cmake\_minimum\_required(VERSION 3.0.2)

project(urg\_node)

find\_package(catkin REQUIRED COMPONENTS

diagnostic\_updater dynamic\_reconfigure laser\_proc message\_generation

nodelet rosconsole roscpp sensor\_msgs std\_msgs std\_srvs tf urg\_c

)

set\_directory\_properties(PROPERTIES COMPILE\_OPTIONS "-std=c++11")

# Dynamic reconfigure support

generate\_dynamic\_reconfigure\_options(cfg/URG.cfg)

add\_message\_files(FILES

Status.msg

)

generate\_messages(

DEPENDENCIES std\_msgs

)

catkin\_package(

INCLUDE\_DIRS include

LIBRARIES urg\_c\_wrapper urg\_node\_driver

CATKIN\_DEPENDS dynamic\_reconfigure laser\_proc message\_runtime nodelet rosconsole

roscpp sensor\_msgs std\_msgs std\_srvs urg\_c

DEPENDS

)

include\_directories(include ${catkin\_INCLUDE\_DIRS})

## Declare a cpp library

add\_library(urg\_c\_wrapper src/urg\_c\_wrapper.cpp)

target\_link\_libraries(urg\_c\_wrapper ${catkin\_LIBRARIES})

# Declare the cpp as a lib also

add\_library(urg\_node\_driver src/urg\_node\_driver.cpp)

add\_dependencies(urg\_node\_driver ${PROJECT\_NAME}\_gencfg ${PROJECT\_NAME}\_gencpp)

target\_link\_libraries(urg\_node\_driver urg\_c\_wrapper ${catkin\_LIBRARIES})

## Declare a cpp executable

add\_executable(urg\_node src/urg\_node.cpp)

target\_link\_libraries(urg\_node urg\_node\_driver ${catkin\_LIBRARIES})

## Declare getID tool

add\_executable(getID src/getID.cpp)

target\_link\_libraries(getID urg\_c\_wrapper ${catkin\_LIBRARIES})

## Mark executables and/or libraries for installation

install(TARGETS urg\_c\_wrapper urg\_node urg\_node\_driver getID

ARCHIVE DESTINATION ${CATKIN\_PACKAGE\_LIB\_DESTINATION}

LIBRARY DESTINATION ${CATKIN\_PACKAGE\_LIB\_DESTINATION}

RUNTIME DESTINATION ${CATKIN\_PACKAGE\_BIN\_DESTINATION}

)

install(

DIRECTORY include/${PROJECT\_NAME}/

DESTINATION ${CATKIN\_PACKAGE\_INCLUDE\_DESTINATION}

)

## Mark cpp header files for installation

install(DIRECTORY include/${PROJECT\_NAME}/

DESTINATION ${CATKIN\_PACKAGE\_INCLUDE\_DESTINATION}

FILES\_MATCHING PATTERN "\*.h"

PATTERN ".svn" EXCLUDE

)

## Install script to set IP addresses

install(PROGRAMS

scripts/set\_urg\_ip.py

DESTINATION ${CATKIN\_PACKAGE\_BIN\_DESTINATION}

)

if(CATKIN\_ENABLE\_TESTING)

find\_package(roslint)

# Ignore long -> int32\_t warning, urg\_c driver uses

# long, etc. Wrapper must also use these

set(ROSLINT\_CPP\_OPTS "--filter=-runtime/references,-runtime/int")

roslint\_cpp(

include/urg\_node/urg\_c\_wrapper.h

include/urg\_node/urg\_node\_driver.h

src/getID.cpp

src/urg\_c\_wrapper.cpp

src/urg\_node\_driver.cpp

src/urg\_node.cpp

)

roslint\_add\_test()

find\_package(roslaunch REQUIRED)

roslaunch\_add\_file\_check(launch/urg\_lidar.launch)

endif()