LOW STATISTIC_KEY_BATTERY_LEVEL_NORMAL	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 abstract classAbsBYDAutoStatisticListener			
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	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 classBYDAutoSpeedDevice		
方法概要		
限定符和返回类型	方法	描述
staticBYDAutoSpeedDevice	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
void	registerListener(AbsBYDAutoSpeedListener l)	监听注册接口
void	unregisterListener(AbsBYDAutoSpeedListenerl)	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	无		
返回值			
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		
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 l)	监听注册接口
void	unregisterListener(AbsBYDAutoEnergyListener l)	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_ECONOMY	经济模式
		ENERGY_OPERATION_SPORT	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
int	原地踩油门发电状态	ENERGY_POWER_GENERATION_INVALID	无效
		ENERGY_POWER_GENERATING	发电中
		ENERGY_POWER_GENERATION_END	发电结束
		ENERGY_POWER_GENERATION_ERROR	无法进入

6.4.5 获取原地踩油门发电功率

API 名称	int getPowerGenerationValue()		
接口描述	获取原地踩油门发电功率		
输入参数	无		
返回值			
数据类型	名称	值	描述
int	原地踩油门发电功率	[ENERGY_POWER_GENERATION_VALUE_MIN, ENERGY_POWER_GENERATION_VALUE_MAX]	[1,31]KW

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	值	描述
int	路面模式	ENERGY_ROAD_SURFACE_KEEP	保持在线
		ENERGY_ROAD_SURFACE_COMMON	普通模式
		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 AbsBYDAutoEnergyListener			
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	onEnergyModeChanged(int mode)	监听整车工作模式	EV/强制 EV/HEV
void	onOperationModeChanged(int mode)	监听整车运行模式	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 classBYDAutoPanoramaDevice		
方法概要		
限定符和返回类型	方法	描述
staticBYDAutoPanoramaDevice	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
int	getBackLineConfig()	获取 sec 倒车线配置状态
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
void	registerListener(AbsBYDAutoPanoramaListener l)	监听注册接口
void	unregisterListener(AbsBYDAutPanoramaListener l)	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
int	全景视频模式	PANORAMA_OUTPUT_SIGNAL_CVBS	CVBS
		PANORAMA_OUTPUT_SIGNAL_LVDS	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		
返回值			
数据类型	名称	值	描述
int	全景打开状态	PANORAMA_WORK_OFF	关闭状态
		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
int	sec 倒车线配置状态	BACK_LINE_NOT_SUPPORT	不支持配置
		BACK_LINE_PAN_INTERNAL	配置为全景内部倒车线
		BACK_LINE_MULTIMEDIA	配置为多媒体倒车线

6.5.5 获取影像输出状态

API 名称	intgetPanoOutputState()		
接口描述	获取全景输出状态		
输入参数	无		
返回值			
数据类型	名称	值	描述
int	输出状态	PANORAMA_OUTPUT_OFF	关闭显示
		PANORAMA_OUTPUT_FRONT	前视
		PANORAMA_OUTPUT_REAR	后视
		PANORAMA_OUTPUT_LEFT	左视
		PANORAMA_OUTPUT_RIGHT	右视
		PANORAMA_OUTPUT_COMP	左右合成

		OSE	
		PANORAMA_OUTPUT_MATCHING	开始匹配
		PANORAMA_OUTPUT_FRONT_LEFT	前视+左视
		PANORAMA_OUTPUT_FRONT_RIGHT	前视+右视
		PANORAMA_OUTPUT_REAR_LEFT	后视+左视
		PANORAMA_OUTPUT_REAR_RIGHT	后视+右视
备注	全景模式、大图模式下、小窗口模式下都能实现 前视，后视、左视，右视、前+左、前+右、后+左、后+右。 前+左、前+右、后+左、后+右，这个当前仅针对横屏状态。 全景模式，实际显示为：全景+ (前+左、前+右、后+左、后+右)。 没有全景配置的车 (右前影像配置)，仅返右视、后视。		

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	名称	值	描述
int	影像的屏幕方向	PANORAMA_ROTATION_HORIZONTAL	横屏
		PANORAMA_ROTATION_VERTICAL	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
int	显示模式	DISPLAY_MODE_PANORAMA	全景模式
		DISPLAY_MODE_FULL_SCREEN	大图模式(全屏)
		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
int	配置	PANORAMA_RF_REVERSE	倒车+右前影像配置
		PANORAMA_ONLINE	全景影像配置
		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 AbsBYDAutPanoramaListener			
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	onPanoOutputStateChanged(int mode)	监听影像输出状态	关闭显示 前视 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 mode)	监听全景打开状态	关闭状态 工作状态
void	onBackLineConfigChanged(int mode)	监听 sec 倒车线配置状态	不支持配置 配置为全景内部倒车线 配置为多媒体倒车线
void	onPanoRotationChanged(int value)	监听影像的屏幕方向的变化	横屏 竖屏
void	onDisplayModeChanged(int mode)	监听显示模式的变化	全景模式 大图模式 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 BYDAutoAcDevice		
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
int	getAcCompressorManualSign()	获取 A/C (压缩机)手动标志
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

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
int	getAcTemperatureControlMode()	获取主副驾驶温度分控方式
int	getAcVentilationState()	获取通风功能设置
int	getRearAcStartState()	获取后空调开启状态
int	setAcControlMode(int setSource,int mode)	设置空调控制方式
int	setAcCycleMode(int setSource,int mode)	设置空调循环方式
int	setAcWindMode(int setSource,int mode)	设置空调出风模式
int	setAcDefrostState(int setSource,int area,int state)	设置前后排空调除霜状态
int	setAcWindLevel(int setSource ,int level)	设置空调风量档位
int	setAcTemperature(int type,int value,int tempSource, int unit)	设置空调温度
int	setAcTemperatureControlMode(int setSource,int mode)	设置主副驾驶温度分控方式
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
void	registerListener(AbsBYDAutoAcListener l)	监听注册接口
void	unregisterListener(AbsBYDAutoAcListener l)	取消注册接口

6.6.2 获取 A/C (压缩机)状态

API 名称	int getAcCompressorMode()		
接口描述	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
int	A/C (压缩机)状态	AC_COMPRESSOR_OFF	压缩机关
		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
int	A/C (压缩机)手动标志	AC_COMPRESSOR_MANUAL_SIGN_OFF	自动控制
		AC_COMPRESSOR_MANUAL_SIGN_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
int	风量手动标志	AC_WINDLEVEL_MANUAL_SIGN_OFF	自动控制
		AC_WINDLEVEL_MANUAL_SIGN_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	值	描述
int	出风模式手动标志	AC_WINDMODE_MANUAL_SIGN_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		
返回值			
数据类型	名称	值	描述
int	空调开启状态	AC_POWER_ON	开启
		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
int	空调控制方式	AC_CTRLMODE_MANUAL	手动
		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
int	空调循环方式	AC_CYCLEMODE_OUTLOOP	外循环
		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	无		
返回值			
数据类型	名称	值	描述
int	空调出风模式	AC_WINDMODE_FACE	1：吹面
		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_FACEFOOTDEFROST	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

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
int	area	除霜开关位置	AC_DEFROST_AREA_FRONT	前除霜开关
			AC_DEFROST_AREA_REAR	后除霜开关
返回值				
数据类型	名称		值	描述
int	所选中的除霜开关的状态		AC_DEFROST_STATE_OFF	除霜状态关闭
			AC_DEFROST_STATE_ON	除霜状态开启
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	

