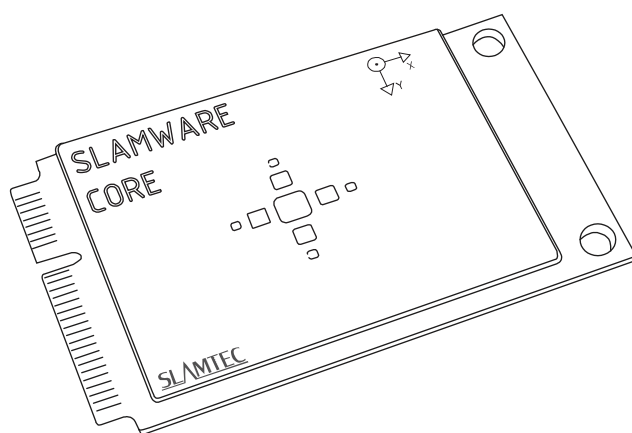


SLAMWARE

模块化自主定位导航解决方案

IOS SDK



目录.....	1
概览.....	3
类.....	4
RLEWRAPPER 类.....	4
RPABSTRACTDEVICE 类.....	4
RPABSTRACTDISCOVER 类.....	5
<RPACTIONPROTOCOL>协议.....	6
<RPBLECONFIGURELISTENER>协议.....	7
RPBLEDEVICE 类.....	7
RPBLEWIFIINFO 类.....	7
RPDEVICEMANAGER 类.....	8
RPDEVICEMANAGER(CONNECT)扩展类.....	9
<RPDISCOVERYDELEGATE>类.....	9
RPFIRMWAREUPDATEINFO 类.....	10
RPFIRMWAREUPDATEPROGRESS 类.....	10
RPHEALTHERROR 类.....	11
RPHEALTHINFO 类.....	12
RPLASERPOINT 类.....	12
RPLASERSCAN 类.....	13
RPLINE 类.....	14
RPLOCATION 类.....	15
RPMAP 类.....	15
<RPMOVEACTIONPROTOCOL>协议.....	17
RPPATH 类.....	17
RPPOINT 类.....	18
RPPOINTF 类.....	18
RPPOSE 类.....	19
RPRECTANGLE 类.....	21
RPRECTANGLEF 类.....	23
RPROTATION 类.....	24
RPSCHEDULETASK 类.....	25
RPSIZE 类.....	26
RPSIZEF 类.....	27
RPSLAMWAREPLATFORMPROTOCOL 协议.....	28
<RPSWEEPMOVEACTIONPROTOCOL>协议.....	37
修订历史.....	38

类列表

类名称	描述
RLEWrapper	类，表示封装。
RPAbstractDevice	类，表示设备。
RPAbstractDiscover	类，表示发现。
<RPActionProtocol>	表示 Action 协议定义。
<RPBleConfigureListener>	表示 Bluetooth configure listener 协议定义。
RPBleDevice	类，表示蓝牙设备。
RPBleWifiInfo	类，表示蓝牙 WiFi 信息。
RPDeviceManager	类，表示设备管理器。
RPDeviceManager[Connect]	类，表示设备管理器连接。
<RPDiscoveryDelegate>	Discovery delegate 协议定义。
RPFirmwareUpdateInfo	类，表示固件更新信息。
RPFirmwareUpdateProgress	类，表示固件更新进程。
RPHealthError	类，表示错误。
RPHealthInfo	类，表示错误信息。
RPLaserPoint	类，表示激光点。
RPLaserScan	类，表示激光扫描。
RPLine	类，表示线。
RPLocation	类，表示位置。
RPMap	类，表示地图。
<RPMoveActionProtocol>	MoveAction 协议定义。
RPPath	类，表示路径。
RPPoint	类，表示点。
RPPointF	类，表示点。
RPPose	类，表示姿态。
RPRectangle	类，表示区域。
RPRectangleF	类，表示区域。
RPRotation	类，表示旋转。
RPScheduleTask	类，表示定时任务。
RPSize	类，表示尺寸。
RPSizeF	类，表示尺寸。
<RPSlamwarePlatformProtocol>	协议，用于定义统一的接口和 SLAMWARE 设备通信。
<RPSweepMoveActionProtocol>	清扫动作协议定义。

