

GRADO EN INGENIERÍA EN TECNOLOGÍA INDUSTRIAL

## TRABAJO FIN DE GRADO

### ***CONTROL DE UN BRAZO MANIPULADOR DE 5 EJES DE LIBERTAD SOBRE UNA PLATAFORMA ROBOTINO***

#### ***DOCUMENTO I - MANUAL DEL USUARIO***

**Alumno/Alumna:** González Jaio, Maialen

**Director/Directora (1):** Casquero Oyarzabal, Oskar

**Director/Directora (2):** Orive Revillas, Darío

**Curso:** 2018-2019

**Fecha:** Bilbao, 20 de junio 2019

El objetivo de este manual es explicar de forma sencilla los pasos necesarios a dar para la puesta en marcha del Robotino y del brazo. Por lo tanto, se asumirá que las instalaciones necesarias de ROS Kinetic y de los paquetes necesarios para el correcto funcionamiento de los equipos se ha realizado previamente.

Debido a que se necesita trabajar en diferentes terminales se utilizará el programa Terminator para la gestión y visualización de las pestañas.

# 1. Configuración inicial

En cada terminal será necesario acceder como usuario root para lo que hay que introducir el siguiente comando:

```
$ ssh root@192.168.1.160  
Contraseña: $ dorp6
```

Una vez dentro del Robotino será necesario acceder al espacio de trabajo:

```
$ cd /home/robotino/catkin_ws/
```

En la primera terminal que se acceda a este directorio será necesario introducir la siguiente línea de comando:

```
$ route add default gw 192.168.1.1
```

Para comprobar que la ruta se ha creado correctamente se debe introducir el siguiente comando que muestra todas las rutas de nuestro equipo:

```
$ route -n
```

Posteriormente, en la primera terminal al igual que en el resto hay que ejecutar los siguientes comandos:

```
$ source ../devel/setup.bash  
$ catkin_make
```

Esta última línea consiste en un compilador, por lo tanto, analizará todo el equipo para comprobar que todos los paquetes están bien instalados. Sin embargo, no realizará las compilaciones del código escrito en Python. Este se comprobará al ejecutar el propio programa.

Una vez que se haya compilado el equipo y no haya errores se procede a la activación del maestro y de los nodos necesarios tanto para el Robotino como en el brazo. Para ello será necesario ejecutar dos archivos de tipo launch en terminales diferentes.

Comando de la primera terminal para la activación del Robotino:

```
$ roslaunch robotino_node robotino_simple_node.launch
```

Comando de la segunda terminal para la activación del brazo:

```
$ roslaunch widowx_arm_controller widowx_arm_controller.launch
```

## 2. Ejecutar un programa

Ahora que están activos los nodos y el maestro se pueden ejecutar los códigos programados en una tercera terminal. Por lo tanto, tras realizar de nuevo los pasos de la configuración inicial, se podrá ejecutar el programa principal introduciendo el siguiente comando:

```
$ rosrun paquete_proyecto base_robotino.py
```

Al ejecutar este comando la plataforma móvil empezará a moverse y a ejecutar el código.

Si en lugar de esta aplicación se quiere ejecutar cualquiera del resto de los códigos creados los pasos a realizar serían los siguientes:

1. Acceder al directorio del código desde el catkin\_ws:

```
$ cd /src/paquete_proyecto/scripts/
```

2. Mostrar los ejecutables disponibles:

```
$ ls
```

3. Modificar o leer el programa:

```
$ nano <nombre_del_ejecutable>.py
```

4. Ejecutar el programa:

```
$ rosrun paquete_proyecto <nombre_del_ejecutable>.py
```

### 3. Crear nuevas funcionalidades

En este apartado se detallarán los pasos a realizar si se desea crear una nueva aplicación para el conjunto de la plataforma con el brazo. En caso de que se desee modificar una aplicación ya existente ya sea para reajustar los parámetros o introducir otros comandos, los pasos a realizar han sido definidos en el apartado anterior.

Si se quiere crear una nueva funcionalidad en Python desde cero en primer lugar hay que posicionarse en el directorio en el que se desee guardar la aplicación, partiendo del catkin\_ws:

```
$ cd /src/paquete_proyecto/scripts/
```

Una vez posicionados dentro del directoria hay que ejecutar la siguiente línea:

```
$ touch <nombre_del_ejecutable>.py
```

Para comprobar que el ejecutable se ha creado correctamente hay que introducir el siguiente comando y observar que entre los resultados obtenidos hay un archivo con el nombre que se ha escrito.

```
$ ls
```

El nombre del archivo estará escrito en color gris por lo que lo siguiente es darle permisos de ejecución al archivo:

```
$ sudo chmod 775 <nombre_del_ejecutable>.py
```

Una vez hecho esto, si se vuelve a ejecutar el anterior comando el color del nombre del archivo pasará a ser verde.

Otra de las opciones para crear o modificar un archivo consiste en realizar una copia de un archivo ya existente. En este caso, no haría falta darle permisos al documento. La línea de comando sería la siguiente:

```
$ cp <nombre_del_ejecutable_orginal>.py <nombre_de_la_copia>.py
```