Documentation: Installation and Setup of ROS2 Humble, PX4, MAVROS, and Gazebo for Drone Simulation

Table of Contents

---

1. Introduction

This document outlines the step-by-step process for installing and configuring a drone simulation environment using ROS2 Humble, PX4, MAVROS, and Gazebo. The goal is to create a functional setup capable of simulating drone operations, including waypoint navigation.

2. Prerequisites

- A computer with a minimum of 8 GB RAM.
- A compatible version of Ubuntu (20.04 recommended).
- Basic knowledge of Linux commands and terminal usage.

---

3. Installation Steps

3.1 Installation of Ubuntu

1. **Download Ubuntu:**
   - Visit the Ubuntu website and download the Ubuntu 20.04 LTS ISO file.
2. **Create a Bootable USB:**
   - Use tools like Rufus (Windows) or Etcher (macOS/Linux) to create a bootable USB drive.
3. **Install Ubuntu:**
   - Boot from the USB drive and follow the installation prompts to install Ubuntu on your computer.

3.2 Installation of ROS2 Humble

1. **Setup the Sources:**

```
sudo apt update
sudo apt install curl gnupg lsb-release
curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.key |
sudo apt-key add -
echo "deb [arch=amd64] http://packages.ros.org/ros2/ubuntu $(lsb_release
-cs) main" | sudo tee /etc/apt/sources.list.d/ros2-latest.list
```

2. **Install ROS2:**

```
sudo apt update
sudo apt install ros-humble-desktop
```

3. **Source the ROS2 Setup:**

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

4. **Install Additional Dependencies:**

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

3.3 Installation of Gazebo 11

1. **Install Gazebo:**

```
sudo apt install gazebo11 libgazebo11-dev
```

2. **Install ROS2 Gazebo Packages:**

```
sudo apt install ros-humble-gazebo-ros-pkgs
```

3.4 Installation of PX4

1. **Clone the PX4 Firmware Repository:**

```
git clone https://github.com/PX4/PX4-Autopilot.git --recursive
cd PX4-Autopilot
```

2. **Install Required Dependencies:**

```
bash ./Tools/setup/ubuntu.sh
```

3.5 Installation of MAVROS

1. **Install MAVROS:**

```
sudo apt install ros-humble-mavros ros-humble-mavros-extras
```

2. **Initialize MAVROS:**

```
sudo ln -s /opt/ros/humble/share/mavros/launch/mavros.launch.py
~/px4_ros_composition/
```

3.6 Building the Workspace

1. **Create a ROS2 Workspace:**

```
mkdir -p ~/ros2_ws/src
cd ~/ros2_ws/src
```

2. **Build the Workspace:**

```
cd ~/ros2_ws
colcon build
source install/setup.bash
```

---

4. Configuration Steps

4.1 Arming the Drone

1. **Start PX4 in SITL Mode with Gazebo:**

```
cd ~/PX4-Autopilot
make px4_sitl gazebo
```

2. **Open a new terminal and set the mode to OFFBOARD:**

```
ros2 service call /mavros/set_mode mavros_msgs/srv/SetMode
"{custom_mode: 'OFFBOARD'}"
```

3. **Arm the Drone:**

```
ros2 service call /mavros/cmd/arming mavros_msgs/srv/CommandBool
"{value: true}"
```

4.2 Using Dummy GPS

1. **Publish Dummy GPS Data:**

```
ros2 topic pub /mavros/setpoint_position/global
sensor_msgs/NavSatFix "{latitude: 47.397742, longitude: 8.545593,
altitude: 500}"
```

4.3 Starting Waypoint Navigation

1. **Run the Waypoint Navigation Node:**

```
ros2 run waypoint_nav waypoint_nav_node
```

4.4 Initializing .world Files

1. **Load a Gazebo World File:**
   Ensure the appropriate world file is specified in your Gazebo launch command or by modifying the `.launch` file accordingly.

---

5. Issues Faced

5.1 Linking ROS2 and Gazebo

- Ensuring proper communication between ROS2 nodes and Gazebo can require specific setup steps. It is essential to verify that both systems are configured to use compatible versions and that the correct namespaces are being used.

5.2 Linking MAVROS and PX4

- Configuring MAVROS to communicate with the PX4 autopilot can involve ensuring the correct parameters are set for the MAVLink communication. Troubleshooting connection issues may require checking the network configuration and confirming that both systems are using the same protocol and baud rate.

5.3 Arming the Drone

- Issues with arming the drone often arise from not meeting the pre-arm checks, such as having a valid GPS signal, sufficient battery level, and being in the correct flight mode. It is essential to monitor the output logs for specific reasons for failure.

5.4 Setting the Waypoints

- Defining and publishing waypoints must be done with precise latitude and longitude values. Any discrepancies in coordinates or failure to publish the data correctly can result in the drone not navigating as expected.

---

6. Troubleshooting

- If the drone fails to arm, check the following:
  - Ensure all pre-arm checks pass (GPS, battery level).
  - Review the console output for any error messages.
  - Confirm that the flight mode is set to OFFBOARD.

---

7. Conclusion

This documentation provides a comprehensive guide to setting up a drone simulation environment using ROS2 Humble, PX4, MAVROS, and Gazebo. Following these steps, users can effectively simulate drone operations, including waypoint navigation and obstacle avoidance. For further development, consider exploring additional ROS2 packages and functionalities tailored for aerial robotics.