RLEWrapper 类

方法

[+ encode](#)

[+ decode](#)

解释

[+ encode](#)

加密。

[+ decode](#)

解密。

RPAbstractDevice 类

类，表示设备。

方法

[-canBeFoundWith:](#)

属性

[int manufacturerId](#)

[int modelId](#)

[NSString *manufacturerName](#)

[NSString *modelName](#)

[NSString *hardwareVersion](#)

[NSString *softwareVersion](#)

[NSString *serialNumber](#)

[NSUUID *deviceId](#)

[NSString *deviceName](#)

解释

[-canBeFoundWith:](#)

是否由一种 `DiscoveryMode` 发现。参数是 `DiscoveryMode` 枚举类型。参见 `DiscoveryMode` 枚举。

`int manufacturerId`

表示生产商 ID 且数据类型为 int。

`int modelId`

表示型号 ID 且数据类型为 int。

`NSString *manufacturerName`

表示生产商名称且数据类型为 NSString。

`NSString *modelName`

表示型号名称且数据类型为 NSString。

`NSString *hardwareVersion`

表示硬件版本号且数据类型为 NSString。

`NSString *softwareVersion`

表示软件版本号且数据类型为 NSString。

`NSString *serialNumber`

表示序列号且数据类型为 NSString。

`NSUUID *deviceId`

表示设备 ID 且数据类型为 NSUUID。

`NSString *deviceName`

表示设备名称且数据类型为 NSString。

RPAbstractDiscover 类

表示 Abstract discover 接口。

方法

[-getStatus:](#)

[-start:](#)

[-stop:](#)

[-getMode](#)

解释

`-getStatus:`

获取状态。

`-start:`

开始。

`-stop:`

停止。

`-getMode`

获取 Discover 类型。类型参考 `DiscoveryMode`。

<RPAActionProtocol>协议

RPAAction 协议定义。

方法

[`-status`](#)

[`-progress`](#)

[`-cancel`](#)

[`-waitUntilDone`](#)

[`-actionName`](#)

解释

`-status`

获取动作状态。

`-progress`

获取动作进程，数据类型为 `double`。

`-cancel`

取消动作。

`-waitUntilDone`

等待动作完成。

`-actionName`

获取动作名称，数据类型为 `NSString`。

<RPBleConfigureListener>协议

RPBleConfigureListener 协议定义。

方法

[-onConfigureSuccess](#)

[-onConfigureFailure:](#)

解释

-onConfigureSuccess

配置成功。

- onConfigureFailure

配置失败。

RPBleDevice 类

类，继承自 RPAbstractDevice 类。表示蓝牙设备。

方法

[-canBeFoundWith:](#)

属性

[CBPeripheral *peripheral](#)

解释

-canBeFoundWith:

是否由一种 DiscoveryMode 发现。参数 DiscoveryMode。参考相关枚举。

[CBPeripheral *peripheral](#)

获取该蓝牙设备的 CBPeripheral 对象。

RPBleWifiInfo 类

类，表示 WiFi 信息。用于配置设备 WiFi。

属性

[NSString *ssid](#)

`NSString *pwd`

解释

`NSString *ssid`

属性，表示 WiFi 的 SSID。

`NSString *pwd`

属性，表示 WiFi 的密码。

RPDeviceManager 类

类，用于发现设备。继承自 RPAbstractDiscover 类。

方法

[-initWithDelegate:](#)

[-connect:withPort:](#)

[-connect:](#)

[-pair:withWifiInfo:withListenter:](#)

属性

[id< RPDiscoveryDelegate > delegate](#)

