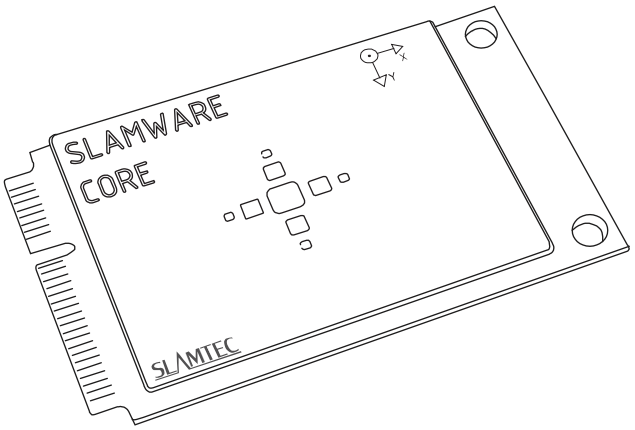
SLAMTEC

SLAMWARE

模块化自主定位导航解决方案

Android SDK



2016-09-30 rev.1.8

www.slamtec.com

上海思岚科技有限公司

包

包名	描述
com.slamtec.slamware	
com.slamtec.slamware.action	
com.slamtec.slamware.discovery	
com.slamtec.slamware.FirmwareUpdate	
com.slamtec.slamware.geometry	
com.slamtec.slamware.robot	

类

类名	描述
AbstractDiscover	类, 表示 abstract discover interface。
AbstractDiscover.BleConfigureListener	类, 表示 BleConfigureListener。
AbstractDiscover.DiscoverStatus	类，表示 DiscoverStatus。
AbstractDiscover.DiscoveryListener	类，表示 DiscoveryListener。
AbstractSlamwarePlatform	类，定义统一的接口用于和 SLAMWARE 设备通讯。
ActionStatus	枚举，列举动作状态。
BleDevice	类，表示 BleDevice。
Device	类，表示设备。
DeviceManager	类，表示管理设备的管理程序。
DiscoveryMode	枚举，列举发现机器人的模式。
FirmwareUpdateInfo	类，表示固件更新信息。
FirmwareUpdateProgress	类，表示固件更新进程。
HealthInfo	类，表示健康状况。
HealthInfo.BaseError	类，表示底盘健康状况和错误信息。
IAction	接口，表示机器人动作。
IMoveAction	接口，表示移动动作。
ISweepMoveAction	接口，表示清扫动作。
LaserPoint	类，表示激光扫描点。
LaserScan	类，表示一次激光扫描。
Line	类，表示线。
Location	类，表示机器人在 3D 空间中的位置。
Map(Robot)	类，表示机器人所在的位置地图。
Map(SDP)	类，表示 SDP 所在的位置地图。
MapKind	枚举，列举地图种类。

MapType	枚举，列举地图数据类型。
MdnsDevice	类，MdnsDevice。
MoveDirection	类，列举人工控制机器人时的方向指令。
NetworkMode	类，表示网络模式。
Path	类，表示路径。
PointF	类，表示 2D 浮点小数数据类型。
Pose	类，表示机器人姿态。
RestartMode	枚举，列举重启模式。
Rotation	类，表示旋转。
ScheduledTask	类，表示预约任务
Size	类，表示整数型的 Size 类别。
SlamwareCorePlatform	类，Abstract Slamware Platform 类的子类。
SlamwareSdpPlatform	类，表示 SDP 扩展平台。
SystemParameters	类，表示系统参数。

AbstractSlamwarePlatform 类

Abstract Slamware Platform 类，定义统一的接口用于和 SLAMWARE 设备通讯。

已知子类：SlamwareCorePlatform

构造器

AbstractSlamwarePlatform()

方法

addScheduledTask(ScheduledTask task)
addWall(Line wall)
addWalls(java.util.List<Line> walls)
clearMap()
clearWallById(int id)
clearWalls()
configureNetwork(int mode, java.util.HashMap<java.lang.String,java.lang.String> options)
deleteScheduledTask(int taskId)
disconnect()
getAvailableMaps()
getBatteryIsCharging()
getBatteryPercentage()
getCurrentAction()
getDCIsConnected()
getDeviceId()
getFirmwareUpdateInfo()
getFirmwareUpdateProgress()
getHardwareVersion()
getKnownArea(MapType type)
getKnownArea(MapType type, MapKind kind)
getLaserScan()
getLocalizationQuality()
getLocation()
getManufacturerId()
getManuFacturerName()
getMap(MapType type, MapKind kind, android.graphics.RectF area)
getMap(MapType type, android.graphics.RectF area)
getMapLocalization()
getMapUpdate()
getModelId()
getModelName()
getNetworkStatus()
getPose()
getRobotHealth()
getScheduledTask(int taskId)

getScheduledTasks()
getSDKVersion()
getSlamwareVersion()
getSoftwareVersion()
getSystemParameter(java.lang.String param)
getWalls()
goHome()
moveBy(MoveDirection direction)
moveTo(java.util.List<Location> locations)
moveTo(java.util.List<Location> locations, boolean appending)
moveTo(java.util.List<Location> locations, boolean appending, boolean isMilestone)
moveTo(Location location)
moveTo(Location location, boolean appending)
moveTo(Location location, boolean appending, boolean isMilestone)
restartModule()
restartModule(RestartMode mode)
rotate(Rotation rotation)
rotateTo(Rotation orientation)
searchPath(Location location)
setMap(Map map)
setMap(Map map, MapType type)
setMap(Map map, MapType type, MapKind kind)
setMapLocalization(boolean v)
setMapUpdate(boolean v)
setPose(Pose pose)
setSystemParameter(java.lang.String param, java.lang.String value)
startFirmwareUpdate()
startSweep()
sweepSpot(Location location)
updateScheduledTask(ScheduledTask task)

