UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Instalación, configuración y test de ROS 2 Humble

Eugenio Ivorra

Objetivos

- En esta práctica se aprenderá a instalar, configurar y ejecutar programas de ROS2. Los objetivos específicos son:
  - Saber qué es ROS2.
  - Instalar ROS2.
  - Crear un workspace.

Índice general

1 Instalación, configuración y test de ROS 2 Humble

1.1 Introducción a ROS 2

1.1.1 ¿Qué es ROS 2?

ROS 2 es un sistema de middleware que proporciona una forma de separar el código de un robot en bloques reutilizables, así como un conjunto de herramientas de comunicación para facilitar la comunicación entre estos bloques. Esto hace que sea más fácil desarrollar, probar y mantener aplicaciones de robots complejas. También proporciona una serie de bibliotecas y herramientas que pueden ayudar a los desarrolladores a crear aplicaciones de robots más rápidamente y con menos esfuerzo. Por ejemplo, ROS 2 incluye bibliotecas para la navegación, la planificación de trayectorias, el control de robots y la visión artificial.

Por último, es de código abierto y se basa en la noción de combinar espacios de trabajo utilizando el entorno de shell. Un "workspace" es un término de ROS para la ubicación en su sistema donde está desarrollando con ROS 2. El espacio de trabajo principal de ROS 2 se llama "underlay". Los espacios de trabajo locales posteriores se denominan "overlays".

1.1.2 ¿Por qué ROS 2?

ROS 2 es el sucesor de ROS 1. Es una plataforma de software de código abierto para robots que proporciona un conjunto de herramientas y bibliotecas para la creación de aplicaciones de robots robustas, escalables y reusables.

ROS 2 es ideal para aplicaciones que requieren una gran cantidad de comunicación entre sus subprogramas, o que tienen funcionalidades que van más allá de un caso de uso muy simple. Por ejemplo, ROS 2 podría ser una buena opción para un robot móvil que se controla con un GPS y una cámara.

1.1.3 ¿Cómo se usa ROS 2?

Para usar ROS 2, los desarrolladores primero deben instalar el software ROS 2 en su sistema. Una vez que el software esté instalado, los desarrolladores pueden comenzar a crear aplicaciones de robots utilizando las herramientas y bibliotecas proporcionadas por ROS 2.

ROS 2 es un sistema de software complejo, pero los desarrolladores pueden aprender a usarlo a través de la documentación, los tutoriales y los ejemplos disponibles en línea.

1.1.4 Ventajas de ROS 2

ROS 2 ofrece una serie de ventajas sobre otros sistemas de middleware para robots, entre las que se incluyen:

- Es una plataforma de código abierto, lo que significa que es gratuito y que su código está disponible para que cualquiera lo revise y modifique.
- Es modular, lo que significa que los desarrolladores pueden elegir las partes de ROS 2 que necesitan para sus aplicaciones.
- Es extensible, lo que significa que los desarrolladores pueden crear sus propias bibliotecas y herramientas para ROS 2.
- Es portable, lo que significa que las aplicaciones de ROS 2 se pueden ejecutar en una variedad de sistemas operativos.
- Es escalable, lo que significa que ROS 2 puede ser utilizado para crear aplicaciones de robots de cualquier tamaño.

1.1.5 Desventajas de ROS 2

ROS 2 es un sistema de software complejo, por lo que puede ser difícil de aprender y usar. Además, ROS 2 es una plataforma relativamente nueva, por lo que puede haber problemas de compatibilidad con algunos robots y hardware.

1.1.6 Conclusión

ROS 2 es una plataforma de software de código abierto potente y flexible que puede ser utilizada para crear aplicaciones de robots robustas, escalables y reusables. ROS 2 es una buena opción para los desarrolladores que buscan un sistema de middleware que les permita crear aplicaciones de robots más rápidamente y con menos esfuerzo.

1.2 Instalación de ROS2

En este tutorial, se explica cómo instalar ROS2 en Ubuntu 22.04. Es posible instalar Ubuntu 22.04 en una máquina virtual o, preferiblemente, en una partición del disco duro.

1.2.1 Preparación del entorno

Primero, necesitarás tener Ubuntu 22.04 instalado. Si aún no tienes Ubuntu instalado, por favor instala la versión 22.04, que puede ser cualquier sabor (Ubuntu, LUbuntu, Mubuntu etc.). Ubuntu22.04

1.2.2 Scripts de instalación

Hemos preparado una serie de scripts de instalación que automatizarán el proceso. Puedes encontrar estos scripts en nuestro repositorio de GitHub, disponible en este enlace.