解释

`-initWithDelegate:`

初始化函数。参数为 RPDiscoveryDelegate。

`-connect:withPort:`

连接到指定的设备。参数为 NSString 类型的 ip 和 int 类型的 port。

`-connect:`

连接到设备。参数为 DiscoveryModeMDNS 类型的 RPAbstractDevice。

`-pair:withWifiInfo:withListenter:`

配置设备连接到指定的 WiFi。参数为 DiscoveryModeBLE 类型的 RPAbstractDevice。

[id< RPDiscoveryDelegate > delegate](#)

Delegate。

RPDeviceManager(Connect)扩展类

方法

[\(nonnull id< RPSlamwarePlatformProtocol >\) + connect:withPort:](#)

[\(nonnull id< RPSlamwarePlatformProtocol >\) + connect:](#)

解释

[\(nonnull id< RPSlamwarePlatformProtocol >\) + connect:withPort:](#)

静态方法。参数为 IP 地址和端口。IP 地址的数据类型为 NSString，端口的数据类型为 int。

[\(nonnull id< RPSlamwarePlatformProtocol >\) + connect:](#)

静态方法。参数是 DiscoveryModeMDNS 的 RPAbstractDevice 对象。

<RPDiscoveryDelegate>类

方法

[-onStartDiscovery:](#)

[-onStopDiscovery:](#)

[-onDiscoveryStatusChanged:withStatus:withError:](#)

[-onDeviceFound:withDevice:](#)

解释

[-onStartDiscovery:](#)

开始发现。参数是 RPAbstractDiscover，可以通过 canBeFoundWith:函数得到是哪种类型的发现协议。

[-onStopDiscovery:](#)

停止发现。参数是 RPAbstractDiscover，可以通过 canBeFoundWith:函数得到是哪种类型的发现协议。

[-onDiscoveryStatusChanged:withStatus:withError:](#)

发现状态改变。参数是 RPAbstractDiscover，DiscoverStatus 类型的状态，和 NSString 类型的 error。

[-onDeviceFound:withDevice:](#)

发现设备。参数是 RPAbstractDiscover 和 RPAbstractDevice。

RPFirmwareUpdateInfo 类

方法

[-init](#)

属性

[current](#)

[latest](#)

[releaseDate](#)

[brief](#)

解释

[-init](#)

初始化函数，数据类型为 NSString。

[current](#)

当前版本，数据类型为 NSString。

[latest](#)

最新版本，数据类型为 NSString。

[releaseDate](#)

发布日期，数据类型为 NSString。

[brief](#)

简要介绍，数据类型为 NSString。

RPFirmwareUpdateProgress 类

属性

[- init](#)

[currentStep](#)

[totalSteps](#)

[currentStepProgress](#)

[currentStepName](#)

解释

- `init`

初始化函数。

`currentStep`

当前步骤，数据类型为 `unsigned int`。

`totalSteps`

所有步骤，数据类型为 `unsigned int`。

`currentStepProgress`

当前步骤进程，数据类型为 `unsigned int`。

`currentStepName`

当前步骤名称，数据类型为 `NSString`。

RPHealthError 类

属性

`errorId`

`errorLevel`

`errorComponent`

`componentErrorCode`

`errorCode`

`errorMessage`

解释

`errorId`

错误消息 id，数据类型为 `int`。

`errorLevel`

错误等级，数据类型为 `BaseErrorLevel`。

`errorComponent`

发生错误的组件，数据类型为 `BaseErrorComponent`。

`componentErrorCode`

组件错误代码，数据类型为 `int`。

`errorCode`

错误代码，数据类型为 int。

`errorMessage`

错误信息，数据类型为 NSString。

RPHealthInfo 类

属性

[`hasWarning`](#)

[`hasError`](#)

[`hasFatal`](#)

[`errors`](#)

解释

`hasWarning`

表示是否有警告类错误。数据类型为布尔值。

`hasError`

表示是否有错误。数据类型为布尔值。

`hasFatal`

表示是否有致命错误。数据类型为布尔值。

`errors`

表示错误列表。数据类型为 NSArray<RPHealthError*>。

RPLaserPoint 类

类，表示激光点。

方法

[`-init`](#)

[`-initWithDistance:andAngle:`](#)

[`-initWithDistance:andAngle:andValid:`](#)