解释

AbstractSlamwarePlatform()

AbstractSlamwarePlatform 是一个抽象类，不能直接构造该类的对象。请使用 DeviceManager 连接设备并获取 AbstractSlamwarePlatform 的对象。

addScheduledTask(ScheduledTask task)

添加预约扫地任务。返回数据类型为布尔值 boolean。

参数：

task-计划任务

`addWall(Line wall)`

添加虚拟墙倒 SLAMWARE。

参数：wall – 需要添加的虚拟墙。

`addWalls(java.util.List<Line> walls)`

添加多个虚拟墙倒 SLAMWARE。

参数：walls – 需要添加的多个虚拟墙。

`clearMap()`

清除当前地图。

`clearWallById(int id)`

移除指定的虚拟墙。

参数：id – 需要移除的虚拟墙的 id。

`clearWalls()`

从 Slamware 清除所有虚拟墙。

```
configureNetwork(int mode,
java.util.HashMap<java.lang.String,java.lang.String> options)
```

配置 Slamware Core 的网络工作模式。数据返回类型布尔值 boolean。

参数

mode – 网络模式

options – 选项

说明

当前支持的工作模式主要有：

- AP 模式（Slamware Core 本身作为一个 WiFi 热点，当用户设备通过 Wifi 或者有线网络连接该 WiFi 热点时，会通过 DHCP 获得一个 IP 地址，而后通过 192.168.11.1 来访问设备，这个模式也是 Slamware Core 出厂的预置模式）

- Station 模式（Slamware Core 本身最为一个 WiFi 设备，连接到其他的 WiFi 热点上。同时 Slamware Core 会自动成为无线网桥，为 High Speed Bus 上的设备分配 IP 地址并提供外网访问服务）
- Disabled 模式（Slamware Core 关闭自身的无线访问功能，只能通过有限网络访问，其 IP 地址、网关和 DNS 服务器均由 API 调用的参数决定）

例子

- 将 Slamware Core 配置成 AP 模式：

```
Platform.configureNetwork(NetworkMode.NetworkModeAp, new HashMap<String, String>());
```

- 让 Slamware Core 连上名字为 Slamtec 的 AP：

```
HashMap<String, String> options = new HashMap<String, String>();
options.put(“ssid”, “Slamtec”);
options.put(“password”, “Password”);
platform.configureNetwork(NetworkMode.NetworkModeStation, options);
```

- 将 Slamware Core 的 IP 地址配置为 192.168.12.13，默认网关为 192.168.12.1，DNS 服务器为：114.114.114.114：

```
HashMap<String, String> options = new HashMap<String, String>();
options.put(“ip”, “192.168.12.13”);
options.put(“mask”: “255.255.255.0”);
options.put(“gateway”: “192.168.12.1”);
options.put(“dns”: “114.114.114.114”);
platform.configureNetwork(NetworkMode.NetworkModeDisabled, options);
```

deleteScheduledTask(int taskId)

是否删除预约的清扫任务。数据返回类型为布尔值 boolean。

参数：

taskId – 任务 id

disconnect()

断开与平台之间的连接。

getAvailableMaps()

