ROS作业

**ROS 乌龟画圆(C + +)**

1．ROS工作空间的建立

创建工作区

1. 打开Home目录，在空白处单击右键，然后单击New Folder创建一个名为catkin\_ws的文件夹

2. 在文件夹catkin\_ws中，创建文件夹src

3.在src目录下，终端输入catkin\_init\_workspace。CMakeLists.txt文件出现在src文件夹中

Code example:

$ mkdir -p ~/catkin\_ws/src

$ cd ~/catkin\_ws/src

$ catkin\_init\_workspace

在catkin\_WS文件夹中的终端输入catkin\_make。在catkin\_WS文件夹中生成build和devel文件夹。

$ cd ~/catkin\_ws/

$ catkin\_make

设置环境变量

设置环境变量是为了让系统知道函数包的位置，以便能够找到它。确保安装脚本正确地覆盖了工作空间ROS\_ PACKAGE\_ path环境变量包含在目录中。

在catkin\_ WS路径下，在终端上输入source devel/setup.bash，以让系统知道函数包放在这个工作区下的catkin\_ws中。

检查环境变量

查看当前环境变量，在terminal中输入echo $ROS\_ PACKAGE\_ PATH

$ source devel/setup.bash

$ echo $ROS\_PACKAGE\_PATH

2、ROS机器人画圆程序的编写

创建功能包

打开终端

$ cd catkin\_ws/src　　　　#Enter workspace

$ catkin\_create\_pkg my\_turtle\_package std\_msgs rospy roscpp

//Compile function pack (Gu Yueju video)

创建C++文件

In my\_turtle\_package/src 创建cpp文件

$ cd my\_turtle\_package/src/

$ touch draw\_circle.cpp

打开draw\_circle.cpp.

$ vi draw\_circle.cpp

--------------------------------------------------

#include "ros/ros.h"

#include<geometry\_msgs/Twist.h> // motion speed structure type geometry\_msgs::Twist definition file

int main(int argc, char \*argv[])

{

ros::init(argc, argv, "vel\_ctrl"); //Initialize the node

ros::NodeHandle n; //Declare a NodeHandle object n and generate a broadcast object vel with n\_pub

ros::Publisher vel\_pub = n.advertise<geometry\_msgs::Twist>("/turtle1/cmd\_vel", 10);

//vel\_pub will broadcast geometry in the topic "/ cmd\_vel" (robot speed control topic)\_ Msgs:: twist type data

ROS\_INFO("draw\_circle start...");//Output display information

while(ros::ok())

{

geometry\_msgs::Twist vel\_cmd; //Declare a geometry\_ Msgs:: object vel of type twist\_ CMD and assign the value of speed to this object

vel\_cmd.linear.x = 2.0; //Front and rear (+ -) m/s

vel\_cmd.linear.y = 0.0; //Left and right (+ -) m/s

vel\_cmd.linear.z = 0.0;

vel\_cmd.angular.x = 0;

vel\_cmd.angular.y = 0;

vel\_cmd.angular.z = 1.8; //Rotation speed of robot, + left turn, - right turn, unit: rad/s

vel\_pub.publish(vel\_cmd); //After assignment, it is sent to the topic "/ cmd\_vel". The core node of the robot will receive and send the past speed value from this topic and forward it to the hardware body for execution

ros::spinOnce();//Calling this function allows other callback functions to be executed (more than the routine does not use callback functions)

}

return 0;

}

修改相关文件

在终端中继续输入如下命令:

$ cd ~/catkin\_ws/src/my\_turtle\_package

$ gedit CMakeLists.txt #Open CMakeLists.txt

在CMakeLists.txt文件中找到##Declare a C++ executable，并在这行之后添加以下内容:

## Declare a C++ executable

## With catkin\_make all packages are built within a single CMake context

## The recommended prefix ensures that target names across packages don't collide

# add\_executable(${PROJECT\_NAME}\_node src/my\_turtle\_package\_node.cpp)

add\_executable(draw\_circle src/draw\_circle.cpp)

target\_link\_libraries(draw\_circle ${catkin\_LIBRARIES})

编译

保存并退出CMakeLists.txt文件。然后在terminal中继续输入以下命令进行编译:

$ cd ~/catkin\_ws/

$ catkin\_make

编译成功。

**3．运行**

1. 打开第一个终端并启动ros

$ roscore

2. 打开第二个终端并启动rosnode

$ rosrun turtlesim turtlesim\_node

3. 打开第三个终端并启动my\_turtle\_package节点

$ cd ~/catkin\_ws/

$ source devel/setup.bash

$ rosrun my\_turtle\_package draw\_circle

小乌龟开始画一个圈。按Ctr+C退出节点，小海龟停止。

实验内容：

1. 理解ROS基本概念。
2. 按照上面的例子复现。（3分）
3. 自行实现小乌龟的运行轨迹。（满分5分）

提交：

单独完成，提交含term用户名（Ubuntu用户名为各自姓名的汉语拼音）的界面截图，cpp源码以及截屏视频。

格式为学号姓名.zip/rar。

注：截屏视频不大于10M，超过视情况扣分。。。

Ref.

http://wiki.ros.org/cn/ROS/Tutorials/CreatingPackage