属性

[float distance](#)

[float angle](#)

[BOOL valid](#)

解释

[-init](#)

初始化函数。

[-initWithDistance:andAngle:](#)

初始化函数。参数为 float 类型的 Distance 和 float 类型的 Angle。

[-initWithDistance:andAngle:andValid:](#)

初始化函数。参数为 float 类型的 Distance , float 类型的 Angle 和 boolean 类型的 Valid。

[float distance](#)

获取 distance。

[float angle](#)

获取 angle。

[BOOL valid](#)

激光点是否有效。

RPLaserScan 类

类，表示激光扫描。

方法

[-initWithLaserPoints:](#)

[-initWithLaserPoints:andPose:](#)

属性

[NSArray<RPLaserPoint*>* laserPoints](#)

[RPPose* pose](#)

解释

`-initWithLaserPoints:`

初始化函数。参数为 NSArray 类型的 RPLaserPoint。

`-initWithLaserPoints:andPose:`

初始化函数。参数为 NSArray<RPLaserPoint>和 RPPose。

`NSArray<RPLaserPoint*>* laserPoints`

获取激光点。

`RPPose* pose`

获取姿态。

RPLine 类

类，表示线。

方法

`-initWithStartPoint:andEndPoint:`

`-initWithStartPoint:andEndPoint:andLineId:`

属性

`RPPointF* startPoint`

`RPPointF* endPoint`

`int lineId`

解释

`-initWithStartPoint:andEndPoint:`

初始化函数。参数为 RPPointF 类型的 StartPoint 和 RPPointF 类型的 EndPoint。

`-initWithStartPoint:andEndPoint:andLineId:`

初始化函数。参数为 RPPointF 类型的 StartPoint , RPPointF 类型的 EndPoint 和 int 类型的 lineId。

`RPPointF* startPoint`

获取起点。

`RPointF* endPoint`

获取终点。

`int lineId`

获取 line id。

RPLocation 类

类，表示位置。

方法

[-init](#)

[-initWithX:andY:andZ:](#)

属性

[float x](#)

[float y](#)

[float z](#)

解释

`-init`

初始化函数。

`-initWithX:andY:andZ:`

初始化函数且以 x , y , z 为参数。

`float x`

获取 x 。

`float y`

获取 y 。

`float z`

获取 z 。

RPMap 类

类，表示地图。

方法

[-initWithOrigin:andDimension:andResolution:andTimestamp:andData:](#)

[-initWithOrigin:andDimension:andResolution:andData:](#)

[-getMapArea](#)

属性

[RPointF* origin](#)

[RSize* dimension](#)

[RPointF* resolution](#)

[long timestamp](#)

[NSData* data](#)

解释

[-initWithOrigin:andDimension:andResolution:andTimestamp:andData:](#)

初始化函数。参数为 RPointF 类型的 origin , RSize 类型的 dimension , RPointF 类型的 resolution , long 类型的 timestamp 和 NSData 类型的 data。

[-initWithOrigin:andDimension:andResolution:andData:](#)

初始化函数。参数为 RPointF 类型的 origin , RSize 类型的 dimension , RPointF 类型的 resolution 和 NSData 类型的 data。

[-getMapArea](#)

获取地图区域。

[RPointF* origin](#)

获取原点。

[RSize* dimension](#)

获取尺寸。

[RPointF* resolution](#)

获取分辨率。

[long timestamp](#)

获取时间戳。

NSData* data

获取数据。

<RPMoveActionProtocol>协议

方法

[-remainingPath](#)

[-remainingMilestones](#)

继承自[<RPAActionProtocol>](#)的方法。

解释

-remainingPath

获取剩余路径。

-remainingMilestones

获取剩余里程碑。

RPPath 类

类，表示路径。

方法

[- init](#)

[-initWithPoints:](#)

属性

[NSArray<RPLocation*>* points](#)

解释

- init

初始化函数。

- initWithPoints:

初始化函数。参数为 points，参数类型为 NSArray<RPLocation>。

[NSArray<RPLocation*>* points](#)

