МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра вычислительные системы и технологии

Лабораторная работа № 1

Введение в программирование алгоритмов управления роботами с использованием симулятора Webots

ОТЧЕТ

по лабораторной работе

по дисциплине

Аппаратное и программное обеспечение

робототехнических систем

Вариант 19

РУКОВОДИТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Гай В.Е.

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Сапожников В.О.

19-ИВТ-3

Работа защищена «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

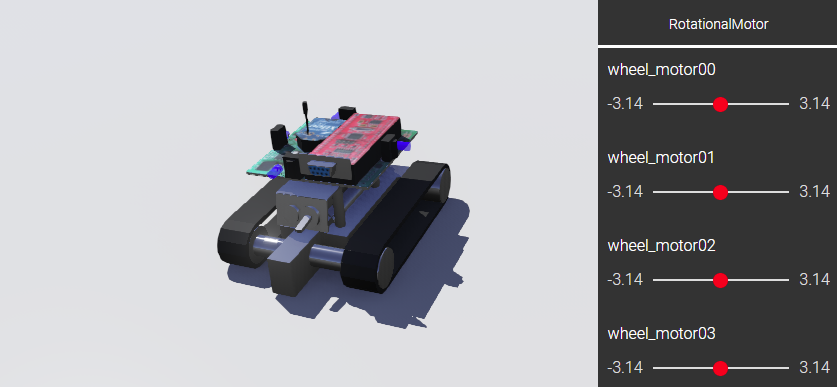
Нижний Новгород 2021

**Цель работы**: получение навыков работы с симулятором Webots

**Задание**

1. Добавить робота, указанного в списке ниже и соответствующего выданному варианту в симулятор Webots
2. Создать для робота контроллер таким образом, чтобы робот двигался не по прямой (для этого можно для разных моторов робота установить разную скорость)

Робот Surveyor SRV-1

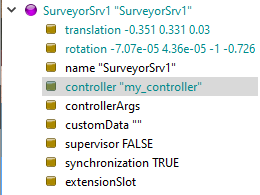


Данный робот имеет 2 гусеницы для каждой из которых используются 5 двигателей.

|  |  |
| --- | --- |
| Левая гусеница | Правая гусеница |
| wheel\_motor00 | wheel\_motor05 |
| wheel\_motor01 | wheel\_motor06 |
| wheel\_motor02 | wheel\_motor07 |
| wheel\_motor03 | wheel\_motor08 |
| wheel\_motor04 | wheel\_motor09 |

Чтобы заставить такого робота двигаться не по прямой достаточно подать разные скорости на гусеницы.

Для этого создадим свой контроллер на ЯП python. Для этого в настройках окружения выберем нашего робота, добавим новый контроллер и выберем его.



Для получения доступа к мотору (как и к другим составляющим робота) необходимо воспользоваться методом getDevice() объекта робот и передать название составляющей в качестве параметра.

Затем мы должны указать желаемую достижимую позицию при помощи метода setPosition() объекта мотор. Укажем значение равное inf для того, чтобы робот всегда был в движении.

И наконец каждому двигателю зададим скорость при помощи метода setVelocity().

# create the Robot instance.

robot = Robot()

# get the time step of the current world.

timestep = int(robot.getBasicTimeStep())

MAX\_SPEED = 10.0

# You should insert a getDevice-like function in order to get the

# instance of a device of the robot. Something like:

# motor = robot.getDevice('motorname')

# ds = robot.getDevice('dsname')

# ds.enable(timestep)

# Левая гусеница

Motor\_r0 = robot.getDevice('wheel\_motor00')

Motor\_r0.setPosition(float('inf'))

Motor\_r0.setVelocity(0.4 \* MAX\_SPEED)

Motor\_r1 = robot.getDevice('wheel\_motor01')

Motor\_r1.setPosition(float('inf'))

Motor\_r1.setVelocity(0.4 \* MAX\_SPEED)

Motor\_r2 = robot.getDevice('wheel\_motor02')

Motor\_r2.setPosition(float('inf'))

Motor\_r2.setVelocity(0.4 \* MAX\_SPEED)

Motor\_r3 = robot.getDevice('wheel\_motor03')

Motor\_r3.setPosition(float('inf'))

Motor\_r3.setVelocity(0.4 \* MAX\_SPEED)

Motor\_r4 = robot.getDevice('wheel\_motor04')

Motor\_r4.setPosition(float('inf'))

Motor\_r4.setVelocity(0.4 \* MAX\_SPEED)

# Правая гусеница

Motor\_r5 = robot.getDevice('wheel\_motor05')

Motor\_r5.setPosition(float('inf'))

Motor\_r5.setVelocity(0.25 \* MAX\_SPEED)

Motor\_r6 = robot.getDevice('wheel\_motor06')

Motor\_r6.setPosition(float('inf'))

Motor\_r6.setVelocity(0.25 \* MAX\_SPEED)

Motor\_r7 = robot.getDevice('wheel\_motor07')

Motor\_r7.setPosition(float('inf'))

Motor\_r7.setVelocity(0.25 \* MAX\_SPEED)

Motor\_r8 = robot.getDevice('wheel\_motor08')

Motor\_r8.setPosition(float('inf'))

Motor\_r8.setVelocity(0.25 \* MAX\_SPEED)

Motor\_r9 = robot.getDevice('wheel\_motor09')

Motor\_r9.setPosition(float('inf'))

Motor\_r9.setVelocity(0.25 \* MAX\_SPEED)

# Main loop:

# - perform simulation steps until Webots is stopping the controller

while robot.step(timestep) != -1:

Таким образом мы заставил нашего робота двигаться по кругу

