目录

目录

[目录 1](#_Toc470621412)

[第一章 概述 2](#_Toc470621413)

[1.1文档目的 2](#_Toc470621414)

[1.2接口类型 2](#_Toc470621415)

[1.3 javascriot库 2](#_Toc470621416)

[第二章 基础接口 2](#_Toc470621417)

[2.1 初始化 3](#_Toc470621418)

[2.1 构造Topic 3](#_Toc470621419)

[2.1 构造控制指令 3](#_Toc470621420)

[2.2 构造导航地图信息 3](#_Toc470621421)

[2.3 构造坐标点信息 4](#_Toc470621422)

[2.4 目标点控制器 4](#_Toc470621423)

[2.5 构造Waypoint信息 4](#_Toc470621424)

[2.6 构造Trajectory信息 4](#_Toc470621425)

[2.7 构造NavCtrl指令 5](#_Toc470621426)

[2.8 构造速度信息 5](#_Toc470621427)

[2.9 TPublish 5](#_Toc470621428)

[2.10 Subscribe 5](#_Toc470621429)

[2.11 Unsubscribe 6](#_Toc470621430)

[第三章 应用接口 6](#_Toc470621431)

[3.1 显示导航地图 6](#_Toc470621432)

[3.2 移除地图 6](#_Toc470621433)

[3.3 切换至建图模式 6](#_Toc470621434)

[3.4 切换至导航模式 7](#_Toc470621435)

[3.5 保存建好的地图 7](#_Toc470621436)

[3.6 保存修改后的地图 7](#_Toc470621437)

[3.7 取消当前导航指令 7](#_Toc470621438)

[3.8 关闭手柄 7](#_Toc470621439)

[3.9 开启手柄 8](#_Toc470621440)

[3.10 添加站点 8](#_Toc470621441)

[3.12 删除站点 8](#_Toc470621442)

[3.13 添加轨迹 8](#_Toc470621443)

[3.14 删除轨迹 8](#_Toc470621444)

[3.15 单点运动 9](#_Toc470621445)

[3.16 前进 9](#_Toc470621446)

[3.17 后退 9](#_Toc470621447)

[3.18 左转 9](#_Toc470621448)

[3.19 右转 9](#_Toc470621449)

[3.20 停止 9](#_Toc470621450)

[第四章 导航状态获取示例 10](#_Toc470621451)

[4.1 状态回调函数1 10](#_Toc470621452)

[4.2 状态回调函数2 11](#_Toc470621453)

[第五章 示例 11](#_Toc470621454)

# 第一章 概述

## 1.1文档目的

略

## 1.2接口类型

文档提供的导航接口基于javascript开发，适用于Web开发

若需要其他语言接口，请参考链接

<https://github.com/RobotWebTools/rosbridge_suite/blob/groovy-devel/ROSBRIDGE_PROTOCOL.md>

## 1.3 javascriot库

easeljs.min.js

eventemitter2.min.js

roslib.min.js

ros2d.min.js

nav2d.min.js

navigation.js

# 第二章 基础接口

使用发布/订阅消息模式，提供一对多的消息发布，解除应用程序耦合

工作空间：NavEvent

## 2.1 初始化

Websocket初始化

@@option {object}

@option.url:服务器地址

@ option.onopen:连接成功回调

@option.onclose:连接关闭回调

@ option.onerror:连接错误回调

|  |  |  |
| --- | --- | --- |
| 函数名 | 参数及说明 | 返回值 |
| Init（option） | Option：初始化参数 | 无 |

## 2.1 构造Topic

构建Topic

@@option {object} 参考 TopicEnum

|  |  |  |
| --- | --- | --- |
| 函数名 | 参数及说明 | 返回值 |
| Topic（option） | Option：topic参数 | topic对象 |

## 2.1 构造控制指令

构造Msg

@data {string} 参考CmdEnum/ShellEnum

|  |  |  |
| --- | --- | --- |
| 函数名 | 参数及说明 | 返回值 |
| Msg（data） | 略 | Mesage对象 |

## 2.2 构造导航地图信息

\*地图信息

\*@@info {object} 地图头信息 包括长、宽

\*@data {Array} 地图各像素值

|  |  |  |
| --- | --- | --- |
| 函数名 | 参数及说明 | 返回值 |
| MapMessage（info, data） | 略 | Mesage对象 |

## 2.3 构造坐标点信息

\*构造坐标点信息

\*@posX {number} 距离原点的水平距离

\*@posY {number} 距离原点的垂直距离

\*@oriZ {number} 角度

\*@oriW {number} 角度

|  |  |  |
| --- | --- | --- |
| 函数名 | 参数及说明 | 返回值 |
| Pose（posX, posY, oriZ, oriW） | 略 | Pose对象 |

