

ROS的重要概念

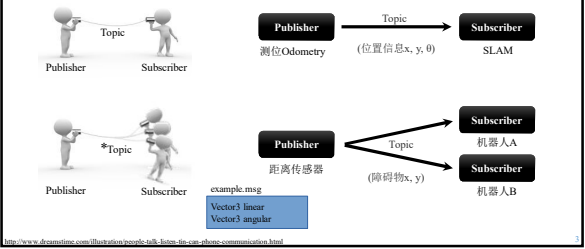
1. ROS 术语
2. 消息通信
3. 消息
4. 名称
5. 坐标变换(TF)
6. 客户端库
7. 异构设备间的通信
8. 文件系统
9. 构建系统

ROS 条款

- **Node (节点)**
 - ROS中运行的最小处理单元。它可以看作是单个可执行程序。在ROS中，系统由多个节点组成。每个节点通过消息通信发送和接收数据。
- **Package (功能包)**
 - 一个或多个节点，用于节点执行的消息等。元功能包(metapackage)是一个具有共同目的的功能包的集合。
- **Message (消息)**
 - 节点之间通过消息(message)来发送和接收数据。消息是诸如integer、floating point和boolean等类型的变量。用户还可以使用诸如消息里包括消息的简单数据结构 或列举消息的消息数组的结构。使用消息的通信方法包括TCPROS、UDPROS等，根据情况使用单向消息发送/接收方式的话题(topic)和双向消息请求(request)/响应(response)方式的服务(service)。

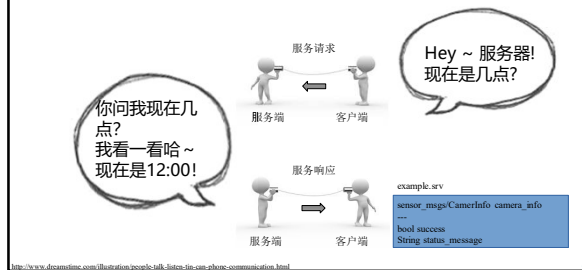
ROS 条款

主题、发布者、订阅者



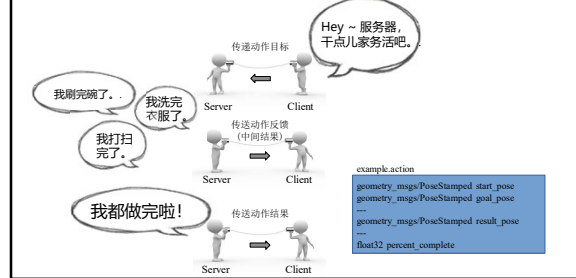
ROS 条款

服务，服务服务器，服务客户端



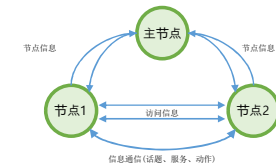
ROS 条款

动作，动作服务器，动作客户端



消息通信的过程

- ROS最基本的特征:
节点间的消息通信!



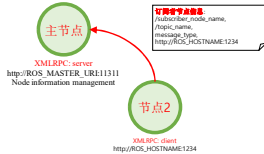
消息通信的过程

1. 运行主节点: XMLRPC(XML-Remote Procedure Call)
- \$ roscore



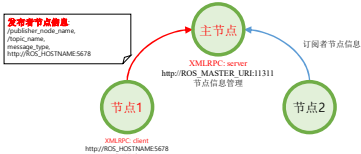
消息通信的过程

2. 运行订阅者节点
- \$roslaunch packagename nodename



消息通信的过程

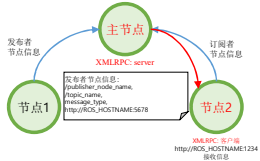
3. 运行发布者节点
- \$roslaunch packagename nodename



消息通信的过程

4. 通知发布者信息

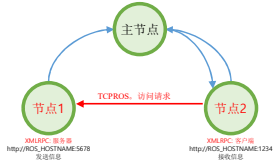
主节点通知订阅节点新的发布者信息。



消息通信的过程

5. 订阅者节点的连接请求

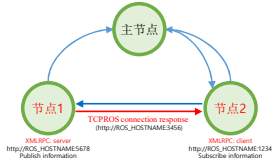
订阅者节点根据从主节点接收的发布者信息，向发布者节点请求直接连接。



消息通信的过程

6. 发布者节点的连接响应

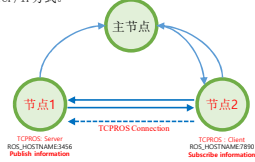
发布者节点将TCP服务器的URL地址和端口作为连接响应发送给订阅者节点



消息通信的过程

7. TCP ROS连接

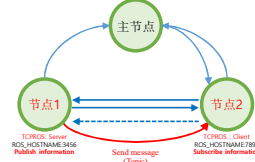
订阅者节点使用TCPROS创建一个与发布者节点对应的客户端，并直接与发布者节点连接。节点间通信使用一种称为TCPROS的TCP/IP方式。



