

```
#!/usr/bin/env python
import rospy
from CodeArchitecture import controllerPID
from sensor_msgs.msg import Imu
from std_msgs.msg import String
from std_msgs.msg import Int32
import numpy as np

arr = np.array([0, 0, 0, 0, 0, 0, 0, 0])
myCon = controllerPID()
firstRun = True

def callback(data):
    #rospy.loginfo(data.data)
    global firstRun
    if firstRun:
        myCon.initializePosition([data.orientation.x, data.orientation.y, data.orientation.z, data.orientation.w], 0)
        firstRun = False
    angles = myCon.convertToAngles([data.orientation.x, data.orientation.y, data.orientation.z, data.orientation.w], 0)
    #rospy.loginfo("%f, %f, %f", angles[0]*180/np.pi, angles[1]*180/np.pi, angles[2]*180/np.pi)
    rospy.loginfo(myCon.getMatrix())
    myCon.update(0, 0, 3, 0, 0, 0, angles[0]*180/np.pi, 0)

def listener():

    # In ROS, nodes are uniquely named. If two nodes with the same
    # name are launched, the previous one is kicked off. The
    # anonymous=True flag means that rospy will choose a unique
    # name for our 'listener' node so that multiple listeners can
    # run simultaneously.
    rospy.init_node('listener', anonymous=True)

    rospy.Subscriber("imu/data", Imu, callback)

    # spin() simply keeps python from exiting until this node is stopped
    #   rospy.spin()

def talker():
    pub0 = rospy.Publisher('motor1', Int32, queue_size=10)
    pub1 = rospy.Publisher('motor2', Int32, queue_size=10)
    pub2 = rospy.Publisher('motor3', Int32, queue_size=10)
    pub3 = rospy.Publisher('motor4', Int32, queue_size=10)
    pub4 = rospy.Publisher('motor5', Int32, queue_size=10)
    pub5 = rospy.Publisher('motor6', Int32, queue_size=10)
    pub6 = rospy.Publisher('motor7', Int32, queue_size=10)
    pub7 = rospy.Publisher('motor8', Int32, queue_size=10)

    #rospy.init_node('talker', anonymous=True)
    rate = rospy.Rate(10) # 10hz
    while not rospy.is_shutdown():

        pub0.publish(arr[0])
        pub1.publish(arr[1])
        pub2.publish(arr[2])
        pub3.publish(arr[3])
        pub4.publish(arr[4])
        pub5.publish(arr[5])
        pub6.publish(arr[6])
        pub7.publish(arr[7])

        rate.sleep()

if __name__ == '__main__':
    listener()
    talker()
```