#!/usr/bin/env python

PACKAGE='astra\_camera'

from dynamic\_reconfigure.parameter\_generator\_catkin import \*

gen = ParameterGenerator()

# TODO Only offer modes supported by known hardware

output\_mode\_enum = gen.enum([ gen.const( "SXGA\_30Hz", int\_t, 1, "1280x1024@30Hz"),

gen.const( "SXGA\_15Hz", int\_t, 2, "1280x1024@15Hz"),

gen.const( "XGA\_30Hz", int\_t, 3, "1280x720@30Hz"),

gen.const( "XGA\_15Hz", int\_t, 4, "1280x720@15Hz"),

gen.const( "VGA\_30Hz", int\_t, 5, "640x480@30Hz"),

gen.const( "VGA\_25Hz", int\_t, 6, "640x480@25Hz"),

gen.const( "QVGA\_25Hz", int\_t, 7, "320x240@25Hz"),

gen.const( "QVGA\_30Hz", int\_t, 8, "320x240@30Hz"),

gen.const( "QVGA\_60Hz", int\_t, 9, "320x240@60Hz"),

gen.const( "QQVGA\_25Hz", int\_t, 10, "160x120@25Hz"),

gen.const( "QQVGA\_30Hz", int\_t, 11, "160x120@30Hz"),

gen.const( "QQVGA\_60Hz", int\_t, 12, "160x120@60Hz"),

gen.const("640400\_30Hz", int\_t, 13, "640x400@30Hz"),

gen.const("320200\_30Hz", int\_t, 14, "320x200@30Hz"),

gen.const("1280800\_7Hz", int\_t, 15, "1280x800@7Hz"),

gen.const("1280800\_30Hz", int\_t, 16, "1280x800@30Hz"),

gen.const("640400\_60Hz", int\_t, 17, "640x400@60Hz")],

"output mode")

video\_stream\_enum = gen.enum([ gen.const( "RGB", bool\_t, True, "RGB video stream preferred"),

gen.const( "IR", bool\_t, False, "IR video stream preferred")],

"preferred video stream mode")

gen.add("rgb\_preferred", bool\_t, 0, "Preferred camera stream", True, edit\_method = video\_stream\_enum)

gen.add("ir\_mode", int\_t, 0, "Video mode for IR camera", 5, 1, 17, edit\_method = output\_mode\_enum)

gen.add("color\_mode", int\_t, 0, "Video mode for color camera", 5, 1, 17, edit\_method = output\_mode\_enum)

gen.add("depth\_mode", int\_t, 0, "Video mode for depth camera", 5, 1, 17, edit\_method = output\_mode\_enum)

gen.add("depth\_registration", bool\_t, 0, "Depth data registration", True)

gen.add("color\_depth\_synchronization", bool\_t, 0, "Synchronization of color and depth camera", False)

gen.add("auto\_exposure", bool\_t, 0, "Auto-Exposure", True)

gen.add("auto\_white\_balance", bool\_t, 0, "Auto-White-Balance", True)

gen.add("data\_skip", int\_t, 0, "Skip N images for every image published (rgb/depth/depth\_registered/ir)", 0, 0, 10)

gen.add("ir\_time\_offset", double\_t, 0, "ir image time offset in seconds", -0.033, -1.0, 1.0 );

gen.add("color\_time\_offset", double\_t, 0, "color image time offset in seconds", -0.033, -1.0, 1.0 );

gen.add("depth\_time\_offset", double\_t, 0, "depth image time offset in seconds", -0.033, -1.0, 1.0 );

gen.add("depth\_ir\_offset\_x", double\_t, 0, "X offset between IR and depth images", 5.0, -20.0, 20.0)

gen.add("depth\_ir\_offset\_y", double\_t, 0, "Y offset between IR and depth images", 4.0, -20.0, 20.0)

gen.add("z\_offset\_mm", int\_t, 0, "Z offset in mm", 0, -200, 200)

gen.add("z\_scaling", double\_t, 0, "Scaling factor for depth values", 1.0, 0.5, 1.5)

gen.add("use\_device\_time", bool\_t, 0, "Use internal timer of OpenNI device", True)

exit(gen.generate(PACKAGE, "Astra", "Astra"))