消息通信的过程

8. 发送消息

- 发布者节点向订阅者节点发送消息。节点间通信使用一种称为TCPROS的TCP/IP方式。



消息通信的过程

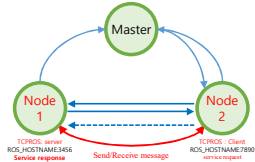
- 在话题模式下，除非连接被终止，否则将连续发送消息。也就是说，连续性。



消息通信的过程

9. 服务请求和响应

- 只执行一次服务请求和服务响应，并断开它们之间的连接。



消息通信的过程

- 与话题模式不同，服务只连接一次，并在执行服务请求和服务响应后断开连接。也就是说，这是一次性的。



理解消息通信概念！

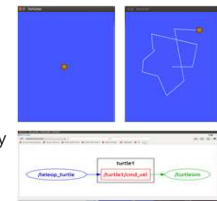
- turtlesim package

- roscore

- roslaunch turtlesim turtlesim_node

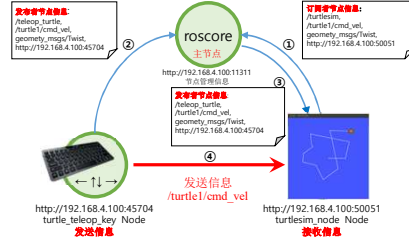
- roslaunch turtlesim turtle_teleop_key

- roslaunch rqt_graph rqt_graph

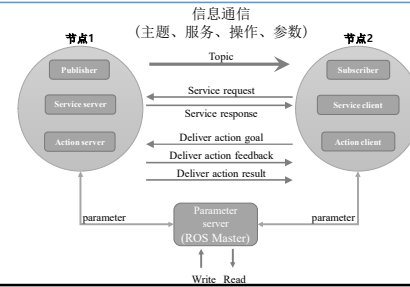


抓住消息传播理念!

• 10. 例子! turtlesim



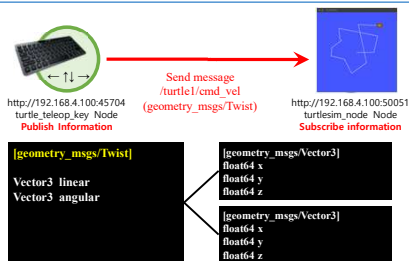
ROS 消息



ROS 消息

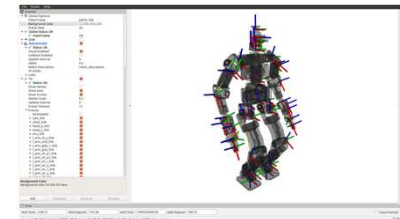
- 消息是一种围绕节点的数据传输
 - Topics, services, and actions all use messages
 - <http://wiki.ros.org/msg>
 - http://wiki.ros.org/common_msgs
- Simple type
 - ex) integer, floating point, boolean
 - http://wiki.ros.org/std_msgs
- A simple data structure containing messages in a message
 - ex) `geometry_msgs/PoseStamped`
 - http://docs.ros.org/api/geometry_msgs/html/msg/PoseStamped.html
- An array data structure in which messages are listed
 - ex) `float32[]` ranges
 - ex) `sensor_msgs/LaserScan`
 - http://docs.ros.org/api/sensor_msgs/html/msg/LaserScan.html

ROS 消息 (ex: geometry_msgs/Twist)



坐标变换 (TF, transform)

- 各关节相对坐标变换
 - 以树结构的形式表示关节之间的关系
 - 我们后面将在移动机器人 (第10章) 和机械手臂 (第13章) 的建模部分详细讨论TF的实例



名称

- 名称
- 节点或消息的唯一标识符 (主题、服务、操作、参数)
 - ROS 支持称为图 (graphs) 的抽象数据类型
- Global
 - Use the name as is or prepend a slash (/) to the name.
- Private
 - Prepend a tilde (~) to the name
- 这将在第7章中详细介绍。在这里, 我们通过修改话题名称来理解名称的用法

