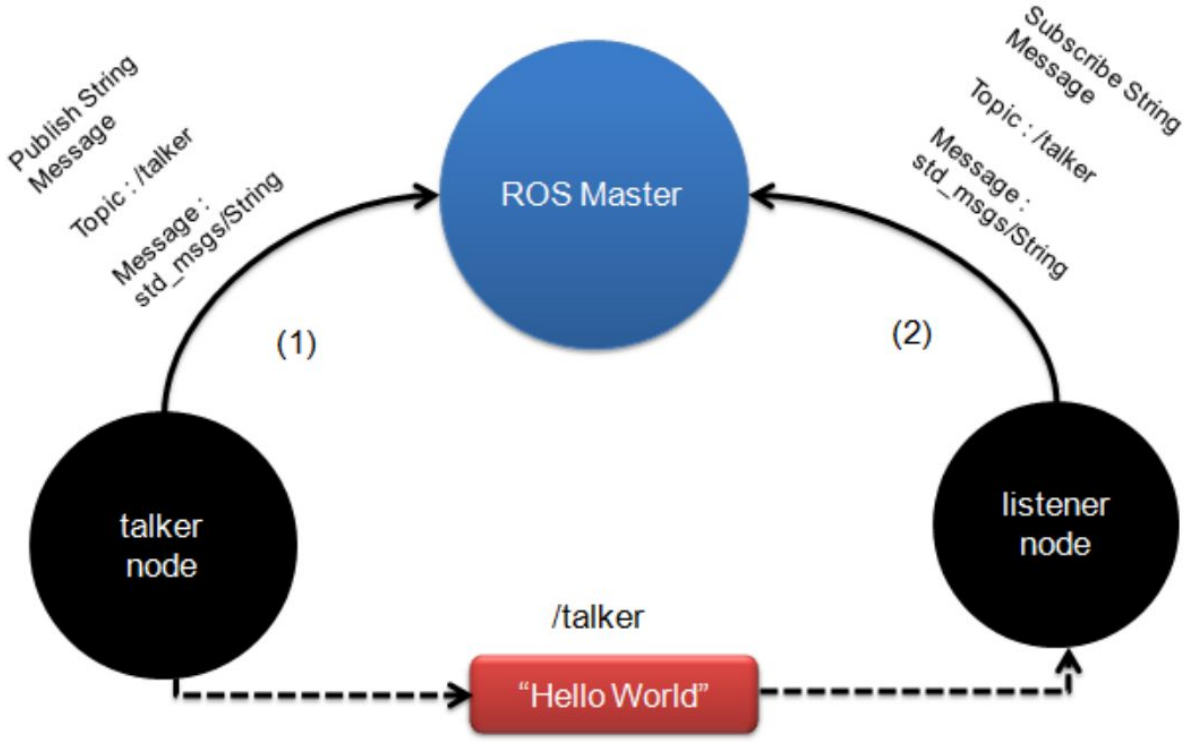
Robotics Lab

ROS(Robot Operating System)

Publisher and Subscriber



Creating a ROS Workspace

To create a ROS Workspace with the name catkin_ws in the home directory,

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

Creating a ROS Package

To create a ROS Package in the previously created workspace we will use **catkin_create_pkg** script.

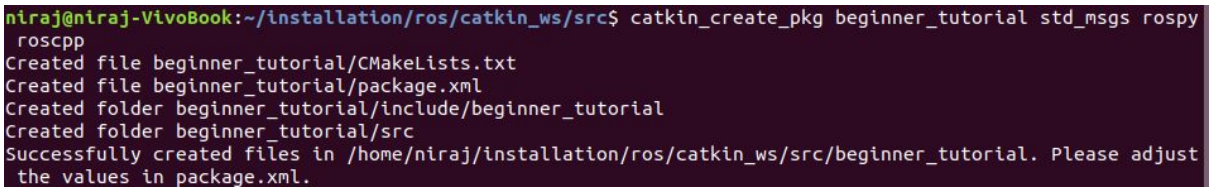
1. Change to source space directory of the workspace

```
$ cd ~/catkin_ws/src
```

2. Use **catkin_create_pkg** script to create a new package '**beginner_tutorials**' which depends on std_msgs, roscpp and rospy
   The syntax to create package is:
   catkin_create_pkg <package_name> [depend1] [depend2] [depend3]...

```
$ catkin_create_pkg beginner_tutorials std_msgs rospy roscpp
```

   This will create a **beginner_tutorials** folder which contains a **package.xml** and a **CMakeLists.txt** which have been partially filled.



3. Build the catkin workspace.

```
$ cd ~/catkin_ws
$ catkin_make
```

4. Source the setup file

```
$ source ~/catkin_ws/devel/setup.bash
```

The package **beginner_tutorials** is created successfully.

Publisher Node

(https://github.com/nirajpahari/ros-basics/blob/master/catkin_ws/src/beginner_tutorials/scripts/talker.py)
Generally all the python scripts are stored in the ‘scripts’ folder in the package.
Let's make a scripts folder in our package:

```
$ roscd beginner_tutorials
$ mkdir scripts
$ cd scripts
```

Now let's create a python file inside the scripts folder and write the publisher code.