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	名称	值	描述
int	area	区域	AC_TEMPERATURE_MAIN	主驾驶温度
			AC_TEMPERATURE_DEPUTY	副驾驶温度
			AC_TEMPERATURE_REAR	后空调
			AC_TEMPERATURE_OUT	车外温度
返回值				
数据类型	名称		值	描述

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

6.6.13 获取默认温度单位

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

int	默认温度单位	AC_TEMPERATURE_UNIT_OF	华氏°F
		AC_TEMPERATURE_UNIT_OC	摄氏°C

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
int	主副驾驶温度分控方式	AC_TEMPCTRL_SEPARATE_OFF	不分控
		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	获取通风功能设置		
输入参数	无		
返回值			
数据类型	名称	值	描述
int	通风功能设置	AC_VENTILATION_STATE_ON	打开
		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	值	描述
int	后空调开启状态	AC_POWER_ON	开启
		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)			
接口描述	设置空调控制方式			
输入参数				
数据类型	字段	名称	值	描述
int	setSource	控制来源	AC_CTRL_SOURCE_VOICE	声控
			AC_CTRL_SOURCE_UI_KEY	触屏/按键
int	mode	空调控制方式	AC_CTRLMODE_MANUAL	手动
			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_VALUE	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			
输入参数				
数据类型	字段	名称	值	描述
int	setSource	控制来源	AC_CTRL_SOURCE_VOICE	声控
			AC_CTRL_SOURCE_UI_KEY	触屏/按键
int	mode	空调循环方式	AC_CYCLEMODE_INLOOP	内循环
			AC_CYCLEMODE_OUTLOOP	外循环
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_VALUE	无效值
		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
int	setSource	控制来源	AC_CTRL_SOURCE_VOICE	声控
			AC_CTRL_SOURCE_UI_KEY	触屏/按键
int	mode	模式	AC_WINDMODE_FACE	1：吹面
			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_FACEFOOTDEFROST	6：吹面+吹脚+除霜
				7：吹面+除霜

			AC_WINDMODE_FACEDEFROST	
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_VALUE	无效值
			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			
输入参数				
数据类型	字段	名称	值	描述
int	setSource	控制来源	AC_CTRL_SOURCE_VOICE	声控
			AC_CTRL_SOURCE_UI_KEY	触屏/按键
int	area	区域	AC_DEFROST_AREA_FRONT	前除霜开关
			AC_DEFROST_AREA_REAR	后除霜开关
int	state	状态	AC_DEFROST_STATE_ON	除霜状态开启
			AC_DEFROST_STATE_OFF	除霜状态关闭

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_VALUE	无效值
		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

API 名称	int setAcWindLevel(int setSource ,int level)			
接口描述	设置空调风量档位			
输入参数				
数据类型	字段	名称	值	描述
int	setSource	控制来源	AC_CTRL_SOURCE_VOICE	声控
			AC_CTRL_SOURCE_UI_KEY	触屏/按键
int	level	风量档位	AC_WINDLEVEL_1	1 ~7，一共 7 个等级的风量
			AC_WINDLEVEL_2	
			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
int	设置状态后的结果		AC_COMMAND_SUCCESS	成功
			AC_COMMAND_INVALID_VALUE	无效值
			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	设置空调温度			
输入参数				
数据类型	字段	名称	值	描述
int	type	空调位置	AC_TEMPERATURE_MAIN	主驾驶空调
			AC_TEMPERATURE_DEPUTY	副驾驶空调
			AC_TEMPERATURE_MAIN_	主副驾 (不分控时有效)

			DEPUTY	
			AC_TEMPERATURE_REAR	后空调
int	value	温度值	[AC_TEMP_IN_FAHRENHEIT_MIN, AC_TEMP_IN_FAHRENHEIT_MAX]°F 或 [AC_TEMP_IN_CELSIUS_MIN, AC_TEMP_IN_CELSIUS_MAX]°C	华氏温度范围[64，91] 摄氏温度范围[17，33]
int	tempSource	控制途径	AC_CTRL_SOURCE_UI_KEY	触屏/按键
			AC_CTRL_SOURCE_VOICE	语音控制
int	unit	温度单位	AC_TEMPERATURE_UNIT_OF	温度单位：华氏温度
			AC_TEMPERATURE_UNIT_OC	温度单位：摄氏度
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_	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				
数据类型	字段	名称	值	描述
int	setSource	控制来源	AC_CTRL_SOURCE_VOICE	声控
			AC_CTRL_SOURCE_UI_KEY	触屏/按键
int	mode	分控方式	AC_TEMPCTRL_SEPARATE_OFF	不分控
			AC_TEMPCTRL_SEPARATE_ON	分控
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_VALUE	无效值
			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
int	setSource	控制来源	AC_CTRL_SOURCE_VOICE	声控
			AC_CTRL_SOURCE_UI_KEY	触屏/按键
int	state	通风功能	AC_VENTILATION_STATE_ON	打开
			AC_VENTILATION_STATE_OFF	关闭
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_VALUE	无效值
			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
int	setSource	控制来源	AC_CTRL_SOURCE_VOICE	声控
			AC_CTRL_SOURCE_UI_KEY	触屏/按键
返回值				
数据类型	名称		值	描述
int	开启空调结果		AC_COMMAND_SUCCESS	成功
			AC_COMMAND_INVALID_VALUE	无效值
			AC_COMMAND_TIMEOUT	超时
			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

int	setSource	控制来源	AC_CTRL_SOURCE_VOICE	声控
			AC_CTRL_SOURCE_UI_KEY	触屏/按键
返回值				
数据类型	名称		值	描述
int	关闭空调结果		AC_COMMAND_SUCCESS	成功
			AC_COMMAND_INVALID_VALUE	无效值
			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
int	setSource	控制来源	AC_CTRL_SOURCE_VOICE	声控
			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_VALUE	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
int	setSource	控制来源	AC_CTRL_SOURCE_VOICE	声控
			AC_CTRL_SOURCE_UI_KEY	触屏/按键
返回值				
数据类型	名称		值	描述
int	关闭后排空调结果		AC_COMMAND_SUCCESS	成功
			AC_COMMAND_INVALID_VALUE	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		

6.6.29 监听方法

类 AbsBYDAutoAcListener			
public abstract class AbsBYDAutoAcListener			
方法概要			
限定符和返回类型	方法	描述	输入
void	onAcCompressorManualSignChanged(int sign)	监听 A/C(压缩机)手动标志的变化	手动标志
void	onAcCompressorModeChanged(int mode)	监听 A/C(压缩机)状态的变化	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
void	onAcDefrostStateChanged(int area,int state)	监听空调除霜方式的变化	区域（前除霜开关， 后除霜开关） 开关的状态
void	onAcRearStarted()	监听后空调开启状态	无

void	onAcRearStoped()	监听后空调关闭状态	无
void	onAcStarted()	监听空调开启状态	无
void	onAcStoped()	监听空调关闭状态	无
void	onAcVentilationStateChanged(int state)	监听通风功能设置的变化	状态
void	onAcWindLevelChanged(int level)	监听风量档位的变化	风量档位
void	onAcWindLevelManualSignChanged(int sign)	监听风量手动标志的变化	手动标志
void	onAcWindModeChanged(int mode)	监听出风模式的变化	出风模式
void	onAcWindModeManualSignChanged(int sign)	监听出风模式手动标志的变化	手动标志
void	onAcWindModeShownStateChanged(int state)	监听空调出风模式显示状态	状态
void	onTemperatureChanged(int area,int value)	监听各区域温度的变化	空调区域 温度值
void	onTemperatureUnitChanged(int unit)	监听温度单位的变化	温度单位
备注	输入值的取值范围参见 get 函数的具体描述		

