ELECENG 3EY4: Electrical System Integration Project

Lab02_Jetson Nano and ROS

Junbo Wang – wangj430 – 400249823

Preet Batra - batrap5 - 400341704

Yichen Lu - luy191 - 400247938

Question 1: Why do we need the dependencies std_msgs, roscpp, and rospy when creating a ROS package?

When creating a ROS (Robot Operating System) package, we need to include certain dependencies to enable communication, data serialization, and other functions in the ROS system. std_msgs provides a set of standard message types commonly used for ROS communication. It indicates that we will use standard message types such as int 8, int 64, string or float. roscpp indicates usage of C++ code. rospy indicates usage of Python code.

Question 2: Explain each line in the command that you used in the terminal: sudo, apt-get, install,ros-melodic-serial, ros-melodic-ackermann-msgs, ros-melodic-rplidar-ros, ros-melodic-realsense2-camera, libusb-dev, libspnav-dev.

sudo: sudo is super user do, which allows executing programs as a superuser

apt-get: apt is known as Advanced Packaging Tool. It is an integrated tool that can be used to install, update, remove and manage software packages on Debian and Ubuntu.

install: a command to install the named package.

ros-melodic-serial: This is a ROS package for the Melodic Morenia distribution, containing serial communication functionality.

ros-melodic-ackermann-msgs: Melodic Morenia's software package can provide ROS messages for vehicles using front-wheel Ackermann steering.

ros-melodic-rplidar-ros: This package is an integration of RPLIDAR with the ROS system.

ros-melodic-realsense2-camera: This package provides a ROS node using the Intel RealSense camera.

libusb-dev: It is the development package of "libusb". "libusb" is a library for USB device access and can compile programs.

libspnav-dev: It is the development package of "libspnav". "libspnav" is a library for interfacing with 3Dconnexion's Space Navigator device.

Question 3: What do you need to do to remove the Wiimote driver? Type the commands in the terminal and confirm with your TA that you have the correct command written down before executing (Hint: use your Linux knowledge from Lab 1). In your report, explain which command you used and why.

The reason for removing the Wiimote driver is because we do not have the library to run it. More specifically, when we clone the repository from GitHub above, a Wiimote driver package will be cloned to the Jetson Nano by default. Therefore, we need to get rid of it.

We use "sudo rm -r wiimote" to remove the Wiimote driver. To delete this folder, firstly, we need to use sudo, which allows executing programs as a superuser. Afterwards, we need to use the “rm -r” command in Linux. The “-r” command represents deleting something recursively, which allows us to fully delete the files inside a directory before deleting the directory itself.

Question 4: In your report, explain which command you used and why you need to build the packages installed earlier.

The command we used is “catkin_make” to build the package. We need to run catkin_make to compile the source code and any dependencies we have. In order to build a package, we must have a catkin workspace containing all projects. Therefore, the installation package needs to be built in advance.

Question 5: Why do you need to edit the .bashrc file in the last step of Objective 5? Explain in detail.

The “setup.bash” script is responsible for setting up the environment of the ROS workspace. Adding the line "source ~/catkin_ws/devel/setup.bash" to the ".bashrc" file lets the system know where my workplace is and ensures that it is automatically set to the ROS environment every time I open a new terminal.

Question 6: What is the purpose of the entries /rosout and /rosout_agg listed in Figure 20? Explain in detail.

The purpose of the /rosout entry is to publish console log messages to the /rosout topic as a standard interface, while the /rosout_agg entry is used to subscribe to an aggregate feed of console log messages that can be received directly from the rosout node.

Question 7: Break down the parts of the command in Figure 21 and explain each function: rostopic pub, /hello, std_msgs/String, “Hello Robot”.

