



En adelante, vamos a modificar el programa de referencia “visionPorComputador_Prog_Ref_No1” (v004) para que corra como un nodo del paquete “test” que llamaremos “my_node”. Para ello hagamos lo siguiente:

Bajar de la página virtual el proyecto de “codeBlocks” denominado “visionPorComputador_Prog_Ref_No1”, asegúrese de que sea la versión v004, del 24 Octubre 2017, 11:30. Póngalo en el Desktop (Escritorio) de su Ubuntu.

Copiar el “main.c” del proyecto “visionPorComputador_Prog_Ref_No1” en el directorio fuente “src” del paquete “test”

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

Copiar “main.c” en el directorio “src” del paquete “test”

```
cp ~/Desktop/visionPorComputador_Prog_Ref_No1/main.c ~/indigo_workspace/sandbox/test/src
```

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

```
cp ~/Escritorio/visionPorComputador_Prog_Ref_No1/main.c ~/indigo_workspace/sandbox/test/src
```

Cierre el terminal

Cambiar el nombre del archivo copiado “main.c” por “my_node.cpp”

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

Ir al directorio fuente del paquete “test”

```
cd ~/indigo_workspace/sandbox/test/src
```

Cambiar el nombre de “main.c” por “my_node.cpp”

```
mv main.c my_node.cpp
```

Cierre el terminal

Copiar el “current_control_parameters.txt” del proyecto “visionPorComputador_Prog_Ref_No1” en el directorio del paquete “test”

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

Copiar “current_control_parameters.txt” en el directorio del paquete “test”

```
cp ~/Desktop/visionPorComputador_Prog_Ref_No1/current_control_parameters.txt ~/indigo_workspace/sandbox/test
```

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

```
cp ~/Escritorio/visionPorComputador_Prog_Ref_No1/current_control_parameters.txt ~/indigo_workspace/sandbox/test
```

Cierre el terminal

Copiar el directorio “input” del proyecto “visionPorComputador_Prog_Ref_No1” en el directorio del paquete “test”

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

Copiar “input” en el directorio del paquete “test”

```
cp -r ~/Desktop/visionPorComputador_Prog_Ref_No1/input ~/indigo_workspace/sandbox/test
```

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

```
cp -r ~/Escritorio/visionPorComputador_Prog_Ref_No1/input ~/indigo_workspace/sandbox/test
```

Cierre el terminal

Copiar el directorio “output” del proyecto “visionPorComputador_Prog_Ref_No1” en el directorio del paquete “test”

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

Copiar “output” en el directorio del paquete “test”

```
cp -r ~/Desktop/visionPorComputador_Prog_Ref_No1/output ~/indigo_workspace/sandbox/test
```

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

```
cp -r ~/Escritorio/visionPorComputador_Prog_Ref_No1/output ~/indigo_workspace/sandbox/test
```

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”

