cmake\_minimum\_required(VERSION 2.8.3)

project(astra\_camera)

find\_package(catkin REQUIRED camera\_info\_manager dynamic\_reconfigure image\_transport nodelet sensor\_msgs roscpp message\_generation)

find\_package(Boost REQUIRED COMPONENTS system thread)

find\_package(libuvc REQUIRED)

#MACHINE = $(shell uname -m)

execute\_process(COMMAND uname -m OUTPUT\_VARIABLE MACHINES )

execute\_process(COMMAND getconf LONG\_BIT OUTPUT\_VARIABLE MACHINES\_BIT )

MESSAGE(STATUS "ORRBEC Machine : ${MACHINES}")

MESSAGE(STATUS "ORRBEC Machine Bits : ${MACHINES\_BIT}")

IF ( (${MACHINES} MATCHES "x86\_64") AND (${MACHINES\_BIT} MATCHES "64") )

set(HOST\_PLATFORM "x64")

# for schroot enviroment

ELSEIF ( (${MACHINES} MATCHES "x86\_64") AND (${MACHINES\_BIT} MATCHES "32") )

set(HOST\_PLATFORM "x86")

ELSEIF ( ${MACHINES} MATCHES "x86" )

ELSEIF ( ${MACHINES} MATCHES "x86" )

set(HOST\_PLATFORM "x86")

ELSEIF ( ${MACHINES} MATCHES "i686" )

set(HOST\_PLATFORM "x86")

ELSEIF ( ${MACHINES} MATCHES "i386" )

set(HOST\_PLATFORM "x86")

ELSEIF ( ${MACHINES} MATCHES "arm" )

set(HOST\_PLATFORM "arm")

ELSEIF ( (${MACHINES} MATCHES "aarch64") AND (${MACHINES\_BIT} MATCHES "64") )

set(HOST\_PLATFORM "arm64")

ELSEIF ( (${MACHINES} MATCHES "aarch64") AND (${MACHINES\_BIT} MATCHES "32") )

set(HOST\_PLATFORM "arm")

ENDIF ()

message(STATUS "ORRBEC : ${HOST\_PLATFORM}")

message(STATUS "libuvc ${libuvc\_VERSION\_MAJOR}.${libuvc\_VERSION\_MINOR}.${libuvc\_VERSION\_PATCH}")

generate\_dynamic\_reconfigure\_options(cfg/Astra.cfg cfg/UVCCamera.cfg)

add\_service\_files(

FILES

GetSerial.srv

GetDeviceType.srv

GetIRGain.srv

GetCameraInfo.srv

GetUVCExposure.srv

GetIRExposure.srv

GetUVCGain.srv

GetUVCWhiteBalance.srv

SetUVCWhiteBalance.srv

SetUVCGain.srv

SetIRExposure.srv

SetIRGain.srv

SetIRFlood.srv

SetLaser.srv

SetLDP.srv

SetUVCExposure.srv

ResetIRGain.srv

ResetIRExposure.srv

SwitchIRCamera.srv

)

generate\_messages(

DEPENDENCIES

std\_msgs sensor\_msgs

)

catkin\_package(

INCLUDE\_DIRS include

LIBRARIES astra\_wrapper

CATKIN\_DEPENDS camera\_info\_manager dynamic\_reconfigure image\_transport nodelet sensor\_msgs roscpp message\_runtime

#DEPENDS libastra

)

add\_definitions(-Dlibuvc\_VERSION\_MAJOR=${libuvc\_VERSION\_MAJOR})

add\_definitions(-Dlibuvc\_VERSION\_MINOR=${libuvc\_VERSION\_MINOR})

add\_definitions(-Dlibuvc\_VERSION\_PATCH=${libuvc\_VERSION\_PATCH})

set(ORBBEC\_OPENNI2\_REDIST "${CMAKE\_CURRENT\_SOURCE\_DIR}/include/openni2\_redist/${HOST\_PLATFORM}")

link\_directories(${ORBBEC\_OPENNI2\_REDIST}

${catkin\_LINK\_DIRS}

)

#MESSAGE(STATUS "ORRBEC : ${ORBBEC\_OPENNI2\_REDIST}")

include\_directories(include

${catkin\_INCLUDE\_DIRS}

${Boost\_INCLUDE\_DIRS}

${CMAKE\_CURRENT\_SOURCE\_DIR}/include/openni2

${libuvc\_INCLUDE\_DIRS}

)

add\_library(astra\_wrapper

src/astra\_convert.cpp

src/astra\_device.cpp

src/astra\_device\_info.cpp

src/astra\_timer\_filter.cpp

src/astra\_frame\_listener.cpp

src/astra\_device\_manager.cpp

src/astra\_exception.cpp

src/astra\_video\_mode.cpp

src/astra\_device\_type.cpp

)

target\_link\_libraries(astra\_wrapper ${catkin\_LIBRARIES} -lOpenNI2 -L${ORBBEC\_OPENNI2\_REDIST}

