#!/usr/bin/env python3

import rospy

# we are sending Int32 message

from std\_msgs.msg import Int32

# this is our node name

nodeName='messagepublisher’

# these are the topic names- Make sure that in the Arduino code, the subscriber nodes are subscribed to these exact topics

topicNameLeftMotor='left motor velocity’

topicNameRightMotor= ‘right motor velocity’

# here, we initialize our subscriber node

# we set "anonymous True" to make sure that the node has a unique name 16# this parameter will add random numbers to the end of the node name

rospy.init\_node(nodeName, anonymous=True)

# here, we are saying that our node is publishing messages to the specified topic names

# we specify the type of messages we want to publish (Int32)

# queue\_size=5 simply means that we limit the number of queued messages if the subscriber cannot receive the messages fast enough

publisherLeftMotor=rospy.Publisher(topicNameLeftMotor, Int32, queue\_size=5)

publisherRightMotor=rospy.Publisher(topicNameRightMotor, Int32, queue\_size=5)

#here we specify the frequency of publishing the messages

#that is, we publish the messages with 1 [Hz]

ratePublisher=rospy.Rate(1)

# here we initialize the integers that we plan to send

# The values are from 0 255

# velocity of the left motor

LeftMotor = 0

# velocity of the right motor

rightMotor = 0

while not rospy.is\_shutdown():

rospy.loginfo(leftMotor)

rospy.loginfo(RightMotor)

leftMotor = int(input("Enter left motor velocity (0-255: "))

rightMotor = int(input("Enter right motor velocity (0-255: "))

publisherLeftMotor.publish(leftMotor)

publisherRightMotor.publish(rightMotor)

ratePublisher.sleep()

#!/usr/bin/env python3

import rospy

# we are sending Int32 message

from std\_msgs.msg import Int32

#this the node name

nodeName='messagesubs'

#these are the topic names

#these topic names have to match the topic names specified in the Arduino code

topicNameLeftEncoder='left\_encoder\_pulses'

topicNameRightEncoder='right\_encoder\_pulses'

#These are the call back functions, that are called when the messages are received

#These functions will print the encoder readings to the screen

def callBackFunctionLeftEncoder(message1):

print("Left encoder pulses: %s" message1.data)

def callBackFunctionRightEncoder(message2):

print("right encoder pulses: %s" message2.data)

# here, we initialize our subscriber node

#we set "anonymous=True" to make sure that the node has a unique name # this parameter will add random numbers to the end of the node name

rospy.init\_node(nodeName, anonymous=True)

# here, we subscribe to the appropriate topics

#we specify the type of the message we want to receive # and we specify the call back functions

rospy.Subscriber(topicNameLeftEncoder, Int32, callBackFunctionLeftEncoder)

rospy.Subscriber(topicNameRightEncoder, Int32, callBackFunctionRightEncoder)

# spin the code

rospy.spin()