



北京进化者机器人科技有限公司  
Beijing Evolver Robotics Co.,LTD

产品名称 Product name	密级 Confidentiality level
进化者机器人控制 SDK 说明文档	
产品版本 Product version	Total 90 pages 共 90 页
4.2	

## 进化者机器人控制 SDK 说明文档

拟制: Prepared by	潘鑫宇	日期: Date	2016/11/11
审核: Reviewed by		日期: Date	
审核: Reviewed by		日期: Date	
批准: Granted by		日期: Date	

北京进化者机器人科技有限公司

版权所有 侵权必究

All rights reserved



## 修订记录 Revision record

日期 Date	修订版本 Revision version	修改描述 Changed Description	作者 Author
2016/11/1	1.0	初稿起草	潘鑫宇
2017/6/30	2.0	新增投影, 净化器, SOS, 电源控制	段萌
2017/7/14	3.0	新增导航, <u>DIY 数据获取</u> , 脚本管理	段萌
2017/8/14	4.0	新增 DIY 控制模块, 迁移 <u>DIY 数据获取</u>	段萌
2017/9/12	4.1	新增获取地图标记点	邢亚凡
2017/9/19	4.2	新增获取昵称、停止 DIY 方法	邢亚凡

### 2.1 工程配置

把 SDK 文件夹里的 Jar 包集成到 lib 下

### 2.2 权限配置

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

## 3 SDK 方法说明

### 3.1 初始化 SDK

控制 SDK 初始化分 2 部, 要调用 connect 和 init 方法, 先 connect 后 init, 详情可参考 Sample  
如果初始化失败, 控制 SDK 将失效, 机器人各个状态获取失败和运动失败, 请确保申请的 AppKey  
为正确并且调用了 connect 和 init 方法

测试账号(仅供测试使用):

```
String  accessKey = "98838811";
```

```
String  accessSecret = "mR07n5fmEC5DDY7yTn55PmdVYJ9bcF9Z";
```

```
String  devId= "568968789";
```

#### 3.1.1 方法名

### M2.1 工程配置

把 2.1 工程配置



把 SDK 文件夹里的 Jar 包集成到 lib 下

## 2.2 权限配置

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

## 3 SDK 方法说明

### 3.1 初始化 SDK

控制 SDK 初始化分 2 部，要调用 connect 和 init 方法，先 connect 后 init，详情可参考 Sample  
如果初始化失败，控制 SDK 将失效，机器人各个状态获取失败和运动失败，请确保申请的 AppKey  
为正确并且调用了 connect 和 init 方法

测试账号(仅供测试使用):

```
String  accessKey = "98838811";  
String  accessSecret = "mR07n5fmEC5DDY7yTn55PmdVYJ9bcF9Z";  
String  devId= "568968789";
```

#### 3.1.1 方法名

Method : connect()

#### 3.1.2 调用示例

```
RobotManager.getInstance(this). connect ();
```

注: 推荐在 application 上的 onCreate()里注册

此方法是和下位机进行通讯，必须调用

#### 3.1.3 方法名

Method : init(String accessKey,String accessSecret, String appKey, String devId ,  
OnInitCompleteListener onInitCompleteListener)

#### 3.1.4 请求参数

参数名称

类型

约束

描述



accessKey

String

1

SDK 开放平台申请的 accessKey

accessSecret

String

1

SDK 开放平台申请的 accessSecret

devId

String

1

SDK 开放平台申请的 devId

onInitCompleteListener

OnInitCompleteListener

1

初始化完成回调

### 3.1.5 调用示例

```
RobotManager.getInstance(this).init(accessKey, accessSecret, devId, this);
```

注：推荐在 application 上的 onCreate() 里注册 s

如果 SDK 正在初始化或者初始化失败，控制 SDK 的功能将不可用

## 3.2. 获取机器人管理实例

### 3.2.1 方法名

Method : getInstance(Context context)



### 3.2.2 请求参数

参数名称

类型

约束

描述

context

Context

1

上下文参数

### 3.2.3 返回值

RobotManager

### 3.2.4 调用示例

RobotManager.getInstance(context)

## 3.3 获取机器 ID

### 3.3.1 方法名

Method : getRobotId

### 3.3.2 返回值

return 机器人 ID(String), 无号码返回 null

### 3.3.3 调用示例

RobotManager.getInstance(mContext).getRobotId();

## 3.4 获取机器昵称

### 3.4.1 方法名

Method : getRobotName

### 3.4.2 返回值

return 机器人昵称(String), 昵称返回小胖



### 3.4.3 调用示例

```
RobotManager.getInstance(mContext). getRobotName ();
```

SDK 文件夹里的 Jar 包集成到 lib 下

## 2.2 权限配置

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

## 3 SDK 方法说明

### 3.1 初始化 SDK

控制 SDK 初始化分 2 部，要调用 connect 和 init 方法，先 connect 后 init，详情可参考 Sample  
如果初始化失败，控制 SDK 将失效，机器人各个状态获取失败和运动失败，请确保申请的 AppKey  
为正确并且调用了 connect 和 init 方法

测试账号(仅供测试使用):

```
String  accessKey = "98838811";  
String  accessSecret = "mR07n5fmEC5DDY7yTn55PmdVYJ9bcF9Z";  
String  devId= "568968789";
```

#### 3.1.1 方法名

Method : connect()

#### 3.1.2 调用示例

```
RobotManager.getInstance(this). connect ();
```

注: 推荐在 application 上的 onCreate()里注册

此方法是和下位机进行通讯，必须调用

#### 3.1.3 方法名

Method : init(String accessKey,String accessSecret, String appKey, String devId ,  
OnInitCompleteListener onInitCompleteListener)

#### 3.1.4 请求参数

参数名称



类型

约束

描述

`accessKey`

String

1

SDK 开放平台申请的 `accessKey`

`accessSecret`

String

1

SDK 开放平台申请的 `accessSecret`

`devId`

String

1

SDK 开放平台申请的 `devId`

`onInitCompleteListener`

OnInitCompleteListener

1

初始化完成回调

### 3.1.5 调用示例

```
RobotManager.getInstance(this).init(accessKey, accessSecret, devId, this);
```

注：推荐在 `application` 上的 `onCreat()`里注册 s

如果 SDK 正在初始化或者初始化失败，控制 SDK 的功能将不可用



## 3.2.获取机器人管理实例

### 3.2.1 方法名

Method : getInstance(Context context)

### 3.2.2 请求参数

参数名称

类型

约束

描述

context

Context

1

上下文参数

### 3.2.3 返回值

RobotManager

### 3.2.4 调用示例

RobotManager.getInstance(context)

## 3.3 获取机器 ID

### 3.3.1 方法名

Method : getRobotId

### 3.3.2 返回值

return 机器人 ID(String), 无号码返回 null

### 3.3.3 调用示例

RobotManager.getInstance(mContext).getRobotId();

## 3.4 获取机器昵称

### 3.4.1 方法名





Method : `getRobotName`

#### 3.4.2 返回值

**return** 机器人昵称(String), 昵称返回小胖

#### 3.4.3 调用示例

```
RobotManager.getInstance(mContext).getRobotName ();
```

ethod : `connect()`

#### 3.1.2 调用示例

```
RobotManager.getInstance(this).connect ();
```

注: 推荐在 `application` 上的 `onCreat()`里注册

此方法是和下位机进行通讯, 必须调用

#### 3.1.3 方法名

Method : `init(String accessKey,String accessSecret, String appKey, String devId ,  
OnInitCompleteListener onInitCompleteListener)`

#### 3.1.4 请求参数

参数名称

类型

约束

描述

`accessKey`

String

1

SDK 开放平台申请的 `accessKey`

`accessSecret`

String



1

SDK 开放平台申请的 accessSecret

devId

String

1

SDK 开放平台申请的 devId

onInitCompleteListener

OnInitCompleteListener

1

初始化完成回调

### 3.1.5 调用示例

```
RobotManager.getInstance(this).init(accessKey, accessSecret, devId, this);
```

注：推荐在 application 上的 onCreate() 里注册 s

如果 SDK 正在初始化或者初始化失败，控制 SDK 的功能将不可用

## 3.2. 获取机器人管理实例

### 3.2.1 方法名

Method : getInstance(Context context)

### 3.2.2 请求参数

参数名称

类型

约束

描述

context

Context



1

上下文参数

### 3.2.3 返回值

RobotManager

### 3.2.4 调用示例

RobotManager.getInstance(context)

## 3.3 获取机器 ID

### 3.3.1 方法名

Method : getRobotId

### 3.3.2 返回值

return 机器人 ID(String), 无号码返回 null

### 3.3.3 调用示例

RobotManager.getInstance(mContext).getRobotId();

## 3.4 获取机器昵称

### 3.4.1 方法名

Method : getRobotName

### 3.4.2 返回值

return 机器人昵称(String), 昵称返回小胖

### 3.4.3 调用示例

RobotManager.getInstance(mContext). getRobotName ();



## 目录

1 文档说明.....	30
1.1 目的.....	30
1.2 范围.....	30
2 工程配置与权限.....	30
2.1 工程配置.....	30
2.2 权限配置.....	30
3 SDK 方法说明.....	30
3.1 初始化 SDK.....	30
3.1.1 方法名.....	31
3.1.2 调用示例.....	31
3.1.3 方法名.....	31
3.1.4 请求参数.....	31
3.1.5 调用示例.....	31
3.2.获取机器人管理实例.....	31
3.2.1 方法名.....	31
3.2.2 请求参数.....	32



---

3.2.3 返回值.....	32
3.2.4 调用示例.....	32
3.3 获取机器 ID.....	32
3.3.1 方法名.....	32
3.3.2 返回值.....	32
3.3.3 调用示例.....	32
3.4 投影仪.....	32
3.4.1 投影仪所有状态.....	33
3.4.2 投影仪状态回调.....	33
3.4.3 获取投影仪状态.....	33
3.4.3.1 方法名.....	33
3.4.3.2 返回值.....	34
3.4.3.3 调用示例.....	34
3.4.4 注册投影仪状态变化监听.....	34
3.4.4.1 方法名.....	34
3.4.4.2 请求参数.....	34
3.4.4.3 调用示例.....	34
3.4.5 取消注册投影仪状态变化监听.....	34
3.4.5.1 方法名.....	34
3.4.5.2 请求参数.....	34
3.4.5.3 调用示例.....	35
3.4.6 投影仪高温状态.....	35
3.4.7 投影仪高温状态回调.....	35
3.4.8 获取投影仪高温状态.....	36
3.4.8.1 方法名.....	36
3.4.8.2 返回值.....	36
3.4.8.3 调用示例.....	36
3.4.9 注册投影仪高温状态变化监听.....	36
3.4.9.1 方法名.....	36



---

3.4.9.2 请求参数.....	36
3.4.9.3 调用示例.....	36
3.4.10 取消注册投影仪高温状态变化监听.....	37
3.4.10.1 方法名.....	37
3.4.10.2 请求参数.....	37
3.4.10.3 调用示例.....	37
3.4.11 投影仪准备状态.....	37
3.4.12 投影仪准备状态回调.....	38
3.4.13 获取投影仪准备状态.....	38
3.4.13.1 方法名.....	38
3.4.13.2 返回值.....	38
3.4.13.3 调用示例.....	38
3.4.13.3 准备状态说明.....	38
3.4.14 注册投影仪高温状态变化监听.....	39
3.4.14.1 方法名.....	39
3.4.14.2 请求参数.....	39
3.4.14.3 调用示例.....	39
3.4.15 取消注册投影仪高温状态变化监听.....	39
3.4.15.1 方法名.....	39
3.4.15.2 请求参数.....	39
3.4.15.3 调用示例.....	40
3.4.16 投影仪控制.....	40
3.4.16.1 获取投影仪管理实例.....	40
3.4.16.1.1 方法名.....	40
3.4.16.1.2 调用示例.....	40
3.4.16.2 打开投影.....	40
3.4.16.2.1 方法名.....	40
3.4.16.2.2 调用示例.....	40
3.4.16.3 关闭投影.....	40



