



A continuación, vamos a crear un nuevo nodo denominado “my_nodo_2”, que consiste en una modificación del nodo “my_node”, para que éste en vez de leer una imagen del disco duro, se suscriba al tópico “/usb_cam/image_raw” por donde el nodo denominado “usb_cam_node” publica las imágenes capturadas por la cámara usb de su laptop, y calcule la componente de intensidad de cada imagen publicada, así como dibuje el segmento lineal y el círculo sobre ella.

Bajar de la página virtual el archivo “my_node_2.cpp” y ponerlo en el Desktop (Escritorio) de Ubuntu.

El nodo “my_node_2” representa una modificación del nodo “my_node”. La modificación consistió en 5 pasos, los cuales están debidamente marcados con la frase “De codeBlocs a ROS” dentro del archivo “my_node_2.cpp”.

Copiar el archivo “my_node_2.cpp” del Desktop (Escritorio) al directorio fuente “src” del paquete “test”

Abrir un nuevo terminal (presione simultáneamente Ctrl+alt+t)

Copiar “my_node_2.cpp” en el directorio “src” del paquete “test”

cp ~/Desktop/my_node_2.cpp ~/indigo_workspace/sandbox/test/src

en el caso de tener el Ubuntu en español, en vez del comando anterior utilice el siguiente:

cp ~/Escritorio/my_node_2.cpp ~/indigo_workspace/sandbox/test/src

Cierre el terminal

Abrir el archivo “CMakeLists.txt” en el directorio del paquete “test” e indicar que debe compilar y enlazar el nuevo nodo

Abrir un nuevo terminal (presione simultáneamente Ctrl+alt+t)

Ir al directorio del paquete “test”

cd ~/indigo_workspace/sandbox/test

Abrir el archivo “CmakeLists.txt”

gedit CMakeLists.txt

Agregar al final del archivo la siguiente línea de código (ver Fig. 17):

rosbuild_add_executable(my_node_2 src/my_node_2.cpp)

Salvar y cerrar el archivo

Cierre el terminal

```
CMakeLists.txt (~/indigo_workspace/sandbox/test) - gedit

cmake_minimum_required(VERSION 2.4.6)
include($ENV{ROS_ROOT}/core/rosbuild/rosbuild.cmake)

# Set the build type. Options are:
# Coverage      : w/ debug symbols, w/o optimization, w/ code-coverage
# Debug         : w/ debug symbols, w/o optimization
# Release       : w/o debug symbols, w/ optimization
# RelWithDebInfo : w/ debug symbols, w/ optimization
# MinSizeRel    : w/o debug symbols, w/ optimization, stripped binaries
#set(ROS_BUILD_TYPE RelWithDebInfo)

rosbuild_init()

#set the default path for built executables to the "bin" directory
set(EXECUTABLE_OUTPUT_PATH ${PROJECT_SOURCE_DIR}/bin)
#set the default path for built libraries to the "lib" directory
set(LIBRARY_OUTPUT_PATH ${PROJECT_SOURCE_DIR}/lib)

# Uncomment if you have defined messages
#rosbuild_genmsg()
# Uncomment if you have defined services
#rosbuild_gensrv()

# Common commands for building c++ executables and libraries
#rosbuild_add_library(${PROJECT_NAME} src/example.cpp)
#target_link_libraries(${PROJECT_NAME} another_library)
#rosbuild_add_boost_directories()
#rosbuild_link_boost(${PROJECT_NAME} thread)
#rosbuild_add_executable(example examples/example.cpp)
#target_link_libraries(example ${PROJECT_NAME})
rosbuild_add_executable(my_node src/my_node.cpp)
rosbuild_add_executable(my_node_2 src/my_node_2.cpp)
```

Fig. 17



Compilar y enlazar el node “my_node_2”

Abrir un nuevo terminal (presione simultáneamente Ctrl+alt+t)

Ejecutar el setup.bash en el nuevo terminal
source ~/indigo_workspace/setup.bash

Ir al directorio del paquete “test”
cd ~/indigo_workspace/sandbox/test

Compilar y enlazar nuevo node
make

Si todo salió bien se debería obtener en rojo la siguiente línea (ve Fig. 18)

Linking CXX executable ../bin/my_node_2

Listo, con esto el nodo “my_node_2” ha sido creado!. No olvide que cada vez que modifique el código del nodo, éste se debe compilar y enlazar nuevamente para que los cambios entren en vigor.

