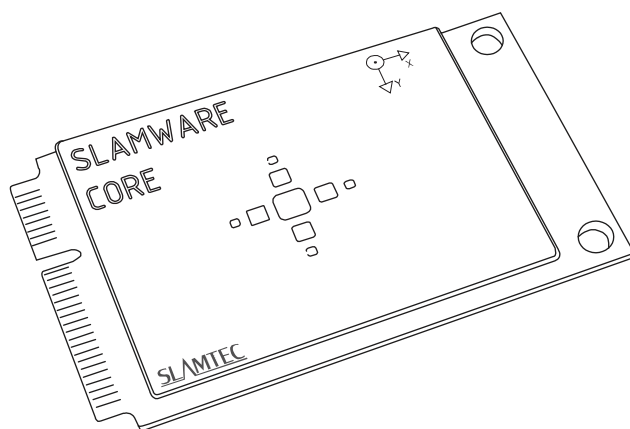


SLAMWARE

Modular Autonomous Robot Localization and Navigation Solution

IOS SDK



CONTENTS	1
OVERVIEW	3
CLASS	4
RLEWRAPPER CLASS REFERENCE.....	4
RPABSTRACTDEVICE CLASS REFERENCE	4
RPABSTRACTDISCOVER CLASS REFERENCE	5
<RPACTIONPROTOCOL> PROTOCOL REFERENCE	6
<RPBLECONFIGURELISTENER> PROTOCOL REFERENCE	7
RPBLEDEVICE CLASS REFERENCE	7
RPBLEWIFIINFO CLASS REFERENCE	7
RPDEVICEMANAGER CLASS REFERENCE.....	8
RPDEVICEMANAGER(CONNECT) CATEGORY REFERENCE	9
<RPDISCOVERYDELEGATE> CLASS REFERENCE.....	9
RPFIRMWAREUPDATEINFO CLASS REFERENCE.....	10
RPFIRMWAREUPDATEPROGRESS CLASS REFERENCE	10
RPHEALTHERROR CLASS REFERENCE.....	11
RPHEALTHINFO CLASS REFERENCE	12
RPLASERPOINT CLASS REFERENCE	12
RPLASERSCAN CLASS REFERENCE	13
RPLINE CLASS REFERENCE	14
RPLOCATION CLASS REFERENCE.....	15
RPMAP CLASS REFERENCE	16
<RPMOVEACTIONPROTOCOL> PROTOCOL REFERENCE.....	17
RPPATH CLASS REFERENCE.....	17
RPPOINT CLASS REFERENCE	18
RPPOINTF CLASS REFERENCE	18
RPPOSE CLASS REFERENCE.....	19
RPRECTANGLE CLASS REFERENCE.....	21
RPRECTANGLEF CLASS REFERENCE.....	23
RPROTATION CLASS REFERENCE	25
RPSCHEDULETASK CLASS REFERENCE	25
RPSIZE CLASS REFERENCE.....	27
RPSIZEF CLASS REFERENCE	27
RPSLAMWAREPLATFORMPROTOCOL REFERENCE.....	28
<RPSWEEPMOVEACTIONPROTOCOL> PROTOCOL REFERENCE.....	37
REVISION HISTORY	39