---

3.4.16.3.1 方法名.....	40
3.4.16.3.2 调用示例.....	41
3.4.16.4 调节亮度.....	41
3.4.16.4.1 方法名.....	41
3.4.16.4.2 请求参数.....	41
3.4.16.4.3 调用示例.....	41
3.4.16.5 梯形校正.....	41
3.4.16.5.1 方法名.....	41
3.4.16.5.2 请求参数.....	41
3.4.16.5.3 调用示例.....	42
3.4.16.6 桌面正投.....	42
3.4.16.6.1 方法名.....	42
3.4.16.6.2 调用示例.....	42
3.4.16.7 桌面背投.....	42
3.4.16.7.1 方法名.....	42
3.4.16.7.2 调用示例.....	42
3.4.16.8 吊装正投.....	43
3.4.16.8.1 方法名.....	43
3.4.16.8.2 调用示例.....	43
3.4.16.9 吊装背投.....	43
3.4.16.9.1 方法名.....	43
3.4.16.9.2 调用示例.....	43
3.5 净化器.....	43
3.5.1 净化器所有状态.....	43
3.5.2 净化器状态变化回调.....	44
3.5.3 获取净化器状态.....	44
3.5.3.1 方法名.....	44
3.5.3.2 返回值.....	44
3.5.3.3 调用示例.....	44



---

3.5.4 注册净化器状态变化监听.....	44
3.5.4.1 方法名.....	44
3.5.4.2 请求参数.....	45
3.5.4.3 调用示例.....	50
3.5.5 取消注册净化器状态变化监听.....	50
3.5.5.1 方法名.....	50
3.5.5.2 请求参数.....	54
3.5.5.3 调用示例.....	54
3.5.6 获取净化器工作时长.....	54
3.5.6.1 方法名.....	54
3.5.6.2 返回值.....	54
3.5.6.3 调用示例.....	54
3.5.7 净化器工作时长回调.....	54
3.5.8 注册净化器工作时长监听.....	55
3.5.8.1 方法名.....	55
3.5.8.2 请求参数.....	55
3.5.8.3 调用示例.....	55
3.5.9 取消注册净化器工作时长监听.....	55
3.5.9.1 方法名.....	55
3.5.9.2 请求参数.....	55
3.5.9.3 调用示例.....	55
3.5.10 获取净化器风机速度.....	56
3.5.10.1 方法名.....	56
3.5.10.2 返回值.....	56
3.5.10.3 调用示例.....	56
3.5.11 净化器控制.....	56
3.5.11.1 获取净化器实例.....	56
3.5.11.1.1 方法名.....	56
3.5.11.1.2 调用示例.....	56





---

3.5.11.2 打开净化器.....	57
3.5.11.2.1 方法名.....	57
3.5.11.2.2 调用示例.....	57
3.5.11.3 关闭净化器.....	57
3.5.11.3.1 方法名.....	57
3.5.11.3.2 调用示例.....	57
3.5.11.4 查询净化器状态.....	57
3.5.11.4.1 方法名.....	57
3.5.11.4.2 调用示例.....	58
3.5.11.5 设置净化器风机速度.....	58
3.5.11.5.1 方法名.....	58
3.5.11.5.2 请求参数.....	58
3.5.11.5.3 调用示例.....	58
3.6 电池.....	58
3.6.1 电量.....	58
3.6.1.1 获取电池电量.....	58
3.6.1.1.1 方法名.....	58
3.6.1.1.2 返回值.....	59
3.6.1.1.3 调用示例.....	59
3.6.1.2 电量状态变化回调.....	59
3.6.1.3 注册机器人电量变化监听.....	59
3.6.1.3.1 方法名.....	59
3.6.1.3.2 请求参数.....	59
3.6.1.3.3 调用示例.....	59
3.6.1.4 取消注册机器人电量变化监听.....	60
3.6.1.4.1 方法名.....	60
3.6.1.4.2 请求参数.....	60
3.6.1.4.3 调用示例.....	60
3.6.2 电池状态.....	60



---

3.6.2.1 电池所有状态.....	60
3.6.2.2 获取电池状态.....	61
3.6.2.2.1 方法名.....	61
3.6.2.2.2 返回值.....	61
3.6.2.2.3 调用示例.....	61
3.6.2.3 注册电池状态变化监听.....	61
3.6.2.3.1 方法名.....	61
3.6.2.3.2 请求参数.....	61
3.6.2.3.3 调用示例.....	61
3.6.2.4 取消注册电池状态变化监听.....	61
3.6.2.4.1 方法名.....	61
3.6.2.4.2 请求参数.....	62
3.6.2.4.3 调用示例.....	62
3.7 头部触摸.....	62
3.7.1 头部触摸所有状态.....	62
3.7.2 头顶触摸状态变化回调.....	62
3.7.3 获取头部触摸状态.....	63
3.7.3.1 方法名.....	63
3.7.3.2 返回值.....	63
3.7.3.3 调用示例.....	63
3.7.4 注册头部触摸状态改变监听.....	63
3.7.4.1 方法名.....	63
3.7.4.2 请求参数.....	63
3.7.4.3 调用示例.....	63
3.7.5 取消注册头部按钮状态改变监听.....	63
3.7.5.1 方法名.....	63
3.7.5.2 请求参数.....	64
3.7.5.3 调用示例.....	64
3.8 面罩状态.....	64



---

3.8.1 面罩所有状态.....	64
3.8.2 获取面罩状态.....	65
3.8.2.1 方法名.....	65
3.8.2.2 返回值.....	65
3.8.2.3 调用示例.....	65
3.8.3 面罩状态变化回调.....	65
3.8.4 注册面罩状态变化监听.....	65
3.8.4.1 方法名.....	65
3.8.4.2 请求参数.....	65
3.8.4.3 调用示例.....	66
3.8.5 取消注册面罩状态变化监听.....	66
3.8.5.1 方法名.....	66
3.8.5.2 请求参数.....	66
3.8.5.3 调用示例.....	66
3.9 托盘状态.....	66
3.9.1 托盘所有状态.....	66
3.9.2 托盘状态变化回调.....	67
3.9.3 获取托盘状态.....	67
3.9.3.1 方法名.....	67
3.9.3.2 返回值.....	67
3.9.3.3 调用示例.....	67
3.9.4 注册托盘状态变化监听.....	67
3.9.4.1 方法名.....	67
3.9.4.2 请求参数.....	67
3.9.4.3 调用示例.....	68
3.9.5 取消注册托盘状态变化监听.....	68
3.9.5.1 方法名.....	68
3.9.5.2 请求参数.....	68
3.9.5.3 调用示例.....	68



---

3.10 头部控制.....	68
3.10.1 获取机器人头部管理实例.....	68
3.10.1.1 方法名.....	68
3.10.1.2 返回值.....	68
3.10.1.3 调用示例.....	68
3.10.2 以默认速度移动到头部的最左端.....	69
3.10.2.1 方法名.....	69
3.10.2.2 调用示例.....	69
3.10.3 以指定的速度移动到头部的最左端.....	69
3.10.3.1 方法名.....	69
3.10.3.2 请求参数.....	69
3.10.3.3 调用示例.....	69
3.10.4 以默认速度移动到头部的最右端.....	69
3.10.4.1 方法名.....	69
3.10.4.2 调用示例.....	69
3.10.5 以指定的速度移动到头部的最右端.....	69
3.10.5.1 方法名.....	69
3.10.5.2 请求参数.....	70
3.10.5.3 调用示例.....	70
3.10.6 以指默认速度移动到头部指定的角度.....	70
3.10.6.1 方法名.....	70
3.10.6.2 请求参数.....	70
3.10.6.3 调用示例.....	70
3.10.7 以指定的速度移动到头部指定的角度.....	70
3.10.7.1 方法名.....	70
3.10.7.2 请求参数.....	70
3.10.7.3 调用示例.....	70
3.10.8 停止头部运动.....	71
3.10.8.1 方法名.....	71



---

3.10.8.2 调用示例.....	71
3.11 双翅控制.....	71
3.11.1 获取机器人翅膀管理实例.....	71
3.11.1.1 方法名.....	71
3.11.1.2 返回值.....	71
3.11.1.3 调用示例.....	71
3.11.2 翅膀以默认速度向上运动.....	72
3.11.2.1 方法名.....	72
3.11.2.2 请求参数.....	72
3.11.2.3 调用示例.....	72
3.11.3 翅膀以指定的速度向上运动.....	72
3.11.3.1 方法名.....	72
3.11.3.2 请求参数.....	72
3.11.3.3 调用示例.....	72
3.11.4 翅膀以默认速度向下运动.....	72
3.11.4.1 方法名.....	72
3.11.4.2 请求参数.....	73
3.11.4.3 调用示例.....	73
3.11.5 翅膀以指定的速度向下运动.....	73
3.11.5.1 方法名.....	73
3.11.5.2 请求参数.....	73
3.11.5.3 调用示例.....	73
3.11.6 翅膀以默认速度向指定角度运动.....	73
3.11.6.1 方法名.....	73
3.11.6.2 请求参数.....	73
3.11.6.3 调用示例.....	74
3.11.7 翅膀以指定的速度向指定角度运动.....	74
3.11.7.1 方法名.....	74
3.11.7.2 请求参数.....	74



---

3.11.7.3 调用示例.....	74
3.11.8 翅膀停止运动.....	74
3.11.8.1 方法名.....	74
3.11.8.2 请求参数.....	74
3.11.8.3 调用示例.....	74
3.12 双轮控制.....	75
3.12.1 获取机器人双轮管理实例.....	75
3.12.1.1 方法名.....	75
3.12.1.2 返回值.....	75
3.12.1.3 调用示例.....	75
3.12.2 以默认速度向前运动.....	75
3.12.2.1 方法名.....	75
3.12.2.2 调用示例.....	75
3.12.3 以指定的速度向前运动.....	75
3.12.3.1 方法名.....	75
3.12.3.2 请求参数.....	75
3.12.3.3 调用示例.....	76
3.12.4 以默认速度向后运动.....	76
3.12.4.1 方法名.....	76
3.12.4.2 调用示例.....	76
3.12.5 以指定的速度向后运动.....	76
3.12.5.1 方法名.....	76
3.12.5.2 请求参数.....	76
3.12.5.3 调用示例.....	76
3.12.6 以默认速度向左运动.....	76
3.12.6.1 方法名.....	76
3.12.6.2 调用示例.....	76
3.12.7 以指定的速度向左运动.....	77
3.12.7.1 方法名.....	77



---

3.12.7.2 请求参数.....	77
3.12.7.3 调用示例.....	77
3.12.8 以默认速度向右运动.....	77
3.12.8.1 方法名.....	77
3.12.8.2 调用示例.....	77
3.12.9 以指定的速度向右运动.....	77
3.12.9.1 方法名.....	77
3.12.9.2 请求参数.....	77
3.12.9.3 调用示例.....	77
3.12.10 双轮停止运动.....	78
3.12.10.1 方法名.....	78
3.12.10.2 调用示例.....	78
3.12.11 双轮禁止功能.....	78
3.12.11.1 双轮禁止开关开启（禁用双轮） .....	78
3.12.11.1.1 方法名.....	78
3.12.11.1.2 调用示例.....	78
3.12.11.2 双轮禁止开关关闭（启用双轮） .....	78
3.12.11.2.1 方法名.....	78
3.12.11.2.2 调用示例.....	78
3.12.11.3 双轮禁止开关所有状态.....	79
3.12.11.4 双轮禁止开关状态变化回调.....	79
3.12.11.5 获取双轮禁止开关状态.....	80
3.12.11.5.1 方法名.....	80
3.12.11.5.2 返回值.....	80
3.12.11.5.3 调用示例.....	80
3.12.11.6 注册双轮禁止开关状态变化监听.....	80
3.12.11.6.1 方法名.....	80
3.12.11.6.2 请求参数.....	80
3.12.11.6.3 调用示例.....	81



