Лабораторная работа 6. ROS Turtlesim

Студент \_\_\_\_\_\_

Вариант 6

Цель работы

Знакомство со средой ROS, работа с пакетом Turtlesim.

Описание работы

С помощью пакета Turtlesim нарисовать черепашкой свой табельный номер в ИСУ. Решений задачи несколько. Небольшие погрешности в выводе ряда цифр допускаются. Выбор цвета фона, цвета и размера ручки произвольный.

Номер ИСУ 280257

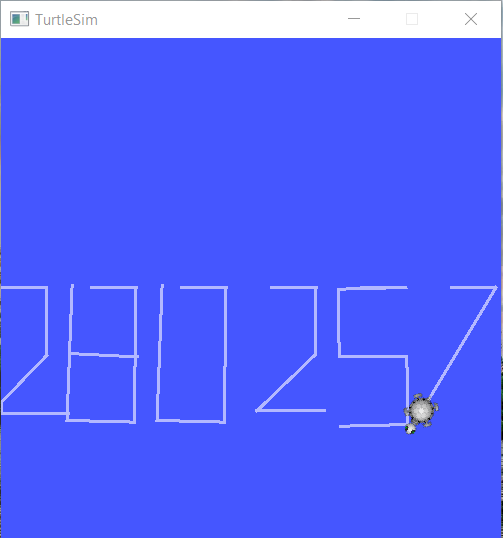


Рис 1. Результат выполнения скрипта mv\_script.py.

**Листинг mv\_script.py.**

#!/usr/bin/env python

import rospy

from geometry\_msgs.msg import Twist

from turtlesim.srv import \*

def forward():

returns

def make\_moves():

rospy.init\_node('move\_turtle', anonymous=True)

pub = rospy.Publisher('/turtle1/cmd\_vel', Twist, queue\_size=10)

rate = rospy.Rate(1)

vel = Twist()

vel.linear.x = 0

vel.linear.y = 0

vel.linear.z = 0

vel.angular.x = 0

vel.angular.y = 0

vel.angular.z = 0

# print 2

new\_turt(-5)

vel = move\_rotate(vel, pub, rate, 0.5, 1, 90, -1)

vel = move\_rotate(vel, pub, rate, 0.5, 1, 45, -1)

vel = move\_rotate(vel, pub, rate, 0.6, 1, 135, 1)

vel = move(vel, pub, rate, 0.5, 1)

# print 8

new\_turt(-3)

for i in range(2):

vel = move\_rotate(vel, pub, rate, 0.5, 1, 90, -1)

vel = move\_rotate(vel, pub, rate, 1, 1, 90, -1)

vel = rotate(vel, pub, rate, 90, -1)

vel = move\_rotate(vel, pub, rate, 0.5, 1, 90, 1)

vel = move(vel, pub, rate, 0.5, 1)

# print 0

new\_turt(-1)

for i in range(2):

vel = move\_rotate(vel, pub, rate, 0.5, 1, 90, -1)

vel = move\_rotate(vel, pub, rate, 1, 1, 90, -1)

# print 2

new\_turt(1)

vel = move\_rotate(vel, pub, rate, 0.5, 1, 90, -1)

vel = move\_rotate(vel, pub, rate, 0.5, 1, 45, -1)

vel = move\_rotate(vel, pub, rate, 0.6, 1, 135, 1)

vel = move(vel, pub, rate, 0.5, 1)

# print 5

new\_turt(3)

vel = move\_rotate(vel, pub, rate, 0.5, 1, 180, -1)

vel = move\_rotate(vel, pub, rate, 0.5, 1, 90, 1)

vel = move\_rotate(vel, pub, rate, 0.5, 1, 90, 1)

vel = move\_rotate(vel, pub, rate, 0.5, 1, 90, -1)

vel = move\_rotate(vel, pub, rate, 0.5, 1, 90, -1)

vel = move(vel, pub, rate, 0.5, 1)

# print 7

new\_turt(5)

vel = move\_rotate(vel, pub, rate, 0.5, 1, 120, -1)

vel = move(vel, pub, rate, 1.1, 1)

def new\_turt(offset):

killTurtle('turtle1')

spawnTurtle(5+offset, 5.544445, 0, 'turtle1')

def move\_rotate(vel, pub, rate, speed, direction, angle, ang\_direction):

vel = move(vel, pub, rate, speed, direction)

vel = rotate(vel, pub, rate, angle, ang\_direction)

return vel

def move(vel, pub, rate, speed, direction):

for i in range(3):

vel.linear.x = direction\*speed

pub.publish(vel)

rate.sleep()

vel.linear.x = 0

pub.publish(vel)

return vel

def rotate(vel, pub, rate, angle, direction):

for i in range(1):

vel.angular.z = direction\*angle \* 3.1415926535 / 180

pub.publish(vel)

rate.sleep()

pub.publish(vel)

vel.angular.z = 0

return vel

if \_\_name\_\_ == "\_\_main\_\_":

try:

spawnTurtle = rospy.ServiceProxy('/spawn', Spawn)

killTurtle = rospy.ServiceProxy('/kill', Kill)

make\_moves()

except rospy.ROSInterruptException:

pass