Class List

Class Name	Description
<u>RLEWrapper</u>	Class, representing wrapper
<u>RPAbstractDevice</u>	Class, representing device
<u>RPAbstractDiscover</u>	Class, representing discovery
<u><RPActionProtocol></u>	Action protocol definition
<u><RPBleConfigureListener></u>	Bluetooth configure listener protocol definition
<u>RPBleDevice</u>	Class, representing Bluetooth device
<u>RPBleWifiInfo</u>	Class, representing the Bluetooth WiFi information
<u>RPDeviceManager</u>	Class, representing device manager
<u>RPDeviceManager[Connect]</u>	Category, representing
<u><RPDiscoveryDelegate></u>	Discovery delegate protocol definition
<u>RPFirmwareUpdateInfo</u>	Class, representing the update information of firmware
<u>RPFirmwareUpdateProgress</u>	Class, representing the update progress of firmware.
<u>RPHealthError</u>	Class, representing error
<u>RPHealthInfo</u>	Class, representing health information
<u>RPLaserPoint</u>	Class, representing laser point
<u>RPLaserScan</u>	Class, representing laser scan
<u>RPLine</u>	Class, representing line
<u>RPLocation</u>	Class, represent location
<u>RPMap</u>	Class, representing map
<u><RPMoveActionProtocol></u>	MoveAction protocol definition
<u>RPPath</u>	Class, representing path
<u>RPPoint</u>	Class, representing point
<u>RPPointF</u>	Class, representing Point
<u>RPPose</u>	Class, representing pose
<u>RPRectangle</u>	Class, representing rectangle
<u>RPRectangleF</u>	Class, representing rectangle
<u>RPRotation</u>	Class, representing rotation
<u>RPScheduleTask</u>	Class, representing schedule task
<u>RPSize</u>	Class, representing Size
<u>RPSizeF</u>	Class, representing Size
<u>RPSlamwarePlatformProtocol</u>	Protocol, defining unified interfaces to interact with SLAMWARE devices
<u><RPSweepMoveActionProtocol></u>	SweepMoveAction protocol definition

RLEWrapper Class Reference

Methods

[+ encode](#)

[+ decode](#)

Details

[+ encode](#)

Encode.

[+ decode](#)

Decode.

RPAbstractDevice Class Reference

Class, represent a device.

Instance Methods

[-canBeFoundWith:](#)

Properties

[int manufacturerId](#)

[int modelId](#)

[NSString *manufacturerName](#)

[NSString *modelName](#)

[int hardwareVersion](#)

[int softwareVersion](#)

[NSString *serialNumber](#)

[NSUUID *deviceId](#)

[NSString *deviceName](#)

Details

[-canBeFoundWith:](#)

Whether the device can be found with discovery mode on.

`int manufacturerId`

Represent the manufacturer id and the data type is int.

`int modelId`

Represent the model id and the data type is int.

`NSString *manufacturerName`

Represent the manufacturer name and the data type is NSString.

`NSString *modelName`

Represent the model name and the data type is NSString.

`int hardwareVersion`

Represent the hardware version and the data type is int.

`int softwareVersion`

Represent the software version and the data type is int.

`NSString *serialNumber`

Represent the serial number and the data type is NSString.

`NSUUID *deviceId`

Represent the device id and the data type is NSUUID.

`NSString *deviceName`

Represent the device name and the data type is NSString.

RPAbstractDiscover Class Reference

Abstract discover interface.

Instance Methods

[-getStatus:](#)

[-start:](#)

[-stop:](#)

[-getMode](#)

Details

-getStatus:

Get status.

-start:

Start.

-stop:

Stop.

-getMode

Get mode.

<RPActionProtocol> Protocol Reference

RPAction protocol definition.

Instance Methods

-status

-progress

-cancel

-waitUntilDone

-actionName

Details

-status

Get the status of the action.

-progress

Get the progress of the action.

-cancel

Cancel the action.

-waitUntilDone

Wait the action to be done.

-actionName

Get the action name.

<RPBleConfigureListener> Protocol Reference

RPBleConfigureListener protocol definition.

Instance Methods

[-onConfigureSuccess](#)

[-onConfigureFailure:](#)

Details

[-onConfigureSuccess](#)

Configuration success.

[-onConfigureFailure:](#)

Configuration failure.

RPBleDevice Class Reference

Class. Represent Bluetooth device.

Instance Methods

[-canBeFoundWith:](#)

Properties

[CBPeripheral *peripheral](#)

Properties inherited from [RPAbstractDevice](#)

Details

[-canBeFoundWith:](#)

Whether the device can be found with discovery mode.

[CBPeripheral *peripheral](#)

Get the Bluetooth device.

RPBleWifiInfo Class Reference

Class, represent the WiFi information.

Instance Methods

[NSString *ssid](#)

[NSString *pwd](#)

Details

[NSString *ssid](#)

Get the ssid.

[NSString *pwd](#)

Get the password.

RPDeviceManager Class Reference

Class, represent the manager to manage the devices.

