#000000cmake\_minimum\_required(VERSION 2.8.3)

#include($ENV{ROS\_ROOT}/core/rosbuild/rosbuild.cmake)

cmake\_minimum\_required(VERSION 2.8.3)

project(wittenstein\_msgs)

find\_package(catkin REQUIRED COMPONENTS roslib

message\_generation std\_msgs sensor\_msgs )

add\_message\_files(

FILES

wittenstein.msg

)

generate\_messages(DEPENDENCIES std\_msgs sensor\_msgs)

catkin\_package(

DEPENDS

CATKIN\_DEPENDS roslib message\_runtime std\_msgs sensor\_msgs

INCLUDE\_DIRS

LIBRARIES

)

include\_directories(

${catkin\_INCLUDE\_DIRS}

)

#add\_dependencies(tactile\_servo\_msgs ${catkin\_EXPORTED\_TARGETS})

# 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})