EE 3EY4
Lab 2

-

Stefan Tosti - Tostis - 400367761 - L05
Luke Hendrikx - Hendrikl - 400388795 - L05
Soham Patel - Pates206 - 400380121 - L05

2024 - 01 - 26

Question 1

`std_msgs:` This package provides standard message types like int8, int64 and string (among others). These identifiers are used to define the data fields in custom messages. When creating a package, including this dependency will also give us the ability to send primitives as message types.

`roscpp:`This is the C++ library for ROS, it gives the package the ability to interface with C++ programs, as well as provides the functionality for creating publisher and subscriber nodes in C++

`rospy:` This is the Python library for ROS, it gives the package the ability to interface with Python programs, as well as provides the functionality for creating publisher and subscriber nodes in Python

Question 2

`Sudo:` Stands for "Super User Do", it is used to execute a command with Super User privileges to run administrative commands

`Apt-get:` This command is used for handling packages, it allows us to search for, manage, update and remove pieces of software on our device

`Install:` This is an argument used with *apt-get* that allows us to install a provided package

`Ros-melodic-serial:` This is a ROS package that provides a library to support serial port communication between computers using C++ RS-232

`Ros-melodic-ackermann-msgs:` This is a ROS package that provides messages for our vehicle, which uses front wheel Ackermann steering

`Ros-melodic-rplidar:` This is a ROS package that provides drivers for RPLiDAR, the LiDAR system being used on the robot

`Ros-melodic-realsense2-camera:` This is a ROS package that provides nodes for using the Intel T265 tracing module, as well as the SR300 and D400 camera

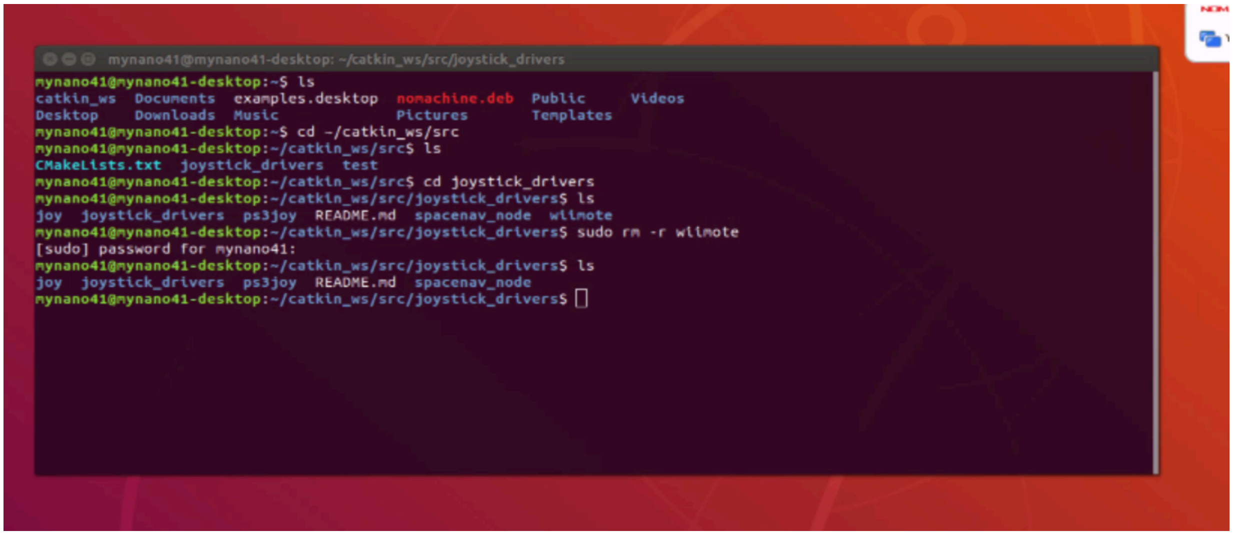
`Libusb-dev:` This is a library for programming USB applications without knowledge of the Linux kernel internals (I.e. provides generic access to USB devices)

`Libspnav-dev:` This is a library for communicating with 6 degree-of-freedom devices (3D input devices)