获取点。

RPoint 类

类，表示点。

方法

[-init](#)

[-initWithX: andY:](#)

属性

[int x](#)

[int y](#)

解释

`-init`

初始化函数。

`-initWithX:andY:`

初始化函数且以 `x` 和 `y` 为参数。

`int x`

获取 `x`。

`int y`

获取 `y`。

RPointF 类

类，表示点。

方法

[-init](#)

[-initWithX:andY:](#)

属性

[float x](#)

[float y](#)

解释

`-init`

初始化函数。

`-initWithX:andY:`

初始化函数且以 `x` 和 `y` 为参数。

`float x`

获取 `x`。

`float y`

获取 `y`。

RPPose 类

类，表示姿态。

方法

`-init`

`-initWithLocation:`

`-initWithRotation:`

`-initWithLocation:andRotation:`

`-initWithX:andY:andZ:andYaw:andPitch:andRoll:`

`-x`

`-setX:`

`-y`

`-setY:`

`-z`

`-setZ:`

`-yaw`

`-setYaw:`

`-pitch`

`-setPitch:`

-roll

-setRoll:

属性

RPLocation* location

RPRotation* rotation

解释

- init

初始化函数。

-initWithLocation:

初始化函数且以 location 为参数。

-initWithRotation:

初始化函数且以 rotation 为参数。

-initWithLocation:andRotation:

初始化函数且以 location 和 rotation 为参数。

-initWithX:andY:andZ:andYaw:andPitch:andRoll:

初始化函数且以 x , y , z , yaw , pitch , roll 为参数。

-x

获取 x。

-setX:

设置 X。

-y

获取 y。

-setY:

设置 Y。

-z

获取 z。

-setZ:

设置 Z。

-yaw

获取 yaw。

-setYaw:

设置 yaw。

-pitch

获取 pitch。

-setPitch:

设置 pitch。

-roll

获取 roll。

-setRoll:

设置 roll。

RPLocation* location

获取 location。

RPRotation* rotation

获取 rotation。

RRectangle 类

类，表示 rectangle。

方法

-init

-initWithOrigin:andSize:

-left

-top

-right

-bottom

-empty

-unionOf:

-intersectionOf:

-area

属性

RPoint* origin

RSize* size

解释

-init

初始化函数。

-initWithOrigin:andSize:

初始化函数。参数为 RPoint 类型的 origin 和 RSize 类型的 size。

-left

左。

-top

上。

-right

右。

-bottom

下。

-empty

Size 是否为空。

-unionOf:

取两个 rectangle 的合集。参数为 RRectangle 类型的 dest。调用之后会改变被调用的对象。

-intersectionOf:

取两个 Rectangle 的交集。参数为 RRectangle 类型的 dest。调用之后会改变被调用的对象。

-area

获取区域且参数为 int 类型的 area。

`RPoint* origin`

获取原点。

`RSize* size`

获取 size。

RRectangleF 类

类，表示 rectangle。

方法

[-init](#)

[-initWithOrigin:andSize:](#)

[-left](#)

[-top](#)

[-right](#)

[-bottom](#)

[-empty](#)

[-unionOf:](#)

[-intersectionOf:](#)

[-area](#)

属性

[RPointF* origin](#)

[RPointF* size](#)

解释

`-init`

初始化函数。

`- initWithOrigin:andSize:`

初始化函数且以 RPointF 类型的原点和 RPointF 类型的 size 为参数。

`-left`

左。

`-top`

上。

`-right`

右。

`-bottom`

下。

`-empty`

Size 是否为空。

`-unionOf:`

取两个 rectangle 的合集。参数为 `RRectangleF` 类型的 `dest`。调用之后都会改变被调用的对象。

`-intersectionOf:`

取两个 Rectangle 的交集。参数为 `RRectangleF` 类型的 `dest`。调用之后都会改变被调用的对象。

`-area`

获取区域。

`RPointF* origin`

获取原点。

`RSizeF* size`

获取 size。

RRotation 类

类，表示旋转。

方法

`-init`

`-initWithYaw:`

`-initWithYaw:andPitch:andRoll:`

属性

[float yaw](#)