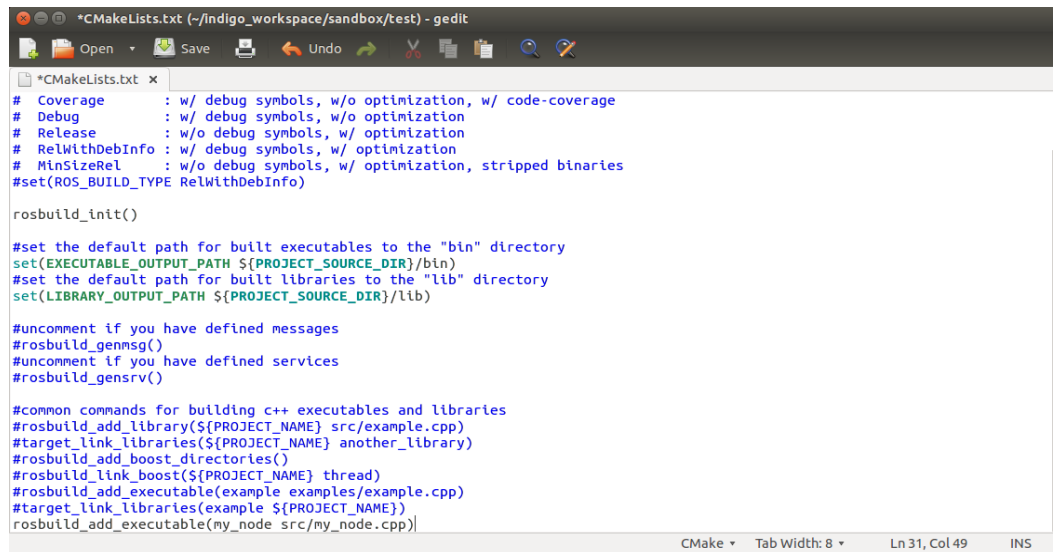
```
gedit CMakeLists.txt
```

Agregar al final del archivo la siguiente línea de código (ver Fig. 12):

```
rosbuild_add_executable(my_node src/my_node.cpp)
```

Salvar y cerrar el archivo

Cierre el terminal



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

Fig. 12

Compilar y enlazar el nuevo nodo

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 nodo

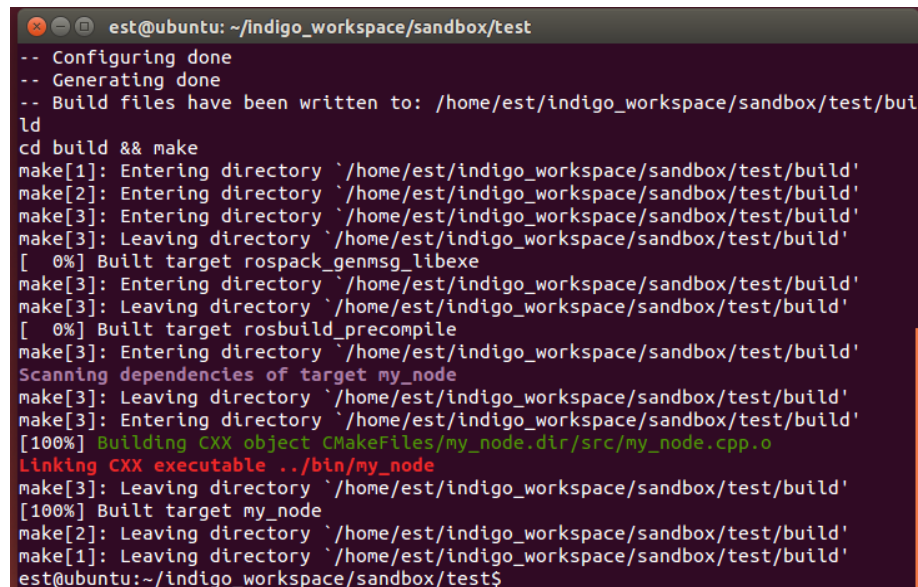
make

Si todo salió bien se debería obtener en rojo la siguiente línea (ver Fig. 13)

Linking CXX executable ../bin/my_node

Listo, con esto el nodo “my_node” ha sido creado!