Question 3

Command used: `sudo rm -r wiimote`

We are recursively removing all of the files in the wiimote directory, found in the joystick_drivers directory that we downloaded from Github. We need to use sudo in order to delete the wiimote as a super user.



```
mynano41@mynano41-desktop:~$ ls
catkin_ws  Documents  examples.desktop  nomachine.deb  Public     Videos
Desktop    Downloads  Music             Pictures       Templates
mynano41@mynano41-desktop:~$ cd ~/catkin_ws/src
mynano41@mynano41-desktop:~/catkin_ws/src$ ls
CMakeLists.txt  joystick_drivers  test
mynano41@mynano41-desktop:~/catkin_ws/src$ cd joystick_drivers
mynano41@mynano41-desktop:~/catkin_ws/src/joystick_drivers$ ls
joy  joystick_drivers  ps3joy  README.md  spacenav_node  wiimote
mynano41@mynano41-desktop:~/catkin_ws/src/joystick_drivers$ sudo rm -r wiimote
[sudo] password for mynano41:
mynano41@mynano41-desktop:~/catkin_ws/src/joystick_drivers$ ls
joy  joystick_drivers  ps3joy  README.md  spacenav_node
mynano41@mynano41-desktop:~/catkin_ws/src/joystick_drivers$ 
```

Question 4

Command used: `catkin_make`

We need to build the previously installed packages because without building the packages, the packages can not be used. Building the packages allows for the packages to be used by the software.

Question 5

Adding this line to the *.bashsrc* file, we allow the *setup.bash* to be sources when opening a new shell. This means that the *catkin_ws* workspace is automatically setup in every new shell, allowing us to use the packages without manually sourcing *setup.bash* each time.

Question 6

`/rosout`: Standard ROS topic to which all ROS nodes publish their log messages. The *rosout* node subscribes to this topic, records these messages in a log file, and then sends these messages to *rosout_agg* which allows for system wide message logging

`/rosout_agg`: This is an aggregated feed published to *rosout.* Instead of connecting to individual ROS nodes to receive console logs, the aggregated feed can be received directly from the *rosout* node

Question 7

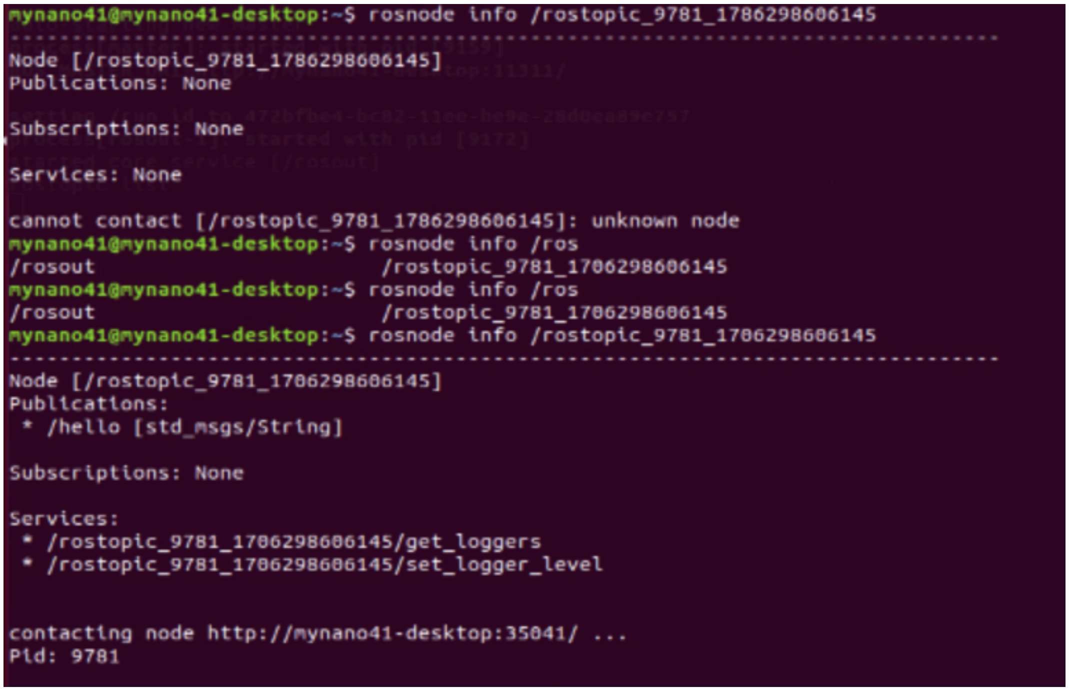
`rostopic pub:` This command instructs ROS to publish a new topic
`/hello:` This is the name of the new topic that we want to publish
`std_msgs/String:` This is the type of our new topic, in our case it is a standard ROS message that represents a string data type
`“Hello Robot”:` This is the actual message to be held in the topic, in our case it is the given string