---

3.12.11.7 取消双轮禁止开关状态变化监听.....	81
3.12.11.7.1 方法名.....	81
3.12.11.7.2 请求参数.....	81
3.12.11.7.3 调用示例.....	81
3.13 控制管理.....	81
3.13.1 获取机器人控制实例.....	81
3.13.1.1 方法名.....	81
3.13.1.2 返回值.....	82
3.13.1.3 调用示例.....	82
3.13.2 设置灯带亮度.....	82
3.13.2.1 方法名.....	82
3.13.2.2 请求参数.....	82
3.13.2.3 调用示例.....	82
3.14 自定义机器人说话.....	82
3.14.1 方法名.....	82
3.14.2 请求参数.....	82
3.14.3 调用示例.....	83
3.15 电源按键状态.....	83
3.15.1 电源按键所有状态.....	83
3.15.2 电源按键状态变化回调.....	83
3.15.3 获取电源按键状态.....	84
3.15.3.1 方法名.....	84
3.15.3.2 返回值.....	84
3.15.3.3 调用示例.....	84
3.15.4 注册电源按键状态变化监听.....	84
3.15.4.1 方法名.....	84
3.15.4.2 请求参数.....	84
3.15.4.3 调用示例.....	84
3.15.5 取消注册托盘状态变化监听.....	85





---

3.15.5.1 方法名.....	85
3.15.5.2 请求参数.....	85
3.15.5.3 调用示例.....	85
3.16 SOS 状态.....	85
3.16.1 SOS 按钮所有状态.....	85
3.16.2 SOS 按钮状态变化回调.....	86
3.16.3 获取 SOS 按钮状态.....	86
3.16.3.1 方法名.....	86
3.16.3.2 返回值.....	86
3.16.3.3 调用示例.....	86
3.16.4 注册 SOS 按钮状态变化监听.....	86
3.16.4.1 方法名.....	86
3.16.4.2 请求参数.....	87
3.16.4.3 调用示例.....	87
3.16.5 取消注册 SOS 按钮状态变化监听.....	87
3.16.5.1 方法名.....	87
3.16.5.2 请求参数.....	87
3.16.5.3 调用示例.....	87
3.17 导航模块.....	87
3.17.1 获取机器人导航管理实例.....	87
3.17.1.1 方法名.....	87
3.17.1.2 返回值.....	87
3.17.1.3 调用示例.....	88
3.17.2 注册导航状态变化监听.....	88
3.17.2.1 方法名.....	88
3.17.2.2 请求参数.....	88
3.17.2.2 调用示例.....	88
3.17.3 取消注册导航状态变化监听.....	89
3.17.3.1 方法名.....	89



---

3.17.3.2 请求参数.....	89
3.17.3.2 调用示例.....	90
3.17.4 导航状态回调.....	91
3.17.5 导航去指定地点.....	92
3.17.5.1 方法名.....	92
3.17.5.2 请求参数.....	92
3.17.5.3 调用示例.....	92
3.17.6 暂停导航.....	92
3.17.6.1 方法名.....	92
3.17.6.2 调用示例.....	93
3.17.7 继续导航.....	93
3.17.7.1 方法名.....	93
3.17.7.2 调用示例.....	93
3.17.8 停止导航.....	93
3.17.8.1 方法名.....	93
3.17.8.2 调用示例.....	93
3.17.9 回充电桩（无监听） .....	93
3.17.9.1 方法名.....	93
3.17.9.2 调用示例.....	93
3.17.10 回充电桩（有监听） .....	93
3.17.10.1 方法名.....	93
3.17.10.2 请求参数.....	94
3.17.10.3 调用示例.....	94
3.17.11 出充电桩（无监听） .....	95
3.17.11.1 方法名.....	95
3.17.11.2 调用示例.....	95
3.17.12 出充电桩（有监听） .....	95
3.17.12.1 方法名.....	95
3.17.12.2 请求参数.....	95



---

3.17.12.3 调用示例.....	95
3.17.13 手动接驳.....	97
3.17.13.1 方法名.....	97
3.17.13.2 调用示例.....	97
3.17.13.3 方法说明.....	97
3.17.14 不依赖地图回充电桩（无监听） .....	97
3.17.14.1 方法名.....	97
3.17.14.2 调用示例.....	97
3.17.15 不依赖地图回充电桩（有监听） .....	97
3.17.15.1 方法名.....	97
3.17.15.2 请求参数.....	97
3.17.15.3 调用示例.....	98
3.18 脚本任务管理模块.....	99
3.18.1 获取机器人脚本任务管理实例.....	99
3.18.1.1 方法名.....	99
3.18.1.2 返回值.....	99
3.18.1.3 调用示例.....	99
3.18.2 脚本状态回调.....	99
3.18.3 开始执行脚本.....	100
3.18.3.1 方法名.....	100
3.18.3.2 请求参数.....	100
3.18.3.3 调用示例.....	100
3.18.4 开始执行脚本（有回调） .....	101
3.18.4.1 方法名.....	101
3.18.4.2 请求参数.....	101
3.18.4.3 调用示例.....	101
3.18.5 停止脚本.....	102
3.18.5.1 方法名.....	102
3.18.5.2 调用示例.....	102



---

3.18.6 重置机器人 (头部、双翅、灯带状态还原) .....	102
3.18.6.1 方法名.....	102
3.18.6.2 调用示例.....	102
3.19 DIY 控制模块.....	102
3.19.1 获取机器人 DIY 控制管理实例.....	103
3.19.1.1 方法名.....	103
3.19.1.2 返回值.....	103
3.19.1.3 调用示例.....	103
3.19.2 获取机器人 DIY 大场景.....	103
3.19.2.1 方法名.....	105
3.19.2.2 返回值.....	105
3.19.2.3 请求参数.....	105
3.19.2.4 调用示例.....	105
3.19.3 获取机器人 DIY 小场景(根据大场景 Id 来获取小场景 id).....	105
3.19.3.1 方法名.....	105
3.19.3.2 返回值.....	105
3.19.3.3 请求参数.....	105
3.19.3.4 调用示例.....	106
3.19.4 获取机器人 DIY 问题(根据小场景 Id 获取问题).....	106
3.19.4.1 方法名.....	106
3.19.4.2 返回值.....	106
3.19.4.3 请求参数.....	106
3.19.4.4 调用示例.....	106
3.19.5 获取机器人 DIY 回答(根据小场景 Id 获取问题).....	107
3.19.5.1 方法名.....	107
3.19.5.2 返回值.....	107
3.19.5.3 请求参数.....	107
3.19.5.4 调用示例.....	107
3.19.6 执行 DIY(根据小场景 Id 执行).....	108



---

3.19.6.1 注意事项.....	108
3.19.6.2 方法名.....	108
3.19.6.3 请求参数.....	108
3.19.6.4 调用示例.....	108
3.19.7 获取机器人 DIY 源数据.....	103
3.19.7.1 方法名.....	103
3.19.7.2 返回值.....	103
3.19.7.3 请求参数.....	103
3.19.7.4 调用示例.....	103



## 1 文档说明

### 1.1 目的

本文档主要介绍机器人控制 SDK（Android 版）使用方法，利用 SDK 可以控制监听机器人运动来实现第三方业务

### 1.2 范围

仅供与进化者公司有深度合作的公司使用

## 2 工程配置与权限

### 2.1 工程配置

把 SDK 文件夹里的 Jar 包集成到 lib 下

### 2.2 权限配置

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

## 3 SDK 方法说明

### 3.1 初始化 SDK

控制 SDK 初始化分 2 部，要调用 connect 和 init 方法，先 connect 后 init，详情可参考 [Sample](#)

如果初始化失败，控制 SDK 将失效，机器人各个状态获取失败和运动失败，请确保申请的 AppKey 为正确并且调用了 connect 和 init 方法

测试账号(仅供测试使用):connect

```
String  accessKey = "98838811";
```

```
String  accessSecret = "mR07n5fmEC5DDY7yTn55PmdVYJ9bcF9Z";
```

```
String  devId= "568968789";
```



### 3.1.1 方法名

Method : connect()

### 3.1.2 调用示例

```
RobotManager.getInstance(this).connect();
```

**注:** 推荐在 application 上的 onCreate()里注册

此方法是和下位机进行通讯，必须调用

### 3.1.3 方法名

Method : init(String accessKey,String accessSecret, String appKey, String devId ,  
OnInitCompleteListener onInitCompleteListener)

### 3.1.4 请求参数

参数名称	类型	约束	描述
accessKey	String	1	SDK 开放平台申请的 accessKey
accessSecret	String	1	SDK 开放平台申请的 accessSecret
devId	String	1	SDK 开放平台申请的 devId
onInitCompleteListener	OnInitCompleteListener	1	初始化完成回调

### 3.1.5 调用示例

```
RobotManager.getInstance(this).init(accessKey, accessSecret, devId, this);
```

**注:** 推荐在 application 上的 onCreate()里注册 s

如果 SDK 正在初始化或者初始化失败，控制 SDK 的功能将不可用

## 3.2.获取机器人管理实例

### 3.2.1 方法名

Method : getInstance(Context context)



### 3.2.2 请求参数

参数名称	类型	约束	描述
context	Context	1	上下文参数

### 3.2.3 返回值

RobotManager

### 3.2.4 调用示例

```
RobotManager.getInstance(context)
```

## 3.3 获取机器 ID

### 3.3.1 方法名

Method : getRobotId

### 3.3.2 返回值

return 机器人 ID(String), 无号码返回 null

### 3.3.3 调用示例

```
RobotManager.getInstance(mContext).getRobotId();
```

## 3.4 获取机器昵称

### 3.4.1 方法名

Method : getRobotName

### 3.4.2 返回值

return 机器人昵称(String), 昵称返回小胖





### 3.4.3 调用示例

```
RobotManager.getInstance(mContext). getRobotName ();
```

## 3.5 投影仪

### 3.5.1 投影仪所有状态

```
/**
 * 投影仪状态
 */
public enum ProjectStatus {
    /** 关闭状态 */
    CLOSED,
    /** 打开状态 */
    OPEN,
    /** 未知状态 */
    UNKNOWN,
}
```

### 3.5.2 投影仪状态回调

```
/**
 * 投影仪状态回调
 */
public interface OnProjectStatusChangedListener {
    void onProjectStatusChanged(ProjectStatusprojectStatus);
}
```

### 3.5.3 获取投影仪状态

#### 3.5.3.1 方法名

Method : getProjectorStatus



### 3.5.3.2 返回值

ProjectStatus

return 投影仪状态

### 3.5.3.3 调用示例

```
RobotManager.getInstance(this).getProjectorStatus ();
```

## 3.5.4 注册投影仪状态变化监听

### 3.5.4.1 方法名

Method : registerProjectorStatusChangedListener(OnProjectStatusChangedListener listener)

### 3.5.4.2 请求参数

参数名称	类型	约束	描述
listener	OnProjectStatusChangedListener	1	投影仪状态变化监听

### 3.5.4.3 调用示例

```
RobotManager.getInstance(this).registerProjectorStatusChangedListener(this);
```

## 3.5.5 取消注册投影仪状态变化监听

### 3.5.5.1 方法名

Method : unregisterProjectorStatusChangedListener (OnProjectStatusChangedListener listener)

### 3.5.5.2 请求参数

参数名称	类型	约束	描述
listener	OnProjectStatusC	1	投影仪状态变化监听



	hangedListener		
--	----------------	--	--

### 3.5.5.3 调用示例

```
RobotManager.getInstance(this).unRegisterProjectorStatusChangedListener (this);
```

### 3.5.6 投影仪高温状态

```
/**
 * 投影仪高温状态
 */
public enum ProjectorHotStatus {
    /** 常温状态 */
    NORMAL_TEMP,
    /** 高温状态 */
    HOT_TEMP,
    /** 未知状态 */
    UNKNOWN,
    /** 初始化失败 */
    FAILURE;

    static ProjectorHotStatus getProjectorHotStatus(int projectorHotStatus) {
        ProjectorHotStatus[] values = ProjectorHotStatus.values();
        return values[projectorHotStatus];
    }
}
```

### 3.5.7 投影仪高温状态回调

```
/**
 * 投影仪高温状态回调
 */
public interface OnProjectorHotStatusChangedListener {
    void onProjectorHotStatusChanged(RobotStatus.ProjectorHotStatus
```



```
projectorHotStatus);  
    }
```

### 3.5.8 获取投影仪高温状态

#### 3.5.8.1 方法名

Method : getProjectorHotStatus

#### 3.5.8.2 返回值

ProjectorHotStatus  
return 投影仪高温状态

#### 3.5.8.3 调用示例

RobotManager.getInstance(this).getProjectorHotStatus();

### 3.5.9 注册投影仪高温状态变化监听

#### 3.5.9.1 方法名

Method :  
registerProjectorHotStatusChangeListener(OnProjectorHotStatusChangeListener  
listener)