获取 SLAMWARE 中可用的地图类型。返回值为地图类型列表。

`getBatteryIsCharging()`

获取电池是否在充电。返回值为布尔值，表明电池是否在充电。

`getBatteryPercentage()`

获取电池剩余电量百分比（从 0~100）。返回值为电池电量百分比。

`getCurrentAction()`

获取机器人当前的动作。返回值为机器人当前正在进行的移动动作。

`getDCIsConnected()`

获取机器人是否连接到电源插座。返回值为布尔值，表明机器人是否连接到充电器。

`getDeviceId()`

获取设备的 UUID。返回值数据类型为 string。

`getFirmwareUpdateInfo()`

请参考 FirmwareUpdateInfo 类

`getFirmwareUpdateProgress()`

请参考 FirmwareUpdateProgress 类

`getHardwareVersion()`

获取设备硬件版本。返回值数据类型为 string。

`getKnownArea(MapType type)`

获取已探索到的地图区域。返回值为已探索的地图区域。

参数：type – 地图数据类型

`getKnownArea(MapType type, MapKind kind)`

获取已探索到的地图区域。返回值为已探索的地图区域。

参数：

type – 地图数据类型

kind – 地图种类

`getLaserScan()`

获取最新的 LASER 扫描。返回值最新的 LASER 扫描。

`getLocalizationQuality()`

获取定位可信度。

`getLocation()`

获取机器人在上述地图坐标系统中的坐标。返回值为机器人位置信息。

`getManufacturerId()`

获取设备制造商 id。返回值数据类型为 int。

`getManuFacturerName()`

获取设备制造商名称。返回值数据类型为 string。

`getMap(MapType type, MapKind kind, android.graphics.RectF area)`

从 SLAMWARE 获取地图数据。返回值为局部地图对象。

参数：

type – 地图数据类型

kind – 地图种类

area – 地图区域

`getMap(MapType type, android.graphics.RectF area)`

从 SLAMWARE 获取地图数据。返回值为局部地图对象。

参数：

type –地图数据类型

area – 地图区域

`getMapLocalization()`

获取 SLAMWARE 是否在进行定位。返回值为布尔值表明 SLAMWARE 是否在进行定位。

`getMapUpdate()`

获取 SLAMWARE 是否更新地图。返回值为布尔值表明 SLAMWARE 是否更新地图。

`getModelId()`

获取设备型号 id。返回数据类型为 int。

`getModelName()`

获取设备型号名称。返回数据类型为 string。

`getNetworkStatus()`

获取网络状况。返回数据类型为 string。

`getPose()`

获取机器人在上述坐标系统中的姿态（包含位置信息和角度信息）。返回值为机器人的姿态。

`getRobotHealth()`

获取机器人健康状态。返回数据为健康状况。

`getScheduledTask(int taskId)`

获取预约任务。

参数：taskId – 任务 id。

`getScheduledTasks()`

获取预约的任务。返回值为预约的任务列表。

`getSDKVersion()`

获取 SLAMWARE SDK 版本。返回值为标明 SLAMWARESDK 版本的字符串。

`getSlamwareVersion()`

获取 SLAMWARE 版本。返回值为标明 SLAMWARE 版本的字符串。

`getSoftwareVersion()`

获取设备的软件版本。返回值数据类型为字符串。

`getSystemParameter(java.lang.String param)`

获取系统参数。返回值为当前参数的值。

参数：:param – 获取的参数。

`getWalls()`

获取当前存在的虚拟墙。返回值为当前存在的虚拟墙列表。

`goHome()`

使机器人返回充电桩（注意：该方法仅适用于支持自动回充功能的机器人版本）。

`moveBy(MoveDirection direction)`

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 –期望机器人移动的方向

`moveTo(java.util.List<Location> locations)`

使机器人移动到一系列指定位置。返回值为执行该项操作的移动动作。

参数：locations – 机器人前往指定位置经过的一系列节点。

`moveTo(java.util.List<Location> locations, boolean appending)`

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：

locations - 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。

`moveTo(java.util.List<Location> locations, boolean appending, boolean isMilestone)`

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：

locations - 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。

`moveTo(Location location)`

使机器人移动到指定地点。返回值为执行该项操作的移动动作。

参数：location – 机器人将要到达的点。

`moveTo(Location location, boolean appending)`

使机器人移动到指定地点。返回值为执行该项操作的移动动作。

参数：

location - 机器人将要到达的点。

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。

`moveTo(Location location, boolean appending, boolean isMilestone)`

使机器人移动到指定地点。返回值为执行该项操作的移动动作。

参数：

location - 机器人将要到达的点。

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 是规划路径到该节点还是直接前往。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。

`restartModule()`

重启 SLAMWARE 模块。

`restartModule(RestartMode mode)`

重启 SLAMWARE 模块。

参数：mode – 重启 SLAMWARE 模块的模式。

`rotate(Rotation rotation)`

使机器人旋转一个角度（不同的角度）。返回值为执行该项操作的移动动作。

参数：rotation – 机器人被要求旋转的弧度。

`rotateTo(Rotation orientation)`

使机器人旋转到特定的朝向。返回值为执行该项操作的移动动作。

参数：orientation – 要求的姿态

`searchPath(Location location)`

在地图中寻找使机器人从当前位置移动到指定目标位置的路径。返回值为一条从机器人的当前位置到指定目标位置的路径。

参数：location – 目标位置

`setMap(Map map)`

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

`setMap(Map map, MapType type)`

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 – 地图数据类型

`setMap(Map map, MapType type, MapKind kind)`

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 – 地图数据类型

kind – 地图种类

`setMapLocalization(boolean v)`

获取是否启用定位功能。

参数：v – 布尔值，表明 SLAMWARE 是否该进行定位。

`setMapUpdate(boolean v)`

获取是否更新地图。

参数：v – 布尔值，表明 SLAMWARE 是否应该更新地图。

`setPose(Pose pose)`

设置机器人的姿态。

参数：pose – 机器人的新姿态

`setSystemParameter(java.lang.String param, java.lang.String value)`

设置系统参数。

参数：

param – 将要进行设置的参数。

value – 将要设置的值。

`startFirmwareUpdate()`

获取是否开始固件更新。返回值为 boolean 表示是否开始更新固件。

`startSweep()`

使机器人开始清扫（注意：该方法仅适用于扫地机版本的 SLAMWARE Core）。返回值为执行该项操作的清扫移动动作。

`sweepSpot(Location location)`

使机器人开始定点清扫（注意：该方法仅适用于扫地机版本的 SLAMWARE Core）。返回值为执行该项操作的清扫移动动作。

`updateScheduledTask(ScheduledTask task)`

更新预约任务列表。

参数：task – 将被更新的预约任务

IAction 接口

已知子接口：IMoveAction, ISweepMoveAction

接口，表示机器人的动作。提供接口来执行该动作。

方法

cancel()
getActionName()
getProgress()
getStatus()
waitUntilDone()

解释

cancel()

取消本次操作。

getActionName()

获取动作名称。返回值为动作名称（string）。

getProgress()

获取动作进程（0~1）。返回值为动作进程（double）。

getStatus()

获取动作状态。返回值为动作状态。

waitUntilDone()

等待动作完成。返回值为完成的动作结果。

IMoveAction IAction 接口

继承自 IAction

已知超接口：IAction

已知子接口：ISweepMoveAction

接口：表示移动动作。

方法

getRemainingMilestones()
getRemainingPath()

解释

getRemainingMilestones()

获取剩余里程碑。返回值为剩余的里程碑。

getRemainingPath()

获取到下一里程碑的剩余路径。返回值为到下一里程碑的剩余路径。

继承自 com.slamtec.slamware.action.IAction 接口的方法如下：

cancel, getActionName, getProgress, getStatus, waitUntilDone

ISweepMoveAction IAction 接口

继承自 IMoveAction

已知超接口：IAction， IMoveAction

接口，表示清扫时的移动动作。

方法

继承自 com.slamtec.slamware.action.IMoveAction 接口的方法如下：

getRemainingMilestones, getRemainingPath

继承自 com.slamtec.slamware.action.IAction 接口的方法如下：

cancel, getActionName, getProgress, getStatus, waitUntilDone

path 类

类，表示一条路径。

构造器

Path()
Path(Path path)
Path(java.util.Vector<Location> points)

