# ROS CONTROL

# 通讯方式

rosbridge

Android

ROS

# 通讯格式

# 启动方式

roslaunch wisebot\_bringup wisebot\_bringup\_bridge.launch

格式：

Roslaunch package\_name file.launch

## 3.1 launch 文件简述

a. launch文件由一对<launch> </launch>包起来，元素都在两个标签之内

b. node 节点格式

<node pkg=”package-name” type=”executable-name” name=”node-name”/>

Node节点也可以显式地给出结束符号

<node pkg=”…”type=”…” name=”…”></node>

在<node pkg=”…”type=”…” name=”…”>和</node>为该节点的子节点

1. 输出到控制台：

设置output属性可以信息输出到控制台，如下所示：

<node pkg="robot\_pose\_publisher" type="robot\_pose\_publisher" name="robot\_pose\_publisher" output="screen">

如果要设置所有的node都输出到控制台，可以在launch的时候使用

Roslaunch --screen packagename launch\_file\_name

1. Include 其他文件

使用include其他文件的时候可以写成如下的形式

<include file=”path-to-launch-file”>

但是这种形式不好的地方就是需要给出文件的绝对路径，在使用的时候显得很麻烦，因此也可以写成如下的形式：

<include file=”$(find packagename)/launch\_filename”/>

1. Argument的使用

X.申明argument

<arg name =”arg\_name”>

Y.指定argument的值

<arg name=”arg\_name” default=”arg\_name”/>

<arg name=”arg\_name” value=”arg\_name”/>

命令行赋值：

Roslaunch packagename launch\_filename arg-name:=arg\_value

使用命令行赋值的时候可以覆盖default的数值，但是不能覆盖value的数值

# 流程

## 4.1 主体流程

app

websocket

回调函数

roslaunch wisebot\_bringup wisebot\_bringup\_bridge.launch

Wisebot\_bridge\_node.cpp main wisebot\_bridge::WisebotBridge wisebot\_bridge(nh, nh\_private);

roslunch

main

Namespace wisebot\_bridge

创建service等待app发送过来的消息

rosbridge

# 5.ROS基础

## 5.1 rospackage

软件包，是ros应用程序代码的组织单元，每个软件包都可以包含程序库、可执行文件、脚本或者其他手动创建的内容； package中的package.xml描述了package中的各项信息：依赖，版本，维护者信息等各项信息。

Rospackage通过命令catkin\_create\_pkg创建，格式如下:

catkin\_create\_pkg packagename dependence1 dependence2 ...

## 5.2 rosnode

# 6.Debug

## 6.1 打印信息

ROS\_INFO(“Debug message.”);

## 6.2 websocket在线测试

<http://www.blue-zero.com/WebSocket/>

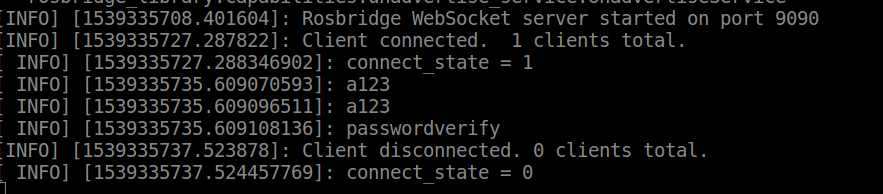
测试example：

默认端口为9090

在线端



打印的log：



6.3 打印topic信息

查看topic list

rostopic list

打印topic信息

rostopic echo /cmd\_vel

参考链接：

<https://www.cnblogs.com/Jessica-jie/p/6961837.html>