#### 3.5.9.2 请求参数

参数名称	类型	约束	描述
listener	OnProjectorHot StatusChangedL istener	1	投影仪高温状态变化监听

#### 3.5.9.3 调用示例

RobotManager.getInstance(this).registerProjectorHotStatusChangeListener(this);



## 3.5.10 取消注册投影仪高温状态变化监听

### 3.5.10.1 方法名

Method :  
unRegisterProjectorHotStatusChangedListener(OnProjectorHotStatusChangedListener listener)

### 3.5.10.2 请求参数

参数名称	类型	约束	描述
listener	OnProjectorHotStatusChangedListener	1	投影仪高温状态变化监听

### 3.5.10.3 调用示例

```
RobotManager.getInstance(this).unRegisterProjectorHotStatusChangedListener(this);
```

## 3.5.11 投影仪准备状态

```
/**  
 * 投影仪准备完成状态  
 */  
public enum ProjectorPrepareCompleteStatus {  
    /** 准备好状态 */  
    COMPLETE,  
    /** 未准备好状态 */  
    UN_COMPLETE,  
    /** 未知状态 */  
    UNKNOWN,  
    /** 初始化失败 */  
    FAILURE;
```



```
static ProjectorPrepareCompleteStatus getProjectorPrepareCompleteStatus(int  
projectorPrepareCompleteStatus) {  
    ProjectorPrepareCompleteStatus[] values =  
    ProjectorPrepareCompleteStatus.values();  
    return values[projectorPrepareCompleteStatus];  
}  
}
```

### 3.5.12 投影仪准备状态回调

```
/**  
 * 投影仪准备完成状态回调  
 */  
public interface OnProjectorPrepareCompleteStatusChangeListener {  
    void OnRobotStateChangeListener(ProjectStatus projectStatus);  
}
```

### 3.5.13 获取投影仪准备状态

#### 3.5.13.1 方法名

Method : getProjectorPrepareCompleteStatus

#### 3.5.13.2 返回值

ProjectorPrepareCompleteStatus

return 投影仪高温状态

#### 3.5.13.3 调用示例

```
RobotManager.getInstance(this).getProjectorPrepareCompleteStatus();
```

#### 3.5.13.3 准备状态说明

判断该功能原因为投影仪有启动时间，如需使用投影仪控制功能，需要等待该状态返回后方可使用。



## 3.5.14 注册投影仪高温状态变化监听

### 3.5.14.1 方法名

Method :  
registerProjectorPrepareCompleteStatusChangedListener(OnProjectorPrepareCompleteStatusChangedListener listener)

### 3.5.14.2 请求参数

参数名称	类型	约束	描述
listener	OnProjectorPrepareCompleteStatusChangedListener	1	投影仪准备状态变化监听

### 3.5.14.3 调用示例

```
RobotManager.getInstance(this).registerProjectorPrepareCompleteStatusChangedListener(this);
```

## 3.5.15 取消注册投影仪高温状态变化监听

### 3.5.15.1 方法名

Method :  
unRegisterProjectorPrepareCompleteStatusChangedListener(OnProjectorPrepareCompleteStatusChangedListener listener)

### 3.5.15.2 请求参数

参数名称	类型	约束	描述
listener	OnProjectorPrepareCompleteStatusChangedListener	1	投影仪准备状态变化监听



	ner		
--	-----	--	--

### 3.5.15.3 调用示例

```
RobotManager.getInstance(this).unRegisterProjectorPrepareCompleteStatusChanged  
Listener(this);
```

## 3.5.16 投影仪控制

### 3.5.16.1 获取投影仪管理实例

#### 3.5.16.1.1 方法名

Method : getProjectorInstance

#### 3.5.16.1.2 调用示例

```
RobotManager.getInstance(this). getProjectorInstance();
```

### 3.5.16.2 打开投影

#### 3.5.16.2.1 方法名

Method : open

#### 3.5.16.2.2 调用示例

```
RobotManager.getInstance(this). getProjectorInstance().open();
```

### 3.5.16.3 关闭投影

#### 3.5.16.3.1 方法名

Method : close





### 3.5.16.3.2 调用示例

```
RobotManager.getInstance(this).getProjectorInstance().close ();
```

### 3.5.16.4 调节亮度

#### 3.5.16.4.1 方法名

Method : changeBrightness(int brightness)

#### 3.5.16.4.2 请求参数

参数名称	类型	约束	描述
brightness	int	1	投影仪亮度 (最小值为 0, 最大值为 5)

#### 3.5.16.4.3 调用示例

```
RobotManager.getInstance(this).getProjectorInstance().changeBrightness (5) ;
```

### 3.5.16.5 梯形校正

#### 3.5.16.5.1 方法名

Method : keystoneCorrection (int angle)

#### 3.5.16.5.2 请求参数

参数名称	类型	约束	描述
angle	int	1	梯形校正角度范围 -28~28 (正 30 度取值 28, 负 30 度取值-28)



#### 3.5.16.5.3 调用示例

```
RobotManager.getInstance(this).getProjectorInstance().keystoneCorrection (20) ;
```

#### 3.5.16.6 桌面正投

##### 3.5.16.6.1 方法名

Method : horizontalLeftOverturn

##### 3.5.16.6.2 调用示例

(无任何变化)

```
RobotManager.getInstance(this).getProjectorInstance().horizontalLeftOverturn () ;
```

#### 3.5.16.7 桌面背投

##### 3.5.16.7.1 方法名

Method : verticalUpOverturn

##### 3.5.16.7.2 调用示例

桌面背投 (左右反转)

```
RobotManager.getInstance(this).getProjectorInstance().verticalUpOverturn () ;
```



### 3.5.16.8 吊装正投

#### 3.5.16.8.1 方法名

Method : horizontalRightOverturn

#### 3.5.16.8.2 调用示例

吊装正投（旋转 180 度（左右上下旋转））

```
RobotManager.getInstance(this).getProjectorInstance().horizontalRightOverturn ();
```

### 3.5.16.9 吊装背投

#### 3.5.16.9.1 方法名

Method : verticalDownOverturn

#### 3.5.16.9.2 调用示例

上下翻转

```
RobotManager.getInstance(this).getProjectorInstance().verticalDownOverturn ();
```

## 3.6 净化器

### 3.6.1 净化器所有状态

```
/**
```

```
 * 净化器状态
```

```
*/
```



```
public enum PurifierStatus {  
    OPEN,  
    CLOSE,  
    INEXISTENCE,  
    FILTER_GAUZE_INEXISTENCE,  
    UNKNOWN,  
}
```

### 3.6.2 净化器状态变化回调

```
/**  
 * 净化器状态变化监听  
 */  
  
public interface OnPurifierStatusChangeListener {  
    void onPurifierStatusChanged(PurifierStatus purifierStatus);  
}
```

### 3.6.3 获取净化器状态

#### 3.6.3.1 方法名

Method : getPurifierStatus

#### 3.6.3.2 返回值

PurifierStatus

return 净化器状态

#### 3.6.3.3 调用示例

```
RobotManager.getInstance(this).getPurifierStatus ();
```

### 3.6.4 注册净化器状态变化监听

#### 3.6.4.1 方法名

Method : registerPurifierStatusChangeListener ( OnPurifierStatusChangeListener



listener)

#### 3.6.4.2 请求参数

参数名称	类型	约束	描述
listener	<p>OnPurifierStatusChangedListen2.1 工程配置</p> <p>把 SDK 文件夹里的 Jar 包集成到 lib 下</p> <p>2.2 权限配置</p> <p>&lt;uses-permission android:name="android.permission.INTERNET"/&gt;</p> <p>3 SDK 方法说明</p> <p>3.1 初始化 SDK</p> <p>控制 SDK 初始化分 2 部，要调用 connect 和 init 方法，先 connect 后 init，详情可参考 Sample</p> <p>如果初始化失败，控制 SDK 将失效，机器人各个状态获取失败和运动失败，请确保申请的 AppKey 为正确并且调用了</p>	1	净化器状态变化监听



	<p>connect 和 init 方法</p> <p>测试账号(仅供测试使用):</p> <pre>String accessKey      = "98838811";  String accessSecret   = "mR07n5fmEC5D DY7yTn55PmdVY J9bcF9Z";  String devId= "568968789";</pre> <p>3.1.1 方法名</p> <p>Method : connect()</p> <p>3.1.2 调用示例</p> <p>RobotManager.getInstance(this). connect ();</p> <p>注：推荐在 application 上的 onCreat()里注册</p> <p>此方法是和下位 机进行通讯，必须 调用</p>		
--	---	--	--



	<p>3.1.3 方法名</p> <p>Method : init(String accessKey,String accessSecret, String appKey, String devId , OnInitCompleteLis tener onInitCompleteList ener)</p> <p>3.1.4 请求参数</p> <p>参数名称 类型 约束 描述</p> <p>accessKey String 1 SDK 开放平台申 请的 accessKey</p> <p>accessSecret String 1 SDK 开放平台申 请的 accessSecret</p>		
--	---	--	--



	<p>devId</p> <p>String</p> <p>1</p> <p>SDK 开放平台申请的 devId</p> <p>onInitCompleteListener</p> <p>OnInitCompleteListener</p> <p>1</p> <p>初始化完成回调</p> <p>3.1.5 调用示例</p> <p>RobotManager.getInstance(this).init(accessKey, accessSecret, devId, this);</p> <p>注：推荐在 application 上的 onCreate()里注册 s</p> <p>如果 SDK 正在初始化或者初始化失败，控制 SDK 的功能将不可用</p> <p>3.2.获取机器人管理实例</p>		
--	--	--	--





	<p>3.2.1 方法名</p> <p>Method : getInstance(Context context)</p> <p>3.2.2 请求参数</p> <p>参数名称 类型 约束 描述</p> <p>context Context 1 上下文参数</p> <p>3.2.3 返回值 RobotManager</p> <p>3.2.4 调用示例 RobotManager.getInstance(context)</p> <p>3.3 获取机器 ID</p> <p>3.3.1 方法名 Method : getRobotId</p> <p>3.3.2 返回值 return 机器人 ID(String), 无号码 返回 null</p>		
--	---	--	--



	<p>3.3.3 调用示例</p> <pre>RobotManager.getInstance(mContext).getRobotId();</pre> <p>3.4 获取机器昵称</p> <p>3.4.1 方法名</p> <p>Method : getRobotName</p> <p>3.4.2 返回值</p> <p>return 机器人昵称 (String), 昵称返回 小胖</p> <p>3.4.3 调用示例</p> <pre>RobotManager.getInstance(mContext).getRobotName ();</pre> <p>er</p>		
--	---	--	--

#### 3.6.4.3 调用示例

```
RobotManager.getInstance(this).registerPurifierStatusChangeListener (this);
```

#### 3.5.5 取消注册净化器状态变化监听

##### 3.6.5.1 方法名

Method : unRegisterPurifierStatusChangedListene (OnPurifierStatu2.1 工程配置  
把 SDK 文件夹里的 Jar 包集成到 lib 下

##### 2.2 权限配置



```
<uses-permissionandroid:name="android.permission.INTERNET"/>
```

### 3 SDK 方法说明

#### 3.1 初始化 SDK

控制 SDK 初始化分 2 部，要调用 connect 和 init 方法，先 connect 后 init，详情可参考 Sample

如果初始化失败，控制 SDK 将失效，机器人各个状态获取失败和运动失败，请确保申请的 AppKey 为正确并且调用了 connect 和 init 方法

测试账号(仅供测试使用):

```
String  accessKey = "98838811";
```

```
String  accessSecret = "mR07n5fmEC5DDY7yTn55PmdVYJ9bcF9Z";
```

```
String  devId= "568968789";
```

##### 3.1.1 方法名

Method : connect()

##### 3.1.2 调用示例

```
RobotManager.getInstance(this).connect ();
```

注：推荐在 application 上的 onCreate()里注册

此方法是和下位机进行通讯，必须调用

##### 3.1.3 方法名

Method : init(String accessKey,String accessSecret, String appKey, String devId ,  
OnInitCompleteListener onInitCompleteListener)