方法

getPoints()
setPoints(java.util.Vector<Location> points)

解释

Path()

创建对象 Path。

Path(Path path)

创建参数为 path 的对象 Path。

Path(java.util.Vector<Location> points)

创建参数为 points 的对象 Path。

getPoints()

获取点。

setPoints(java.util.Vector<Location> points)

设置点。

ActionStatus 枚举

public enum ActionStatus, 继承自 java.lang.Enum<ActionStatus>, 表示动作的状态。

枚举常量

WAITING_FOR_START
RUNNING
FINISHED
PAUSED
STOPPED
ERROR

解释

WAITING_FOR_START

动作已创建但未开始。

RUNNING

动作正在进行。

FINISHED

动作成功完成。

PAUSED

动作已暂停。

STOPPED

动作已停止。

ERROR

动作遇到错误。

MoveDirection 枚举

public enum MoveDirection, 继承自 java.lang.Enum<MoveDirection>, 人工控制机器人时要求机器人移动的方向。

枚举常量

FORWARD
BACKWARD
TURN_RIGHT
TURN_LEFT

解释

FORWARD

当前移动动作为向前。

BACKWARD

当前移动动作为向后。

TURN_RIGHT

当前移动动作为向右。

TURN_LEFT

当前移动动作为向左。

AbstractDiscover.BleConfigureListener 接口

封闭类：AbstractDiscover

方法

onConfigureSuccess()
onConfigureFailure(java.lang.String error)

解释

onConfigureSuccess()

配置成功。

onConfigureFailure(java.lang.String error)

配置失败。

AbstractDiscover 类

public abstract class AbstractDiscover, abstract discover 接口。

已知子类：DeviceManager

嵌套类

AbstractDiscover.BleConfigureListener
AbstractDiscover.DiscoverStatus
AbstractDiscover.DiscoveryListener

构造器

AbstractDiscover()

方法

getMode()
setListener(AbstractDiscover.DiscoveryListener listener)
getStatus(DiscoveryMode mode)
start(DiscoveryMode mode)
stop(DiscoveryMode mode)

解释

AbstractDiscover()

创建对象 AbstractDiscover。

`getMode()`

获取模式。

`setListener(AbstractDiscover.DiscoveryListener listener)`

设置 listener。

`getStatus(DiscoveryMode mode)`

获取状态。

`start(DiscoveryMode mode)`

开始。

`stop(DiscoveryMode mode)`

结束。

AbstractDiscover.DiscoveryListener 类

封闭类：`AbstractDiscover`

构造器

`DiscoveryListener()`

方法

`onStartDiscovery(AbstractDiscover discover)`
`onStopDiscovery(AbstractDiscover discover)`
`onDiscoveryError(AbstractDiscover discover, java.lang.String error)`
`onDeviceFound(AbstractDiscover discover, Device device)`

解释

`DiscoveryListener()`

创建对象 DiscoveryListener。

`onStartDiscovery(AbstractDiscover discover)`

Discovery 扫描开始的回调函数。

`onStopDiscovery(AbstractDiscover discover)`

Discovey 扫描停止的回调函数。

`onDiscoveryError(AbstractDiscover discover, java.lang.String error)`

Discovery 错误的回调函数。

`onDeviceFound(AbstractDiscover discover, Device device)`

发现设备的回调函数。

BleDevice 类

继承自 Device

构造器

`BleDevice(BluetoothDevice device)`

方法

`BluetoothDevice getDevice()`
`canBeFoundWith(DiscoveryMode mode)`

Methods inherited from class com.slamtec.slamware.discovery.Device: getDeviceId, getDeviceName, getHardwareVersion, getManufactureId, getManufactureName, getModelId, getModelName, getSerialNumber, getSoftwareVersion, setDeviceId, setDeviceName, setHardwareVersion, setManufactureId, setManufactureName, setModelId, setModelName, setSerialNumber, setSoftwareVersion

解释

`BleDevice(BluetoothDevice device)`

创建对象 BleDevice。

`getDevice()`

获取设备 device。

`canBeFoundWith(DiscoveryMode mode)`

在 discovery 模式开启的状态下能否发现设备。

由 Device 类中的 canBeFoundWith 指定

Device 类

public abstract class Device, 表示一台设备。

已知子类：

BleDevice, MdnsDevice

构造器

Device()

方法

getManufactureId()
setManufactureId(int manufactureId)
getModelId()
setModelId(int modelId)
getManufactureName()
setManufactureName(java.lang.String manufactureName)
getModelName()
setModelName(java.lang.String modelName)
getHardwareVersion()
setHardwareVersion(int hardwareVersion)
getSoftwareVersion()
setSoftwareVersion(int softwareVersion)
getSerialNumber()
setSerialNumber(java.lang.String serialNumber)
canBeFoundWith(DiscoveryMode mode)
getDeviceId()
setDeviceId(java.util.UUID deviceId)
getDeviceName()
setDeviceName(java.lang.String deviceName)

解释

Device()

创建对象 Device。

getManufactureId()

获取 Manufacture id。

setManufactureId(int manufactureId)

