Robotics and AI Department

Task 1(robot-arm-ros)

The steps below uses ROS melodic , 1.14.11 version which runs on ubuntu 18.04.5.

Installing ROS melodic

The following steps assumes that the user have Ubuntu 18.04 installed.

Setup the system to accept software from packages.ros.org.

```
$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -
sc) main" > /etc/apt/sources.list.d/ros-latest.list'
```

Setup the keys:

```
$ curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc |
sudo apt-key add –
```

Before the installation, first check whether the Debian package index is up-to-date:

```
$ sudo apt update
```

Desktop-Full Installation command:

```
$ sudo apt install ros-melodic-desktop-full
```

To automatically add ROS environment variables to your bash session every time a new shell is launched:

```
$ echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc
$ source ~/.bashrc
```

Install and initialize system dependencies in ROS:

```
$ sudo apt install python-rosdep
$ sudo rosdep init
$ rosdep update
```

To start the ROS (master node)

```
$ roscore
```

Preparing ROS

Setup the workspace which is where ROS projects are built and stored:

```
$ mkdir -p ~/catkin_ws/src
$ cd ~/catkin_ws/
$ catkin_make
```

Where `catkin_ws` is the name of the workspace, and `catkin_make` is used to the build the project and packages inside the source folder.

```
$ echo "source ~/catkin_ws/devel/setup.bash" >> ~/.bashrc

$ source ~/.bashrc
```

Robot arm package

Adding `arduino_robot_arm` package to `src` folder:

```
$ cd ~/ catkin_ws /src
$ sudo apt install git
$ git clone https://github.com/smart-methods/arduino_robot_arm
```

Dependencies

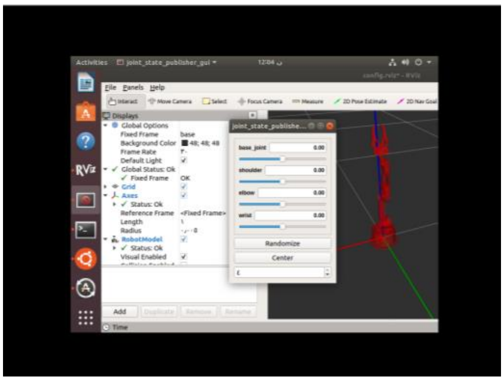
```
$ cd ~/ catkin_ws
$ rosdep install --from-paths src --ignore-src -r -y
$ sudo apt-get install ros-melodic-moveit
$ sudo apt-get install ros-melodic-joint-state-publisher ros-melodic-joint-state-publisher-gui
$ sudo apt-get install ros-melodic-gazebo-ros-control joint-state-publisher
$ sudo apt-get install ros-melodic-ros-controllers ros-melodic-ros-control
```

Compilation

```
$ catkin_make
```

Controlling the robot arm using joint_state_publisher:

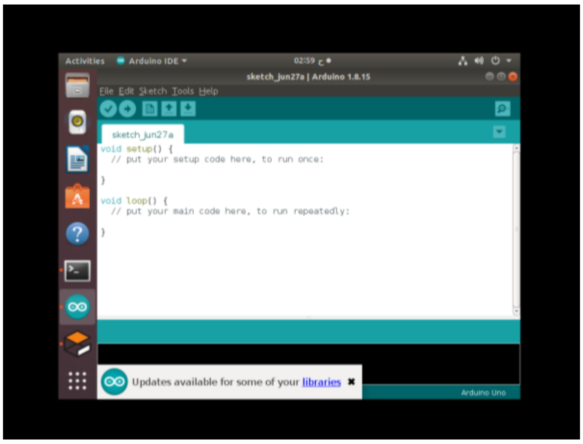
```
$ roslaunch robot_arm_pkg check_motors.launch
```



Global Options
Fixed Frame: base
Background Color: 48; 48; 48
Default Light: ✓
Global Status: Ok
Fixed Frame: OK
Grid: ✓
Axes: ✓
Status: Ok
Reference Frame: <Fixed Frame>
RobotModel: ✓
Status: Ok
Visual Enabled: ✓

base_joint 0.00
shoulder 0.00
elbow 0.00
wrist 0.00

Arduino IDE Ubuntu



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Updates available for some of your libraries

1- Install rosserial for Arduino:

```
$ sudo apt-get install ros-melodic-rosserial-arduino
$ sudo apt-get install ros-melodic-rosserial
```

2- Install ros_lib into the Arduino environment:

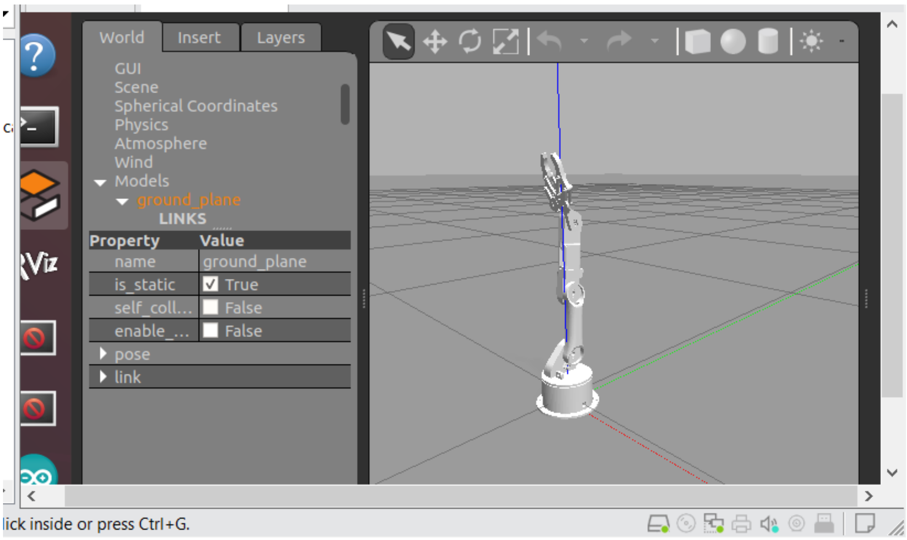
```
$ cd ~/Arduino/libraries
$ rm -rf ros_lib
$ rosrun rosserial_arduino make_libraries.py .
```

`Arduino` is the directory where the Linux Arduino environment saves the sketches.

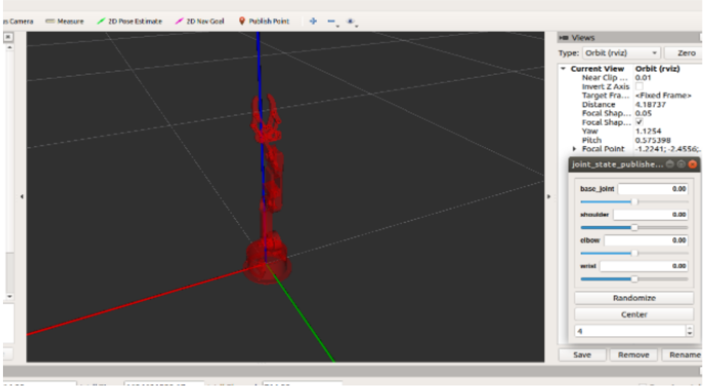
3- Upload the Arduino code.

Start simulating the robot arm using RViz and Gazebo; the motors are controlled via joint_state_publisher, as can be seen below:

```
$ roslaunch robot_arm_pkg check_motors.launch
$ roslaunch robot_arm_pkg check_motors_gazebo.launch
$ rosrun robot_arm_pkg joint_states_to_gazebo.py
```



Property	Value
name	ground_plane
is_static	☑ True
self_coll...	□ False
enable_...	□ False



Type: Orbit (rviz) Zero

Current View	Orbit (rviz)
Near Clip ...	0.01
Invert Z Axis	□
Target Fra...	<Fixed Frame>
Distance	4.18737
Focal Shap...	0.05
Focal Shap...	✓
Yaw	1.1254
Pitch	0.575398
Focal Point	-1.2241; -2.4556;.

base_joint	0.00
shoulder	0.00
elbow	0.00
wrist	0.00

4

MoveIt in RViz

MoveIt Used for kinematics, motion planning, trajectory processing and controlling the robot

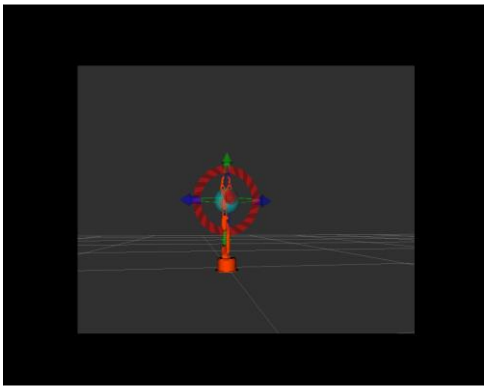
```
$ roslaunch moveit_setup_assistant setup_assistant.launch
```

To run the robot arm using MoveIt package:

```
$ roslaunch moveit_pkg demo.launch
```

To launch MoveIt with Gazebo simulator use the following command:

```
$ roslaunch moveit_pkg demo_gazebo.launch
```



Connecting with Arduino:

```
$ roslaunch moveit_pkg demo.launch
$ rosrun rosserial_python serial_node.py _port:=/dev/ttyUSB0 _baud:=115200
```