6.7 空气质量类

6.7.1 方法概要

类 BYDAutoPM2p5Device		
public class BYDAutoPM2p5Device		
方法概要		
限定符和返回类型	方法	描述
static BYDAutoPM2p5 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(AbsBYDAutoPM2p5Listener l)	监听注册接口
void	unregisterListener(AbsBYDAutPM2p5 Listener l)	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	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
int[]	int[0]车内检测状态	PM2P5_STATE_ON/PM2P5_STATE_OFF	开启/关闭
	int[1]车外检测状态	PM2P5_STATE_ON/PM2P5_STATE_OFF	开启/关闭

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[]	int[0]车内 PM2.5 等级	PM2P5_LEVEL_INVALID PM2P5_LEVEL_EXCELLENT PM2P5_LEVEL_GOOD PM2P5_LEVEL_LOW_GRADE PM2P5_LEVEL_MIDDLE PM2P5_LEVEL_HEAVY PM2P5_LEVEL_SERIOUS	0x0：无效 0x1：优 0x2：良 0x3：轻度 0x4：中度 0x5：重度 0x6：严重
	int[1]车外 PM2.5 等级	同上	同上

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[]	int[0]车内 PM2.5 数值	[PM2P5_VALUE_MIN ,PM2P5_VALUE_MAX]	{0 3000} ug/m³
	int[1]车外 PM2.5 数值	同上	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			
返回类型	方法	描述	输入
void	onPM2p5CheckStateChanged(int state_in,int state_out))	监听车内/车外 PM2.5 检测状态	
void	onPM2p5LevelChanged(int level_in,int level_out)	监听车内/车外 PM2.5 等级	
void	onPM2p5ValueChanged(int value_in,int value_out)	监听车内/车外 PM2.5 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
int	getEnginePower()	获取功率(发动机和电机)
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(AbsBYDAutoEngineListener l)	监听注册接口
void	unregisterListener(AbsBYDAutEngineListener l)	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	获取发动机排量		
输入参数	无		
返回值			
数据类型	名称	值	描述
double	排量	[ENGINE_DISPLACEMENT_MIN, ENGINE_DISPLACEMENT_MAX]	[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		
返回值			
数据类型	名称	值	描述
String	发动机型号	0x00：371QA 0x01：473QB 0x02：473QC 0x03：473QD	型号名称

		0x04：476ZQA 0x05：483QA 0x06：483QB 0x07：483QB CNG 0x08：487ZQA 0x09：488QA 0x0A:4G15 0x0B:4G18 0x0C:4G69 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
int	功率 (发动机和发电机)	[ENGINE_POWER_MIN, 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
int	转速	[ENGINE_POWER_MIN, 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	值	描述
int	冷却液位	ENGINE_COOLANT_LEVEL_LOW	过低
		ENGINE_COOLANT_LEVEL_NORMAL	正常

6.8.7 获取燃油油位信号

API 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
int	油位	[ENGINE_OIL_MIN, ENGINE_OIL_MAX]	[0,254]

6.8.8 监听方法

监听类 AbsBYDAutoEngineListener			
public class AbsBYDAutoEngineListener			
当监听的对象数值发生变化时，推送给用户			
方法概要			
返回类型	方法	描述	输入
void	onEngineSpeedChanged(int value)	监听发动机转速变化	发动机转速
void	onEngineCoolantLevelChanged(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	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 classBYDAutoGearboxDevice		
方法概要		
限定符和返回类型	方法	描述
staticBYDAutoGearboxDevice	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	getGearboxAutoModeType()	获取自 动变 速箱档位
int	getGearboxManualModeLevel()	获 取手 动变 速箱档位
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(AbsBYDAutoGearboxListener l)	监听注册接口
void	unregisterListener(AbsBYDAutGearboxLi	取消注册接口

	stener l)	

6.9.2 获取变速箱型号

API 名称	String getGearboxCode()		
接口描述	获取变速箱型号		
输入参数	无		
返回值			
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 0x09：5RT10 0x0A：5RT14 0x0B：6T25 0x0C：6DT25	型号名称

		0x0D：6DT33 0x0E：6T18 Ox0F：6DT35	

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	无		
返回值			
数据类型	名称	值	描述
int	变速箱类型	GEARBOX_TYPE_MT	MT,手动档
		GEARBOX_TYPE_AMT	AMT，半自动变速箱
		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
int	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
		GEARBOX_AUTO_MODE_D	D - 自动档前进
		GEARBOX_AUTO_MODE_S	S
		GEARBOX_AUTO_MODE_M	M

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
int	手动变速箱档位	GEARBOX_REAL_LEVEL_D	前进挡
		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
int	制动液位	GEARBOX_BRAKE_FLUID_LEVEL_LOW	过低
		GEARBOX_BRAKE_FLUID_LEVEL_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
int	驻车制动开关状态	GEARBOX_PARK_BREAK_SWITCH_VALID	驻车有效

		GEARBOX_PARK_BREAK_SWITCH_INVALID	驻车无效

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
int	制动踏板状态	GEARBOX_BREAK_PADAL_PRESS	踩下
		GEARBOX_BREAK_PADAL_NOT_PRESS	未踩下

6.9.9 监听方法

监听类 AbsBYDAutoGearboxListener		
public class AbsBYDAutoGearboxListener		
当监听的对象数值发生变化时，推送给用户		
方法概要		
返回类型	方法	描述
void	onGearboxAutoModeTypeChanged(int level)	监听自动变速箱档位变化

void	onGearboxManualModeLevelChanged(int 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 BYDAutoDoorLockDevice		
方法概要		
限定符和返回类型	方法	描述
Static BYDAutoDoorLockDevice	getInstance(Context con)	获取实例
int	getDoorLockStatus(int area)	获取各门锁状态
void	registerListener(AbsBYDAutoDoorLockListener l)	监听注册接口
void	unregisterListener(AbsBYDAutoDoorLockListener l)	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
int	area	门锁区域	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_FRONT	右前门门锁状态
			DOOR_LOCK_AREA_RIGHT_REAR	右后门门锁状态
			DOOR_LOCK_AREA_BACK	后背门门锁状态
			DOOR_LOCK_AREA_CHILDLOCK_LEFT	左儿童锁位置信息
			DOOR_LOCK_AREA_CHILDLOCK_RIGHT	右儿童锁位置信息
返回值				
数据类型	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 abstract class AbsBYDAutoDoorLockListener			
方法概要			
限定符和返回类型	方法	描述	输入
void	onDoorLockStatusChanged(int area, int state)	监听各 区域门锁状态的变化	区域 (左前门门锁，左后门门锁，右前门门锁，右后门门锁，后背门门锁，左儿童锁， 右儿童锁) 门锁的状态(闭锁，解锁)
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 l)	监听注册接口
void	unregisterListener(AbsBYDAutoLightListener l)	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
int	灯光 AUTO 档开关状态	LIGHT_ON	打开
		LIGHT_OFF	关闭

6.11.3 获取各车灯状态

API 名称	int getLightStatus(int type)

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
int	type	车灯类型	LIGHT_SIDE	小灯
			LIGHT_LOW_BEAM	近光灯
			LIGHT_HIGH_BEAM	远光灯
			LIGHT_LEFT_TURN_SIGNAL	左转向灯
			LIGHT_RIGHT_TURN_SIGNAL	右转向灯
			LIGHT_FRONT_FOG	前雾灯
			LIGHT_REAR_FOG	后雾灯
			LIGHT_FOOT	照脚灯(外后视镜下的灯)
返回值				
数据类型	名称		值	描述
int	车灯状态		LIGHT_ON	打开
			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	名称	值	描述
int	AFS 开关状态（自适应转向大灯系统）	LIGHT_ON	打开
		LIGHT_OFF	关闭