Instance Methods

[-initWithDelegate:](#)

[-connect:withPort:](#)

[-connect:](#)

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

Instance Methods inherited from [RPAbstractDiscover](#)

Properties

[id< RPDiscoveryDelegate > delegate](#)

Details

[-initWithDelegate:](#)

Initialization function, and the parameter is RPDiscoverDelegate.

[-connect:withPort:](#)

Connect with specified device. And parameters are ip in NSString type and port in int type.

[-connect:](#)

Connect with device. And the parameter is RPAbstractDevice.

`-pair:withWifiInfo:withListener:`

Pair device and connect to specified WiFi. And parameters are RPAbstractDevice, RPBleWifiInfo, RPBleConfigureListener.

`id< RPDDiscoveryDelegate > delegate`

Delegate.

RPDeviceManager(connect) Category Reference

Method

`(nonnull id< RPSlamwarePlatformProtocol >) + connect:withPort:`
`(nonnull id< RPSlamwarePlatformProtocol >) + connect:`

Details

`(nonnull id< RPSlamwarePlatformProtocol >) + connect:withPort:`

Static method. Parameters are IP address and portal. The data type of IP address is NSString; the data type of portal is int.

`(nonnull id< RPSlamwarePlatformProtocol >) + connect:`

Static method. Parameter is the RPAbstractDevice object of DiscoveryModeMDNS.

<RPDiscoveryDelegate> Class Reference

Instance Methods

`-onStartDiscovery:`

`-onStopDiscovery:`

`-onDiscoveryStatusChanged:withStatus:withError:`

`-onDeviceFound:withDevice:`

Details

`-onStartDiscovery:`

Start discovery.

`-onStopDiscovery:`

Stop discovery.

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

Discovery status changed.

[-onDeviceFound:withDevice:](#)

Found device.

RPFirmwareUpdateInfo Class Reference

Method

[-init](#)

Properties

[current](#)

[latest](#)

[releaseDate](#)

[brief](#)

Details

[-init](#)

Initialization function. The data type is NSString.

[current](#)

Current version. The data type is NSString.

[latest](#)

The latest version. The data type is NSString.

[releaseDate](#)

The release data. The data type is NSString.

[brief](#)

The brief introduction. The data type is NSString.

RPFirmwareUpdateProgress Class Reference

Properties

[- init](#)

[currentStep](#)

[totalSteps](#)[currentStepProgress](#)[currentStepName](#)

Details

- `init`

Initialization function.

`currentStep`

Current step. The data type is unsigned int.

`totalSteps`

All the steps. The data type is unsigned int.

`currentStepProgress`

The progress of current step. The data type is unsigned int.

`currentStepName`

The name of current step. The data type is NSString.

RPHealthError Class Reference

Properties

[errorId](#)

[errorLevel](#)

[errorComponent](#)

[componentErrorCode](#)

[errorCode](#)

[errorMessage](#)

Details

`errorId`

The ID of error message. And the data type is int.

`errorLevel`

Error level. And the data type is BaseErrorLevel.

[errorComponent](#)

Component which has error. And the data type is BaseErrorComponent.

[componentErrorCode](#)

The component error code. And the data type is int.

[errorCode](#)

Error code. And the data type is int.

[errorMessage](#)

Error message. And the data type is NSString.

RPHealthInfo Class Reference

Properties

[hasWarning](#)

[hasError](#)

[hasFatal](#)

[errors](#)

Details

[hasWarning](#)

Whether there is warning. The data type is Boolean.

[hasError](#)

Whether there is error. The data type is Boolean.

[hasFatal](#)

Whether there is fatal error. The data type is Boolean.

[errors](#)

The error list. The data type is NSArray<RPHealthError*>.

RPLaserPoint Class Reference

Class, represent laser point.

Instance Methods

[-init](#)

[-initWithDistance:andAngle:](#)

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

Properties

[float distance](#)

[float angle](#)

[BOOL valid](#)

Details

[-init](#)

Initialization function.

[-initWithDistance:andAngle:](#)

Initialization function. And parameters are Distance in float type and Angle in float type.

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

Initialization function. And parameters are Distance in float type, Angle in float type and Valid in Boolean type.

[float distance](#)

Get the distance.