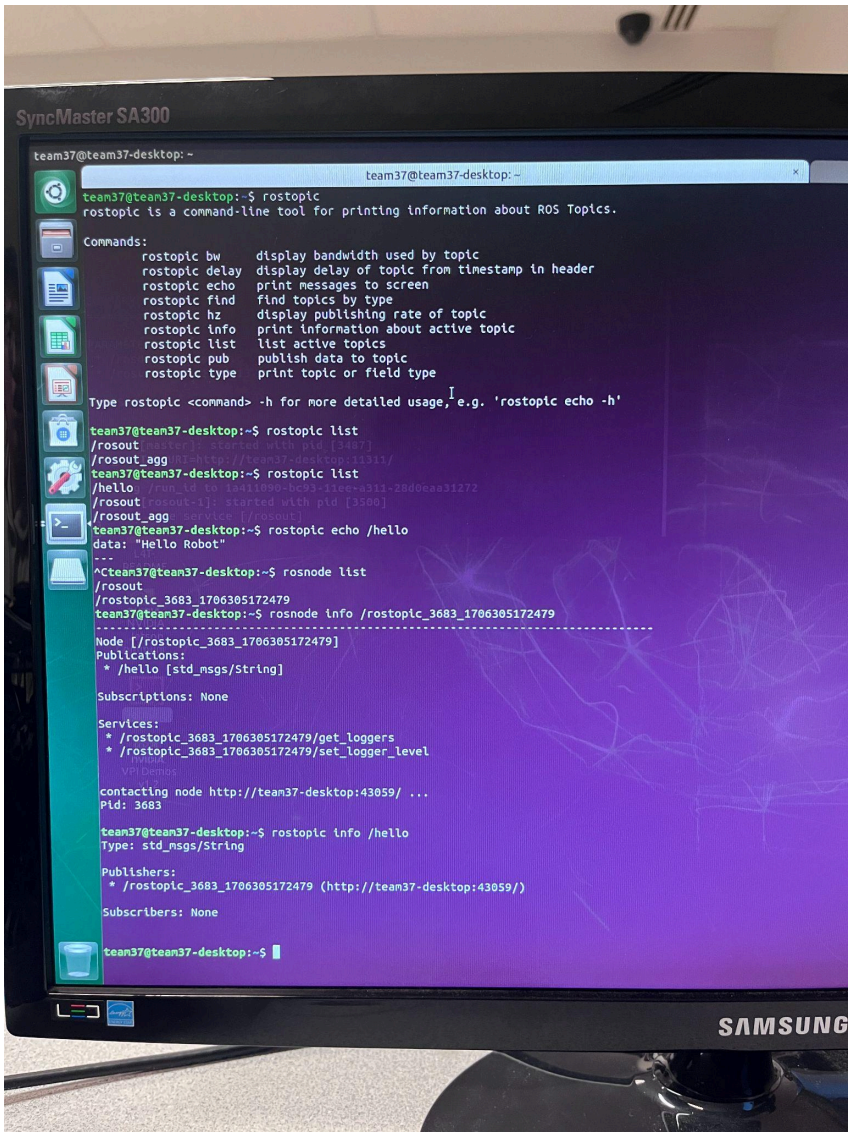
rostopic pub: rostopic pub is used to publish a ROS topic. It allows users to manually publish messages to topics that can be subscribed to by other ROS nodes.

/hello: /hello is the name of the ROS topic to which the message will be published. In ROS, a topic is a channel through which nodes communicate with each other.

std_msgs/String: "std_msgs/String" specifies the message type published to the topic. "std_msgs" is a standard ROS package containing common message types. "String" represents a string of characters, which can be any text.

"Hello Robot": It is the actual message published to the "/hello" topic, represented as a string literal.

Question 8: Write and explain in your own words a summary of what you did in this objective and what you learned from it. Include screenshots of the output of the rosnode info/rostopic/_.... command and of the rostopic info /hello command.



In objective 6, we open a new terminal window by pressing the keys (Ctrl + Alt + T) and enter the roscore command, which will start ROS Master. After that, we enter the rostopic list command, which will display the entries /rosout and /rosout_agg, which are built-in functions of ROS for reporting and collecting error messages. In another new terminal, we publish a ROS topic using rostopic pub and use the entire command rostopic pub /hello std_msgs/String “Hello Robot” to send messages to other nodes in the ROS system. If we want to see the message in rostopic, we can enter rostopic echo

/hello to successfully publish the topic and receive the message. Then to stop viewing this topic, we press Ctrl + C.

When we want to find the node that published the message, we will use the rosnode list command. Since we published the /hello topic in rostopic, the ROS system will continue to create nodes. To view node details, enter rosnode info /rostopic_3683_1706305172479. Alternatively, we can type rostopic info /hello to get the same information for the /hello topic.