Cierre el terminal



```
est@ubuntu: ~/.indigo_workspace/sandbox/test
-- Configuring done
-- Generating done
-- Build files have been written to: /home/est/indigo_workspace/sandbox/test/build
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'
Scanning dependencies of target my_node
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.dir/src/my_node.cpp.o
Linking CXX executable ../bin/my_node
make[3]: Leaving directory `/home/est/indigo_workspace/sandbox/test/build'
[100%] Built target my_node
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. 13



Correr el nuevo nodo

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)

Ejecutar el setup.bash en el nuevo terminal

source ~/indigo_workspace/setup.bash

Ir al directorio del paquete “test”

cd ~/indigo_workspace/sandbox/test

Correr el nuevo nodo (ver Fig. 15)

roslaunch test my_node

No cierre el terminal

Felicitaciones, acaba de correr el programa de referencia “visionPorComputador_Prog_Ref_No1” como un nodo llamado “my_node” en ROS Indigo !!. Los resultados los apreciará tanto en el terminal como en el directorio “output” (ver Fig. 15), el cual se encuentra en el directorio del paquete “test” (~/indigo_workspace/sandbox/test).

Cierre ambas terminales

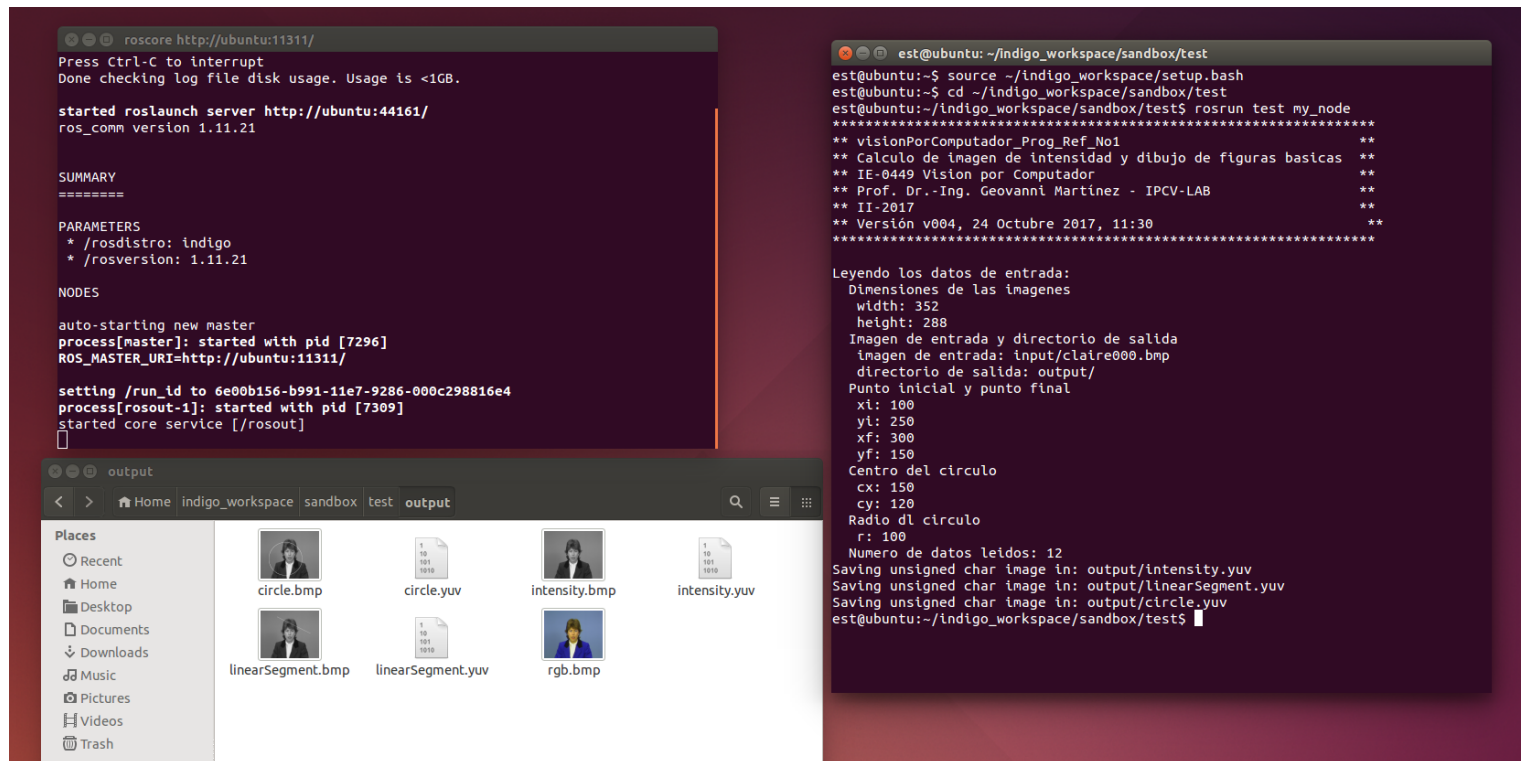


Fig. 15