Ros2 Learning Document

First session: 3 Oct. 2024.

Notes:

Nodes:

You can start a talker node (sends data) and another listener node (receives data). The data flows between nodes.

A node is program you run (python, c++, etc..)

“rqt_graph” shows u graph of ur nodes and stuff

Nodes can do MORE than send and receive data.

Nodes seems to be programs doing they own ting (they can do basically anything, host web servers show graphs blah blah blah)

Creating nodes:

Ros2 workspace installing global stuff and custom ros2 workspace stuff https://www.youtube.com/watch?v=3GbrKQ7G2P0&list=PLLSegLrePWgJudpPUof4-nVFHGkB62Izy&index=3

Creating packages:

You throw nodes in packages?

-navigation packages

-camera packages

-etc..

Packages can depend on each other

Packages are either python or c++ (nodes written in python or c++)

Creating packages:

Written in Python (or c++) using OOP.

Topics:

Topics are ways for nodes to communicate. So basically nodes send data to a topic, and every node subscribed to that topic can hear whatever is sent, its like a group chat

Service:

Request and response operation

Used for computation for example

Send data from lidar node to processor node compute shits etc...

Questions:

- What is turutlesim?
- What is colcon? Also argcomplete/autocomplet
- What is bash script?
- What is mkdir command? Make folder
- What is ls command
- What is workspace gonna look like with actual hardware?
- Ask gpt, what exactly am I gonna do with ros ?
- Should I write packages or nodes with python or c++
- What packages am I gonna use?
- Topics vs services?

Second session: 9 Oct. 2024.

Notes:

To start working:

1. Setup workspace:
   a) Create file (mkdir) as ros workspace
   b) In the file create another file called src
   c) In the src file we write our nodes (nodes in packages)
   d) Then you write “colcon build” in the ws file (this like hiring a secretary or smth)
   e) Then u gedit .bashrc and copy the install setup.bash directory.

2. Create packages:
   a) Go to the src file
   b) Create packages “ros2 pkg create pkg_name –build-type ament_pythin –dependencies rclpy” rclpy is ros2 py library
   c) “code .” to go to the package py file u just created
   d) You create the nodes inside the folder that has the same name as the package but the one that is nested
   e) You go back to ws folder and you colcon build to register pkg

   (creating nodes in next session)

Questions:

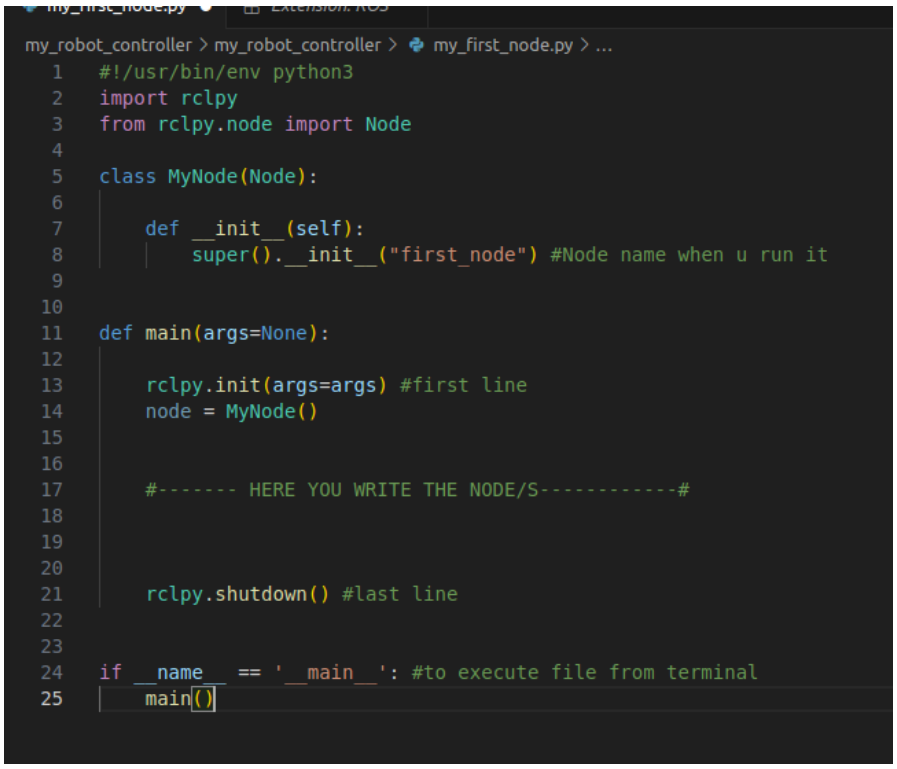
- why u gedit .bashrc and copy the install setup.bash directory. And each time?
- When to source ~/.bashrc ? when to colcon build if u have symlink?

