

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 \_\_\_\_\_

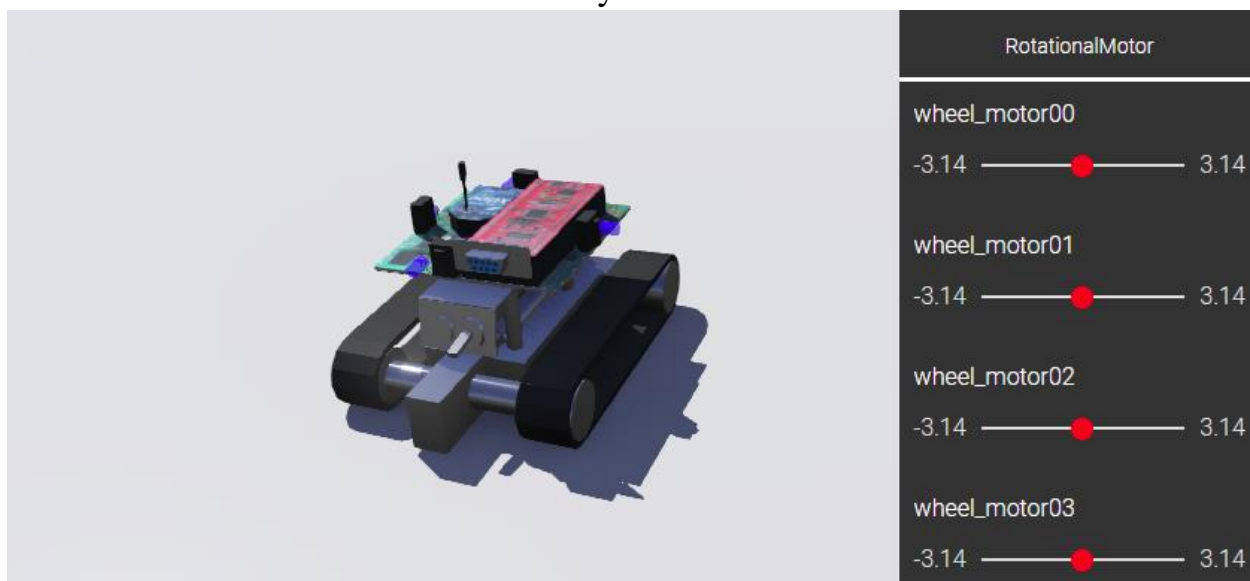
Нижний Новгород 2021

**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

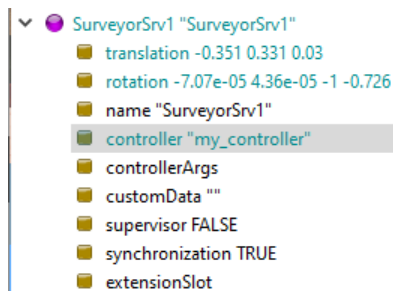


Данный робот имеет 2 гусеницы для каждой из которых используются 5 двигателей.

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 `robot` 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