##### 3.1.4 请求参数

参数名称

类型

约束

描述



accessKey

String

1

SDK 开放平台申请的 accessKey

accessSecret

String

1

SDK 开放平台申请的 accessSecret

devId

String

1

SDK 开放平台申请的 devId

onInitCompleteListener

OnInitCompleteListener

1

初始化完成回调

### 3.1.5 调用示例

```
RobotManager.getInstance(this).init(accessKey, accessSecret, devId, this);
```

注：推荐在 application 上的 onCreate() 里注册 s

如果 SDK 正在初始化或者初始化失败，控制 SDK 的功能将不可用

## 3.2. 获取机器人管理实例

### 3.2.1 方法名

Method : getInstance(Context context)

### 3.2.2 请求参数



参数名称

类型

约束

描述

context

Context

1

上下文参数

### 3.2.3 返回值

RobotManager

### 3.2.4 调用示例

RobotManager.getInstance(context)

## 3.3 获取机器 ID

### 3.3.1 方法名

Method : getRobotId

### 3.3.2 返回值

return 机器人 ID(String), 无号码返回 null

### 3.3.3 调用示例

RobotManager.getInstance(mContext).getRobotId();

## 3.4 获取机器昵称

### 3.4.1 方法名

Method : getRobotName

### 3.4.2 返回值

return 机器人昵称(String), 昵称返回小胖

### 3.4.3 调用示例



```
RobotManager.getInstance(mContext). getRobotName ();
```

```
sChangeListener listener)
```

### 3.6.5.2 请求参数

参数名称	类型	约束	描述
listener	OnPurifierStatus ChangeListener	1	净化器状态变化监听

### 3.6.5.3 调用示例

```
RobotManager.getInstance(this).unRegisterPurifierStatusChangeListener (this);
```

## 3.6.6 获取净化器工作时长

### 3.6.6.1 方法名

Method : getPurifierWorkTime

### 3.6.6.2 返回值

PurifierWorkTime

return 净化器工作时长（单位：分钟）

### 3.6.6.3 调用示例

```
RobotManager.getInstance(this).getPurifierWorkTime();
```

## 3.6.7 净化器工作时长回调

```
/**
```

```
    * 净化器工作时长变化监听
```

```
*/
```

```
public interface OnPurifierWorkTimeStatusChangeListener {
```

```
void onPurifierWorkTimeStatusChanged(PurifierWorkTime);}
```



## 3.6.8 注册净化器工作时长监听

### 3.6.8.1 方法名

Method : registerPurifierWorkTimeListener ( (OnPurifierWorkTimeStatusChanged listener)

### 3.6.8.2 请求参数

参数名称	类型	约束	描述
listener	OnPurifierWorkTimeStatusChanged	1	净化器工作时长变化监听

### 3.6.8.3 调用示例

```
RobotManager.getInstance(this).registerPurifierWorkTimeListener(this);
```

## 3.6.9 取消注册净化器工作时长监听

### 3.6.9.1 方法名

Method : unRegisterPurifierWorkTimeListener ( (OnPurifierWorkTimeStatusChanged listener)

### 3.6.9.2 请求参数

参数名称	类型	约束	描述
listener	OnPurifierWorkTimeStatusChanged	1	净化器工作时长变化监听

### 3.6.9.3 调用示例

```
RobotManager.getInstance(this).unRegisterPurifierWorkTimeListener(this);
```



### 3.6.10 获取净化器风机速度

#### 3.6.10.1 方法名

Method : getPurifierFanSpeed

#### 3.6.10.2 返回值

PurifierFanSpeed

return 净化器速度 (最小值为 0, 最大值为 255)

#### 3.6.10.3 调用示例

```
RobotManager.getInstance(this).getPurifierFanSpeed();
```

### 3.6.11 净化器控制

#### 3.6.11.1 获取净化器实例

##### 3.6.11.1.1 方法名

Method : getPurifierInstance

##### 3.6.11.1.2 调用示例

```
RobotManager.getInstance(this).getPurifierInstance();
```





## 3.6.11.2 打开净化器

### 3.6.11.2.1 方法名

Method : open

### 3.6.11.2.2 调用示例

```
RobotManager.getInstance(this).getPurifierInstance().open();
```

## 3.6.11.3 关闭净化器

### 3.6.11.3.1 方法名

Method : close

### 3.6.11.3.2 调用示例

```
RobotManager.getInstance(this).getPurifierInstance().close();
```

## 3.6.11.4 查询净化器状态

### 3.6.11.4.1 方法名

Method : query



#### 3.6.11.4.2 调用示例

```
RobotManager.getInstance(this).getPurifierInstance().query();
```

#### 3.6.11.5 设置净化器风机速度

##### 3.6.11.5.1 方法名

Method : setFanSpeed

##### 3.6.11.5.2 请求参数

参数名称	类型	约束	描述
speed	int	1	净化器风机速度（最小值为 0，最大值为 255）

##### 3.6.11.5.3 调用示例

```
RobotManager.getInstance(this).getPurifierInstance().setFanSpeed(120);
```

## 3.7 电池

### 3.7.1 电量

#### 3.7.1.1 获取电池电量

##### 3.7.1.1.1 方法名

Method : getBatteryLevel



#### 3.7.1.1.2 返回值

int

return 电池电量[0-100], 未获取到返回 -10000

#### 3.7.1.1.3 调用示例

```
RobotManager.getInstance(this).getBatteryLevel ();
```

#### 3.7.1.2 电量状态变化回调

```
/**  
 * 电量状态变化监听  
 */  
public interface OnBatteryLevelChangedListener {  
    void onBatteryLevelChanged(int batteryLevel);  
}
```

#### 3.7.1.3 注册机器人电量变化监听

##### 3.7.1.3.1 方法名

Method : registerBatteryLevelChangedListener(OnBatteryLevelChangedListener listener)

##### 3.7.1.3.2 请求参数

参数名称	类型	约束	描述
listener	OnBatteryLevelChangedListener	1	电量变化监听

##### 3.7.1.3.3 调用示例

```
RobotManager.getInstance(this).registerBatteryLevelChangedListener (this);
```



### 3.7.1.4 取消注册机器人电量变化监听

#### 3.6.1.4.1 方法名

Method : unregisterBatteryLevelChangedListener (OnBatteryLevelChangedListener listener)

#### 3.6.1.4.2 请求参数

参数名称	类型	约束	描述
listener	OnBatteryLevelChangedListener	1	电量变化监听

#### 3.6.1.4.3 调用示例

```
RobotManager.getInstance(this).unRegisterBatteryLevelChangedListener (this);
```

## 3.7.2 电池状态

### 3.7.2.1 电池所有状态

```
/**
 * 电池状态
 */
public enum BatteryStatus {
    /** 未充电 */
    NOT_CHARGING,
    /** 充电中 */
    CHARGINGING,
    /** 未知状态 */
    UNKNOWN,
}
```



### 3.7.2.2 获取电池状态

#### 3.7.2.2.1 方法名

Method : getBatteryStatus

#### 3.7.2.2.2 返回值

BatteryStatus

return 电池状态

#### 3.7.2.2.3 调用示例

```
RobotManager.getInstance(this).getBatteryStatus();
```

### 3.7.2.3 注册电池状态变化监听

#### 3.7.2.3.1 方法名

Method : registerBatteryStatusChangeListener (OnBatteryLevelChangedListener listener)

#### 3.7.2.3.2 请求参数

参数名称	类型	约束	描述
listener	OnBatteryStatusChangeListener	1	电池状态监听

#### 3.7.2.3.3 调用示例

```
RobotManager.getInstance(this).registerBatteryStatusChangeListener (this);
```

### 3.7.2.4 取消注册电池状态变化监听

#### 3.7.2.4.1 方法名

Method : unregisterBatteryStatusChangeListener (OnBatteryLevelChangedListener listener)



#### 3.7.2.4.2 请求参数

参数名称	类型	约束	描述
listener	OnBatteryStatusChangeListener	1	电池状态监听

#### 3.7.2.4.3 调用示例

```
RobotManager.getInstance(this).unRegisterBatteryStatusChangeListener (this);
```

## 3.8 头部触摸

### 3.8.1 头部触摸所有状态

```
/**
 * 头部触摸状态
 */
public enum HeadTouchStatus{
    /** 触摸中 */
    TOUCHING,
    /** 未触摸 */
    NOT_TOUCH,
    /** 未知状态 */
    UNKNOWN,
}
```

### 3.8.2 头顶触摸状态变化回调

```
/**
 * 头顶触摸状态变化监听
 */
public interface OnHeadTouchStatusChangedListener {
    void onHeadTouchStatusChanged(HeadTouchStatus headTouchStatus);
}
```



}

### 3.8.3 获取头部触摸状态

#### 3.8.3.1 方法名

Method : getHeadTouchStatus

#### 3.8.3.2 返回值

HeadTouchStatus

return 头顶按钮状态

#### 3.8.3.3 调用示例

```
RobotManager.getInstance(this).getHeadTouchStatus ();
```

### 3.8.4 注册头部触摸状态改变监听

#### 3.8.4.1 方法名

Method :

registerOnHeadTouchStatusChangeListener (OnHeadTouchStatusChangeListener  
listener)

#### 3.8.4.2 请求参数

参数名称	类型	约束	描述
listener	OnHeadTouchStatusCh angedListener	1	头部状态监听

#### 3.8.4.3 调用示例

```
RobotManager.getInstance(this).registerOnHeadTouchStatusChangeListener(this);
```

### 3.8.5 取消注册头部按钮状态改变监听

#### 3.8.5.1 方法名

Method

:



unRegisterOnHeadTouchStatusChangedListener(OnHeadTouchStatusChangedListener listener)

#### 3.8.5.2 请求参数

参数名称	类型	约束	描述
listener	OnHeadTouchStatusChangedListener	1	电池状态监听

#### 3.8.5.3 调用示例

```
RobotManager.getInstance(this).unRegisterOnHeadTouchStatusChangedListener  
(this);
```

## 3.9 面罩状态

### 3.9.1 面罩所有状态

```
/**  
 * 面罩状态  
 */  
public enum MaskStatus {  
    /** 关闭 */  
    CLOSED,  
    /** 中间 */  
    FLOAT,  
    /** 打开 */  
    OPENED,  
    /** 未知状态 */  
    UNKNOWN,  
}
```





## 3.9.2 获取面罩状态

### 3.9.2.1 方法名

Method : getMaskStatus

### 3.9.2.2 返回值

MaskStatus

return 面罩状态

### 3.9.2.3 调用示例

```
RobotManager.getInstance(this).getMaskStatus ();
```

## 3.9.3 面罩状态变化回调

```
/**  
 * 面罩状态变化监听  
 */  
  
public interface OnMaskStatusChangedListener {  
void onMaskStatusChanged(MaskStatus maskStatus);  
}
```

## 3.9.4 注册面罩状态变化监听

### 3.9.4.1 方法名

Method : registerMaskStatusChangedListener (OnMaskStatusChangedListener listener)

### 3.9.4.2 请求参数

参数名称	类型	约束	描述
listener	OnMaskStatusC hangedListener	1	面罩状态监听



### 3.9.4.3 调用示例

```
RobotManager.getInstance(this).registerMaskStatusChangedListener (this);
```

### 3.9.5 取消注册面罩状态变化监听

#### 3.9.5.1 方法名

Method : unRegisterMaskStatusChangedListener (OnMaskStatusChangedListener listener)

#### 3.9.5.2 请求参数

参数名称	类型	约束	描述
listener	OnMaskStatusChangedListener	1	面罩状态监听

#### 3.9.5.3 调用示例

```
RobotManager.getInstance(this).unRegisterMaskStatusChangedListener (this);
```

## 3.10 托盘状态

### 3.10.1 托盘所有状态

```
/**
 * 托盘状态
 */
public enum SalverStatus {
    /** 关闭 */
    CLOSED,
    /** 打开 */
    OPENED,
    /** 未知状态 */
    UNKNOWN,
```



```
}
```

### 3.10.2 托盘状态变化回调

```
/**
```

```
 * 托盘状态变化监听
```

```
*/
```

```
public interface OnSalverStatusChangedListener {  
void onSalverStatusChanged(SalverStatus salverStatus);  
}
```

### 3.10.3 获取托盘状态

#### 3.10.3.1 方法名

Method : getSalverStatus

#### 3.10.3.2 返回值

SalverStatus

return 托盘状态

#### 3.10.3.3 调用示例

```
RobotManager.getInstance(this).getSalverStatus ();
```

### 3.10.4 注册托盘状态变化监听

#### 3.10.4.1 方法名

Method : registerSalverStatusChangedListener (OnSalverStatusChangedListener listener)