6.11.5 监听方法

监听类 AbsBYDAutoLightListener			
public class AbsBYDAutoLightListener			
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(int state)	监听 AFS 开关状态变化	打开/关闭
备注	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 BYDAutoSafetyBeltDevice		
方法概要		
限定符和返回类型	方法	描述
Static BYDAutoSafetyBeltDevice	getInstance(Context con)	获取实例
int	getSafetyBeltStatus(int area)	获取各位置安全带状态
int	getPassengerStatus(int area)	获取各位置的乘客状态
void	registerListener(AbsBYDAutoSafetyBeltListener l)	监听注册接口
void	unregisterListener(AbsBYDAutoSafetyBeltListener l)	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

输入参数				
数据类型	字段名	名称	值	描述
int	area	安全带区域	SAFETY_BELT_AREA_MAIN	驾驶员安全带
			SAFETY_BELT_AREA_DEPUTY	副驾驶安全带
			SAFETY_BELT_AREA_SECOND_ROW_SEAT_LEFT	左后二排安全带
			SAFETY_BELT_AREA_SECOND_ROW_SEAT_RIGHT	右后二排安全带
			SAFETY_BELT_AREA_SECOND_ROW_SEAT__MID	中后二排安全带
返回值				
数据类型	名称		值	描述
int	安全带状态		SAFETY_BELT_STATE_INVALID	无效
			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

int	area	座椅区域	SAFETY_BELT_PASSENGER_DEPUTY	副驾驶位座椅
			SAFETY_BELT_PASSENGER_SECOND_ROW_SEAT_LEFT	左后二排座椅
			SAFETY_BELT_PASSENGER_SECOND_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	名称		值	描述
int	乘客状态		SAFETY_BELT_PASSENGER_STATE_INVALID	无效
			SAFETY_BELT_PASSENGER_STATE_SOMEBODY	有人
			SAFETY_BELT_PASSENGER_STATE_NOBODY	无人

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			
void	onSafetyBeltStatusChanged(int area, int state)	监听各 位置安全带状态的变化	区域 (驾驶员安全带，副驾驶安全带，左后二排安全带，右后二排安全带，中后二排安全带) 安全带的状态(扣上，解开)
void	onPassengerStatusChanged(int area, int state)	监听各 位置乘客状态	座椅区域(副驾驶位座椅，左后二排座椅，右后二排座椅，中后二排座椅) 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
int[]	getAllRadarProbeStates()	获取所有位置的探头状态
int	getReverseRadarSwitchState()	获取倒车雷达开关状态
void	registerListener(AbsBYDAutoRadarListener l)	监听注册接口
void	unregisterListener(AbsBYDAutoRadarListener l)	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			
输入参数				
数据类型	字段名	名称	值	描述
int	area	探头位置	RADAR_AREA_LEFT	左探头
			RADAR_AREA_LEFT_FRONT	左前探头
int	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
			RADAR_AREA_RIGHT	右探头
			RADAR_AREA_RIGHT_FRONT	右前探头

			RADAR_AREA_FRONT_RIGHT_MID	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
int	探头状态		RADAR_COMMAND_INVALID_VALUE	无效
			RADAR_PROBE_STATE_ABNORMAL	传感器异常
			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	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] 右前探头状态 state[2] 左后探头状态 state[3] 右后探头状态 state[4] 左探头状态 state[5] 右探头状态 state[6] 前左中探头状态 state[7] 前右中探头状态
		RADAR_PROBE_STATE_ABNORMAL	
		RADAR_PROBE_STATE_SAFE	
		RADAR_PROBE_STATE_GREEN	
		RADAR_PROBE_STATE_YELLOW	
		RADAR_PROBE_STATE_RED	
		RADAR_COMMAND_FAILED	

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	值	描述
int	倒车雷达开关状态	RADAR_REVERSE_SWITCH_ON	打开
		RADAR_REVERSE_SWITCH_OFF	关闭

6.13.5 监听方法

类 AbsBYDAutoRadarListener			
public abstract class AbsBYDAutoRadarListener			
方法概要			
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，右探头，右前探头，前右中位置探头，右 后 探头)； state：探头状态(传感器异常，安全，绿色警告，黄，红)
void	onReverseRadarSwitchStateChanged(int state)	监听倒 车雷达开关状态	打开/关闭
备注	输入值的取值范围参见 get 函数的具体描述		

6.14 充电类

6.14.1 方法概要

类BYDAutoChargingDevice		
public classBYDAutoChargingDevice		
方法概要		
限定符和返回类型	方法	描述
Static BYDAutoChargingDevice	getInstance(Context con)	获取实例
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 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
int	getBatteryManagementDeviceState()	获取动力电池管理器与外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
int[]	getChargingScheduleTime()	获取预约充电倒计时(时，分)
void	registerListener(AbsBYDAutoChargingListener l)	监听注册接口
void	unregisterListener(AbsBYDAutoChargingListener l)	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		
返回值			
数据类型	名称	值	描述
int	车载充电器故障状态	CHARGING_FAULT_STATE_NORMAL	正常
		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	值	描述
int	车载充电器工作状态	CHARGING_WORK_STATE_READY	准备就绪
		CHARGING_WORK_STATE_START	充电开始
		CHARGING_WORK_STATE_FINISH	充电结束
		CHARGING_WORK_STATE_TERMINATE	充电终止

6.14.4 获取本次累计总充电量

API 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,CHARGING_CAPACITY_MAX]KWH	[0, 65.534]KWH

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	值	描述
int	充电模式	CHARGING_TYPE_DEFAULT	默认
		CHARGING_TYPE_AC	交流充电
		CHARGING_TYPE_VTOG	VTOG 充电
		CHARGING_TYPE_GB_DC	国标直流充电
		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	无		
返回值			
数据类型	名称	值	描述
int[]	当前充满电剩余时间：小时	[REST_HOUR_MIN,REST_HOUR_MAX]h	time[0]: [0,254]h

	当前充满电剩余时间：分钟	[MINUTE_MIN,MINUTE_MAX]min	time[1]: [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
int	type	充电口盖	CHARGING_CAP_AC	交流充电口盖
			CHARGING_CAP_DC	直流充电口盖
返回值				
数据类型	名称		值	描述
int	充电口盖开关状态		CHARGING_CAP_STATE_ON	打开
			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
int	充电口电锁执行状态反馈	CHARGING_PORT_STATE_LOCK_FINISH	开锁完成
		CHARGING_PORT_STATE_LOCK_INVALID	开锁失效
		CHARGING_PORT_STATE_UNLOCK_FINISH	闭锁完成
		CHARGING_PORT_STATE_UNLOCK_INVALID	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	无		
返回值			
数据类型	名称	值	描述
int	放电请求状态	DISCHARGE_STATE_NON	无放电请求
		DISCHARGE_STATE_HOUSEHOLD_APPLIANCE	家用设备
		DISCHARGE_STATE_THREE_PHASE_EQUIPMENT	三相设备