${Boost\_LIBRARIES} )

add\_dependencies(astra\_wrapper ${PROJECT\_NAME}\_gencfg ${PROJECT\_NAME}\_generate\_messages\_cpp)

add\_library(astra\_driver\_lib

src/astra\_driver.cpp

src/astra\_device\_type.cpp

)

target\_link\_libraries(astra\_driver\_lib astra\_wrapper ${catkin\_LIBRARIES} ${Boost\_LIBRARIES} )

add\_dependencies(astra\_driver\_lib ${PROJECT\_NAME}\_gencfg ${PROJECT\_NAME}\_generate\_messages\_cpp)

add\_library(astra\_camera\_nodelet

ros/astra\_camera\_nodelet.cpp

)

target\_link\_libraries(astra\_camera\_nodelet astra\_driver\_lib ${catkin\_LIBRARIES} ${Boost\_LIBRARIES} )

add\_dependencies(astra\_camera\_nodelet ${PROJECT\_NAME}\_gencfg ${PROJECT\_NAME}\_generate\_messages\_cpp)

add\_executable(astra\_camera\_node

ros/astra\_camera\_node.cpp

)

target\_link\_libraries(astra\_camera\_node astra\_driver\_lib ${catkin\_LIBRARIES} ${Boost\_LIBRARIES} )

add\_dependencies(astra\_camera\_node ${PROJECT\_NAME}\_gencfg ${PROJECT\_NAME}\_generate\_messages\_cpp)

add\_executable(astra\_list\_devices

src/list\_devices.cpp

)

target\_link\_libraries(astra\_list\_devices astra\_wrapper)

add\_executable(astra\_test\_wrapper test/test\_wrapper.cpp )

target\_link\_libraries(astra\_test\_wrapper astra\_wrapper ${Boost\_LIBRARIES})

if (UNIX AND NOT APPLE)

add\_executable(astra\_usb\_reset src/usb\_reset.c)

set(ADDITIONAL\_EXECUTABLES "astra\_usb\_reset")

endif()

add\_executable(camera\_node src/libuvc\_camera/main.cpp

src/libuvc\_camera/camera\_driver.cpp

src/astra\_device\_type.cpp)

target\_link\_libraries(camera\_node ${libuvc\_LIBRARIES} ${Boost\_LIBRARIES} ${catkin\_LIBRARIES})

add\_dependencies(camera\_node ${PROJECT\_NAME}\_gencfg ${PROJECT\_NAME}\_generate\_messages\_cpp)

add\_library(libuvc\_camera\_nodelet src/libuvc\_camera/nodelet.cpp

src/libuvc\_camera/camera\_driver.cpp

src/astra\_device\_type.cpp)

target\_link\_libraries(libuvc\_camera\_nodelet ${libuvc\_LIBRARIES} ${Boost\_LIBRARIES} ${catkin\_LIBRARIES})

add\_dependencies(libuvc\_camera\_nodelet ${PROJECT\_NAME}\_gencfg ${PROJECT\_NAME}\_generate\_messages\_cpp)

install(TARGETS astra\_wrapper astra\_camera\_nodelet astra\_camera\_node astra\_list\_devices astra\_driver\_lib ${ADDITIONAL\_EXECUTABLES}

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

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

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

)

install(FILES include/openni2\_redist/${HOST\_PLATFORM}/libOpenNI2.so

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

)

install(DIRECTORY include/openni2\_redist/${HOST\_PLATFORM}/OpenNI2

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

)

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

# DESTINATION ${CATKIN\_PACKAGE\_INCLUDE\_DESTINATION}

#)

# add xml file

install(FILES astra\_nodelets.xml

DESTINATION ${CATKIN\_PACKAGE\_SHARE\_DESTINATION}

)

install(FILES 56-orbbec-usb.rules

DESTINATION ${CATKIN\_PACKAGE\_LIB\_DESTINATION}/${PROJECT\_NAME}

)

install(DIRECTORY scripts

DESTINATION ${CATKIN\_PACKAGE\_LIB\_DESTINATION}/${PROJECT\_NAME}

)

install(FILES 56-orbbec-usb.rules

DESTINATION ${CATKIN\_PACKAGE\_SHARE\_DESTINATION}

)

install(DIRECTORY scripts

DESTINATION ${CATKIN\_PACKAGE\_SHARE\_DESTINATION}

)

install(TARGETS camera\_node libuvc\_camera\_nodelet

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

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

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

)