Para usar estos scripts, sigue los pasos que se describen a continuación:

1. Clona el repositorio de GitHub en tu máquina local:

```
git clone https://github.com/euivmar/ros2_setup_scripts_ubuntu.git
```

2. Accede al directorio del repositorio:

```
cd ros2_setup_scripts_ubuntu
```

3. Dale permisos de ejecución al script ‘run.sh‘ y ejecútalo:

```
chmod +x run.sh
./run.sh
```

4. Haz lo mismo con el script ‘installTurtlebot3SIM‘:

```
chmod +x installTurtlebot3SIM
./installTurtlebot3SIM
```

5. Finalmente, haz lo mismo con el script ‘installTurtlebot3SIM_WS‘:

```
chmod +x installTurtlebot3SIM_WS
./installTurtlebot3SIM_WS
```

Siguiendo estos pasos, deberías tener un entorno ROS2 completamente configurado y listo para usar para la asignatura.

1.2.3 Test de la instalación

Para comprobar que la instalación ha ido bien, mi recomendación es lo primero reiniciar el ordenador y a continuación ejecutar el siguiente comando:

```
ros2 doctor
```

El resultado debería de ser `"All 5 checks passed"`.

1.3 Software Auxiliar Recomendado

Recomendamos instalar el editor de código `Visual Studio Code` con el complemento de ROS, así como la terminal `Terminator` para facilitar el desarrollo en ROS.

1.3.1 Visual Studio Code

Visual Studio Code (VS Code) es un editor de código fuente desarrollado por Microsoft. Ofrece características como resaltado de sintaxis, autocompletado inteligente, y la capacidad de instalar extensiones para ampliar su funcionalidad.

Instalación en Ubuntu:

```
sudo snap install code --classic
```

Ejecución: Para ejecutar Visual Studio Code, puedes hacer clic en su icono desde el menú de aplicaciones o ejecutar el siguiente comando en la terminal:

```
code
```

1.3.2 Plugin de ROS para VS Code

Este complemento proporciona un conjunto de herramientas útiles para el desarrollo de ROS, como autocompletado para mensajes ROS, conexión fácil con servidores ROS y mucho más.

Instalación: Abra VS Code, vaya a la pestaña de Extensiones y busque “ROS”. Instale la extensión ofrecida por Microsoft.

1.3.3 Terminator

Terminator es una aplicación de terminal que proporciona múltiples ventanas, pestañas, y una serie de otras mejoras sobre la terminal estándar de Ubuntu.

Instalación en Ubuntu:

```
sudo apt update
sudo apt install terminator
```

Estos programas, cuando se utilizan juntos, ofrecen un entorno de desarrollo potente y altamente personalizable para ROS.

2 Usar Colcon para compilar paquetes y crear nuestro espacio de trabajo

2.1 Introducción a Colcon

Colcon es una herramienta versátil y de código abierto para la construcción de proyectos ROS 2. Se destaca por su capacidad para manejar desde pequeños estudios hasta grandes sistemas de producción. Su configuración adaptable y su compatibilidad con diversas herramientas de compilación, como CMake, Ninja y Meson, le permiten funcionar en sistemas operativos como Linux, macOS y Windows. Aunque es relativamente nuevo y pueden surgir problemas de compatibilidad, su eficiencia, flexibilidad y la oportunidad de colaboración comunitaria lo convierten en una elección destacada para desarrolladores en robótica.

2.2 Crear un espacio de trabajo (Workspace)

Primero, creamos un directorio (`ros2_ws`) para contener nuestro espacio de trabajo.

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

2.2.1 Clonar repositorio de ejemplo

Clonamos el repositorio de ejemplos en el directorio `src` del espacio de trabajo.

```
git clone https://github.com/euivmar/ROS2_examples.git
```

2.2.2 Compilar el workspace

Para compilar el espacio de trabajo (workspace) hay que estar en la raíz del directorio y ejecutar:

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

El argumento `-symlink-install` permite modificar los archivos de Python sin necesidad de recompilar.

2.2.3 Cargar el workspace

Es necesario que hagamos un source del nuevo workspace para que ros2 encuentre los nuevos paquetes. Si quisiéramos tener el workspace siempre cargado tendríamos que añadir la linea en el `.bashrc` con la ruta absoluta.

```
source install/local_setup.bash
```

2.2.4 Ejecutar test

Para ejecutar los test en los paquetes que acabamos de construir:

```
colcon test
```

Saldrán algunos warnings de deprecated, pero no son relevantes.

2.2.5 Ejecutar unos ejemplos

Para probar un ejemplo que hemos compilado podemos ejecutar en una terminal `(terminal 1)` :

```
ros2 run examples_rclpy_minimal_publisher publisher_member_function
```

Y en la otra terminal `(terminal 2)`, haciendo previamente el source si no lo hemos añadido a nuestro .bashrc

```
ros2 run examples_rclpy_minimal_subscriber subscriber_member_function
```

Si todo va bien la `(terminal 1)` debería de enviar "Hola mundo" con un contador a la `(terminal 2)` que lo recibe y lo muestra.

Bibliografía

Web de los tutoriales de ROS (2023). https://docs.ros.org/en/humble/Tutorials.html.

Web de ROS2 Humble (2023). https://docs.ros.org/en/humble/index.html.