#### 3.10.4.2 请求参数

参数名称	类型	约束	描述
listener	OnSalverStatusChangedListener	1	托盘状态监听



### 3.10.4.3 调用示例

```
RobotManager.getInstance(this).registerSalverStatusChangedListener (this);
```

## 3.10.5 取消注册托盘状态变化监听

### 3.10.5.1 方法名

Method : unRegisterSalverStatusChangedListener (OnSalverStatusChangedListener listener)

### 3.10.5.2 请求参数

参数名称	类型	约束	描述
listener	OnSalverStatus ChangedListene r	1	托盘状态监听

### 3.10.5.3 调用示例

```
RobotManager.getInstance(this).unRegisterSalverStatusChangedListener (this);
```

## 3.11 头部控制

### 3.11.1 获取机器人头部管理实例

#### 3.11.1.1 方法名

Method : getHeadInstance

#### 3.11.1.2 返回值

HeadManager

return 机器人头部管理实例

#### 3.11.1.3 调用示例

```
RobotManager.getInstance(this).getHeadInstance ();
```



### 3.11.2 以默认速度移动到头部的最左端

#### 3.11.2.1 方法名

Method : moveLeft

#### 3.11.2.2 调用示例

```
RobotManager.getInstance(this).getHeadInstance ().moveLeft ();
```

### 3.11.3 以指定的速度移动到头部的最左端

#### 3.11.3.1 方法名

Method : moveLeft (int speed)

#### 3.11.3.2 请求参数

参数名称	类型	约束	描述
speed	int	1	头部移动速度

#### 3.11.3.3 调用示例

```
RobotManager.getInstance(this).getHeadInstance ().moveLeft (50);
```

### 3.11.4 以默认速度移动到头部的最右端

#### 3.11.4.1 方法名

Method : moveRight

#### 3.11.4.2 调用示例

```
RobotManager.getInstance(this).getHeadInstance ().moveRight ();
```

### 3.11.5 以指定的速度移动到头部的最右端

#### 3.11.5.1 方法名

Method : moveRight (int speed)



### 3.11.5.2 请求参数

参数名称	类型	约束	描述
speed	int	1	头部移动速度

### 3.11.5.3 调用示例

```
RobotManager.getInstance(this).getHeadInstance ().moveRight (50);
```

## 3.11.6 以指默认速度移动到头部指定的角度

### 3.11.6.1 方法名

Method : moveToByAngle (int angle)

### 3.11.6.2 请求参数

参数名称	类型	约束	描述
angle	int	1	头部转动角度

### 3.11.6.3 调用示例

```
RobotManager.getInstance(this).getHeadInstance ().moveToByAngle(50);
```

## 3.11.7 以指定的速度移动到头部指定的角度

### 3.11.7.1 方法名

Method : moveToByAngle ((int angle, int speed)

### 3.11.7.2 请求参数

参数名称	类型	约束	描述
angle	int	1	头部转动角度
speed	int	1	头部移动速度

### 3.11.7.3 调用示例

```
RobotManager.getInstance(this).getHeadInstance ().moveToByAngle(50, 50);
```



### 3.11.8 停止头部运动

#### 3.11.8.1 方法名

Method : stop

#### 3.11.8.2 调用示例

```
RobotManager.getInstance(this).getHeadInstance ().stop ();
```

## 3.12 双翅控制

@param movePart 运动部位

WingManager.MOVE\_WING:表示双翅

WingManager.MOVE\_LEFT\_WING:表示左翅

WingManager.MOVE\_RIGHT\_WING:表示右翅

@param speed 运动速度 (0-100)

@param angle 翅膀角度 (0-90 度)

### 3.12.1 获取机器人翅膀管理实例

#### 3.12.1.1 方法名

Method : getWingInstance

#### 3.12.1.2 返回值

WingManager

return 机器人翅膀管理实例

#### 3.12.1.3 调用示例

```
RobotManager.getInstance(this).getWingInstance ();
```



## 3.12.2 翅膀以默认速度向上运动

### 3.12.2.1 方法名

Method : moveUp (intmovePart)

### 3.12.2.2 请求参数

参数名称	类型	约束	描述
movePart	int	1	翅膀运动部位

### 3.12.2.3 调用示例

```
RobotManager.getInstance(this).getWingInstance                .moveUp  
(WingManager.MOVE_WING);
```

## 3.12.3 翅膀以指定的速度向上运动

### 3.12.3.1 方法名

Method : moveUp (intmovePart)

### 3.12.3.2 请求参数

参数名称	类型	约束	描述
movePart	int	1	翅膀运动部位
speed	Int	1	翅膀运动速度

### 3.12.3.3 调用示例

```
RobotManager.getInstance(this).getWingInstance                .moveUp  
(WingManager.MOVE_WING, 50);
```

## 3.12.4 翅膀以默认速度向下运动

### 3.12.4.1 方法名

Method : moveDown (intmovePart)





#### 3.12.4.2 请求参数

参数名称	类型	约束	描述
movePart	int	1	翅膀运动部位

#### 3.12.4.3 调用示例

```
RobotManager.getInstance(this).getWingInstance().moveDown  
(WingManager.MOVE_WING);
```

### 3.12.5 翅膀以指定的速度向下运动

#### 3.12.5.1 方法名

Method : moveDown (intmovePart, int speed)

#### 3.12.5.2 请求参数

参数名称	类型	约束	描述
movePart	int	1	翅膀运动部位
speed	Int	1	翅膀运动速度

#### 3.12.5.3 调用示例

```
RobotManager.getInstance(this).getWingInstance().moveDown(WingManager.MOVE_WI  
NG, 50);
```

### 3.12.6 翅膀以默认速度向指定角度运动

#### 3.12.6.1 方法名

Method : moveToByAngle (intmovePart, int angle)

#### 3.12.6.2 请求参数

参数名称	类型	约束	描述
movePart	int	1	翅膀运动部位
angle	int	1	翅膀运动角度



### 3.12.6.3 调用示例

```
RobotManager.getInstance(this).getWingInstance                .moveToByAngle  
(WingManager.MOVE_WING, 90);
```

## 3.12.7 翅膀以指定的速度向指定角度运动

### 3.12.7.1 方法名

Method : moveToByAngle (intmovePart, int angle)

### 3.12.7.2 请求参数

参数名称	类型	约束	描述
movePart	int	1	翅膀运动部位
angle	int	1	翅膀运动角度
speed	int	1	翅膀运动速度

### 3.12.7.3 调用示例

```
RobotManager.getInstance(this).getWingInstance                .moveToByAngle  
(WingManager.MOVE_WING, 90, 50);
```

## 3.12.8 翅膀停止运动

### 3.12.8.1 方法名

Method : stop(intmovePart)

### 3.12.8.2 请求参数

参数名称	类型	约束	描述
movePart	int	1	翅膀运动部位

### 3.11.8.3 调用示例

```
RobotManager.getInstance(this).getWingInstance                .stop  
(WingManager.MOVE_WING);
```



## 3.13 双轮控制

### 3.13.1 获取机器人双轮管理实例

#### 3.13.1.1 方法名

Method : getWheelInstance

#### 3.13.1.2 返回值

WheelManager

return 机器人底部双轮管理实例

#### 3.13.1.3 调用示例

```
RobotManager.getInstance(this).getWheelInstance ();
```

### 3.13.2 以默认速度向前运动

#### 3.13.2.1 方法名

Method : moveFront

#### 3.13.2.2 调用示例

```
RobotManager.getInstance(this).getWheelInstance ().moveFront;
```

### 3.13.3 以指定的速度向前运动

#### 3.13.3.1 方法名

Method : moveFront (int speed)

#### 3.13.3.2 请求参数

参数名称	类型	约束	描述
speed	int	1	双轮运动速度（最小为 0， 最大为 400）



### 3.13.3.3 调用示例

```
RobotManager.getInstance(this).getWingInstance().moveFront (100);
```

### 3.13.4 以默认速度向后运动

#### 3.13.4.1 方法名

Method : moveBack

#### 3.13.4.2 调用示例

```
RobotManager.getInstance(this).getWingInstance () .moveBack ();
```

### 3.13.5 以指定的速度向后运动

#### 3.13.5.1 方法名

Method : moveBack (int speed)

#### 3.13.5.2 请求参数

参数名称	类型	约束	描述
speed	int	1	双轮运动速度（最小为 0， 最大为 400）

#### 3.13.5.3 调用示例

```
RobotManager.getInstance(this).getWingInstance () .moveBack (100);
```

### 3.13.6 以默认速度向左运动

#### 3.13.6.1 方法名

Method : moveLeft

#### 3.13.6.2 调用示例

```
RobotManager.getInstance(this).getWingInstance () .moveLeft ();
```



### 3.13.7 以指定的速度向左运动

#### 3.13.7.1 方法名

Method : moveLeft (int speed)

#### 3.13.7.2 请求参数

参数名称	类型	约束	描述
speed	int	1	双轮运动速度（最小为 0， 最大为 400）

#### 3.13.7.3 调用示例

```
RobotManager.getInstance(this).getWingInstance ().moveLeft (100);
```

### 3.13.8 以默认速度向右运动

#### 3.13.8.1 方法名

Method : moveRight

#### 3.13.8.2 调用示例

```
RobotManager.getInstance(this).getWingInstance ().moveRight ();
```

### 3.13.9 以指定的速度向右运动

#### 3.13.9.1 方法名

Method : moveRight (int speed)

#### 3.13.9.2 请求参数

参数名称	类型	约束	描述
speed	int	1	双轮运动速度 (最小为 0, 最大为 400)

#### 3.13.9.3 调用示例

```
RobotManager.getInstance(this).getWingInstance ().moveRight (100);
```



### 3.13.10 双轮停止运动

#### 3.13.10.1 方法名

Method : stop

#### 3.13.10.2 调用示例

```
RobotManager.getInstance(this).getWingInstance ().stop ();
```

### 3.13.11 双轮禁止功能

#### 3.13.11.1 双轮禁止开关开启（禁用双轮）

##### 3.13.11.1.1 方法名

Method : disableWheel

##### 3.13.11.1.2 调用示例

```
RobotManager.getInstance(this).disableWheel();
```

#### 3.13.11.2 双轮禁止开关关闭（启用双轮）

##### 3.13.11.2.1 方法名

Method : enableWheel

##### 3.13.11.2.2 调用示例

```
RobotManager.getInstance(this).enableWheel();
```



### 3.13.11.3 双轮禁止开关所有状态

```
/**
 * 双轮禁止开关状态(ENABLE: 禁止功能关闭  DISABLE: 禁止功能开启)
 */
public enum WheelSwitchStatus {
    /** 开启状态 */
    ENABLE,
    /** 关闭状态 */
    DISABLE,
    /** 未知状态 */
    UNKNOWN,
    /** 初始化失败 */
    FAILURE;
    static WheelSwitchStatus getWheelSwitchStatus(int wheelSwitchStatus) {
        WheelSwitchStatus[] values = WheelSwitchStatus.values();
        return values[wheelSwitchStatus];
    }
}
```

### 3.13.11.4 双轮禁止开关状态变化回调

```
/**
 * 双轮禁止开关的状态变化回调
 */
public interface OnWheelSwitchStatusChangedListener{
    void onWheelSwitchStatusChanged(RobotStatus.WheelSwitchStatus
wheelSwitchStatus);
}
```



### 3.13.11.5 获取双轮禁止开关状态

#### 3.13.11.5.1 方法名

Method : getWheelSwitchStatus

#### 3.13.11.5.2 返回值

WheelSwitchStatus

return 双轮禁止开关状态

#### 3.13.11.5.3 调用示例

```
RobotManager.getInstance(this).getWheelSwitchStatus();
```

### 3.13.11.6 注册双轮禁止开关状态变化监听

#### 3.13.11.6.1 方法名

Method: registerWheelSwitchChangeListener ( OnWheelSwitchStatusChangeListener listener)