[float angle](#)

Get the angle.

[BOOL valid](#)

Whether the laser point is valid.

RPLaserScan Class Reference

Class, represent laser scan.

Instance Methods

[-initWithLaserPoints:](#)

[-initWithLaserPoints:andPose:](#)

Properties

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

[RPPose* pose](#)

Details

[-initWithLaserPoints:](#)

Initialization function. And parameter is RPLaserPoint in NSArray type.

[-initWithLaserPoints:andPose:](#)

Initialization function. And parameters are NSArray<RPLaserPoint>and RPPose.

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

Get laser points.

[RPPose* pose](#)

Get pose.

RPLine Class Reference

Class, represent line.

Instance Methods

[-initWithStartPoint: andEndPoint:](#)

[-initWithStartPoint:andEndPoint:andLineId:](#)

Properties

[RPPointF* startPoint](#)

[RPPointF* endPoint](#)

[int lineId](#)

Details

[-initWithStartPoint:andEndPoint:](#)

Initialization function. And parameters are StartPoint in RPPointF type and EndPoint in RPPointF type.

-initWithStartPoint:andEndPoint:andLineId:

Initialization function. And parameters are StartPoint in RPPointF type, EndPoint in RPPointF type and LineId in int type.

RPPointF* startPoint

Get start point.

RPPointF* endPoint

Get end point.

int lineId

Get line id.

RPLocation Class Reference

Class, represent location.

Instance Methods

-init

-initWithX:andY:andZ:

Properties

float x

float y

float z

Details

-init

Initialization function.

-initWithX:andY:andZ:

Initialization function with x, y and z as parameters.

float x

Get x.

float y

Get y.

float z

Get z.

RPMap Class Reference

Class, represent map.

Instance Methods

-

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

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

[-getMapArea](#)

Properties

[RPPointF* origin](#)

[RPSize* dimension](#)

[RPPointF* resolution](#)

[long timestamp](#)

[NSData* data](#)

Details

-

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

Initialization function. And parameters are Origin in RPPointF type, Dimension in RPSize type, Resolution in RPPointF type, Timestamp in long type and Data in NSData type.

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

Initialization function. And parameters are Origin in RPPointF type, Dimension in RPSize type, Resolution in RPPointF type and Data in NSData type.

[-getMapArea](#)

Get map area.

[RPPointF* origin](#)

Get origin.

`RPSize* dimension`

Get dimension.

`RPPointF* resolution`

Get resolution.

`long timestamp`

Get timestamp.

`NSData* data`

Get data.

<RPMoveActionProtocol> Protocol Reference

Instance Methods

[`-\(RPPath*\) remainingPath`](#)

[`-\(RPPath*\) remainingMilestones`](#)

Instances Methods inherited from [`<RPActionProtocol>`](#)

Details

[`-\(RPPath*\) remainingPath`](#)

Get the remaining path.

[`-\(RPPath*\) remainingMilestones`](#)

Get the remaining milestones.

RPPath Class Reference

Class, represent a path.

Instance Methods

[`-\(init\)`](#)

[`-\(initWithPoints:\)`](#)

Properties

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

Details

`-init`

Initialization function.

`-initWithPoints:`

Initialization function. And parameter is points in NSArray<RPLocation>.

`NSArray<RPLocation*>* points`

Get points.

RPPoint Class Reference

Class, represent point.

Instance Methods

`-init`

`-initWithX:andY:`

Properties

`int x`

`int y`

Details

`-init`

Initialization function.

`-initWithX:andY:`

Initialization function with x and y as parameters.

`int x`

Get x with int as parameters.

`int y`

Get y with int as parameters.

RPPointF Class Reference

Class, represent PointF.

Instance Methods

[-init](#)

[-initWithX:andY:](#)

Properties

[float x](#)

[float y](#)

Details

[-init](#)

Initialization function.

[-initWithX:andY:](#)

Initialization function with x and y as parameters.

[float x](#)

Get x with float as parameters.

[float y](#)

Get y with float as parameters.

RPPose Class Reference

Class, represent pose.