Node	Relative (基本)	Global	Private
/node1	bar -> /bar	/bar -> /bar	~bar -> /node1/bar
/wg/node2	bar -> /wg/bar	/bar -> /bar	~bar -> /wg/node2/bar
/wg/node3	foo/bar -> /wg/foo/bar	/foo/bar -> /foo/bar	~foo/bar -> /wg/node3/foo/bar

客户端库

- 支持多种编程语言
 - roscpp, rospy, roslisp
 - rosjava, roscs, roseus, rosgo, roshask, rosnodejs, RobotOS.jl, roslua, PhaROS, rosR, rosruby, Unreal-Ros-Plugin
- [MATLAB for ROS](#)
- [LabVIEW for ROS](#)

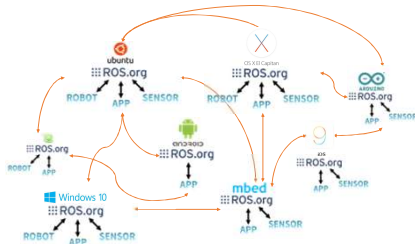


<http://wiki.ros.org/Client%20libraries>

<https://commons.wikimedia.org/wiki/File:Prog-languages.png>

24

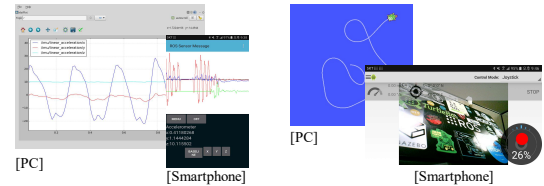
异构设备之间的通信



24

异构设备之间的通信

- Example 1: 远程传输图像 (see Chapter 8, Camera)
- Example 2: 在PC上检查Android智能手机的加速值([APP](#))
- Example 3: 用Android智能手机控制Turtlebot3 ([APP](#))



27

ROS 命令

- ROS 命令概述
- ROS shell命令
- ROS 执行命令
- ROS 信息命令
- ROS catkin命令
- ROS 功能包命令

28

ROS shell 命令

命令	重要度	命令释义	详细说明
roscd	★★★	ros+cd(changes directory)	移动到指定的ROS功能包目录
rosls	★☆☆	ros+ls(lists files)	显示ROS功能包的文件和目录
rosed	★☆☆	ros+ed(editor)	编辑ROS功能包文件
roscp	★☆☆	ros+cp(copies files)	复制ROS功能包文件
rospd	☆☆☆	ros+pushd	添加目录至ROS目录索引
rostd	☆☆☆	ros+directory	显示ROS目录索引中的目录

29

ROS 执行命令

命令	重要度	命令释义	详细说明
roscore	★★★	ros+core	- master (ROS 名称服务) - roscore (日记记录) - parameter server (参数管理)
roslaunch	★★★	ros+run	运行节点
roslaunch	★★★	ros+launch	运行多个节点及设置运行选项
rosclean	★★☆	ros+clean	检查或删除ROS日志文件

30

ROS 信息命令

命令	重要度	命令释义	详细说明
rostopic	★★★	ros+topic	确认ROS话题信息
rosservice	★★★	ros+service	确认ROS服务信息
roscall	★★★	ros+node	确认ROS节点信息
rosparam	★★★	ros+param(parameter)	确认和修改ROS参数信息
rosviz	★★★	ros+bag	记录和回放ROS消息
rosmv	★★☆	ros+msg	显示ROS消息类型
rosvr	★★☆	ros+srv	显示ROS服务类型
rosversion	★☆☆	ros+version	显示ROS功能包的版本信息
roswtf	☆☆☆	ros+wtf	检查ROS系统

31

ROS catkin 命令

命令	重要度	详细说明
catkin_create_pkg	★★★	自动生成功能包
catkin_make	★★★	基于catkin构建系统的构建
catkin_eclipse	★★☆	对于用catkin构建系统生成的功能包进行修改，使其能在 Eclipse环境中使用
catkin_prepare_release	★★☆	发布时用到的日志整理和版本标记
catkin_generate_changelog	★★☆	在发布时生成或更新CHANGELOG.rst文件
catkin_init_workspace	★★☆	初始化catkin构建系统的工作目录
catkin_find	★★☆	搜索catkin

