## כלי ROS

***catkin\_make***

***catkin\_create\_pkg [package] <depend1> <depend2>...***

***rosverison -d***

***rostopic* <bw>** - display bandwidth used by topic  
 **<echo>** - print message to screen  
 <find> - find topics by type  
 <hz> - display publishing rate of topic  
 <info> - print information about active topic  
 <list> - list active topics  
 <pub> - publish data to topic  
 <type> - print topic type

***rosrun*** <--prefix cmd> <--debug> [PACKAGE] [EXE] <args>

***rosnode*** <ping> - test connectivity  
 <list> - list active nodes  
 <info> - print information about node  
 <machine> - list nodes running on a particular machine or list of machines  
 <kill> - kill a running node  
 <cleanup> - purge registration information of unreachable nodes

***roscore*** <-p [port]> <-v>

***rosmaster <--core> <-p [port]>***

***rospack <depends>  
 <find>  
 <plugins>  
 ....***

***rosmsg <show>***- Show message declaration ***<list>***- List all messages ***<md5>*** - display message md5sum ***<package>*** - List messages in a package ***<packages>*** - List packages that contains messages

***rossrv <show>***- Show service declaration ***<list>***- List all services ***<md5>*** - display service md5sum ***<package>*** - List services in a package ***<packages>*** - List packages that contains services

***rosservice <args>*** - print service arguments ***<call>*** - call the service with the provided args ***<find>*** - find service by service type ***<info>* -** print information about service ***<list>*** - list active services ***<type>*** - print service type ***<uri>*** - print service ROSRPC uri

roslaunch <options> [package] [launch file] <args>

--files - print list files loaded by launch file

--args==NODE\_NAME - print command line arguments for node  
 --nodes - print list of nod names in launch file  
 --find-node=NODE\_NAME - find launch file that node is defined in

--child==NAME - Run as child service 'NAME'

--screen - force output of all local nodes to screen

-u URI / --server\_uri=URI - URI of server

rosparam <get>  
 <set>

<list>

<load>

<dump>

<delete>

***rqt***

***rviz***

## Main Services

ניתן להצגה ע"י rosservice list

rosservice call   
 /gazebo/pause\_physics  
 /gazebo/unpause\_physics

/gazebo/reset\_world

/gazebo/reset\_simulator

/gazebo/spawn\_gazebo\_model

/gazebo/spawn\_sdf\_model

/gazebo/spawn\_urdf\_model

/gazebo/delete\_model

## דוגמאות

rosrun gazebo\_ros spawn\_model -file ~/.gazebo/models/quadrotor/model.sdf -gazebo -model [name] -x [x] -y [y]

rosrun gazebo spawn\_model -file [urdf file] -urdf -model [model name]

rosservice call /gazebo/delete\_model '{model\_name: [name]}'

rosservice call /myquadrotor1/shutdown

rosservice call /myquadrotor2/engage

rosservice call /gazebo/set\_model\_state '{model\_state: {model\_name: [name], pose: { position: { x: 0, y: 0, z: 1}, orientation: {x: 0 ...} }, twist: { linear: {x: ..}, angular: { x: ...} }, reference\_frame: world } }'

rosservice call gazebo/get\_world\_properties

rosservice call gazebo/get\_model\_properties '{ model\_name: [name] }'

rostopic pub /wind geometry\_msgs/Vector3 -- '5' '0' '0'

rostopic pub -1 /turtule/cmd\_level geometery\_msgs/Twist -- '[2,0,0]' '[0,0,1]'

## ROSCPP

### אוסף פקודות

* ros::init

- register at core

- set up remappings

-set up networking

* ros::NodeHandle - interface to topics, services and parameters
* ros::NodeHandle::subscribe([Topic name], [queue size], [callback], <[transport hints]>
* ros::NodeHandle::advertise <[msg type]>([Topic name], [queue size],<IsLatch>)
* ros::ServiceServer service = [ros::NodeHandle].advertiseService
* ros::ServiceClient client = [ros::NodeHandle].serviceClient
* client.call
* ros::spin, ros::spinOnce
* ros::getGlobalCallbackQueue()->callAvailable(ros::WallDuration(0.1)); <= ros::spin
* ros::shutdown
* ros::ok
* <ros::Rate>.sleep()
* ROS\_INFO
* ROS\_ERROR
* boost::bind - use member function as callbacks

example : boost::bind (Listener::laserCb, this, -1);