[float pitch](#)

[float roll](#)

解释

`-init`

初始化函数。

`-initWithYaw:`

初始化函数且以 yaw 为参数。

`-initWithYaw:andPitch:andRoll:`

初始化函数且以 yaw , pitch 和 roll 为参数。

`float yaw`

获取 yaw。

`float pitch`

获取 pitch。

`float roll`

获取 roll。

RPScheduleTask 类

属性

[int id](#)

[int hour](#)

[int minute](#)

[int year](#)

[int month](#)

[int day](#)

[int maxDuration](#)

[BOOL enabled](#)

[int weekRepeat](#)

[NSString task](#)

解释

`int id`

定时任务 ID，数据类型为 int。

`int hour`

小时，数据类型为 int。

`int minute`

分钟，数据类型为 int。

`int year`

年，数据类型为 int。

`int month`

月，数据类型为 int。

`int day`

日，数据类型为 int。

`int maxDuration`

任务最长执行时间，数据类型为 int。

`BOOL enabled`

是否启用定时任务，数据类型为布尔值。

`int weekRepeat`

重复。不重复为 0，周日为 1，周一为 2，周二为 4，周三为 8，周四为 16，周五为 32，周六为 64，每天为 127。

[NSString task](#)

定时任务名字，不能为空。数据类型为 NSString。

RPSize 类

类，表示 size。

方法

[-init](#)

[-initWithWidth:andHeight:](#)

属性

[int width](#)

[int height](#)

解释

`-init`

初始化函数。

`-initWithWidth:andHeight:`

初始化函数且以 `int` 类型的宽度和 `int` 类型的高度为参数。

`int width`

获取宽度。

`int height`

获取高度。

RPSizeF 类

类，表示 size。

方法

[-init](#)

[-initWithWidth:andHeight:](#)

属性

[float width](#)

[float height](#)

解释

`-init`

初始化函数。

`-initWithWidth:andHeight:`

初始化函数且以 `float` 类型的宽度和 `float` 类型的高度为参数。

float width

获取宽度。

float height

获取高度。

RPSlamwarePlatformProtocol 协议

方法

[-disconnect](#)

[- deviceId](#)

[- manufacturerId](#)

[- manufacturerName](#)

[- modelId](#)

[- modelName](#)

[- hardwareVersion](#)

[- softwareVersion](#)

[-availableMaps](#)

[-getMapWithMapType:inArea:ofMapKind:](#)

[-setMapWithMap:ofMapType:andMapKind:](#)

[-getKnownAreaOfMapType:andMapKind:](#)

[-clearMap](#)

[-location](#)

[-pose](#)

[-setPose:](#)

[-mapLocalization](#)

[-setMapLocalization:](#)

[-mapUpdate](#)

[-setMapUpdate:](#)

[-localizationQuality](#)

-moveToLocations:andAppendingToCurrentTask:andSetAsMilestones:
-moveToLocation:andAppendingToCurrentTask:andSetAsMilestones:
-moveBy:
-rotateToOrientation:
-rotateBy:
-currentAction
-searchPathToLocation:
-batteryPercentage
-batteryIsCharging
-dcIsConnected
-slamwareVersion
-sdkVersion
-laserScan
-walls
-addWall:
-addWalls:
-clearWallById:
-clearWalls
-startSweep
-sweepSpot:
-goHome
-restartModuleWithMode:
-setSystemParameterNamed:withValue:
-valueOfSystemParameterNamed:
- getFirmwareUpdateInfo
- startFirmwareUpdate
- getFirmwareUpdateProgress

- [getScheduledTasks](#)
- [addScheduledTask:](#)
- [getScheduledTaskWithId:](#)
- [updateScheduledTask:](#)
- [deleteScheduledTaskWithId:](#)
- [getRobotHealth](#)
- [clearRobotHealth](#)

