ROS2

Introduction

ROS2 is the opensource software development frameware or the middleware for the robotics platform.It is developed for fast prototyping and research project in various educational and research propose.The purpose of migrating from ros1 to ros2 are:

1. Now the update for the ROS1 will be stopped by 2025 so we can't find the additional functionality which we may want later after 2025.
2. As ROS1 is deprived of various security issues, we cannot get the strong security connection between two devices which may run ros1 independently. So, ROS2 provide all these functionality as it is using DDS as network protocols for all communications compared to UDP/TCP protocal integrated on ROS1.
3. ROS2 doesn't need the mediater like ros master as in ROS1 due to which we can build several connections over the network which can't be hampered by the ros master.
4. ROS2 even perform better on the lossy or weak network connections.
5. It has multi platform support on windows, linux and MACOS.
6. It has even backward compatibility with ros1 vis ros-bridge library.

Installation on Linux 20.04 version

There are various distro-version available for ros2 such as foxy, dashing, humble etc. So for now humble is the latest developed version, so we are going to install humble on our ubuntu os.
In order to install all the required dependencies along with ros2-humble we need to download the following script from repositories and run it.

```
sudo chmod 755 ./install_ros2_foxy.sh
bash ./install_ros2_foxy.sh
```

If the follwing script doesn't work.Manually run the below command one by one.

Set locale:

```
locale  # check for UTF-8

sudo apt update && sudo apt install locales
sudo locale-gen en_US en_US.UTF-8
sudo update-locale LC_ALL=en_US.UTF-8 LANG=en_US.UTF-8
export LANG=en_US.UTF-8

locale  # verify settings
```

Setup sources:

```
sudo apt install software-properties-common
sudo add-apt-repository universe
sudo apt update && sudo apt install curl
sudo curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -o /usr/share/keyrings/ros-archive-keyring.gpg
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/ros-archive-keyring.gpg] http://packages.ros.org/ros2/ubuntu $(. /etc/os-release && echo $UBUNTU_CODENAME) main" | sudo tee /etc/apt/sources.list.d/ros2.list > /dev/null
```

ROS2

Install ROS2 package:

```
sudo apt update
sudo apt upgrade
sudo apt install ros-humble-desktop python3-argcomplete
```

Environmental setup:
We can also add these command to ~/.bashrc file so that we don't need to manually run these cmd every time we open the terminal and source via source ~/.bashrc cmd.

```
source /opt/ros/humble/setup.bash
export ROS_DOMAIN_ID=30 ## need to be same as the robot when connected to same server
```

Now, we may required additional package such gazebo and various localization and navigation packages which are given below.

- Install gazebo11 for 3d robotics simulator which may be needed to simulate the various pre-built environments available in the ros2 package already.

  ```
  sudo apt-get install ros-humble-gazebo-*
  ```

- Install cartographer package or slam-toolbox package which are needed for slam(simultaneous localization and mapping purpose) i.e building the map which are needed for localization for later purpose. Slam-toolbox provides various pre-built in package like online-async slam and offline-sync for mapping functionality. We can use either one of the package for out purposes.

  ```
  sudo apt install ros-humble-slam-toolbox
  sudo apt install ros-humble-cartographer
  sudo apt install ros-humble-cartographer-ros
  ```

- Install Navigation2 package which consists of various nodes needed for trejectory planning, obstacle avoidance, local and global cost map, amcl package for localization and of course the differential_drive controller for sending the actuation command to the wheels of the robots.

  ```
  sudo apt install ros-humble-navigation2
  sudo apt install ros-humble-nav2-bringup
  ```

- Install turtlebot3 package which consists of a small differential drive robot integrated with the sensors such as lidar, encoder and imu etc which may be needed for visualizing the real robot in real time.

  ```
  source ~/.bashrc
  sudo apt install ros-humble-dynamixel-sdk
  sudo apt install ros-humble-turtlebot3-msgs
  sudo apt install ros-humble-turtlebot3
  ```

ROS2

Install Lidar package:
We need to install the lidar package based on its type on the ros2 which gives the scan range data to localize the robot in the real time. Currently we have the YdLidar available here so, we need to install the ydlidar ros2 driver which is the standard ros2 package for the ydlidar devices. As ydlidar_ros2_driver depends on the ydlidar_sdk library so we need to install and build the library before it.

```
cd $HOME
git clone https://github.com/YDLIDAR/YDLidar-SDK.git
mkdir YDLidar-SDK/build
cd YDLidar-SDK/build
cmake ..
make
sudo make install
```

Now we would install and build the ydlidar_ros2_driver.

```
cd $HOME
mkdir -p colcon_ws/src && cd colcon_ws/src
git clone https://github.com/YDLIDAR/ydlidar_ros2_driver.git
cd ~/colcon_ws
colcon build –symlink-install
source ./install/setup.bash
source ~/colcon_ws/install/setup.bash
```

Make sure to source the above setup.bash file in the ~/.bashrc file also inorder to source the above command every time we open the new terminal.Inorder to create the serial_port alias we need to run the following command.

```
chmod 0777 src/ydlidar_ros2_driver/startup/*
sudo sh src/ydlidar_ros2_driver/startup/initenv.sh
```

To run the lidar after connecting to the ros2 environment, run the following command then it will start visualizing the scan data in the rviz2.

```
ros2 launch ydlidar_ros2_driver ydlidar_launch.py
ros2 launch ydlidar_ros2_driver ydlidar_launch_view.py
```