## 2.4 目标点控制器

\*目标点控制器

\*@pose {object} 目标坐标点

|  |  |  |
| --- | --- | --- |
| 函数名 | 参数及说明 | 返回值 |
| Goal（pose） | 略 | Goal对象 |

## 2.5 构造Waypoint信息

\*构造Waypoint信息

\*@name {string} waypoint 标记名

\*@pose {object} 坐标点

\*@radius {number} 半径

\*@timeout {number} 超时时间 单位 MS

\*@keep {string} 未能正常到达目标点后动作 LOOP 再一次尝试 NONE 不再尝试

\*@mode {string} 参考WaypointMode说明

|  |  |  |
| --- | --- | --- |
| 函数名 | 参数及说明 | 返回值 |
| WaypointMessage（name, pose, radius, timeout, keep, mode） | 略 | Waypoint对象 |

## 2.6 构造Trajectory信息

\*构造Trajectory信息

\*@name {string} trajectory 标记名

\*@waypoints {Array} waypoint集合

|  |  |  |
| --- | --- | --- |
| 函数名 | 参数及说明 | 返回值 |
| TrajectoryMessage（name, waypoints） | 略 | Trajectory对象 |

## 2.7 构造NavCtrl指令

\*构造Waypoint与Trajectory的控制指令

\*@control {number} 1:启动 2:停止

\*@goal\_name {string} 需要执行的waypoint/trajectory标记名

|  |  |  |
| --- | --- | --- |
| 函数名 | 参数及说明 | 返回值 |
| NavCtrlMessag（control, goal\_name） | 略 | NavCtrl对象 |

## 2.8 构造速度信息

\*构造速度信息

\*@linear\_x {number} 线速度

\*@angular\_z {number} 角速度

|  |  |  |
| --- | --- | --- |
| 函数名 | 参数及说明 | 返回值 |
| TwistMessage（linear\_x, angular\_z） | 略 | Twist对象 |

## 2.9 TPublish

\*发布Topic

\*@topic {object} Topic函数构造生成

\*@data {object} 各种消息的构造函数生成 例如：Msg、WaypointMessage、TrajectoryMessage、TwistMessage

|  |  |  |
| --- | --- | --- |
| 函数名 | 参数及说明 | 返回值 |
| TPublish（topic, data） | 略 | 无 |

## 2.10 Subscribe

\*订阅Topic

\*@topic {object} 参考TopicEnum类型

\*@callback {function} 回调

|  |  |  |
| --- | --- | --- |
| 函数名 | 参数及说明 | 返回值 |
| Subscrib（topic, callback） | 略 | 无 |

## 2.11 Unsubscribe

\*取消订阅Topic

\*@topic {object} Subscribe的返回值

\*@callback {function} 订阅的回调函数

|  |  |  |
| --- | --- | --- |
| 函数名 | 参数及说明 | 返回值 |
| Unsubscribe（topic, callback） | 略 | 无 |

# 第三章 应用接口

## 3.1 显示导航地图

\*显示地图

\*@width {number} 画布宽度

\*@height {number} 画布高度

\*@div {string} 画布父容器ID

|  |  |  |
| --- | --- | --- |
| 函数名 | 参数及说明 | 返回值 |
| ShowMap（width, height, div） | 略 | Viewer对象 |

## 3.2 移除地图

\*移除地图

\*@div {string} 画布父容器ID

|  |  |  |
| --- | --- | --- |
| 函数名 | 参数及说明 | 返回值 |
| RemoveMap（div） | 略 | Viewer对象 |

