**ROS Tutoriales**

Comprobar las variables de entorno del creadas por ROS

$ printenv | grep ROS

Después de crear un espacio de trabajo (directorio) ejecutamos la instrucción que se ejecuta la primera vez y crea un fichero texto en y dos directorios y .

$ mkdir -p ~/catkin\_ws/src

$ cd ~/catkin\_ws/

$ catkin\_make

Ejecuta la instrucción ya que se van a alterar las variables de entorno de nuestro actual sobre el alojado en .

$ source devel/setup.bash

Comprobamos el que los tutoriales de la versión de estén instalados.

$ sudo apt-get install ros-lunar-ros-tutorials

**Paquete**

Es la organización software del código de . Cada paquete contiene librerías, ejecutables, y artefactos.

**Manifiesto**

Tiene extensión y es una descripción del paquete y sirve para definir dependencias entre paquetes y capturar meta información sobre el paquete como la versión, licencia…

La instrucción sirve para obtener información sobre un paquete.

$ rospack find [package\_name]

La instrucción te lleva directamente al directorio de

La instrucción te muestra los ficheros y carpetas del directorio

Los paquetes se crean sobre directorios de trabajos .

La instrucción básica para crear un paquete es hacer uno de

$ catkin\_create\_pkg name\_pkg depend1 depend2 …

Podemos comprobar que en el fichero asociado al paquete creado contiene las dependencias marcadas en la instrucción

Para comprobar las dependencias de un paquete lo podemos hacer así.

$ rospack depends1 name\_pkg

Dentro del fichero podemos añadir una descripción del paquete…

$ <description>…</description>

$ <maintainer email=”…”>…</maintainer>

$ <licence>...</licence>

La instrucción combina las llamadas a y . Se ejecuta sobre el directorio de trabajo.

**Nodo**

Es un ejecutable que usa para comunicarse con otros nodos.

**Mensajes**

Tipo de dato usado por cuando está suscrito o publica en un tema.

**Tema**

Lugar donde los nodos publican mensajes.

**Master**

Nombre del servicio de .

**Rosout**

Salidas de texto y error de .

Para arrancar usamos el que incluye el , y los parámetros de configuración del servidor.

$ roscore

La estructura general de hace uso de un cliente servidor en terminales por lo que es necesario abrir un nuevo para cada nodo.

Para obtener información sobre los nodos que se están ejecutando lo podemos hacer con la instrucción .

$ rosnode info /name\_exe

Para ejecutar un nodo debemos especificar el paquete en el que se encuentra y el nombre del ejecutable. El paquete representa una carpeta.

$ rosrun [package\_name] [node\_name]

El paquete nos permite visualizar un grafo de la comunicación con los anteriormente explicados. Para instalarlo:

$ sudo apt-get install ros-lunar-rqt

$ sudo apt-get install ros-lunar-rqt-common-plugins

Ejecutamos el paquete con la siguiente instrucción.

$ rosrun rqt\_graph rqt\_graph

La instrucción es usada para obtener información sobre los . para saber el tipo de dato que hay que enviar por el , para listar los activos. para publicar información a un . para obtener información sobre los reportes publicados.

Para listar los servicios que nos ofrece usamos la siguiente instrucción.

$ rosservice list

Para llamar a un servicio de usamos la siguiente instrucción.

rosservice call [service] [args]

Para alterar los parámetros del servidor se usa

da la posibilidad de un usar una interface llamada que permite visualizar los nodos. Se arranca de la siguiente forma.

$ rosrun rqt\_console rqt\_console

$ rosrun rqt\_logger\_level rqt\_logger\_level

Para hacer un inicio de en un paquete con un arranque preestablecido usamos la siguiente instrucción.

$ roslaunch [package] [filename.launch]

El arranque se especifica sobre el fichero . Sirve para guardar pruebas realizadas.

La instrucción sirve para editar ficheros de un paquete. Se puede cambiar el editor por defecto de .