		DISCHARGE_STATE_THREE_PHASE_VEHICLE	三相车辆
		DISCHARGE_STATE_POWER_SYSTEM	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
int	充电器状态	CHARGING_CHARGER_STATE_NOT_CONNECTED	未连接
		CHARGING_CHARGER_STATE_CONNECTED	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
int	充电枪连接状态	CHARGING_GUN_STATE_CONNECTED_NONE	未连接
		CHARGING_GUN_STATE_CONNECTED_AC	交流充电枪连接
		CHARGING_GUN_STATE_CONNECTED_DC	直流充电枪连接
		CHARGING_GUN_STATE_CONNECTED_AC_DC	交流与直流充电枪同时连接
		CHARGING_GUN_STATE_CONNECTED_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

	前总电压和总电流相乘)	CHARGING_POWER_MAX]	

6.14.13 获取动力电池管理器与外接充电设备当前状态

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
int	动力电池管理器与外接充电设备当前状态	CHARGING_BATTERY_STATE_READY	准备就绪
		CHARGING_BATTERY_STATE_CHARGING	正在充电
		CHARGING_BATTERY_STATE_CHARG_FINISH	充电结束
		CHARGING_BATTERY_STATE_DISCHARG	正在放电
		CHARGING_BATTERY_STATE_CHARG_TERMINATE	充电终止
		CHARGING_BATTERY_STATE_BREAKDOWN_C10	C10 充电柜故障
		CHARGING_BATTERY_STATE_BREAK	充电枪连接故障

		DOWN_CHARGING_GUN	
		CHARGING_BATTERY_STATE_BREAKDOWN_CHARGER	车载充电器故障
		CHARGING_BATTERY_STATE_BREAKDOWN_AC	交流外充设备故障
		CHARGING_BATTERY_STATE_SCHEDULE	预约等待
		CHARGING_BATTERY_STATE_DISCHARG_CBU	正在给整车放电
		CHARGING_BATTERY_STATE_TIMEOUT	与充电柜通讯超时
		CHARGING_BATTERY_STATE_DISCHARG_FINISH	放电结束

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
int	定时充电功能状态	CHARGING_STATE_ENABLE	定时充电允许
		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_INVALID	无效态
		CHARGING_SCHEDULE_STATE_CANCEL	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_REMOTE	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

int	充电枪未插提醒状态	CHARGING_GUN_STATE_ON	提醒
		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
int[]	预约充电倒计时（小时）	[HOUR_MIN, HOUR_MAX]h	time[0]: 0-23 时
	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 abstract class AbsBYDAutoChargingListener			
方法概要			
限定符和返回类型	方法	描述	输入

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)	监听车载充电器工作状态的变化	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	onChargingPortLockRebackStateChang	监听充电口电	state(开锁完成，开锁

	ed(int state)	锁执行反馈的变化	失效， 闭锁完成，闭锁失效)
void	onDischargeRequestStateChanged(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)	监听充电枪连接状态的变化	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	onChargingPowerChanged(double value)	监听充电功率的变化	功率
void	onBatteryManagementDeviceStateChanged(int state)	监听动力电池管理器与外接充电设备当前状态的变化	state(准备就绪,正在充电,充电结束,正在放电,充电终止,C10 充电柜故障,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)
void	onChargingScheduleEnableStateChanged(int state)	监听定时充电功能状态的变化	state(充电允许，充电不允许)
void	onChargingScheduleStateChanged(int state)	监听预约充电状态的变化	state(无效态,取消预约充电, 无预约充电,本地预约充电,远程预约充电)
void	onChargingGunNotInsertedStateChanged(int state)	监听充电枪未插提醒状态的变化	state(提醒,不提醒)
void	onChargingScheduleTimeChanged(int hour,int min)	监听预约充电倒计时的变化	hour: 0-23 时 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

类 BYDAutoTyreDevice

public class BYDAutoTyreDevice		
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
int	getTyreBatteryState()	获取胎压系统电池电量状态
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
void	registerListener(AbsBYDAutoTyreListener l)	监听注册接口
void	unregisterListener(AbsBYDAutoTyreListener l)	取消注册接口

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
int	area	轮胎位置	TYRE_COMMAND_AREA_LEFT_FRONT	左前轮胎
			TYRE_COMMAND_AREA_RIGHT_FRONT	右前轮胎
			TYRE_COMMAND_AREA_LEFT_REAR	左后轮胎
			TYRE_COMMAND_AREA_RIGHT_REAR	右后轮胎
返回值				
数据类型	名称		值	描述
int	探头状态		TYRE_AIR_LEAK_STATE_NORMAL	正常
			TYRE_AIR_LEAK_STATE_QUICK	快速漏气
			TYRE_AIR_LEAK_STATE_SLOW	慢速漏气
			TYRE_COMMAND_INVALID_VALUE	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
int	胎压系统电池电量状态	TYRE_BATTERY_STATE_NORMAL	正常
		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	名称	值	描述
int	area	轮胎位置	TYRE_COMMAND_AREA_LEFT_FRONT	左前轮胎
			TYRE_COMMAND_AREA_RIGHT_FRONT	右前轮胎
			TYRE_COMMAND_AREA_LEFT_REAR	左后轮胎
			TYRE_COMMAND_AREA_RIGHT_REAR	右后轮胎
返回值				
数据类型	名称		值	描述

int	轮胎压力状态	TYRE_PRESSURE_STATE_NORMAL	正常
		TYRE_PRESSURE_STATE_OVERPRESSURE	过压
		TYRE_PRESSURE_STATE_UNDERPRESSURE	欠压
		TYRE_COMMAND_INVALID_VALUE	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			
输入参数				
数据类型	字段名	名称	值	描述
int	area	轮胎位置	TYRE_COMMAND_AREA_LEFT_FRONT	左前轮胎
			TYRE_COMMAND_AREA_RIGHT_FRONT	右前轮胎
			TYRE_COMMAND_AREA_LEFT_REAR	左后轮胎
			TYRE_COMMAND_AREA_RIGHT_REAR	右后轮胎

返回值			
数据类型	名称	值	描述
int	轮胎压力值	[TYRE_PRESSURE_VALUE_MIN,TYRE_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	名称	值	描述
int	area	轮胎位置	TYRE_COMMAND_AREA_LEFT_FRONT	左前轮胎
			TYRE_COMMAND_AREA_RIGHT_FRONT	右前轮胎
			TYRE_COMMAND_AREA_LEFT_REAR	左后轮胎
			TYRE_COMMAND_AREA_RIGHT_REAR	右后轮胎
返回值				
数据类型	名称		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_VALUE	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
int	胎压系统状态	TYRE_SYSTEM_STATE_NORMAL	正常
		TYRE_SYSTEM_STATE_SELF_CHECKING	自检
		TYRE_SYSTEM_STATE_SIGNAL_ANOMAL	信号异常
		TYRE_SYSTEM_STATE_BREAKDOWN	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_NORMAL	正常
		TYRE_TEMPERATURE_STATE_SUPER_HIGH	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 abstract class AbsBYDAutoTyreListener			
方法概要			
限定符和返回类型	方法	描述	输入
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)
void	onTyreBatteryStateChanged(int state)	监听胎压系统电池电量状态的变化	状态(正常，低电量)
void	onTyrePressureStateChanged(int area,int state)	监听各轮胎压力状态的变化	轮胎位置(左前轮胎，右前轮胎，左后轮胎，右后轮胎) 压力状态(正常，过压，欠压)
void	onTyrePressureValueChanged(int area,int value)	监听各轮胎压力值的变化	轮胎位置(左前轮胎，右前轮胎，左后轮胎，右后轮胎) 压力值([0,4094]kpa)
void	onTyreSignalStateChanged(int area,int state)	监听各轮胎信号状态的变化	轮胎位置(左前轮胎，右前轮胎，左后轮胎，右后轮胎) 信号状态(正常，异常)

void	onTyreSystemStateChanged(int state)	监听胎压系统状态的变化	系统状态(正常，自检，信号异常，故障，屏蔽状态)
void	onTyreTemperatureStateChanged(int state)	监听胎压系统温度状态的变化	温度状态(正常，超高温，高温，胎温显示休眠)
备注	输入值的取值范围参见 get 函数的具体描述		