### הודעות

\* הודעות יוגדרו בספריית [package]/msg/\*.msg

\* הודעה יכולה להכיל את הטיפוסים הבאים: int, float, string, time, duration, array

example: Point.msg

float64 x

float64 y

\* תהליך יצירת הודעה חדשה: <http://wiki.ros.org/msg>

\* ספריות של הודעות/שירותים ב ROS:

**opt/ros/indigo/share/**

common\_msgs - contains messages that are widely used by ROS :

- actionlib\_msgs -

- diagnostic\_msgs

- geometry\_msgs

- nav\_msgs

- sensor\_msgs

- shape\_msgs

-stereo\_msgs

-trajecoty\_msgs

-visualization\_msgs

gateway\_msgs

geographic\_msgs

gazebo\_msgs

hector\_nav\_msgs

hector\_uav\_msgs

map\_msgs

rosgraph\_msgs

tf2\_geometry\_msgs

### שירותים (services)

<http://wiki.ros.org/roscpp/Overview/Services>

שירותים יוגדרו בתוך ספריית [package]/srv/\*.srv

\* קריאה לservice:

ros::service::call([Service name], [request])

או

ros::ServiceClient client = nh.serviceClient<[msg type]>([Service name])

client.call ([server])

\* הפצה שירות

ros::ServiceServer nh.advertiseService(const std::string& service, callback>);

פונקציית הcallback תופעל עם בקשה לשירות. מבנה הפונקציה:

bool callback(MReq& request, MRes& response);

פונקציית הcallback יכולה להיות כפונקציה גלובלית, כפונקציה בתוך מחלקה או בתור operator() (Functor object)

\* ניתן להוסיף להוסיף מידע נוסף (metadata) לבקשה ובצורה זאת לנהל coockies בצד השרת. Service Connection Headers

### Parameter Server

<http://wiki.ros.org/roscpp/Overview/Parameter%20Server>

nh.getParam("/global\_name", global\_name));

ros::param::get("/global\_name", global\_name))

nh.param<std::string>("default\_param", default\_param, "default\_value");

nh.setParam("/global\_param", 5);

ros::param::set("/global\_param", 5);

nh.setParam("relative\_param", "my\_string");

nh.setParam("bool\_param", **false**);

nh.hasParam("my\_param"))

ros::param::has("my\_param"))

nh.deleteParam("my\_param");

ros::param::del("my\_param");

ros::param::get("~private\_name", param);

nh.searchParam("bar", key))

XmlRpc::XmlRpcValue my\_list;

nh.getParam("my\_list", my\_list);

### Timers

<http://wiki.ros.org/Clock>

<http://wiki.ros.org/roscpp_tutorials/Tutorials/Timers>

<http://wiki.ros.org/roscpp/Overview/Timers>

### \* שעון ROS

### ros::Timer ros::NodeHandle::createTimer(ros::Duration period, <callback>, bool oneshot = false)

מבנה ה-callback:

void callback(const ros::TimerEvent&);

מבנה ה TimerEvent:

ros::Time last\_expected -

ros::Time last\_real

ros::Time current\_expected

ros::Time current\_real

ros::WallTime profile.last\_duration

\* שעון אמיתי (זמן מערכת הפעלה)

ros::WallTimer timer = nh.createWallTimer(ros::WallDuration(0.1), callback);

\* פונקציות שעון:

ros::Time time = ros::Time::now();

ros::Duration d = ros::Duration(1, 0);

### serialization

<http://wiki.ros.org/roscpp/Overview/MessagesSerializationAndAdaptingTypes>