1. **CONFIGURAR UN ESPACIO DE TRABAJO – CATKIN WORKSPACE**

Se ingresa los siguiente comandos:



Al crear el espacio de trabajo con el comando *catkin\_make*, se tiene lo siguiente:

Texto

Descripción generada automáticamente

Ahora, se tienen que abrir el siguiente archivo:

Escribir “*source ~/catkin\_ws/devel/setup.bash”* el cual es el entorno que se ha creado.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. **CREAR UN PAQUETE LLAMADO TURTLESIM\_TEST PACKAGE**

Texto

Descripción generada automáticamente

Con el comando *catkin\_create\_pkg* se creó el paquete. Además, se instaló también las librerías que se usarán para las siguientes indicaciones de la tarea.

Posteriormente, se compiló el paquete con el comando *catkin\_make.*

Texto

Descripción generada automáticamente

Ahora se tiene el paquete dentro de src:

Texto

Descripción generada automáticamente

1. **CREAR UN NUEVO MENSAJE “SPEED”, DEL TIPO FLOAT32 LLAMADO DATA.**

Se crea la carpeta de mansajes personalizados en el paquete que ya cree y también se crea el mensaje llamado Speed con el dato y su tipo indicado.



Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. **MODIFICAR EL ARCHIVO CMAKELISTS.TXT PARA ASEGURARSE QUE EL NUEVO MENSAJE ES COMPILADO.**

En el archivo CMakeLists.txt viene comentado las líneas de ejemplo para configurar los mensajes personalizados que queramos crear en el paquete, por lo que se hacen los siguientes cambios:

Texto

Descripción generada automáticamente

Lo anterior se cambia por:

Texto

Descripción generada automáticamente

Después se cambia:

Texto

Descripción generada automáticamente

Por:

Texto

Descripción generada automáticamente

Y, además, se agrega también:

Texto

Descripción generada automáticamente

Ahora se compila con *catkin\_make.*

1. **CREAR UN ARCHIVO *.LAUNCH* QUE EJECUTE UN NODO LLAMADO *TURTLESIM\_NODE* DENTRO DEL NAMESPACE /MY\_TURTLE. CREE TAMBIÉN UN ARCHIVO .LAUNCH PARA LANZAR *RQT\_PLOT* MOSTRANDO LOS VALORES DE X,Y Y THETA DE LA POSICIÓN DE LA TORTUGA.**

Se crea la carpeta “launch” y dentro se crean los archivos .launch, como se muestra:

Texto

Descripción generada automáticamente

En el archivo *turtlesim.launch*, que tiene como objetivo ejecutar el nodo *turtlesim\_node*, contiene lo siguiente:

Texto

Descripción generada automáticamente

Se inicia ROS:

Texto

Descripción generada automáticamente

Y se ejecuta: Texto

Descripción generada automáticamente

Y se lo siguiente:

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

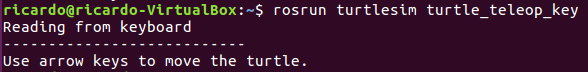
El archivo *turtlesim\_plot.launch,* que tiene como objetivo dibujar los valores de x,y,theta, contiene lo siguiente:

Texto

Descripción generada automáticamente

Ahora, para que se vea las líneas de los valores, se ejecutó también:





Con el objetivo de mover la tortuga con las teclas.

Ahora, se ejecuta *turtlesim\_plot.launch*:

Texto

Descripción generada automáticamente

y se ve lo siguiente:

Interfaz de usuario gráfica

Descripción generada automáticamente

1. Cree una secuencia de comandos bash para publicar un valor para el mensaje Speed, ingresado por la línea de comando en /my\_turtle/speed usando el tipo de mensaje recién definido. Utilice una velocidad angular de 1.55 para girar 90°.
2. Grabe los comandos de su script con rosbag, reprodúzcalos y grabe el movimiento resultante.