#### 3.13.11.6.2 请求参数

参数名称	类型	约束	描述
listener	OnWheelSwitchStatusChangeListener	1	双轮禁止开关状态变化监听





#### 3.13.11.6.3 调用示例

```
RobotManager.getInstance(this).registerWheelSwitchChangeListener(this);
```

#### 3.13.11.7 取消双轮禁止开关状态变化监听

##### 3.13.11.7.1 方法名

Method : unRegisterWheelSwitchChangeListener(OnWheelSwitchStatusChangedListener listener)

##### 3.13.11.7.2 请求参数

参数名称	类型	约束	描述
listener	OnWheelSwitchStatusChangedListener	1	双轮禁止开关状态变化监听

##### 3.13.11.7.3 调用示例

```
RobotManager.getInstance(this).unRegisterWheelSwitchChangeListener(this);
```

### 3.14 控制管理

#### 3.14.1 获取机器人控制实例

##### 3.14.1.1 方法名

Method : getControlInstance



### 3.14.1.2 返回值

ControlManager

return 机器人控制实例

### 3.14.1.3 调用示例

```
RobotManager.getInstance(this).getControlInstance ();
```

## 3.14.2 设置灯带亮度

### 3.14.2.1 方法名

Method : setLightBeltBrightness (int brightness)

### 3.14.2.2 请求参数

参数名称	类型	约束	描述
brightness	int	1	灯带亮度（最小值为 0，最大值为 255）

### 3.14.2.3 调用示例

```
RobotManager.getInstance(this).getControlInstance ().setLightBeltBrightness (100);
```

## 3.15 自定义机器人说话

### 3.15.1 方法名

Method : executeSpeak (String content)

### 3.15.2 请求参数

参数名称	类型	约束	描述
content	String	1	机器人说话内容



### 3.14.3 调用示例

```
RobotManager.getInstance(this).executeSpeak("我是进化者机器人公司制造的小胖");
```

## 3.16 电源按键状态

### 3.16.1 电源按键所有状态

```
/**
 * 电源键状态
 */
public enum PowerKeyStatus {
    /** 按下状态 */
    DOWN,
    /** 抬起状态 */
    UP,
    /** 未知状态 */
    UNKNOWN,
    /** 初始化失败 */
    FAILURE;

    static PowerKeyStatus getPowerKeyStatus(int powerKeyStatus) {
        PowerKeyStatus[] values = PowerKeyStatus.values();
        return values[powerKeyStatus];
    }
}
```

### 3.16.2 电源按键状态变化回调

```
/**
 * 电源按键状态变化监听
 */
```



```
public interface OnPowerKeyStatusChangedListener {  
    void onPowerKeyStatusChanged(RobotStatus.PowerKeyStatus powerKeyStatus);  
}
```

### 3.16.3 获取电源按键状态

#### 3.16.3.1 方法名

Method : getPowerKeyStatus

#### 3.16.3.2 返回值

PowerKeyStatus

return 电源按键状态

#### 3.16.3.3 调用示例

```
RobotManager.getInstance(this).getPowerKeyStatus();
```

### 3.16.4 注册电源按键状态变化监听

#### 3.16.4.1 方法名

Method : registerPowerKeyStatusChangedListener(OnPowerKeyStatusChangedListener listener)

#### 3.16.4.2 请求参数

参数名称	类型	约束	描述
listener	OnPowerKeyStatusChangedListener	1	电源按键状态监听

#### 3.16.4.3 调用示例

```
RobotManager.getInstance(this).registerPowerKeyStatusChangedListener(this);
```



## 3.16.5 取消注册托盘状态变化监听

### 3.16.5.1 方法名

Method :  
unRegisterPowerKeyStatusChangedListener(OnPowerKeyStatusChangedListenerlistener)

### 3.16.5.2 请求参数

参数名称	类型	约束	描述
listener	OnPowerKeySta tusChangedListe nerlistener	1	电源按键状态监听

### 3.16.5.3 调用示例

```
RobotManager.getInstance(this).unRegisterPowerKeyStatusChangedListener(this);
```

## 3.17 SOS 状态

### 3.17.1 SOS 按钮所有状态

```
/**  
 * SOS 状态  
 */  
public enum SosKeyStatus {  
    /** 按下状态 */  
    DOWN,  
    /** 抬起状态 */  
    UP,  
    /** 未知状态 */  
    UNKNOWN,  
    /** 初始化失败 */
```



FAILURE;

```
static SosKeyStatus getSosKeyStatus(int sosKeyStatus) {  
    SosKeyStatus[] values = SosKeyStatus.values();  
    return values[sosKeyStatus];  
}  
}
```

### 3.17.2 SOS 按钮状态变化回调

```
/**  
 * SOS 按钮状态变化监听  
 */  
public interface OnSosStatusChangedListener {  
    void onSosStatusChanged(RobotStatus.SosKeyStatus sosKeyStatus);  
}
```

### 3.17.3 获取 SOS 按钮状态

#### 3.17.3.1 方法名

Method : `getPowerKeyStatus`

#### 3.17.3.2 返回值

`getSosStatus`  
return SOS 状态

#### 3.17.3.3 调用示例

```
RobotManager.getInstance(this).getSosStatus();
```

### 3.17.4 注册 SOS 按钮状态变化监听

#### 3.17.4.1 方法名

Method : `registerSosStatusChangedListener(OnSosStatusChangedListener)`



#### 3.17.4.2 请求参数

参数名称	类型	约束	描述
listener	OnSosStatusC hangedListener	1	SOS 按钮状态监听

#### 3.17.4.3 调用示例

```
RobotManager.getInstance(this).registerSosStatusChangedListener(this);
```

### 3.17.5 取消注册 SOS 按钮状态变化监听

#### 3.17.5.1 方法名

Method : unRegisterSosStatusChangedListener(OnSosStatusChangedListener)

#### 3.17.5.2 请求参数

参数名称	类型	约束	描述
listener	OnSosStatusCha ngedListener	1	SOS 按钮状态监听

#### 3.17.5.3 调用示例

```
RobotManager.getInstance(this).unRegisterSosStatusChangedListener(this);
```

## 3.18 导航模块

### 3.18.1 获取机器人导航管理实例

#### 3.18.1.1 方法名

Method : getNavigationInstance

#### 3.18.1.2 返回值

NavigationManager

return 机器人导航管理实例



### 3.18.1.3 调用示例

```
RobotManager.getInstance(this).getNavigationInstance();
```

## 3.18.2 注册导航状态变化监听

### 3.18.2.1 方法名

Method : registerOnNavigationStateChangeListener(OnNavigationStateChangeListener mOnNavigationStateChangeListener)

### 3.18.2.2 请求参数

参数名称	类型	约束	描述
mOnNavigationStateChangeListener	OnNavigationStateChangeListener	1	导航状态变化监听

### 3.18.2.2 调用示例

```
RobotManager.getInstance(this).getNavigationInstance().registerOnNavigationStateChangeListener(this);
```

```
@Override
```

```
public void onNavigationStart() {  
    Log.d(TAG, "onNavigationStart: ");  
}
```

```
@Override
```

```
public void onNavigationPause() {  
    Log.d(TAG, "onNavigationPause: ");  
}
```

```
@Override
```





```
public void onNavigationContinue() {  
    Log.d(TAG, "onNavigationContinue: ");  
}  
  
@Override  
public void onNavigationStop() {  
    Log.d(TAG, "onNavigationStop: ");  
}  
  
@Override  
public void onNavigationSuccess() {  
    Log.d(TAG, "onNavigationSuccess: ");  
}  
  
@Override  
public void onNavigationFail(int reason) {  
    Log.d(TAG, "onNavigationFail: ");  
}
```

### 3.18.3 取消注册导航状态变化监听

#### 3.18.3.1 方法名

Method : unregisterOnNavigationStateChangeListener(OnNavigationStateChangeListener  
mOnNavigationStateChangeListener)

#### 3.18.3.2 请求参数

参数名称	类型	约束	描述
mOnNavigationS tateChangeListen	OnNavigationSt ateChangeListen	1	导航状态变化监听



er	er		
----	----	--	--

### 3.18.3.2 调用示例

```
RobotManager.getInstance(this).getNavigationInstance().unRegisterOnNavigationSta  
teChangeListener(this);
```

```
@Override
```

```
public void onNavigationStart() {  
    Log.d(TAG, "onNavigationStart: ");  
}
```

```
@Override
```

```
public void onNavigationPause() {  
    Log.d(TAG, "onNavigationPause: ");  
}
```

```
@Override
```

```
public void onNavigationContinue() {  
    Log.d(TAG, "onNavigationContinue: ");  
}
```

```
@Override
```

```
public void onNavigationStop() {  
    Log.d(TAG, "onNavigationStop: ");  
}
```

```
@Override
```

```
public void onNavigationSuccess() {  
    Log.d(TAG, "onNavigationSuccess: ");  
}
```



```
@Override  
public void onNavigationFail(int reason) {  
    Log.d(TAG, "onNavigationFail: ");  
}
```

### 3.18.4 导航状态回调

```
**  
* 导航状态监听  
*/  
public interface OnNavigationStateChangeListener {  
    /**  
     * 导航开始  
     */  
    void onNavigationStart();  
  
    /**  
     * 导航暂停  
     */  
    void onNavigationPause();  
  
    /**  
     * 导航继续  
     */  
    void onNavigationContinue();  
  
    /**  
     * 导航停止  
     */  
    void onNavigationStop();
```



```
/**
 * 导航成功
 */
void onNavigationSuccess();

/**
 * 导航失败
 */
void onNavigationFail(int reason);
}
```

### 3.18.5 导航去指定地点

#### 3.18.5.1 方法名

Method : startNavigation(float endX, float endY, float endAngle)

#### 3.18.5.2 请求参数

参数名称	类型	约束	描述
endX	float	1	终点 X 坐标
endY	float	1	终点 Y 坐标
endAngle	float	1	终点角度

#### 3.18.5.3 调用示例

```
RobotManager.getInstance(this).getNavigationInstance().startNavigation(407, 405, 1);
```

### 3.18.6 暂停导航

#### 3.18.6.1 方法名

Method : pauseNavigation()



### 3.18.6.2 调用示例

```
RobotManager.getInstance(this).getNavigationInstance().pauseNavigation();
```

## 3.18.7 继续导航

### 3.18.7.1 方法名

Method : continueNavigation()

### 3.18.7.2 调用示例

```
RobotManager.getInstance(this).getNavigationInstance().continueNavigation();
```

## 3.18.8 停止导航

### 3.18.8.1 方法名

Method : stop()

### 3.18.8.2 调用示例

```
RobotManager.getInstance(this).getNavigationInstance().stop();
```

## 3.18.9 回充电桩（无监听）

### 3.18.9.1 方法名

Method : goToCharging()

### 3.18.9.2 调用示例

```
RobotManager.getInstance(this).getNavigationInstance().goToCharging();
```

## 3.18.10 回充电桩（有监听）

### 3.18.10.1 方法名

Method :

goToCharging(OnNavigationStateChangeListener mOnNavigationStateChangeListener)



### 3.18.10.2 请求参数

参数名称	类型	约束	描述
mOnNavigationStateChangeListener	OnNavigationStateChangeListener	1	导航状态变化监听

### 3.18.10.3 调用示例

```
RobotManager.getInstance(this).getNavigationInstance().goToCharging(this);
```

```
@Override
```

```
public void onNavigationStart() {  
    Log.d(TAG, "onNavigationStart: ");  
}
```

```
@Override
```

```
public void onNavigationPause() {  
    Log.d(TAG, "onNavigationPause: ");  
}
```

```
@Override
```

```
public void onNavigationContinue() {  
    Log.d(TAG, "onNavigationContinue: ");  
}
```

```
@Override
```

```
public void onNavigationStop() {  
    Log.d(TAG, "onNavigationStop: ");  
}
```

```
@Override
```



```
public void onNavigationSuccess() {  
    Log.d(TAG, "onNavigationSuccess: ");  
}  
  
@Override  
public void onNavigationFail(int reason) {  
    Log.d(TAG, "onNavigationFail: ");  
}
```

### 3.18.11 出充电桩（无监听）

#### 3.18.11.1 方法名

Method : outChargingPile()

#### 3.18.11.2 调用示例

```
RobotManager.getInstance(this).getNavigationInstance().outChargingPile();
```

### 3.18.12 出充电桩（有监听）

#### 3.18.12.1 方法名

Method :

outChargingPile(OnNavigationStateChangeListener mOnNavigationStateChangeListener)