设置 manufacture id。

`getModelId()`

获取 mode id。

`setModelId(int modelId)`

设置 mode id。

`getManufactureName()`

获取 manufacture name。

`setManufactureName(java.lang.String manufactureName)`

设置 manufacture name。

`getModelName()`

获取 mode name。

`setModelName(java.lang.String modelName)`

设置 mode name。

`getHardwareVersion()`

获取 hard ware version。

`setHardwareVersion(int hardwareVersion)`

设置 hardware version。

`getSoftwareVersion()`

获取 software version。

`setSoftwareVersion(int softwareVersion)`

设置 software version。

`getSerialNumber()`

获取 serial number。

`setSerialNumber(java.lang.String serialNumber)`

设置 serial number。

`canBeFoundWith(DiscoveryMode mode)`

在 discovery 模式开启的状态下能否发现设备。

`getDeviceId()`

获取 device id。

`setDeviceId(java.util.UUID deviceId)`

设置 device id。

`getDeviceName()`

获取 device name。

`setDeviceName(java.lang.String deviceName)`

设置 device name。

DeviceManager 类

继承自 `AbstractDiscover`

The manager to manage devices。

Nested classes/interfaces inherited from class

com.slamtec.slamware.discovery.AbstractDiscover:

AbstractDiscover.BleConfigureListener,

AbstractDiscover.DiscoverStatus,

AbstractDiscover.DiscoveryListener

构造器

`DeviceManager(Context context)`

方法

`connect(java.lang.String host, int port)`
`connect(Device device)`
`pair(Device device, java.lang.String wifiSSID, java.lang.String wifiPassword, AbstractDiscover.BleConfigureListener listener)`
`setListener(AbstractDiscover.DiscoveryListener listener)`
`getStatus(DiscoveryMode mode)`
`start(DiscoveryMode mode)`
`stop(DiscoveryMode mode)`
`DiscoveryMode getMode()`

解释

`DeviceManager(Context context)`

创建对象 DeivceManager。

`connect(java.lang.String host, int port)`

直接连接到 SLAMWARE Core（该方法常用于 Android 设备通过高速总线直接连接到 SLAMWARE Core）。返回值为连接到的平台。

参数：

host – 设备主机（通常是 192.168.11.1）

port – 端口

`connect(Device device)`

连接到指定的基于 SLAMWARE 的设备。返回值为连接到的设备。

参数：

device – 准备连接的设备

`pair(Device device,java.lang.String wifiSSID, java.lang.String wifiPassword, AbstractDiscover.BleConfigureListener listener)`

通过 SSID 和密码匹配 SLAMWARE 设备。

参数：

device – 准备配对的设备

wifiSSID – WiFi 的 SSID

wifiPassword – WiFi 的密码

listener – 配置监听器

`setListener(AbstractDiscover.DiscoveryListener listener)`

由 AbstractDiscover 类中的 setListener 指定

`getStatus(DiscoveryMode mode)`

由 AbstractDiscover 类中的 getStatus 指定。

`start(DiscoveryMode mode)`

由 AbstractDiscover 类中的 start 指定

`stop(DiscoveryMode mode)`

由 AbstractDiscover 类中的 stop 指定

`getMode()`

由 AbstractDiscover 类中的 getMode 指定

MdnsDevice 类

继承自 `Device`

构造器

`MdnsDevice(java.lang.String addr, int port)`

方法

`getAddr()`
`getPort()`
`canBeFoundWith(DiscoveryMode mode)`

解释

`MdnsDevice(java.lang.String addr, int port)`

创建对象 MdnsDevice。

`getAddr()`

获取地址。

`getPort()`

获取端口。

`canBeFoundWith(DiscoveryMode mode)`

由 Device 类中的 canBeFoundWith 指定

AbstractDiscover.DiscoverStatus 枚举

封闭类

AbstractDiscover

枚举常量

STOPPED
WORKING
ERROR

解释

STOPPED

停止。

WORKING

工作。

ERROR

错误。

DiscoveryMode 枚举

表明机器人如何被发现。

枚举常量

BLE
MDNS

解释

BLE

BLE 模式。

MDNS

FirmwareUpdateInfo 类

public class FirmwareUpdateInfo 表示固件更新信息。

构造器

FirmwareUpdateInfo(java.lang.String current, java.lang.String latest, java.lang.String releaseDate, java.lang.String brief)

方法

getBrief()
getCurrent()
getLatest()
getReleaseDate()

解释

FirmwareUpdateInfo(java.lang.String current, java.lang.String latest, java.lang.String releaseDate, java.lang.String brief)

创建对象 FirmwareUpdateInfo 且参数为 current/latest/releasedDate/brief。

getBrief()

获取简明信息, 数据类型为字符串 string。

getCurrent()

获取当前固件版本, 数据类型为字符串 string。

getLatest()

获取最新固件版本, 数据类型为字符串 string。

getReleaseDate()

获取固件发布日期, 数据类型为字符串 string。

FirmwareUpdateProgress 类

public class FirmwareUpdateProgress 表示固件更新进程。

构造器

FirmwareUpdateProgress(int currentStep, int totalStep, int currentStepProgress, java.lang.String currentStepName)

方法

getCurrentStep()
getCurrentStepName()
getCurrentStepProgress()
getTotalStep()

解释

FirmwareUpdateProgress(int currentStep, int totalStep, int currentStepProgress, java.lang.String currentStepName)

创建对象 FirmwareUpdateProgress 且参数为 currentStep/totalStep/currentStepProgress/currentStepName。

getCurrentStep()

获取当前步骤，数据类型为 int。

getCurrentStepName()

获取当前步骤名称，数据类型为字符串 string。

getCurrentStepProgress()

获取当前步骤进程，数据类型为 int。

getTotalStep()

获取所有步骤，数据类型为 int。

Line 类

public class Line,表示一条线。

构造器

```
Line(int segmentId, PointF startPoint, PointF endPoint)
Line(int segmentId, float startX, float startY, float endX, float endY)
Line(Line line)
Line(PointF startP, PointF endP)
```

方法

```
getStartPoint()
setStartPoint(PointF startPointF)
getEndPoint()
setEndPoint(PointF endPoint)
getStartX()
getStartY()
getEndX()
getEndY()
getSegmentId()
setSegmentId(int segmentId)
```

