cmake\_minimum\_required(VERSION 3.0.2)

project(wittenstein)

## Compile as C++11, supported in ROS Kinetic and newer

add\_compile\_options(-std=c++11)

## Find catkin macros and libraries

## if COMPONENTS list like find\_package(catkin REQUIRED COMPONENTS xyz)

## is used, also find other catkin packages

find\_package(catkin REQUIRED COMPONENTS

roscpp

std\_msgs

wittenstein\_msgs

)

# Generate messages in the 'msg' folder

# add\_message\_files(

# FILES

# )

## Generate services in the 'srv' folder

# add\_service\_files(

# FILES

# Service1.srv

# Service2.srv

# )

## Generate actions in the 'action' folder

# add\_action\_files(

# FILES

# Action1.action

# Action2.action

# )

## Generate added messages and services with any dependencies listed here

# generate\_messages(

# DEPENDENCIES

# std\_msgs

# )

## Generate dynamic reconfigure parameters in the 'cfg' folder

# generate\_dynamic\_reconfigure\_options(

# cfg/DynReconf1.cfg

# cfg/DynReconf2.cfg

# )

catkin\_package(

INCLUDE\_DIRS include

LIBRARIES ${PROJECT\_NAME}

CATKIN\_DEPENDS roscpp std\_msgs wittenstein\_msgs

#DEPENDS system\_lib

)

###########

## Build ##

###########

## Specify additional locations of header files

## Your package locations should be listed before other locations

include\_directories(

${catkin\_INCLUDE\_DIRS}

include

)

#add\_executable(wittenstein\_main src/main.cpp src/HexFT.cpp)

#target\_link\_libraries(wittenstein\_main ${catkin\_LIBRARIES})

add\_executable(wittenstein\_main src/main.cpp src/HexFT.cpp)

add\_dependencies(wittenstein\_main ${${PROJECT\_NAME}\_EXPORTED\_TARGETS} ${catkin\_EXPORTED\_TARGETS})

target\_link\_libraries(wittenstein\_main

${catkin\_LIBRARIES}

)

add\_executable(wittenstein\_listener src/listener\_wittenstein.cpp)

add\_dependencies(wittenstein\_listener ${${PROJECT\_NAME}\_EXPORTED\_TARGETS} ${catkin\_EXPORTED\_TARGETS})

target\_link\_libraries(wittenstein\_listener

${catkin\_LIBRARIES}

)