6.16 仪表类

6.16.1 方法概要

类 BYDAutoInstrumentDevice		
public classBYDAutoInstrumentDevice		
方法概要		
限定符和返回类型	方法	描述
staticBYDAutoInstrumentDevice	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
void	registerListener(AbsBYDAutoInstrumentListener l)	监听注册接口
void	unregisterListener(AbsBYDAutoInstrumentListener l)	取消注册接口

6.16.2 获取故障提示信息

API 名称	int getMalfunctionInfo(int typeName)			
接口描述	获取故障提示信息			
输入参数				
数据类型	字段名	名称	值	描述
int	type	类型	MALFUNCTION_MACHINE_OIL_LOW_PRESSURE	机油压力低
			MALFUNCTION_PARKING_BRAKE	驻车制动故障
			MALFUNCTION_CHARGING_SYSTEM	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_LEAK	快速漏气
			MALFUNCTION_HIGH_WATER_TEMPERATURE	水温高
			MALFUNCTION_ELECTRIC_PARKING_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
			MALFUNCTION_HIGH_MOTOR_TEMPERATURE	电机过高
			MALFUNCTION_BATTERY	电池故障
			MALFUNCTION_HIGH_BATTERY_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
			MALFUNCTION_SMART_KEY	智能钥匙系统告警灯
			MALFUNCTION_FRONT_BELT	前排座椅安全带未系提示
返回值				
数据类型	名称		值	描述
int	故障状态		DEVICE_HAS_THE_MALFUNCTION	有故障/提示
			DEVICE_NOT_HAS_THE_MALFUNCTION	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
int	蜂鸣器状态	INSTRUMENT_ALARM_BUZZLE_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
int	unitName	类型	TEMPERATURE_UNIT	温度
			PRESSURE_UNIT	气压
			FUEL_CONSUMPTION_AND_DISTANCE_UNIT	油耗距离
			POWER_UNIT	功率
返回值				
数据类型	名称		值	描述
int	单位		DEGREE_CENTIGRADE/DEGREE_FAHRENHEIT	单位(温度) 摄氏度/华氏温度
			BAR/PSI/KPA	单位(气压) 0x1:bar 0x2:psi 0x3:kpa
			L_P_100KM_AND_KM/	单位(油耗距离)

		KM_P_L_AND_KM/ MPG_GB_AND_MILE/ MPG_US_AND_MILE/ KWH_P_100KM_AND_KM/ KWH_P_100MI_AND_MILE	0x1:L/100km,km 0x2:km/L,km 0x3:mpg(GB),mile 0x4:mpg(US),mile 0x5：KWH/100km，km 0x6:KWH/100Mi，mile
		KW/HP	单位(功率) 0x1:kW 0x2:HP
备注	油耗距离单位目前仅支持 1、3、4		

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

int	typeName	类型	MAINTENANCE_TIME	保养时间
			MAINTENANCE_MILEAGE	保养里程
返回值				
数据类型	名称		值	描述
int	保养时间		[MAINTENANCE_TIME_DAY_MIN, MAINTENANCE_TIME_DAY_MAX]	[0,720]天
	保养里程		[MAINTENANCE_MILEAGE_KILOMETER_MIN, MAINTENANCE_MILEAGE_KILOMETER_MAX]	[0,20000]km

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
int	unitName	类型	TEMPERATURE_UNIT	温度
			PRESSURE_UNIT	气压
			FUEL_CONSUMPTION_AND_DISTANCE_UNIT	油耗距离
			POWER_UNIT	功率

int	unitValue	值	DEGREE_CENTIGRADE/DEGREE_FAHRENHEIT	单位(温度) 摄氏度/华氏温度
			BAR/PSI/KPA	单位(气压) 0x1:bar 0x2:psi 0x3:kpa
			L_P_100KM_AND_KM/ KM_P_L_AND_KM/ MPG_GB_AND_MILE/ MPG_US_AND_MILE/ KWH_P_100KM_AND_KM/ KWH_P_100MI_AND_MILE	单位(油耗距离) 0x1:L/100km,km 0x2:km/L,km 0x3:mpg(GB),mile 0x4:mpg(US),mile 0x5: KWH/100km，km 0x6:KWH/100Mi，mile
			KW/HP	单位(功率) 0x1:kW 0x2:HP

返回值			
数据类型	名称	值	描述
int	设置结果	INSTRUMENT_COMMAND_SUCCESS	成功
		INSTRUMENT_COMMAND_FAILED;	失败
		INSTRUMENT_COMMAND_INVALID;	无效输入
		INSTRUMENT_COMMAND_TIMEOUT;	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	设置保养时间/里程			
输入参数				
数据类型	字段名	名称	值	描述
int	typeName	类型	MAINTENANCE_TIME	保养时间
			MAINTENANCE_MILEAGE	保养里程
int	infoValue	值	[MAINTENANCE_TIME_DAY_MIN, MAINTENANCE_TIME_DAY_MAX]	[0,720]天
			[MAINTENANCE_MILEAGE_KILOME	[0,20000]km

			TER_MIN, MAINTENANCE_MILEAGE_KILOMETER_MAX]	
返回值				
数据类型	名称		值	描述
int	设置结果		INSTRUMENT_COMMAND_SUCCESS	成功
			INSTRUMENT_COMMAND_FAILED;	失败
			INSTRUMENT_COMMAND_INVALID;	无效输入
			INSTRUMENT_COMMAND_TIMEOUT;	超时

6.16.8 获取外接充电量

API 名称	double getExternalChargingPower()		
接口描述	获取外接充电量		
输入参数	无		
返回值			
数据类型	名称	值	描述
double	外接充电量	[EXTERNAL_CHARGING_POWER_MIN,	{0.0, 10000.0} kw.h

		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 classAbsBYDAutoInstrumentListener			
方法概要			
限定符和返回类型	方法	描述	输入
void	onMalfunctionInfoChanged(int typeName, int hasMalfunction)	监听故障提示信息	类型和是否故障
void	onAlarmBuzzleStateChange(int state)	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	onExternalChargingPowerChanged(double value)	监听外接充电量的变化	外接充电量
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 classBYDAutoTimeDevice		
方法概要		
限定符和返回类型	方法	描述
staticBYDAutoTimeDevice	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 weekday)	设置年月日、星期
int	setTime(int hour,int minute,int second)	设置时分秒
int	setTimeFormat(int value)	设置时制
void	registerListener(AbsBYDAutoTimeListener l)	监听注册接口
void	unregisterListener(AbsBYDAutoTimeListener 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[]	int[0] 年	[TIME_YEAR_MIN,TIME_YEAR_MAX]	[2001,2255]
	int[1] 月	[TIME_MONTH_MIN,TIME_MONTH_MAX]	[1,12]
	int[2] 日	[TIME_DAY_MIN,TIME_DAY_MAX]	[1,31]
	int[3] 时	[TIME_HOUR_MIN,TIME_HOUR_MAX]	[0,23]
	int[4] 分	[TIME_MINUTE_MIN,TIME_MINUTE_MAX]	[0,59]
	int[5] 秒	[TIME_SECOND_MIN,TIME_SECOND_MAX]	[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