Third session: 10 Oct. 2024.

Creating nodes:

a. We already have a pkg in src file (python pkg)
b. go inside the pkg
c. then inside the nested file of same pkg name
d. “touch file.py” to create the node
e. then u make the file executable “chmod +x file.py”
f. at top of the py code u write “#!/usr/bin/env python3” to tell interpreter to use py3

g. import the py ros library “import rclpy”

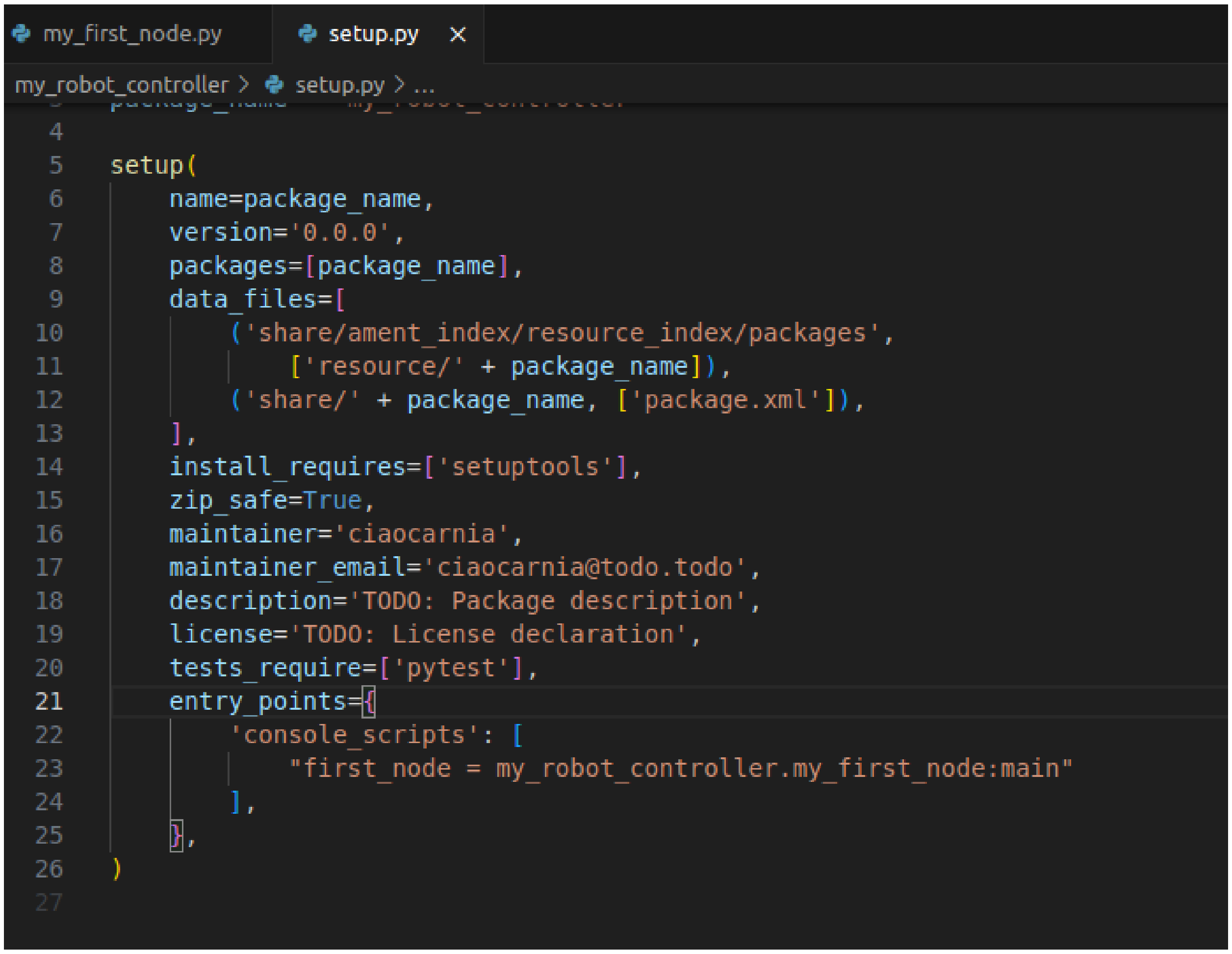


my_robot_controller > my_robot_controller > my_first_node.py > ...

```python
#!/usr/bin/env python3
import rclpy
from rclpy.node import Node

class MyNode(Node):

    def __init__(self):
        super().__init__("first_node") #Node name when u run it


def main(args=None):

    rclpy.init(args=args) #first line
    node = MyNode()


    #------- HERE YOU WRITE THE NODE/S-----------#



    rclpy.shutdown() #last line


if __name__ == '__main__': #to execute file from terminal
    main()
```

e. to functionalize the node so it runs in ros/using ros functions we need to install it now

h. you got to setup.py and write this in the console script



```python

setup(
    name=package_name,
    version='0.0.0',
    packages=[package_name],
    data_files=[
        ('share/ament_index/resource_index/packages',
            ['resource/' + package_name]),
        ('share/' + package_name, ['package.xml']),
    ],
    install_requires=['setuptools'],
    zip_safe=True,
    maintainer='ciaocarnia',
    maintainer_email='ciaocarnia@todo.todo',
    description='TODO: Package description',
    license='TODO: License declaration',
    tests_require=['pytest'],
    entry_points={
        'console_scripts': [
            "first_node = my_robot_controller.my_first_node:main"
        ],
    },
)

```

k. then you “colcon build” then “src ~/.bashrc”