32

ROS 功能包命令

命令	重要度	命令释义	详细说明
rospack	★★★	ros+pack(age)	查看与ROS功能包相关的信息
roscpp	★★☆	ros+install	安装ROS附加功能包
roscpp	★★☆	ros+dep(endencies)	安装该功能包的依赖性文件
roscpp	☆☆☆	ros+locate	ROS功能包信息相关命令
roscpp	☆☆☆	ros+create-pkg	自动生成ROS功能包(用于旧的roscpp系统)
roscpp	☆☆☆	ros+make	构建 ROS 功能包(用于旧的roscpp系统)

33

RViz (ROS 可视化工具)

ROS 工具

- ROS的三维可视化工具
 - 传感器数据可视化
 - 激光距离传感器 (LDS) 的距离数据
 - 来自深度相机的点云数据，如“RealSense”、“Kinect”、“Xtion”等。
 - 摄像机图像数据
 - IMU传感器的惯性数据……
- 表示机器人配置和计划运动
 - URDF (统一机器人描述格式)
- Navigation导航
- Manipulation操纵
- Tele-operation远程操作

RViz 案例 #1

- Point cloud data of 'Kinect'

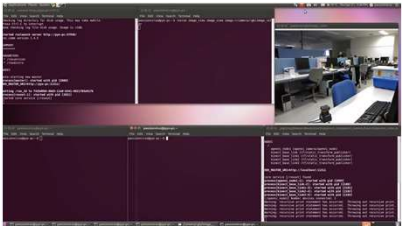


<https://youtu.be/DqOlpZBOpxY>

35

RViz 案例 #2

- Distance data of LASER distance sensor(LDS)

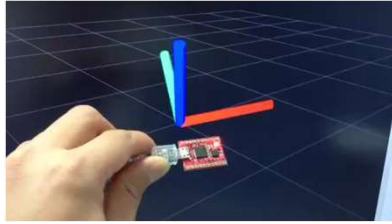


<https://youtu.be/qjtoAJ1wzB6s>

36

RViz 案例 #3

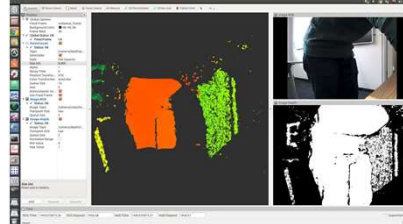
- Inertia data of IMU sensor



<https://youtu.be/j5v5JKppQo>

RViz 案例 #4

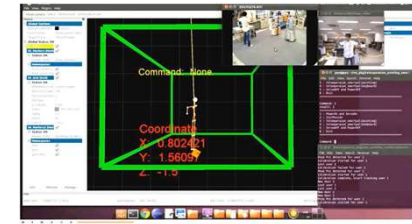
- Video of point cloud, color, depth of 'RealSense'



<https://youtu.be/J6kgPEzY4s>

RViz 案例 #5

- Show skeleton and direction of a person

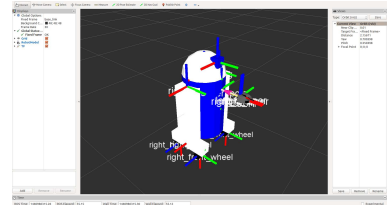


https://youtu.be/ah8uNv9c_Q

RViz 案例 #6

- R2-D2 robot model

```
$ sudo apt-get install ros-kinetic-urdf-tutorial
$ roslaunch urdf_tutorial_display.launch model:=S$(find urdf_tutorial)/urdf/05-visual.urdf
```



RViz 案例 #7

- Environment model, robot, path



https://youtu.be/9fba1AD1c_4

RViz 案例 #8

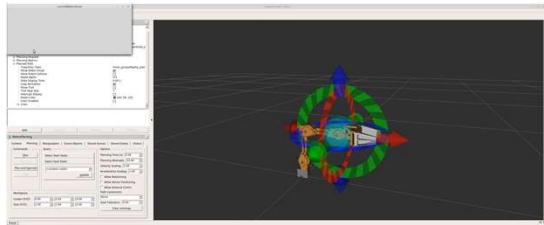
- Map display, navigation, and assign destination



<https://youtu.be/VYIMywcYALU>

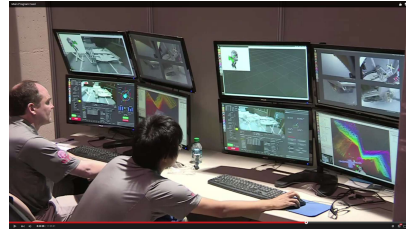
RViz 案例#9

- IK target position and path display using interactive markers



RViz 案例#10

- Disaster relief robots (2015 DARPA Robotics Challenge)