Create a new .py file named **talker.py** in the scripts folder and add the following code.

```
#!/usr/bin/env python

import rospy
from std_msgs.msg import Int16

# to randomise the integer message
import random


def talker():

   # Create a publisher object -> Publisher(<topic name>, <message type>, queue_size)
   publisher = rospy.Publisher("talker", Int16, queue_size=10)

   # Initialize the ros node -> init_node(<node name>)
   rospy.init_node("talker_node")

   # publisher rate in Hz
   rate = rospy.Rate(10)

   # Publish until the node is not closed
```

```
    while not rospy.is_shutdown():

        message = random.randint(0, 5)

        # publish the message
        publisher.publish(message)

        # Logging the info in the console
        rospy.loginfo(message)
        rospy.loginfo("Message published successfully")

        # Wait until the rate defined
        rate.sleep()


if __name__ == '__main__':
    try:
        talker()
    except rospy.ROSInterruptException:
        pass
```

Subscriber Node

(https://github.com/nirajpahari/ros-basics/blob/master/catkin_ws/src/beginner_tutorials/scripts/listener.py)

Create a new .py file named **listener.py** in the scripts folder and add the following code.

```
#!/usr/bin/env python

import rospy
from std_msgs.msg import Int16

# This function is called when the subscriber gets the message from
the publisher
```

```
def callback(data):

   rospy.loginfo("Printing the data sent by publisher")
   rospy.loginfo(data.data)


def listener():

   # Declare the subscriber to subscribe the topic talker which
publishes integer
   # message
   rospy.Subscriber('talker', Int16, callback)

   # Initialize the ros node for subscriber
   rospy.init_node("listener_node")

   rospy.loginfo("Listening...")

   # This command keeps python from existing until the node is
stopped
   rospy.spin()


if __name__ == '__main__':
   listener()
```

Make the .py files executable

```
$ chmod +x talker.py
$ chmod +x listener.py
```

Building the nodes

Go to the catkin workspace and run catkin_make

```
$ cd ~/catkin_ws
$ catkin_make
```

Now source the devel space of the workspace

```
$ source devel/setup.bash
```

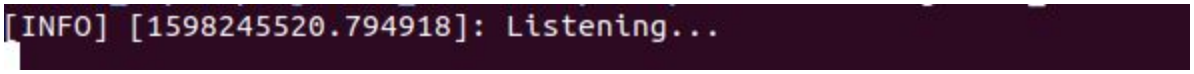
Starting the nodes

Start the ROS master

```
$ roscore
```

Now run the subscriber and listener nodes in separate terminal tabs

```
$ rosrun beginner_tutorials listener.py
```




```
$ rosrun beginner_tutorials talker.py
```

```
[INFO] [1598245612.392616]: Message published successfully
[INFO] [1598245612.492944]: Message published successfully
[INFO] [1598245612.593119]: Message published successfully
[INFO] [1598245612.693036]: Message published successfully
[INFO] [1598245612.793412]: Message published successfully
[INFO] [1598245612.893075]: Message published successfully
[INFO] [1598245612.993345]: Message published successfully
[INFO] [1598245613.093369]: Message published successfully
[INFO] [1598245613.193595]: Message published successfully
[INFO] [1598245613.293327]: Message published successfully
[INFO] [1598245613.393866]: Message published successfully
[INFO] [1598245613.493017]: Message published successfully
[INFO] [1598245613.593278]: Message published successfully
```

If you check the listener node again, now it receives the message published by publisher

```
[INFO] [1598245648.329453]: Listening...
[INFO] [1598245650.987062]: Printing the data sent by publisher
[INFO] [1598245650.990696]: Hello this is the message from talker
[INFO] [1598245651.087422]: Printing the data sent by publisher
[INFO] [1598245651.091198]: Hello this is the message from talker
[INFO] [1598245651.187354]: Printing the data sent by publisher
[INFO] [1598245651.190797]: Hello this is the message from talker
[INFO] [1598245651.286964]: Printing the data sent by publisher
[INFO] [1598245651.291594]: Hello this is the message from talker
[INFO] [1598245651.386919]: Printing the data sent by publisher
[INFO] [1598245651.391670]: Hello this is the message from talker
[INFO] [1598245651.486543]: Printing the data sent by publisher
[INFO] [1598245651.490127]: Hello this is the message from talker
```

Exercise:

1. Create a subscriber that subscribes to a topic “name_talk” which is of the type String.
2. Create a publisher to publish to a topic “name_talk”. Publish your name as a string message.

Tips

Checklist for creating a basic publisher

1. Import rospy and required message types

2. Create a function with same name as the filename
3. Create a publisher object with topic to be published, message type for the topic and queue size
4. Initialize the rospy node for publisher
5. Define the rate of publishing data
6. Loop until shutdown
7. Publish the message
8. Wait until the rate defined
9. Create the main function to run the above publisher function and catch ROSInterruptException

Checklist for creating a basic subscriber

1. Import rospy and required message types
2. Create a callback function ie. the function that will be called when the subscriber receives the message
3. Create a subscriber function with same name as the filename
4. Create a subscriber object with the topic to subscribe, message type for the topic and the name of callback function
5. Initialize the rospy node for subscriber
6. Use .spin() function to keep the node from exiting
7. Create the main function to run the subscriber function defined above.