#### 3.18.12.2 请求参数

参数名称	类型	约束	描述
mOnNavigationStateChangeListener	OnNavigationStateChangeListener	1	导航状态变化监听

#### 3.18.12.3 调用示例

```
RobotManager.getInstance(this).getNavigationInstance().outChargingPile(this);  
@Override
```



```
public void onStart() {  
    Log.d(TAG, "onStart: ");  
}  
  
@Override  
public void onPause() {  
    Log.d(TAG, "onPause: ");  
}  
  
@Override  
public void onResume() {  
    Log.d(TAG, "onResume: ");  
}  
  
@Override  
public void onStop() {  
    Log.d(TAG, "onStop: ");  
}  
  
@Override  
public void onSuccess() {  
    Log.d(TAG, "onSuccess: ");  
}  
  
@Override  
public void onFailure(int reason) {  
    Log.d(TAG, "onFailure: ");  
}
```





### 3.18.13 手动接驳

#### 3.18.13.1 方法名

Method : connectChargingPile()

#### 3.18.13.2 调用示例

```
RobotManager.getInstance(this).getNavigationInstance().connectChargingPile();
```

#### 3.18.13.3 方法说明

使用方式：将机器人电极与充电桩电机触碰，调用该方法来接驳充电

### 3.18.14 不依赖地图回充电桩（无监听）

#### 3.18.14.1 方法名

Method : goToChargingWithoutMap()

#### 3.18.14.2 调用示例

```
RobotManager.getInstance(this).getNavigationInstance().goToChargingWithoutMap(  
);
```

### 3.18.15 不依赖地图回充电桩（有监听）

#### 3.18.15.1 方法名

Method :

goToChargingWithoutMap(OnNavigationStateChangeListener  
mOnNavigationStateChangeListener)

#### 3.18.15.2 请求参数

参数名称	类型	约束	描述
mOnNavigationS tateChangeListen er	OnNavigati onStateCha ngeListener	1	导航状态变化监听



### 3.18.15.3 调用示例

```
RobotManager.getInstance(this).getNavigationInstance().goToChargingWithoutMap(this);
```

```
@Override
```

```
public void onNavigationStart() {  
    Log.d(TAG, "onNavigationStart: ");  
}
```

```
@Override
```

```
public void onNavigationPause() {  
    Log.d(TAG, "onNavigationPause: ");  
}
```

```
@Override
```

```
public void onNavigationContinue() {  
    Log.d(TAG, "onNavigationContinue: ");  
}
```

```
@Override
```

```
public void onNavigationStop() {  
    Log.d(TAG, "onNavigationStop: ");  
}
```

```
@Override
```

```
public void onNavigationSuccess() {  
    Log.d(TAG, "onNavigationSuccess: ");  
}
```

```
@Override
```



```
public void onNavigationFail(int reason) {  
    Log.d(TAG, "onNavigationFail: ");  
}
```

### 3.18.16 获取地图标记点

#### 3.18.14.1 方法名

Method : `getAllPoint ()`

#### 3.18.14.2 调用示例

```
RobotManager.getInstance(this).getNavigationInstance().getAllPoint();
```

注意：没有标记点时将返回 `null`

## 3.19 脚本任务管理模块

### 3.19.1 获取机器人脚本任务管理实例

#### 3.19.1.1 方法名

Method : `getGroupInstance`

#### 3.19.1.2 返回值

`GroupManager`

return 机器人脚本任务管理实例

#### 3.19.1.3 调用示例

```
RobotManager.getInstance(this).getGroupInstance();
```

### 3.19.2 脚本状态回调

/\*\*



\* 脚本任务回调

\*/

```
public interface OnGroupTaskExecuteListener {  
    void onStart();  
    void onStop();  
}
```

### 3.19.3 开始执行脚本

#### 3.19.3.1 方法名

Method : execute(String tasksContent)

#### 3.19.3.2 请求参数

参数名称	类型	约束	描述
tasksContent	String	1	脚本内容

#### 3.19.3.3 调用示例

```
try {  
    RobotManager.getInstance(this).getGroupInstance().execute("{\n" +  
        "    \"actions\": [\n" +  
        "        {\n" +  
        "            \"wheel\": {\n" +  
        "                \"direction\": \"back_distance\",\n" +  
        "                \"speed\": 250,\n" +  
        "                \"distance\": 1000\n" +  
        "            }\n" +  
        "        }\n" +  
        "    ]\n" +  
        "});  
    } catch (Exception e) {  
        e.printStackTrace();  
    }
```



```
}
```

### 3.19.4 开始执行脚本（有回调）

#### 3.19.4.1 方法名

Method : execute(String tasksContent,  
OnGroupTaskExecuteListener mOnGroupTaskExecuteListener)

#### 3.19.4.2 请求参数

参数名称	类型	约束	描述
tasksContent	String	1	脚本内容
mOnGroupTaskExecuteListener	OnGroupTaskExecuteListener	1	脚本任务回调

#### 3.19.4.3 调用示例

```
try {  
    RobotManager.getInstance(this).getGroupInstance().execute("{\n" +  
        "    \"actions\": [\n" +  
        "        {\n" +  
        "            \"wheel\": {\n" +  
        "                \"direction\": \"front_distance\",\n" +  
        "                \"speed\": 250,\n" +  
        "                \"distance\": 1000\n" +  
        "            }\n" +  
        "        }\n" +  
        "    ]\n" +  
        "}", new GroupManager.OnGroupTaskExecuteListener() {  
        @Override
```



```
        public void onStart() {  
            Log.d(TAG, "onStart: ");  
        }  
  
        @Override  
        public void onStop() {  
            Log.d(TAG, "onStop: ");  
        }  
    });  
} catch (Exception e) {  
    e.printStackTrace();  
}
```

### 3.19.5 停止脚本

#### 3.19.5.1 方法名

Method : stop()

#### 3.19.5.2 调用示例

```
RobotManager.getInstance(this).getGroupInstance().stop();
```

### 3.19.6 重置机器人（头部、双翅、灯带状态还原）

#### 3.19.6.1 方法名

Method : reset()

#### 3.19.6.2 调用示例

```
RobotManager.getInstance(this).getGroupInstance().reset();
```



## 3.20 DIY 控制模块

### 3.20.1 获取机器人 DIY 控制管理实例

#### 3.20.1.1 方法名

Method : `getDiyControlInstance`

#### 3.20.1.2 返回值

`DiyControlManager`

return 机器人 DIY 控制管理实例

#### 3.20.1.3 调用示例

```
RobotManager.getInstance(this).getDiyControlInstance();
```

### 3.20.2 获取机器人 DIY 源数据

#### 3.20.2.1 方法名

Method : `getAllDiyData(Context mContext)`

#### 3.20.2.2 返回值

String return JSON 字符串

#### 3.20.2.3 请求参数

参数名称	类型	约束	描述
mContext	Context	1	上下文

#### 3.20.2.4 调用示例

```
try {  
RobotManager.getInstance(this).getDiyControlInstance().getAllDiyData(mContext);  
} catch (Exception e) {  
e.printStackTrace();  
}
```



```
}
```

结果:

```
[
```

```
{
```

```
  "data":[
```

```
    {
```

```
      "answers":[
```

```
        {
```

```
        "content":{"file_md5":"","file_name":"","file_path":"","time_tasks":[{"actions":[{"action_
content":"3","action_name":"摇头
","action_time":"3.6","action_pclId":-1,"action_type":0,"id":0}], "expressions":[{"expressio
n_name":"傲慢","expression_id":1}], "text":"我叫 ROBOT_NICKNAME_VALUE, 我将尽我
的 一 切 给 主 人 带 来 欢 乐
","frequency":0,"stop_time":7,"start_time":0,"time":7}], "file_url":"","name":"","file_size":0
,"total_time":7,"type":3}"
```

```
        }
```

```
      ],
```

```
      "smallId":"1",
```

```
      "questions":[
```

```
        {
```

```
          "content":"该怎么称呼你"
```

```
        }
```

```
      ]
```

```
    },
```

```
  ],
```

```
  "id":"3",
```

```
  "name":"第三大场景"
```

```
}
```





]

### 3.20.3 获取机器人 DIY 大场景

#### 3.20.3.1 方法名

Method : public List<DiyItemScene> getItemScene(Context context)

#### 3.20.3.2 返回值

List return 大场景对象 list

#### 3.20.3.3 请求参数

参数名称	类型	约束	描述
context	Context	1	上下文

#### 3.20.3.4 调用示例

```
try {  
    RobotManager.getInstance(context).getDiyControlInstance().getItemScene(context);  
} catch (Exception e) {  
    e.printStackTrace();  
}
```

### 3.20.4 获取机器人 DIY 小场景(根据大场景 Id 来获取小场景 id)

#### 3.20.4.1 方法名

Method : public List<String> getSmallScene(Context context, String relationId)

#### 3.20.4.2 返回值

List return 小场景 list

#### 3.20.4.3 请求参数

参数名称	类型	约束	描述
------	----	----	----



context	Context	1	上下文
relationId	String	1	大场景 Id

#### 3.20.4.4 调用示例

```
try {  
String  
smallSceneIdList=RobotManager.getInstance(context).getDiyControlInstance().getS  
mallScene(context,1);  
Log.d(TAG, "smallSceneIdList:" + smallSceneIdList);  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}  
结果: smallSceneIdList:[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 18, 18, 20,  
20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32]
```

#### 3.20.5 获取机器人 DIY 问题(根据小场景 Id 获取问题)

##### 3.20.5.1 方法名

Method : public String getQuestions(Context context, String smallId)

##### 3.20.5.2 返回值

String return 小场景问题

##### 3.20.5.3 请求参数

参数名称	类型	约束	描述
context	Context	1	上下文
smallId	String	1	小场景 Id

##### 3.20.5.4 调用示例

```
try {  
String
```



```
questions=RobotManager.getInstance(context).getDiyControlInstance().getQuestions(
context,2);
Log.d(TAG, "question:" + questions);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

结果:

question: 你是谁

### 3.20.6 获取机器人 DIY 回答(根据小场景 Id 获取问题)

#### 3.20.6.1 方法名

Method : public String getAnswers(Context context, String smallId)

#### 3.20.6.2 返回值

String return 小场景答案

#### 3.20.6.3 请求参数

参数名称	类型	约束	描述
context	Context	1	上下文
smallId	String	1	小场景 Id

#### 3.20.6.4 调用示例

```
try {
String
answers=RobotManager.getInstance(context).getDiyControlInstance().getAnswers(co
ntext,2);
Log.d(TAG, "answers:" + answers);
}
```



```
} catch (Exception e) {  
    e.printStackTrace();  
}  
结果:  
answers:{"file_md5":"","file_name":"","file_path":"","time_tasks":[{"actions":[{"action_content":"3","action_name":"摇头","action_time":"3.6","action_pcId":-1,"action_type":0,"id":0}],expressions":[{"expression_name":"闭嘴","expression_id":2}],text:"我是能给您带来欢乐的机器人 ROBOT_NICKNAME_VALUE","frequency":0,"stop_time":6,"start_time":0,"time":6}],file_url":"","name":"","file_size":0,"total_time":6,"type":3}
```

### 3.20.7 执行 DIY(根据小场景 Id 执行)

#### 3.20.7.1 注意事项

★ 调用该方法必须是在面罩关闭的情况下调用,在打开面罩的情况下调用该方法将不执行相应指令。

#### 3.20.7.2 方法名

Method : public void startRobotDIY(String id,Context context)

#### 3.20.7.3 请求参数

参数名称	类型	约束	描述
Id	String	1	小场景 Id
context	Context	1	上下文

#### 3.20.7.4 调用示例

```
try {  
String
```



```
RobotManager.getInstance(context).getDiyControlInstance().startRobotDIY("2",context);  
    } catch (Exception e) {  
        e.printStackTrace();  
    }
```

## 3.20.8 停止执行 DIY

### 3.20.8.1 方法名

Method : stopDiy (Context mContext)

### 3.20.8.2 请求参数

参数名称	类型	约束	描述
mContext	Context	1	上下文

### 3.20.8.3 调用示例

```
try {  
  
RobotManager.getInstance(getApplicationContext()).getDiyControlInstance().stopDiy(getApplicationContext());  
    } catch (Exception e) {  
        e.printStackTrace();  
    }
```