API 名称	int setDate(int year,int month,int day,int weekday)			
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
int	year	年	[TIME_YEAR_MIN, TIME_YEAR_MAX]	[2001,2255]
	month	月	[TIME_MONTH_MIN, TIME_MONTH_MAX]	[1,12]
	day	日	[TIME_DAY_MIN, TIME_DAY_MAX]	[1,31]
	weekday	星期	[TIME_WEEKDAY_MIN, TIME_WEEKDAY_MAX]	[1,7]
返回值				
数据类型	名称		值	描述

int	设置结果	TIME_COMMAND_SUCCESS	成功
		TIME_COMMAND_FAILED;	失败
		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
int	hour	时	[TIME_HOUR_MIN,TIME_HOUR_MAX]	[0,23]
	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
int	设置结果		ENERGY_COMMAND_SUCCESS	成功
			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
int	value	时制	TIME_FORMAT_24H_ON	24H
			TIME_FORMAT_24H_OFF	12H
返回值				
数据类型	名称		值	描述
int	设置结果		ENERGY_COMMAND_SUCCESS	成功
			ENERGY_COMMAND_FAILED;	失败
			ENERGY_COMMAND_INVALID;	无效输入
			ENERGY_COMMAND_TIMEOUT;	超时
			ENERGY_COMMAND_BUSY.	系统忙

6.17.7 监听方法

类 AbsBYDAutoTimeListener

public abstract class AbsBYDAutoTimeListener			
方法概要			
限定符和返回类型	方法	描述	输入
void	onTimeChanged(int[] time)	监听时间变化	year,month,day,hour,minute,second 变化
void	onTimeFormatChanged(int value)	监听时制变化	时制 12H/24H
备注	输入值的取值范围参见 get 函数的具体描述		

6.18 车辆设置类

6.18.1 方法概要

类 BYDAutoSettingDevice			
public classBYDAutoSettingDevice			
方法概要			
限定符和返回类型	方法	描述	值 (get 的返回值， set 的输入值)
staticBYDAutoSettin	getInstance(Context con)	获取实例	返回

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

int	setPM25TimeCheck(int value)	(开/关，默认关)	SET_OFF
高压			
int	getEnergyFeedback()	能量回馈强度设置(标准/较大)，默认标准	SET_DR_ENERGY_FB_STANDARD, SET_DR_ENERGY_FB_LARGE
int	setEnergyFeedback(int value)		
int	getSOCTarget()	SOC 目标点设置，可设置范围 15%~70%，默认 25%	[SET_DR_SOC_TARGET_MIN ,SET_DR_SOC_TARGET_MAX]
int	setSOCTarget(int value)		
信息媒体			
int	getChargingPort()	充电枪电锁工作模式(启用防盗、停用防盗)，默认启用防盗	SET_ON SET_OFF
int	setChargingPort(int value)		
int	getAutoExternalRearMirrorFollowUpSwitch()	外后视镜随动(开或关)，默认开	SET_ON SET_OFF
int	setAutoExternalRearMirrorFollowUpSwitch(int value)		
int	getLockOff()	开锁方式(四门、仅驾驶	SET_CAR_LOCK_OFF_4

int	setLockOff(int value)	员侧)，默认四门	DOORS SET_CAR_LOCK_OFF_ DR_SIDE
int	getLanguage()(仅支持简体中文)	语言(简体中文，繁体中文，English，俄语，阿拉伯语， 默认简体中文)	SET_LANGUAGE_SIMPLE_CHINESE SET_LANGUAGE_COMPLEX_ CHINESE SET_LANGUAGE_ENGLISH SET_LANGUAGE_RUSSIAN SET_LANGUAGE_ARABIC
	setLanguage(int value) (仅支持简体中文)		
int	getOverspeedLock()	超速闭锁(开或关)，默认开	SET_OFF SET_ON
int	setOverspeedLock(int value)		
int	getSafeWarnState()	获取安全警告标志位(警告/不警告， “请切换到 P 挡域”满足时的提示)	SET_ON:警告 SET_OFF:不警告
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（Comfort）、运动（Sport）)，默认舒适，交互需要区分是否有全地形	SET_DR_ST_ASSIS_COMFORT SET_DR_ST_ASSIS_SPORT
int	setSteerAssis(int value)		
辅电			
int	getRearViewMirrorFlip()	倒车外后视镜翻转(开或关)，默认开	SET_OFF、SET_ON
int	setRearViewMirrorFlip (int value)		
int	getDriverSeatAutoReturn()	驾驶员座椅自动回位(开或关)，默认开	SET_OFF、SET_ON
int	setDriverSeatAutoReturn(int value)		
int	getSteerPositionAutoReturn()	转向盘位置自动恢复(开或关)，默认开	SET_OFF、SET_ON
int	setSteerPositionAutoReturn(int value)		
int	getRemoteControlUpwin	遥控升窗(开或关)，默认	SET_OFF、SET_ON

	dowState()	开，不带防夹时无此项	
int	setRemoteControlUpwindowState(int value)		
int	getRemoteControlDownwindowState()	遥控降窗(开或关)，默认关	SET_OFF、SET_ON
int	setRemoteControlDownwindowState(int value)		
int	getLockCarRiseWindow()	锁车自动关窗(开或关)，默认关，不带防夹时无此项	SET_OFF、SET_ON
int	setLockCarRiseWindow(int value)		
int	getMicroSwitchLockWindowState()	长按微动开关闭锁升窗(开或关)，默认开，不带防夹时无此项	SET_OFF、SET_ON
int	setMicroSwitchLockWindowState(int value)		
int	getMicroSwitchUnlockWindowState()	长按微动开关解锁降窗(开或关)，默认关	SET_OFF、SET_ON
int	setMicroSwitchUnlockWindowState(int value)		
int	getBackHomeLightDelayValue()	回家照明延时（10s、20s、30s、40s、 50s、60s），默认 10s,0s 表示	[SET_CAR_LIGHT_DELAY_VALUE_MIN, SET_CAR_LIGHT_DELA
int	setBackHomeLightDelayV		

	alue(int value)	关闭	Y_VALUE_MAX]
int	getLeftHomeLightDelayValue()	离家照明延时（10s、20s、30s、40s、 50s、60s），默认 10s，0s 表示关闭	[SET_CAR_LIGHT_DELAY_VALUE_MIN, SET_CAR_LIGHT_DELAY_VALUE_MAX]
int	setLeftHomeLightDelayValue(int value)		
int	getBackDoorElectricMode()	后背门电动功能设置状态（电动或手动）	SET_OFF--manual SET_ON--electric
int	setBackDoorElectricMode(int value)		
void	registerListener(AbsBYDAutoSettingListener l)	监听注册接口	无
void	unregisterListener(AbsBYDAutoSettingListenerl)	取消注册接口	无
备注	set 函数的返回值包括： 成功=SETTING_COMMAND_SUCCESS 失败=SETTING_COMMAND_FAILED 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,表示无此设置项		

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	值	描述
String	feature	功能	FEATURE_OVERSPEED_LOCKING	超速闭锁
			FEATURE_BACK_DOOR	电动后背门
			FEATURE_REARVIEW_MIRROR_FOLLOW_UP	外后视镜随动(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	结果		DEVICE_HAS_THE_FEATURE	有
			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

int	是否在线	SET_OFF	不在线(未检测到/没有该模块)
		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 AbsBYDAutoSettingListener		
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