解释

`Line(int segmentId, PointF startPoint, PointF endPoint)`

创建对象 Line 且 segment id，start point 和 end point 为指定值。

`Line(int segmentId, float startX, float startY, float endX, float endY)`

创建对象 Line 且 segment id, startX, startY, endX, endy 为指定值。

`Line(Line line)`

创建对象 Line 且以 Line 为参数。

`Line(PointF startP, PointF endP)`

创建对象 Line 且 startP 和 endP 为指定值。

`getStartPoint()`

获取起始点。

`setStartPoint(PointF startPointF)`

设置起始点。

`getEndPoint()`

获取结束点。

`setEndPoint(PointF endPoint)`

设置结束点。

`getStartX()`

获取 start x。

`getStartY()`

获取 star y。

`getEndX()`

获取 end x。

`getEndY()`

获取 end y。

`getSegmentId()`

获取 segment id。

`setSegmentId(int segmentId)`

设置 Segment id。

PointF 类

public class PointF,表示一个 2d 浮点数据类型。

构造器

`PointF()`
`PointF(float x, float y)`
`PointF(PointF rhs)`

方法

`getX()`
`setX(float x)`

getY()
setY(float y)

解释

PointF()

创建对象 PointF。

PointF(float x, float y)

创建对象 PointF 且 x，y 为指定值。

PointF(PointF rhs)

创建对象 PointF 且以 PointF 为参数。

getX()

获取 X。

setX(float x)

设置 X。

getY()

获取 Y。

setY(float y)

设置 Y。

Size 类

public class Size， 表示整数型的 size 数据类型。

构造器

Size()
Size(int width, int height)
Size(Size rhs)

方法

getWidth()
setWidth(int width)
getHeight()
setHeight(int height)

解释

`Size()`

创建对象 Size。

`Size(int width, int height)`

创建对象 Size 且 width 和 height 为指定值。

`Size(Size rhs)`

创建对象 Size 且以 Size 为参数。

`getWidth()`

获取 width。

`setWidth(int width)`

设置 width。

`getHeight()`

获取 height。

`setHeight(int height)`

设置 height。

HealthInfo 类

public class HealthInfo，表示机器人的健康状况。

嵌套类

HealthInfo.BaseError

构造器

HealthInfo()
HealthInfo(boolean warning, boolean error, boolean fatal, java.util.ArrayList<HealthInfo.BaseError> errors)

方法

getErrors()
isError()
isFatal()
isWarning()
setError(boolean error)
setErrors(java.util.ArrayList<HealthInfo.BaseError> errors)
setFatal(boolean fatal)
setWarning(boolean warning)

解释

HealthInfo()

创建对象 HealthInfo()。

HealthInfo(boolean warning, boolean error, boolean fatal, java.util.ArrayList<HealthInfo.BaseError> errors)

创建对象 HealthInfo()，参数为 warning/error/fatal/错误列表。

getErrors()

获取错误信息。返回值为错误信息列表。

isError()

获取是否是错误信息。返回数据类型为 Boolean。

isFatal()

获取是否是致命错误信息。返回数据类型为 Boolean。

`isWarning()`

获取是否是警告信息。返回数据类型为 Boolean。

`setError(boolean error)`

是否设置错误信息。

参数：error – 将要处理的错误信息

`setErrors(java.util.ArrayList<HealthInfo.BaseError> errors)`

设置为错误信息列表。

参数：errors – 错误信息列表

`setFatal(boolean fatal)`

设置为致命错误信息。

参数： fatal – 致命错误

`setWarning(boolean warning)`

设置为警告。

参数： warning – 警告类错误信息

HealthInfo.BaseError 类

封闭类： `HealthInfo`

Fields

`BaseErrorComponentMotion`

`BaseErrorComponentPower`

`BaseErrorComponentSensor`

`BaseErrorComponentSystem`

`BaseErrorComponentUnknown`

`BaseErrorComponentUser`

`BaseErrorLevelError`

`BaseErrorLevelFatal`

BaseErrorLevelUnknown

BaseErrorLevelWarn

构造器

BaseError()

BaseError(int id, int errorCode, int errorLevel, int errorComponent, int componentErrorCode, java.lang.String errorMessage)

方法

getComponentErrorCode()

getErrorCode()

getErrorComponent()

getErrorLevel()

getErrorMessage()

getId()

setComponentErrorCode(int componentErrorCode)

setErrorCode(int errorCode)

setErrorComponent(int errorComponent)

setErrorLevel(int errorLevel)

setErrorMessage(java.lang.String errorMessage)

setId(int id)

解释

BaseErrorComponentMotion

机器人底盘运动错误。

BaseErrorComponentPower

机器人底盘充电错误。

BaseErrorComponentSensor

机器人底盘传感器错误。

`BaseErrorComponentSystem`

机器人底盘系统错误。

`BaseErrorComponentUnknown`

机器人底盘未知错误。

`BaseErrorComponentUser`

机器人底盘用户错误。

`BaseErrorLevelError`

底盘错误等级为错误。

`BaseErrorLevelFatal`

底盘错误等级为致命错误。

`BaseErrorLevelUnknown`

底盘错误等级为未知错误。

`BaseErrorLevelWarn`

底盘错误等级为警告错误。

`BaseError()`

创建对象 BaseError（ ）

`BaseError(int id, int errorCode, int errorLevel, int errorComponent, int componentErrorCode, java.lang.String errorMessage)`

创建对象 BaseError（ ）。参数为错误代码（error code），错误等级（error level），错误组件（error component），组件错误代码（componentErrorCode），错误信息（errorMessage）。