## 3.3 切换至建图模式

\*切换至建图模式

|  |  |  |
| --- | --- | --- |
| 函数名 | 参数及说明 | 返回值 |
| Gmapping（） | 略 | 无 |

## 3.4 切换至导航模式

\*切换至导航模式

|  |  |  |
| --- | --- | --- |
| 函数名 | 参数及说明 | 返回值 |
| Navigation（） | 略 | 无 |

## 3.5 保存建好的地图

\*保存建好的地图 和shell\_feedback 状态回调配合使用

|  |  |  |
| --- | --- | --- |
| 函数名 | 参数及说明 | 返回值 |
| SaveMap（） | 略 | 无 |

## 3.6 保存修改后的地图

\*保存修改后的地图

\*@info {object} 地图头信息

\*@data {Array} 地图各像素点值

|  |  |  |
| --- | --- | --- |
| 函数名 | 参数及说明 | 返回值 |
| SaveAsMapEdit（） | 略 | 无 |

## 3.7 取消当前导航指令

\*取消当前导航指令

|  |  |  |
| --- | --- | --- |
| 函数名 | 参数及说明 | 返回值 |
| Cancel（） | 略 | 无 |

## 3.8 关闭手柄

\*关闭手柄指令

|  |  |  |
| --- | --- | --- |
| 函数名 | 参数及说明 | 返回值 |
| CloseHandle（） | 略 | 无 |

## 3.9 开启手柄

\*开启手柄指令

|  |  |  |
| --- | --- | --- |
| 函数名 | 参数及说明 | 返回值 |
| OpenHandle（） | 略 | 无 |

## 3.10 添加站点

\*站点添加

\*@wapoint {object} 站点信息 WaypointMessage

|  |  |  |
| --- | --- | --- |
| 函数名 | 参数及说明 | 返回值 |
| WayPointAdd（waypoint） | 略 | 无 |

## 3.12 删除站点

\*站点删除

\*@wapoint {object} 站点信息 WaypointMessage

|  |  |  |
| --- | --- | --- |
| 函数名 | 参数及说明 | 返回值 |
| WayPointRemove（waypoint） | 略 | 无 |

## 3.13 添加轨迹

\*轨迹添加

\*@trajectory {object} 站点信息 TrajectoryMessage

|  |  |  |
| --- | --- | --- |
| 函数名 | 参数及说明 | 返回值 |
| TrajectoryAdd（waypoint） | 略 | 无 |

## 3.14 删除轨迹

\*轨迹删除

\*@trajectory {object} 站点信息 TrajectoryMessage

|  |  |  |
| --- | --- | --- |
| 函数名 | 参数及说明 | 返回值 |
| TrajectoryRemove（waypoint） | 略 | 无 |

## 3.15 单点运动

\* 单点运动 导航至指定地点

\* @pose {Pose} 坐标点

\* @goalCallback {Fuction} 运动状态回调

|  |  |  |
| --- | --- | --- |
| 函数名 | 参数及说明 | 返回值 |
| goPostion（pose, goalCallback） | 略 | 无 |

## 3.16 前进

|  |  |  |
| --- | --- | --- |
| 函数名 | 参数及说明 | 返回值 |
| goFront（） | 略 | 无 |

## 3.17 后退

|  |  |  |
| --- | --- | --- |
| 函数名 | 参数及说明 | 返回值 |
| goBack（） | 略 | 无 |

## 3.18 左转

|  |  |  |
| --- | --- | --- |
| 函数名 | 参数及说明 | 返回值 |
| goLeft（） | 略 | 无 |

## 3.19 右转

|  |  |  |
| --- | --- | --- |
| 函数名 | 参数及说明 | 返回值 |
| goRight（） | 略 | 无 |

## 3.20 停止

|  |  |  |
| --- | --- | --- |
| 函数名 | 参数及说明 | 返回值 |
| goStop（） | 略 | 无 |

# 第四章 导航状态获取示例

## 4.1 状态回调函数1

地图状态 手柄开关 软件更新状态 log日志获取等

NavEvent.Subscribe(NavEvent.TopicEnum.shell\_feedbackTopic, function (data) {

console.log(data);

var temp = data.data.split(/[ :]/);

switch (temp[0]) {

case "joy\_on":

break;

case "joy\_off":

break;

\*保存map地图成功，并通过cmdTopic 发布load\_map,save\_map\_edit消息

case "save\_map":

NavEvent.Publish(NavEvent.TopicEnum.cmdTopic, NavEvent.CmdEnum.LoadMap, NavEvent.CmdEnum.SaveMapEdit);

break;

\*保存map\_edit地图成功，并通过cmdTopic 发布load\_map\_edit消息

case "save\_map\_edit":

NavEvent.Publish(NavEvent.TopicEnum.cmdTopic, NavEvent.CmdEnum.LoadMapEdit);

break;

\*修改map\_edit地图成功

case "save\_as\_map\_edit":

//TODO

break;

case "version":

break;

case "user\_status":

break;

case "user\_auth":

break;

case "software\_update":

break;

case "dbparam":

break;

case "roslog":

break;

}

});

## 4.2 状态回调函数2

模式切换状态 获取

NavEvent.Subscribe(NavEvent.TopicEnum.diagnosticsTopic, function (data) {

console.log(data);

for (var i = 0; i < data.status.length; i++) {

if (data.status[i].name == "/Other/ros\_mode") {

switch (data.status[i].message) {

case NavEvent.NavigationModeEnum.Navigation:

break;

case NavEvent.NavigationModeEnum.Gmapping:

break;

case NavEvent.NavigationModeEnum.Coverting:

break;

default:

break;

}

return;

}

}

});

# 第五章 示例

$(function () {

NavEvent.Init({

url: "ws://192.168.0.251:9090",

onopen: function () {

console.log("success");

},

onclose: function () {

console.log("close");

}

});

})