	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
void	onChargingPortSwitchChanged(int state)	监听充电枪电锁工作模式变化 (启用防盗、停用防盗)
void	onAutoExternalRearMirrorFollowUpSwitchChanged(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)
void	onLanguageChanged(int value) （仅支持简体中文）	监听语言(简体中文，繁体中文，English，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））

void	onRearViewMirrorFlipSwitchChanged(int state)	监听倒车外后视镜翻转开关变化
void	onDriverSeatAutoReturnSwitchChanged(int state)	监听驾驶员座椅自动回位开关变化
void	onSteerPositionAutoReturnSwitchChanged(int state)	监听转向盘位置自动恢复开关变化
void	onRemoteControlUpwindowStateChanged(int state)	监听遥控升窗开关变化，不带防夹时无此项
void	onControlWindowSwitchChanged(int state)	监听遥控降窗开关变化
void	onLockCarRiseWindowChanged(int state)	监听锁车自动关窗开关变化，不带防夹时无此项
void	onMicroSwitchLockWindowStateChanged(int state)	监听长按微动开关闭锁升窗开关变化，不带防夹时无此项
void	onMicroSwitchUnlockWindowStateChanged(int state)	监听长按微动开关解锁降窗开关变化
void	onBackHomeLightDelayValueChanged(int value)	监听回家照明延时时间变化（10s、20s、30s、40s、 50s、60s），0s 表示关闭
void	onLeftHomeLightDelayValueChanged(int value)	监听离家照明延时时间变化（10s、20s、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		
public class BYDAutoSensorDevice		
方法概要		
限定符和返回类型	方法	描述
static BYDAutoSensorDevice	getInstance(Context con)	获取实例
int	getLightIntensity()	获取光照(环境光)强度等级
void	registerListener(AbsBYDAutoSensorListener l)	监听注册接口
void	unregisterListener(AbsBYDAutoSensorListener 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
int	等级	LIGHT_ILLUM_LEVEL1	大于 230Lux
		LIGHT_ILLUM_LEVEL2	大于 180Lux，小 230Lux
		LIGHT_ILLUM_LEVEL3	大于 130Lux，小 180Lux
		LIGHT_ILLUM_LEVEL4	大于 80Lux，小 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)	监听光照强度变化	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 BYDAutoMultimediaDevice		
方法概要		
限定符和返回类型	方法	描述
static BYDAutoMultimediaDevice	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)	
void	unregisterListener(AbsBYDAutoMultimediaListener l)	取消注册接口
备注	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
int	播放类型	MULTIMEDIA_TYPE_AM	AM
		MULTIMEDIA_TYPE_FM	FM
		MULTIMEDIA_TYPE_CD	CD
		MULTIMEDIA_TYPE_VCD	VCD
		MULTIMEDIA_TYPE_DVD	DVD
		MULTIMEDIA_TYPE_TV	TV
		MULTIMEDIA_TYPE_AUDIO_OFF	Audio Off (音响关闭)
		MULTIMEDIA_TYPE_AUX	AUX

		MULTIMEDIA_TYPE_USB_AUDIO	USB 音频
		MULTIMEDIA_TYPE_USB_VIDEO	USB 视频
		MULTIMEDIA_TYPE_SD_AUDIO	SD 音频
		MULTIMEDIA_TYPE_SD_VIDEO	SD 视频
		MULTIMEDIA_TYPE_HD_AUDIO	HD（硬盘)音频
		MULTIMEDIA_TYPE_HD_VIDEO	HD（硬盘)视频
		MULTIMEDIA_TYPE_LOCAL_AUDIO	本地音频
		MULTIMEDIA_TYPE_LOCAL_VIDEO	本地视频
		MULTIMEDIA_TYPE_BT	蓝牙音乐
		MULTIMEDIA_TYPE_ROBOT	Robot
		MULTIMEDIA_TYPE_INVAID	无效
备注	媒体中心未启动，或者被用户从近期任务清除，则返回 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

API 名称	int getPlayMode()		
接口描述	获取播放模式		
输入参数	无		
返回值			
数据类型	名称	值	描述
int	播放模式	MULTIMEDIA_PLAY_MODE_SINGLE_REPEAT	单曲重复播放
		MULTIMEDIA_PLAY_MODE_RANDOM	随机播放
		MULTIMEDIA_PLAY_MODE_PREVIEW	预览播放
		MULTIMEDIA_PLAY_MODE_SCAN	SCAN 状态
		MULTIMEDIA_PLAY_MODE_STEREO	广播播放立体声频道
		MULTIMEDIA_PLAY_MODE_ALL_REPEAT	全部重复播放

		MULTIMEDIA_PLAY_MODE_INVAID	无效
备注	媒体中心未启动，或者被用户从近期任务清除，则返回MULTIMEDIA_PLAY_MODE_INVAID， 否则返回媒体中心当前的播放模式。 FM： MULTIMEDIA_PLAY_MODE_STEREO 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	int getPlayState()		
接口描述	获取播放状态		
输入参数	无		
返回值			
数据类型	名称	值	描述
int	播放状态	MULTIMEDIA_STATE_PLAY	播放
		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
MediaInfo	String fileName		文件名称
	String artistName		歌手名称
	String albumName		专辑名称
备注	媒体中心未启动，或者被用户从近期任务清除，返回 NULL。 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

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
int	mode	MODE_RADIO	FM
		MODE_MUSIC	音乐
		MODE_VIDEO	视频
int	action	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_PLAY_PRE	音乐/视频播放上一首，FM 搜索上一台
		ACTION_PLAY_NEXT	音乐/视频播放下一首，FM 搜索下一台
		ACTION_SET_PLAY_PATTERN	音乐设置播放模式(循环，随机，单曲)，视频设置全屏半屏播放

		ACTION_AUTO_SEARCH		FM 自动搜索
		ACTION_CANCEL_RADIO_SEARCH		FM 停止搜台
MediaControlParam	param	PARAM_SOURCE	SOURCE_LOCAL	本地音乐/视频
			SOURCE_USB	U 盘音乐/视频
			SOURCE_SD	SD 音乐/视频
			SOURCE_BTMUSIC	蓝牙音乐
		PARAM_PATTERN	PATTERN_CYCLE	音乐全部
			PATTERN_RANDOM	音乐随机
			PATTERN_SINGLE	音乐单曲
			PATTERN_HALF_SCREEN	视频半屏播放
			PATTERN_FULL_SCREEN	视频全屏播放
		PARAM_RADIO_FREQ	FM 频道值，int 类型	FM 频道值

		PARAM_FILE_NAME	歌曲/视频名称， string 类型	歌曲/视频名称
		PARAM_ARTIST_NAME	艺术家名称，string 类型	歌曲所属艺术家名称
返回值				
数据类型	名称	值		描述
int	控制结果	MULTIMEDIA_COMMAND_SUCCESS		成功
		MULTIMEDIA_COMMAND_FAILED		失败
备注	MediaControlParam 是封装好的类型，里面的内容为键值对。 针对不同的 mode+action, param 可以填充一项或者多项，也可以不填充。			

6.20.7监听方法

类 AbsBYDAutoMultimediaListener			
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	方法	描述	输入

型			
void	onMediaTypeChanged(int type)	监听播放类型的变化	播放类型
void	onPlayModeChanged (int mode)	监听播放模式的变化	播放模式
void	onPlayStateChanged (int state)	监听播放状态的变化	播放状态
void	onPlayMediaInfoChanged (MediaInfo mediaInfo)	监听当前播放的音视频信息的变化	当前播放的音视频信息
备注	输入值的取值 参见 get 函数的具体描述。 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的变化时，也会同步通知变化。		