<https://www.youtube.com/user/DARPArc>

Rviz安装与测试

- RViz Installation

```
$ sudo apt-get install ros-kinetic-rviz
```

* If you installed 'ros-kinetic-desktop-full', RViz will be installed by default

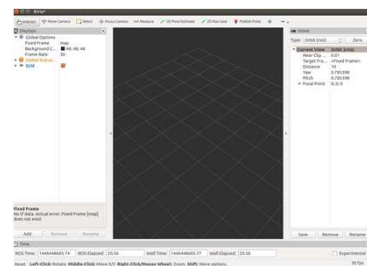
- Run RViz

```
$ roslaunch rviz rviz
```

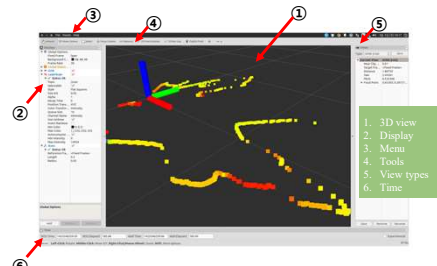
or

```
$ rviz
```

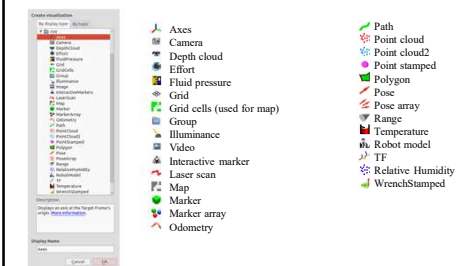
RViz 初始屏幕(尚未配置)



RViz 的屏幕配置(for LDS)



RViz 的显示类型(在“显示”菜单中单击“添加”)



用 ‘RViz’ , 传感器和机器人 相关数据可视化变得非常简单!

RQT:用于ROS的插件式综合GUI工具

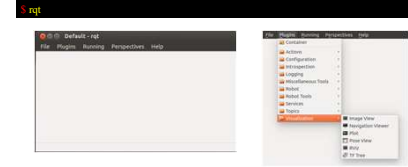
- 从ROS Fuerte版本开始, 现有的“rxbag”、“rxplot”、“rxgraph”等已与“rqt”合并。它现在是ROS的综合GUI工具, 带有诸如“rqt_bag”、“rqt_plot”、“rqt_graph”等插件
- 由于“rqt”是用“Qt”开发的, 用户可以自由添加和开发插件
- 让我们来看看‘rqt_image_view’、‘rqt_graph’、‘rqt_plot’、‘rqt_bag’, 它们是‘rqt’的代表性插件
- 此外, 还有一些插件, 如
- rqt_action, rqt_gui, rqt_plot, rqt_runtime_monitor, rqt_bag, rqt_gui_cpp, rqt_pose_view, rqt_rviz, rqt_bag_plugins, rqt_gui_py, rqt_publisher, rqt_service_caller, rqt_capabilities, rqt_image_view, rqt_py_common, rqt_shell, rqt_console, rqt_launch, rqt_py_console, rqt_srv, rqt_controller_manager, rqt_logger_level, rqt_reconfigure, rqt_tf_tree, rqt_dep, rqt_moveit, rqt_robot_dashboard, rqt_toprqt_ez_publisher, rqt_msg, rqt_robot_monitor, rqt_toprqt_graph, rqt_nav_view, rqt_robot_steering, rqt_web, etc. (wow.. —;;)

RQT 安装和测试

• RQT Installation

```
$ sudo apt-get install ros-kinetic-rqt ros-kinetic-rqt-common-plugins
```

• RQT Run



RQT Plug-in #1

1. Action

- Action Type Browser | Check the data structure of action type

2. Configuration

- Dynamic Reconfigure | Change the GUI setting value to change the setting value provide by the nodes
- Launch | GUI version of 'roslaunch'

3. Introspection

- Node Graph | Graph view showing relationship diagrams and message flow of running nodes
- Package Graph | Graph view showing node dependencies
- Process Monitor | Check CPU utilization, memory usage, and number of threads of running nodes

4. Logging

- Bag | ROS data logging
- Console | Check for messages such as warning, error that occur on the nodes
- Logger Level | Select and display logger information such as Debug, Info, Warn, Error, Fatal