Instance Methods

[-init](#)

[-initWithLocation:](#)

[-initWithRotation:](#)

[-initWithLocation:andRotation:](#)

[-initWithX:andY:andZ:andYaw:andPitch:andRoll:](#)

[-x](#)

[-setX:](#)

[-y](#)

-setY:

-z

-setZ:

-yaw

-setYaw:

-pitch

-setPitch:

-roll

-setRoll:

Properties

RPLocation* location

RPRotation* rotation

Details

-init

Initialization function.

-initWithLocation:

Initialization function with location as parameters.

-initWithRotation:

Initialization function with rotation as parameters.

-initWithLocation:andRotation:

Initialization function with location and rotation as parameters.

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

Initialization function with x, y, z, yaw, pitch, roll as parameters.

-x

Get x.

-setX:

Set X.

-y

Get y.

-setY:

Set Y.

-z

Get z.

-setZ:

Set Z.

-yaw

Get yaw.

-setYaw:

Set yaw.

-pitch

Get pitch.

-setPitch:

Set pitch.

-roll

Get roll.

-setRoll:

Set roll.

RPLocation* location

Get location.

RPRotation* rotation

Get rotation.

RPRectangle Class Reference

Class, represent the RPRectangle.

Instance Methods

-init

-initWithOrigin:andSize:

-left

-top

-right

-bottom

-empty

-unionOf:

-intersectionOf:

-area

Properties

RPoint* origin

RSize* size

Details

-init

Initialization function.

-initWithOrigin:andSize:

Initialization function. And parameters are Origin in RPoint type and Size in RSize type.

-left

Left.

-top

Top.

-right

Right.

-bottom

Bottom.

-empty

Whether the size is empty.

[-unionOf:](#)

Get the union of two rectangles. And parameter is dest in RPRectangle type. The invoked object will be changed after calling this function.

[-intersectionOf:](#)

Get the intersection of two rectangles. And parameter is dest in RPRectangle type. The invoked object will be changed after calling this function.

[-area](#)

Get area. And parameter is area in int type.

[RPPoint* origin](#)

Get origin.

[RPSize* size](#)

Get size.

RPRectangleF Class Reference

Class, represent the RPRectangleF.

Instance Methods

[-init](#)

[-initWithOrigin:andSize:](#)

[-left](#)

[-top](#)

[-right](#)

[-bottom](#)

[-empty](#)

[-unionOf:](#)

[-intersectionOf:](#)

[-area](#)

Properties

[RPPointF* origin](#)

RPSizeF* size

Details

-init

Initialization function.

-initWithOrigin:andSize:

Initialization function. And parameter are Origin in RPPointF type and Size in RPPointF type.

-left

Left.

-top

Top.

-right

Right.

-bottom

Bottom.

-empty

Whether the size is empty.

-unionOf:

Get the union of two rectangles. And parameter is dest in RPRectangleF type. The invoked object will be changed after calling this function.

-intersectionOf:

Get the intersection of two rectangles. And parameter is dest in RPRectangleF type. The invoked object will be changed after calling this function.

-area

Get area.

RPPointF* origin

Get origin.

RPSizeF* size

Get size.

RPRotation Class Reference

Class, represent rotation.

Instance Methods

[-init](#)

[-initWithYaw:](#)

[-initWithYaw:andPitch:andRoll:](#)

Properties

[float yaw](#)

[float pitch](#)

[float roll](#)

Details

[-init](#)

Initialization function.

[-initWithYaw:](#)

Initialization function with yaw as parameters.

[-initWithYaw:andPitch:andRoll:](#)

Initialization function with yaw, pitch and roll as parameters.

[float yaw](#)

Get yaw.

[float pitch](#)

Get pitch.

[float roll](#)

Get roll.

RPScheduleTask Class Reference

Properties

[int id](#)

int hour

int minute

int year

int month

int day

int maxDuration

BOOL enabled

int weekRepeat

NSString task

Details

int id

The ID of schedule task. And the data type is int.

int hour

Hour. And the data type is int.

int minute

Minute. And the data type is int.

int year

Year. And the data type is int.

int month

Month. And the data type is int.

int day

Day. And the data type is int.

int maxDuration

The time maximum of task execution. And the data type is int.