`getComponentErrorCode()`

获取组件错误代码。返回数据类型为 int。

`getErrorCode()`

获取错误代码。返回数据类型为 int。

`getErrorComponent()`

获取错误组件。返回数据类型为 int。

`getErrorLevel()`

获取错误等级。返回数据类型为 int。

`getErrorMessage()`

获取错误信息。返回数据类型为 string。

`getId()`

获取错误 id。返回数据类型为 int。

`setComponentErrorCode(int componentErrorCode)`

设置组件错误代码。返回值为组件错误代码。数据类型为 int。

`setErrorCode(int errorCode)`

设置错误代码。返回值为错误代码。数据类型为 int。

`setErrorComponent(int errorComponent)`

设置错误组件。返回值为错误组件。数据类型为 int。

`setErrorLevel(int errorLevel)`

设置错误等级。返回值为错误等级。数据类型为 int。

`setErrorMessage(java.lang.String errorMessage)`

设置错误信息。返回值为错误信息。数据类型 string。

`setId(int id)`

设置 Id。返回值为 Id。数据类型为 int。

LaserPoint 类

public class LaserPoint, 表示一个激光扫描点。

构造器

`LaserPoint()`
`LaserPoint(float distance, float angle)`
`LaserPoint(float distance, float angle, boolean valid)`
`LaserPoint(LaserPoint rhs)`

方法

解释

LaserPoint()

创建对象 LaserPoint。

LaserPoint(float distance, float angle)

创建对象 LaserPoint 且距离和角度为指定值。

LaserPoint(float distance, float angle, boolean valid)

创建对象 LaserPoint 且距离，角度和有效性为指定值。

LaserPoint(LaserPoint rhs)

创建对象 LaserPoint 且以 LaserPoint 为参数。

getDistance()

获取距离。

setDistance(float distance)

设置距离。

getAngle()

获取角度。

setAngle(float angle)

设置角度。

isValid()

测量是否有效。

setValid(boolean valid)

将值设为有效。

LaserScan 类

public class LaserScan，表示激光扫描。

构造器

LaserScan()
LaserScan(java.util.Vector<LaserPoint> laserPoints)
LaserScan(java.util.Vector<LaserPoint> laserPointsPose pose)
LaserScan(LaserScan rhs)

方法

getLaserPoints()
setLaserPoints(java.util.Vector<LaserPoint> laserPoints)
getPose()
setPose(Pose pose)

解释

LaserScan()

创建对象 LaserScan。

LaserScan(java.util.Vector<LaserPoint> laserPoints)

创建对象 LaserScan 且以 LaserPoints 为参数。

LaserScan(java.util.Vector<LaserPoint> laserPointsPose pose)

创建对象 LaserScan 且以 LaserPointPose 为参数。

LaserScan(LaserScan rhs)

创建对象 LaserScan 且以 LaserScan 为参数。

getLaserPoints()

获取激光点。

setLaserPoints(java.util.Vector<LaserPoint> laserPoints)

设置激光点。

getPose()

获取姿态。

`setPose(Pose pose)`

设置姿态。

Location 类

public class Location，表示 3d 空间中机器人的位置。

构造器

`Location()`
`Location(float x, float y, float z)`
`Location(Location rhs)`

方法

`distanceTo(Location that)`
`getX()`
`setX(float v)`
`getY()`
`setY(float v)`
`getZ()`
`setZ(float v)`

解释

`Location()`

创建对象 Location。

`Location(float x, float y, float z)`

创建对象 Location 且 x，y，z 为指定值。

`Location(Location rhs)`

创建对象 Location 且以 Location 为参数。

`distanceTo(Location that)`

获取到 Location 的距离。

`getX()`

获取 X 值。

`setX(float v)`

设置 X 值。

getY()

获取 Y 值。

setY(float v)

设置 Y 值。

getZ()

获取 Z 值。

setZ(float v)

设置 Z 值。

Map 类

public class Map，表示地图。

构造器

Map(PointF origin, Size dimension, PointF resolution, long timestamp, byte[] data)

方法

getOrigin()
setOrigin(PointF origin)
getDimension()
setDimension(Size dimension)
getResolution()
setResolution(PointF resolution)
getTimestamp()
setTimestamp(long timestamp)
getMapArea()
getData()
setData(byte[] data)

解释

Map(PointF origin, Size dimension, PointF resolution, long timestamp, byte[] data)

创建对象 Map。

getOrigin()

获取 origin。

`setOrigin(PointF origin)`

设置 origin。

`getDimension()`

获取 dimension。

`setDimension(Size dimension)`

设置 dimensions。

`getResolution()`

获取 resolution。

`setResolution(PointF resolution)`

设置 resolution。

`getTimestamp()`

获取 time stamp。

`setTimestamp(long timestamp)`

设置 time stamp。

`getMapArea()`

获取 map area。

`getData()`

获取 data。

`setData(byte[] data)`

设置 data。

NetworkMode 类

Fields

`NetworkModeAP`
`NetworkModeStation`
`NetworkModeWifiDisabled`

构造器

NetworkMode()

解释

NetworkModeAP

网络模式为 AP。

NetworkModeStation

网络模式为 Station。

NetworkModeWifiDisabled

Wifi 禁用的网络模式。

NetworkMode()

创建对象 NetworkMode()。

Pose 类

public class Pose，表示机器人姿态。

构造器

Pose()
Pose(Location loc, Rotation rot)
Pose(float x, float y, float z, float yaw, float roll, float pitch)
Pose(Pose rhs)

方法

Location getLocation()
setLocation(Location location)
getRotation()
setRotation(Rotation rotation)
getX()
setX(float v)
getY()
setY(float v)
getZ()
setZ(float v)
getYaw()
setYaw(float v)
getRoll()
setRoll(float v)

getPitch()
setPitch(float v)

解释

Pose()

创建对象 Pose。

Pose(Location loc, Rotation rot)