Question 8

1.) In this objective we started up the Robot Operating System (ROS) using *roscore*.
2.) After this, we created a topic that published our string “Hello Robot” using *rostopic pub /hello std_msgs/String “Hello Robot"*.
3.) Then, we printed the data from the topic onto the terminal using *rostopic echo /hello*.
4.) We then used *rosnode list* to figure out which node was being used to publish the topic
5.) After we found the node, we used *rosnode info /rostopic_.....* to find out information about the node.
6.) An alternative approach to step 5.) would be *rostopic info /hello*.

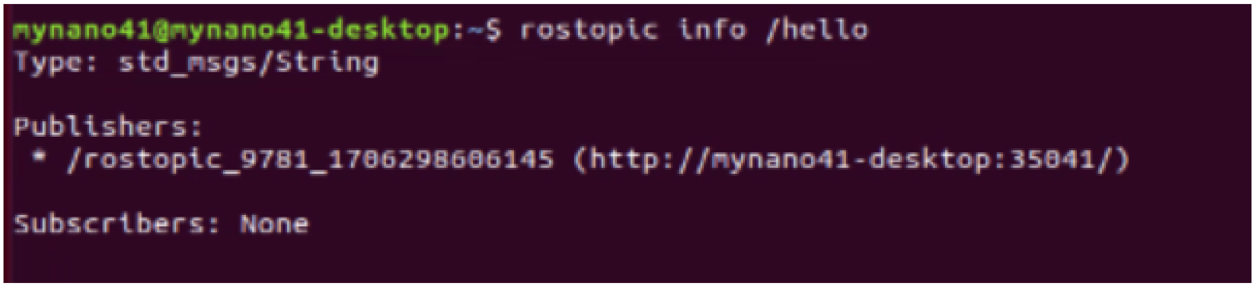
From this objective we learned about nodes and topics in ROS. We familiarized ourself with the various commands that can be used to boot up ROS, create a topic, print data from a topic, and list nodes.

The output of *rosnode info /rostopic/ ...*



```
mynano41@mynano41-desktop:~$ rosnode info /rostopic_9781_1786298606145
--------------------------------------------------------------------------------
Node [/rostopic_9781_1786298606145]
Publications: None

Subscriptions: None

Services: None

cannot contact [/rostopic_9781_1786298606145]: unknown node
mynano41@mynano41-desktop:~$ rosnode info /ros
/rosout                          /rostopic_9781_1706298606145
mynano41@mynano41-desktop:~$ rosnode info /ros
/rosout                          /rostopic_9781_1706298606145
mynano41@mynano41-desktop:~$ rosnode info /rostopic_9781_1706298606145
--------------------------------------------------------------------------------
Node [/rostopic_9781_1706298606145]
Publications:
 * /hello [std_msgs/String]

Subscriptions: None

Services:
 * /rostopic_9781_1706298606145/get_loggers
 * /rostopic_9781_1706298606145/set_logger_level


contacting node http://mynano41-desktop:35041/ ...
Pid: 9781
```

The output for *rostopic info /hello*



```
mynano41@mynano41-desktop:~$ rostopic info /hello
Type: std_msgs/String

Publishers:
 * /rostopic_9781_1706298606145 (http://mynano41-desktop:35041/)

Subscribers: None
```