Fourth session: Getting SLAM A3 RPLIDAR up and running

Prerequisite:

-install ros2

-run `sudo usermod -a -G dialout` **and restart pc.**

1- **Create a ros2 workspace**

```
mkdir -p ~/ros2_ws/src

cd ~/ros2_ws/src
```

2- **Source ros2 setup for the workspace (so you can run ROS2 commands):**

a. Open bashsrc file

```
nano  ~/.bashsrc
```

b. Source ROS2 for the workspace by adding this line in the end (don't forget to save)

```
source /opt/ros/foxy/setup.bash
```

3- **Clone the package into the src file**

```
git clone https://github.com/Slamtec/sllidar_ros2.git
```

4- **Go back to the workspace**

```
cd ~/ros2_ws
```

5- **confirm workspace build (ig each time you edit the ws u write this)**

```
colcon build --symlink-install
```

6- **Source the setup file (this time the workspace file not the ros2 setup file)**

```
source install/setup.bash
```

Note: If you don’t want having to source the workspace each time then access the bashrc file and write this at the end:

```
Source  ~/<work space name>/install/setup.bash
```

After this make sure to run the basrc file (source it):

```
source ~/.bashrc
```

7- Go back to workspace root and run the command

```
ros2 launch sllidar_ros2 view_sllidar_a3_launch.py
```

Questions:

- Why do we keep sourcing files (workspace, ros2), what is “sourcing” exactly? How do I have to source the ros2 file for a specific workspace into the bashrc file so the workspace doesn’t need ros2 source every terminal? What about the workspace file? The source install setup bash one.

Answer:

- We source the ros2 file `source /opt/ros/foxy/setup.bash` to be able to run ros2 commands (telling terminal this is a ros session)
- We source the workspace file because we need to idk what to tell you