RQT Plug-in #2

5. Miscellaneous Tools

- Python Console | Python console screen
- Shell | Activate shell
- Web | Activate web browser

6. Robot

- Depending on the robot, add a plug-in such as a dashboard

7. Robot Tools

- Controller Manager | Plug-in required to control the controller
- Diagnostic Viewer | Check robot device and error
- MoveIt! Monitor | Check 'MoveIt!' data used in robot arm planning
- Robot Steering | Robot adjustment GUI tool, used in remote control to steer the robot
- Runtime Monitor | Check for errors and warning on nodes in real time

RQT Plug-in #3

8. Services

- Service Caller | Connect to the running service server and request service
- Service Type Browser | Check the data structure of the service type

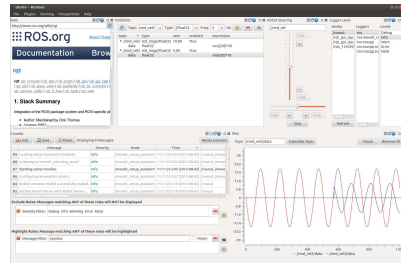
9. Topics

- Easy Message Publisher | Publish topic in GUI environment
- Topic Publisher | Create and publish topic
- Topic Type Browser | Check the data structure of the topic type
- Topic Monitor | Check the information of selected topic

10. Visualization

- Image View | Check image data of camera
- Navigation Viewer | Check location and target point of robot navigation
- Plot | 2D data plot GUI plug-in, 2D data plotting
- Pose View | Show current TF location and model location
- RViz | RViz plug-in which is 3D visualization tool
- TF Tree | Graph view showing tf relation as a tree structure

RQT 案例



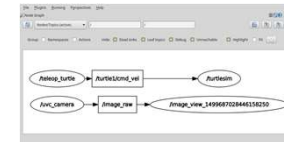
RQT 练习 #1: rqt_image_view

```
$ roslaunch uvc_camera uvc_camera_node
$ rqt ( Select [Plugins] -> [Visualization] -> [Image View] in menu )
or
$ rqt_image_view
```



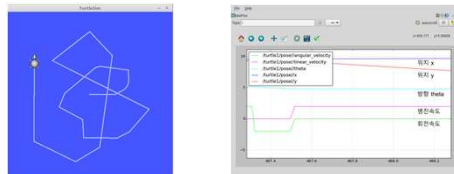
RQT 练习 #2: rqt_graph

```
$ roslaunch turtlesim turtlesim_node
$ roslaunch turtlesim turtle_teleop_key
$ roslaunch uvc_camera uvc_camera_node
$ roslaunch image_view image_view image:=image_raw
$ rqt ( Select [Plugins] -> [Introspection] -> [Node_Graph] in menu )
or
$ rqt_graph
```



RQT 练习 #3: rqt_plot

```
$ roslaunch turtlesim turtlesim_node
$ roslaunch turtlesim turtle_teleop_key
$ rqt ( Select [Plugins] -> [Visualization] -> [Plot] in menu )
or
$ rqt_plot /turtle1/pose/
```



RQT 练习 #4: rqt_bag

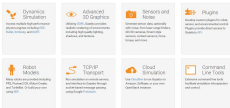
```
$ roslaunch uvc_camera uvc_camera_node
$ rosbag record /image_raw
$ rqt ( Select [Plugins] -> [Logging] -> [Bag] in menu )
or
$ rqt_bag
```



三维模拟器: Gazebo

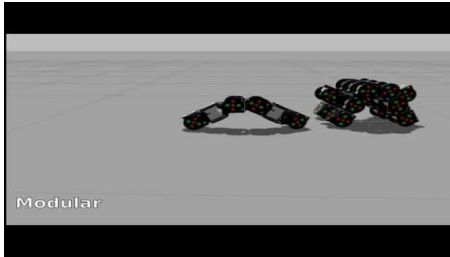
Gazebo

- Gazebo是具有物理引擎、机器人模型、传感器、环境模型等功能的三维仿真平台。它可以帮助您获得与真实环境中的**数据相似**的数据。
- Gazebo被认为是最近引进的开放式模拟器中最好的模拟器。此外，它还被选为**DARPA机器人挑战赛**的官方模拟器。
- 与ROS**高度兼容**



61

Gazebo



https://youtu.be/R3xUKYcG_bc

62

关键点?
如果你需要模拟器**,**
那么建议使用ROS & **Gazebo!!**