创建对象 Pose 且 loc 和 ros 为指定值。

Pose(float x, float y, float z, float yaw, float roll, float pitch)

创建对象 Pose 且 x，y，z，yaw，roll，pitch 为指定值。

Pose(Pose rhs)

创建对象 Pose 且以 Pose 为参数。

Location getLocation()

获取 Location。

setLocation(Location location)

设置 Location。

getRotation()

获取 rotation。

setRotation(Rotation rotation)

设置 rotation。

getX()

获取 X。

setX(float v)

设置 X。

getY()

获取 Y。

`setY(float v)`

设置 Y。

`getZ()`

获取 Z。

`setZ(float v)`

设置 Z。

`getYaw()`

获取 yaw。

`setYaw(float v)`

设置 yaw。

`getRoll()`

获取 roll。

`setRoll(float v)`

设置 roll。

`getPitch()`

获取 pitch。

`setPitch(float v)`

设置 pitch。

Rotation 类

public class Rotation。

构造器

`Rotation()`
`Rotation(float yaw)`
`Rotation(float yaw, float pitch, float roll)`
`Rotation(Rotation rhs)`

方法

`getYaw()`

setYaw(float yaw)
getRoll()
setRoll(float roll)
getPitch()
setPitch(float pitch)

解释

Rotation()

创建对象 Rotation。

Rotation(float yaw)

创建对象 Rotation 且 yaw 为指定值。

Rotation(float yaw, float pitch, float roll)

创建对象 Rotation 且 yaw，pitch，roll 为指定值。

Rotation(Rotation rhs)

创建对象 Rotation 且以 Rotation 为参数。

getYaw()

获取 yaw。

setYaw(float yaw)

设置 Yaw。

getRoll()

获取 roll。

setRoll(float roll)

设置 roll。

getPitch()

获取 pitch。

setPitch(float pitch)

设置 pitch。

ScheduledTask 类

构造器

ScheduledTask(int taskId, java.lang.String task, int weekRepeat, boolean enabled, int maxDuration, int year, int month, int day, int hour, int minute)

方法

getDay()
getHour()
getMaxDuration()
getMinute()
getMonth()
getTask()
getTaskId()
getWeekRepeat()
getYear()
isEnabled()
setDay(int day)
setEnabled(boolean enabled)
setHour(int hour)
setMaxDuration(int maxDuration)
setMinute(int minute)
setMonth(int month)
setTask(java.lang.String task)
setTaskId(int taskId)
setWeekRepeat(int weekRepeat)
setYear(int year)

解释

ScheduledTask(int taskId, java.lang.String task, int weekRepeat, boolean enabled, int maxDuration, int year, int month, int day, int hour, int minute)

创建对象 ScheduledTask()。

getDay()

获取时间信息中的日期，返回数据类型为 int。

getHour()

获取时间信息中的小时，返回数据类型为 int。

getMaxDuration()

获取最大持续时间，返回数据类型为 int。

`getMinute()`

获取时间信息中的 分钟，返回数据类型为 int。

`getMonth()`

获取时间信息中的 月份， 返回数据类型为 int。

`getTask()`

获取任务,返回数据类型为 string。

`getTaskId()`

获取任务 id， 返回数据类型为 int。

`getWeekRepeat()`

获取每周重复清扫信息，返回数据类型为 int。

`getYear()`

获取时间信息中的 年， 返回数据类型为 int。

`isEnabled()`

获取是否开启预约任务， 返回数据类型为 boolean。

`setDay(int day)`

设置日期.

参数：day – 日期

`setEnabled(boolean enabled)`

是否开启。

参数：enabled – 开启

`setHour(int hour)`

设置小时。

参数：hour – 小时

`setMaxDuration(int maxDuration)`

设置最长持续时间。

参数：maxDuration – 最长持续时间。

`setMinute(int minute)`

设置分钟。

参数：minute – 分钟

`setMonth(int month)`

设置月份。

参数： month – 月份

`setTask(java.lang.String task)`

设置任务。

参数：task – 任务名称

`setTaskId(int taskId)`

设置任务 id。

参数：taskId – 任务 id

`setWeekRepeat(int weekRepeat)`

设置周重复。

参数：weekRepeat – 周重复

`setYear(int year)`

设置年份。

参数：year – 年

SystemParameters 类

public class SystemParameters。

构造器

`SystemParameters()`

Fields

SYSPARAM_ROBOT_SPEED
SYSVAL_ROBOT_SPEED_HIGH
SYSVAL_ROBOT_SPEED_MEDIUM
SYSVAL_ROBOT_SPEED_LOW

解释

SystemParameters()

创建对象 SystemParameters。

SYSPARAM_ROBOT_SPEED

机器人速度。

SYSVAL_ROBOT_SPEED_HIGH

机器人高速度。

SYSVAL_ROBOT_SPEED_MEDIUM

机器人中速度。

SYSVAL_ROBOT_SPEED_LOW

机器人低速度。

MapKind 枚举

枚举常量

EXPLORE_MAP
SWEEP_MAP

解释

EXPLORE_MAP

利用 SLAM 算法创建的地图。

SWEEP_MAP

清扫操作使用的地图（仅适用于扫地机版本机器人）。

MapType 枚举

public enum MapType

枚举常量

BITMAP_8BIT

解释

BITMAP_8BIT

位图，每个像素是一个带符号的 8bit integer。

RestartMode 枚举

枚举常量

SOFT
HARD

解释

SOFT

软重启模块，仅重启模块中的核心部分。

HARD

硬重启模块，整个模块重新启动，可能会花费较长的时间。

日期	版本	描述
2016-05-03	0.1	初始版本
2016-06-07	1.8	更新封面图片
2016-09-30	1.8	添加部分类定义