BOOL enabled

Enable the schedule task or not. And the data type is Boolean.

int weekRepeat

Repeat the schedule task.

0 for not repeating the schedule task; 1 for repeating the schedule task on Sunday; 2 for Monday; 4 for Tuesday; 8 for Wednesday; 16 for Thursday; 32 for Friday; 64 for Saturday; 127 for each day.

NSString task

The name of the schedule task. The data type is NSString and cannot be null value.

RPSize Class Reference

Class, represent size.

Instance Methods

[-init](#)

[-initWithWidth:andHeight:](#)

Properties

[int width](#)

[int height](#)

Details

-init

Initialization function.

-initWithWidth:andHeight:

Initialization function. And parameters are width and height in int type.

int width

Get width with int as parameters.

int height

Get height with int as parameters.

RPSizeF Class Reference

Class, represent size.

Instance Methods

[-init](#)

[-initWithWidth:andHeight:](#)

Properties

[float width](#)

[float height](#)

Details

-init

Initialization function.

-initWithWidth:andHeight:

Initialization function. And parameters are width and height in float type.

float width

Get width with float as parameters.

float height

Get height with float as parameters.

RPSlamwarePlatformProtocol Reference

Instance Methods

[-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

Details

-disconnect

Disconnect.

- deviceId

Get the device ID. The return value is ID and the data type is NSUUID.

- manufacturerId

Get the device manufacturer ID. The return value is ID and the data type is integer.

- manufacturerName

Get the manufacturer name. The data type is NSString.

- modelId

Get the device model ID. The data type of the return value is integer.

- modelName

Get the name of the device model. The data type of the return value is NSString.

- hardwareVersion

Get the hardware version information. The data type of the return value is NSString.

- softwareVersion

Get the software version information. The data type of the return value is NSString.

-availableMaps

Get available maps.

-getMapWithMapType:inArea:ofMapKind:

Get map with map type and kind as parameters.

-setMapWithMap:ofMapType:andMapKind:

Upload map data to the Slamware. (Notice: should be used with setPose, and with map update and localization stopped)

Parameters

map: the map object.

type: the data type of the map.

kind: the kind of the map.

-getKnownAreaOfMapType:andMapKind:

Get the known area of the map. The return value is the explored area of the map.

Parameters

type: the data type of the map.

kind: the kind of the map.

-clearMap

Clear current map.

-location

Get the position of robot in the map coordinate system. The return value is the location of the robot.

-pose

Get the pose of the robot (including location and rotation). The return value is the pose of the robot.

-setPose:

Set the pose of the robot.

Parameters

_pose: the new pose of the robot.

-mapLocalization

Get if the Slamware is doing localization. The return value is a boolean to indicate if the Slamware is doing localization.

-setMapLocalization:

Enable or disable localization.

Parameters

v: a boolean to indicate if the Slamware should do localization.

-mapUpdate

Get if the Slamware is updating map. The return value is a boolean to indicate if the Slamware is updating map.

-setMapUpdate:

Enable or disable map update.

Parameters

v: a boolean to indicate if the Slamware should update map.

-localizationQuality

Represent whether the localization information is valid.

-moveToLocations:andAppendingToCurrentTask:andSetAsMilestones:

Make robot move to a series of points. The return value is the move action to manipulate this operation.

Parameters

locs: the points to visit.

appending: a boolean to indicate if Slamware should clear current tasks or append these point to the visit list.

isMilestone A boolean to indicate if Slamware should plan a route to the points or go directly to the point.

-moveToLocation:andAppendingToCurrentTask:appending andSetAsMilestones:

Make robot move to a specific point. The return value is the move action to manipulate this operation.

Parameters

loc: the point to visit

appending: a boolean to indicate if Slamware should clear current tasks or append these point to the visit list

isMilestone A boolean to indicate if Slamware should plan a route to the points or go directly to the point.

-moveBy:

Manual control robot's movement (notice: this action will not do any obstacle avoidance). You have to invoke this API repeat to keep the robot move, and call MoveAction.cancel() to stop the movement in time, or the robot will stop after a period of last moveBy call.

The return value is the move action to manipulate this operation.