解释

-disconnect

断开连接。

- deviceId

获取设备 id。返回值为设备 id，数据类型为 NSUUID。

- manufacturerId

获取设备制造商 id。返回数据类型为 integer。

- manufacturerName

制造商名称。返回数据类型为 NSString。

- modelId

获取设备型号 id。返回的数据类型为 integer。

- modelName

获取设备型号名称。返回值为标明型号名称的 NSString。

- hardwareVersion

获取硬件版本信息。返回值为标明硬件版本信息的 NSString。

- softwareVersion

获取软件版本信息。返回值为标明软件版本信息的 NSString。

-availableMaps

获取 Slamware 中可用的地图类型。返回值为地图类型列表，注意请将类型列表成员转换为 RMap 类型枚举。

-getMapWithMapType:inArea:ofMapKind:

获取地图且以地图数据类型，地图区域及种类为参数。返回值为部分地图对象。

参数

type：地图的数据类型

rect：地图所要求的区域

kind：地图的种类

-setMapWithMap:ofMapType:andMapKind:

上传地图数据到 SLAMWARE。（注意：应与 setPose 配套使用并确保地图未处于更新或定位状态。）

参数

Map：地图。

Type：地图的数据类型。

Kind：地图的种类。

-getKnownAreaOfMapType:andMapKind:

获取地图的已知区域。返回值为已探索的地图上的区域。

参数

Type：地图的类型。

Kind：地图的种类。

-clearMap

清除当前地图。

-location

在地图坐标系统中获取机器人的位置。返回值为机器人的位置。

-pose

获取机器人的姿态（包含位置信息和角度信息）。返回值为机器人的姿态。

-setPose:

设置机器人的姿态。

参数

pose：机器人的新姿态

-mapLocalization

获取机器人是否在进行定位。返回值为布尔值，表明机器人是否在进行定位。

-setMapLocalization:

启用或停用定位功能。

参数

V：布尔值，用于决定 SLAMWARE 是否进行定位功能。

-mapUpdate

获取 SLAMWARE 是否更新地图。返回值为布尔值，表明 SLAMWARE 是否在更新地图。

-setMapUpdate:

启用或停用地图更新。

参数

V：布尔值，表明 SLAMWARE 是否应该更新地图。

-localizationQuality

获取定位信息可信度。

-moveToLocations:andAppendingToCurrentTask:andSetAsMilestones:

使机器人移动到一系列的指定位置。返回值为执行该项操作的移动动作。

参数

Locs：机器人前往指定位置经过的一系列节点。

Appending：布尔值，用于决定 SLAMWARE 是清除当前任务建立新的点还是将新的点添加到已有的节点列表中。

isMilestone：- 布尔值，用于决定 SLAMWARE 是规划路径到一系列节点还是直接前往。当这个参数为 true 时，机器人会将上述点视作关键点，通过路径搜索的方式前往目的地；当参数为 false 时，会被视作普通点，不会启用路径搜索功能。

- moveToLocation:andAppendingToCurrentTask:andSetAsMilestones:

使机器人移动到指定点，返回值为执行该操作的动作。

参数

Loc：将要移动到的点。

Appending：布尔值，用于决定 SLAMWARE 是清除当前任务建立新的点还是将新的点添加到已有的节点列表中。

isMilestone：布尔值，用于决定 SLAMWARE 是规划路径到一系列节点还是直接前往。当这个参数为 true 时，机器人会将上述点视作关键点，通过路径搜索的方式前往目的地；当参数为 false 时，会被视作普通点，不会启用路径搜索功能

- moveBy:

人工控制机器人的移动。返回值为执行该项操作的移动动作。

（注意：在此状态下，机器人不会进行避障）。需要反复调用该函数来保持机器人的移动状态并调用 MoveAction.cancel() 函数来及时使机器人停止移动，否则机器人会在持续调用 moveBy 函数一段时期后停止移动。

参数

Direction：要求机器人移动的方向，参考 RPMoveDirection 枚举。

-rotateToOrientation:

使机器人旋转到指定的姿态。返回值为执行该操作的动作。

参数

Orientation：要求的姿态。

-rotateBy:

使机器人旋转到指定的角度（有差异的）。返回值为执行该操作的动作。

参数

Offset：机器人要求被旋转的弧度。

-currentAction

获取机器人当前的动作。

-searchPathToLocation:

在地图中搜寻路径从机器人当前位置到指定位置。返回值为从机器人当前位置到指定位置的路径。

参数

Location：指定位置。

-batteryPercentage

获取电池的剩余电量（从 0~100）。返回值为电池剩余电量的百分比，数据类型为 int。

-batteryIsCharging

获取电池是否在充电。返回值为一个 boolean 值，表明电池是否在充电。

-dcIsConnected

获取机器人是否连接到了电源插座。返回值为一个 boolean 值，表明机器人是否连接到充电器。

-slamwareVersion

获取 SLAMWARE 版本。返回值为标明 SLAMWARE 版本的 NSString。

-sdkVersion

获取 SLAMWARE SDK 的版本。返回值为标明 SLAMWARE SDK 版本的 NSString。

-laserScan

获取最近的激光扫描。返回值为最近的激光扫描。

-walls

获取已经存在的虚拟墙。返回值为已存在的虚拟墙列表。

-addWall:

添加虚拟墙到 Slamware。

参数

Wall：准备添加的虚拟墙。

-addWalls:

添加多个虚拟墙到 Slamware。

参数

Walls：准备添加的多个虚拟墙。

-clearWallById:

清除指定的虚拟墙。

参数

wallId：准备清除的虚拟墙 id。

-clearWalls

从 SLAMWARE 清除所有的虚拟墙。

-startSweep

使机器人开始清扫（注意：此方法仅适用于扫地机版本机器人）。返回值为执行该操作的清扫动作。

-sweepSpot:

使机器人进行定点清扫（注意：此方法仅适用于扫地机版本机器人）。返回值为执行该操作的清扫动作。

-goHome

使机器人回到充电座（注意：该方法仅适用于支持自动返回功能版本的机器人）。返回值为执行该操作的动作。

-restartModuleWithMode:

重启 SLAMWARE 模块。

参数

Mode：重启 SLAMWRE 模块的模式。

-setSystemParameterNamed:withValue:

设置系统参数。

参数

Name：准备设置的系统参数名称。

Value：准备为系统参数设置的值。

-valueOfSystemParameterNamed:

获取系统参数。返回值为参数的当前值。

参数

Name：准备获取的参数名称。

- getFirmwareUpdateInfo

获取固件更新信息。返回值为固件更新信息。

- startFirmwareUpdate

开始固件更新。

- `getFirmwareUpdateProgress`

获取固件更新进程。返回值为固件更新进程。

- `getScheduledTasks`

获取预约任务信息。返回值为标明预约任务的数组。

- `addScheduledTask:`

添加预约任务。返回值的数据类型为布尔值。YES 表示添加成功，NO 表示添加失败。

参数

task：将要添加的预约任务。

- `getScheduledTaskWithId:`

获取指定的预约任务信息。

参数

Id：预约任务的 id

- `updateScheduledTask:`

更新预约任务。返回值为更新的预约任务。

参数

task：添加的预约任务

- `deleteScheduledTaskWithId:`

删除预约任务。返回值的数据类型为布尔值。YES 表示删除成功，NO 表示删除失败。

参数

Id：准备删除的预约任务 id

- `getRobotHealth`

获取机器人健康状况。返回值为 RPHealthInfo

- `clearRobotHealth`

清除错误信息。

参数

errorCode：RPHealthError 类中的 errorCode

<RPSweepMoveActionProtocol>协议

方法

[-getAvailableSweepMaps](#)

[-getSweepMap:withArea:](#)

[-getSweepMapArea:](#)

解释

[-getAvailableSweepMaps](#)

获取可用的清扫地图。

[-getSweepMap:withArea:](#)

获取清扫地图。

[-getSweepMapArea:](#)

获取清扫地图区域。

日期	版本	描述
2016-05-12	0.1	初始版本
2016-06-07	1.8	更新封面图片
2016-11-04	1.8	添加最新部分类定义