Cierre el terminal

```
est@ubuntu: ~/indigo_workspace/sandbox/test
ld
cd build && make
make[1]: Entering directory `/home/est/indigo_workspace/sandbox/test/build'
make[2]: Entering directory `/home/est/indigo_workspace/sandbox/test/build'
make[3]: Entering directory `/home/est/indigo_workspace/sandbox/test/build'
make[3]: Leaving directory `/home/est/indigo_workspace/sandbox/test/build'
[ 0%] Built target rospack_genmsg_libexe
make[3]: Entering directory `/home/est/indigo_workspace/sandbox/test/build'
make[3]: Leaving directory `/home/est/indigo_workspace/sandbox/test/build'
[ 0%] Built target rosbuilt_precompile
make[3]: Entering directory `/home/est/indigo_workspace/sandbox/test/build'
make[3]: Leaving directory `/home/est/indigo_workspace/sandbox/test/build'
[ 50%] Built target my_node
make[3]: Entering directory `/home/est/indigo_workspace/sandbox/test/build'
Scanning dependencies of target my_node_2
make[3]: Leaving directory `/home/est/indigo_workspace/sandbox/test/build'
make[3]: Entering directory `/home/est/indigo_workspace/sandbox/test/build'
[100%] Building CXX object CMakeFiles/my_node_2.dir/src/my_node_2.cpp.o
Linking CXX executable ../bin/my_node_2
make[3]: Leaving directory `/home/est/indigo_workspace/sandbox/test/build'
[100%] Built target my_node_2
make[2]: Leaving directory `/home/est/indigo_workspace/sandbox/test/build'
make[1]: Leaving directory `/home/est/indigo_workspace/sandbox/test/build'
est@ubuntu:~/indigo_workspace/sandbox/test$
```

Fig. 18



Seguidamente, se ejecutará el nodo “usb_cam_node”, el cual publica las imágenes capturadas por la cámara USB de su laptop por el tópico “usb_cam/image_raw”, y el nodo “my_node_2” para que se suscriba a ese tópico, calcule la componente de intensidad de cada imagen publicada, así como dibuje un segmento lineal y un círculo sobre cada una de ellas.

Abrir un nuevo terminal (presione simultáneamente Ctrl+alt+t)

Correr el ROS maestro

roscore

No cierre el terminal

Abrir un nuevo terminal (presione simultáneamente Ctrl+alt+t)

Colocar en el servidor de parámetros de ROS el parámetro “usb_cam/pixel_format” en su valor correcto

rosparam set usb_cam/pixel_format yuyv

Colocar en el servidor de parámetros de ROS el parámetro “video_device” con el nombre de dispositivo usado por su cámara USB (por defecto es /dev/video0)

rosparam set usb_cam/video_device /dev/video0

En caso de que no funcione, hay que buscar el nombre de dispositivo usado por su cámara USB. En el mismo terminal ejecute los siguientes pasos a), b) y c)

a) Ir a la carpeta de dispositivos /dev

cd /dev

b) Listar todos los dispositivos (archivos)

ls

c) Buscar el que tenga como nombre “videoX”, donde X es un número entero, como por ejemplo video1, y colocarlo en el parámetro “video_device” del servidor de parámetros de ROS; siguiendo el ejemplo (video1):

rosparam set usb_cam/video_device /dev/video1

Correr el nodo “usb_cam_node”

Antes de este: `roslaunch usb_cam usb_cam_node`

No cierre el terminal

En una terminal nueva digite:

`$source ~/catkin_ws/devel/setup.bash`

y en la misma digite:

`$roslaunch usb_cam usb_cam_node`
(esto con el fin de activar la cámara)

Después en una terminal nueva continúe con lo que está debajo de las líneas punteadas

Abrir un nuevo terminal (presione simultáneamente Ctrl+alt+t)

Ejecutar el setup.bash en el nuevo terminal

source ~/indigo_workspace/setup.bash

Ir al directorio del paquete “test”

cd ~/indigo_workspace/sandbox/test

Ejecutar el nodo “my_node_2”

roslaunch test my_node_2

Enjoy!. Como verá a cada imagen publicada se le calcular la componente de intensidad y se dibuja sobre ella un segmento lineal y un círculo.

No cierre el terminal

Cuando haya terminado cierre todas los terminales

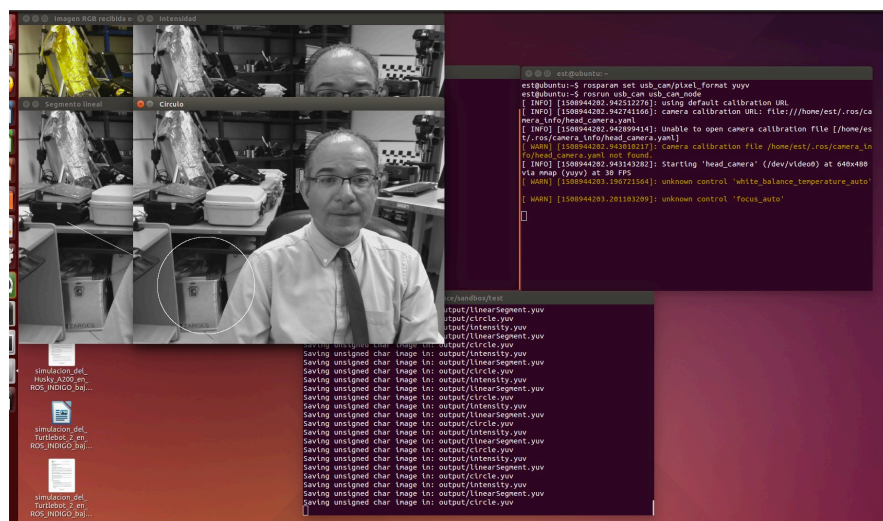


Fig. 19