Parameters

direction: which type of movement you want the robot do.

-rotateToOrientation:

Make robot rotate a specific pose. The return value is the move action to manipulate this operation.

Parameters

orientation: required pose.

-rotateBy:

Make robot rotate a specific angle (differential). The return value is the move action to manipulate this operation.

Parameters

offset: the rad the robot required to rotate.

-currentAction

Get robot current action.

-searchPathToLocation:

Search a path in the map from robot's current position to the required location. The return value is a path from robot's current location to the target location.

Parameters

location: the target location.

-batteryPercentage

Get the left percentage of the battery (from 0 ~ 100). The return value is the battery percentage.

-batteryIsCharging

Get if the battery is charging. The return value is a boolean to indicate if the battery is charging.

-dcIsConnected

Get if the robot is _connected with an outlet. The return value is a boolean to indicate if the robot is _connected to the charger.

-slamwareVersion

Get the version of Slamware. The return value is the version string of the Slamware.

-sdkVersion

Get the version of Slamware SDK. The return value is the version string of the Slamware SDK.

-laserScan

Get the most recent LASER scan. The return value is the most recent LASER scan.

-walls

Get existing virtual walls. The return value is a list of existing virtual walls.

-addWall:

Add a virtual wall to Slamware.

Parameters

wall: the virtual wall to add

-addWalls:

Add a set of virtual walls to Slamware.

Parameters

walls: virtual walls to add.

-clearWallById:

Remove specific virtual wall.

Parameters

wallId: the id to the virtual wall to remove.

-clearWalls

Remove all virtual walls from Slamware.

-startSweep

Make robot to start sweep (Notice: This method is only available on Slamware Core Vacuum Robot Edition). The return value is the sweep move action to manipulate this operation.

-sweepSpot:

Make robot to sweep a particular area (Notice: This method is only available on Slamware Core Vacuum Robot Edition). The return value is the sweep move action to manipulate this operation.

-goHome

Make robot go back to the charging base (Notice: This method is only available on robots which support auto home feature). The return value is the move action to manipulate this operation.

-restartModuleWithMode:

Restart the Slamware module

Parameters

mode: the mode to restart Slamware module

-setSystemParameterNamed:withValue:

Set system parameter.

Parameters

name: The parameter to set.

value: The value you want to set.

-valueOfSystemParameterNamed:

Get system parameter. And the return value is the current value of the parameter.

Parameters

Name: the parameter to get.

- getFirmwareUpdateInfo

Get the firmware update information. The return value is the firmware update information.

- startFirmwareUpdate

Start the firmware update.

- getFirmwareUpdateProgress

Get the firmware update progress. The return value is firmware update progress.

- getScheduledTasks

Get the information of scheduled tasks. The return value data type is an array.

- addScheduledTask:

Add scheduled task or not. The return value data type is Boolean. YES for adding successfully; NO for failing to add the scheduled task.

Parameters:

task: task to be added in scheduled task list.

- `getScheduledTaskWithId:`

Get the information of a specified scheduled task.

Parameters:

id: the ID of the schedule task

- `updateScheduledTask:`

Update scheduled task. The return value is the updated schedule task.

Parameters:

task: task to be added in scheduled task.

- `deleteScheduledTaskWithId:`

Delete schedule task. The data type of the return value is Boolean. YES for deleting successfully; NO for failing to delete the task.

Parameters:

Id: The ID of a task which is going to be deleted.

- `getRobotHealth`

Get the health status of the robot. The return value is RPHealthInfo.

- `clearRobotHealth`

Clear the error message.

Parameters:

errorCode in errorCode : RPHealthError class.

<RPSweepMoveActionProtocol> Protocol Reference

Instance Methods

[-getAvailableSweepMaps](#)

[-getSweepMap:withArea:](#)

[-getSweepMapArea:](#)

Details

-getAvailableSweepMaps

Get available sweep maps.

-getSweepMap:withArea:

Get sweep map.

-getSweepMapArea:

Get sweep map area.

Date	Version	Description
2016-05-12	0.1	Initial version
2016-06-07	1.8	Added the SLAMWARE core image in the cover
2016-11-04	1.8	Added the definition of latest classes