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»

ОТЧЁТ

По лабораторной работе №3

по дисциплине «Аппаратное и программное обеспечение   
роботизированных систем»

«Программное обеспечение  
роботизированных систем»

ПРОВЕРИЛ:

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

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ильин М.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Савин Д.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Полякова И.

**17-В-1**

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

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

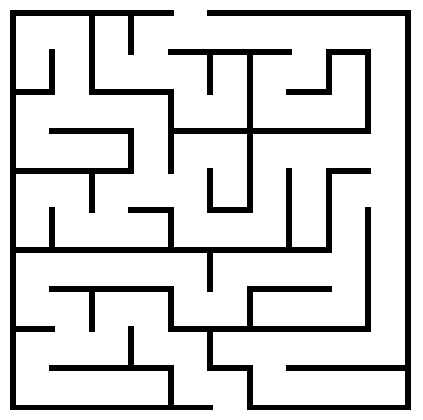
Нижний Новгород

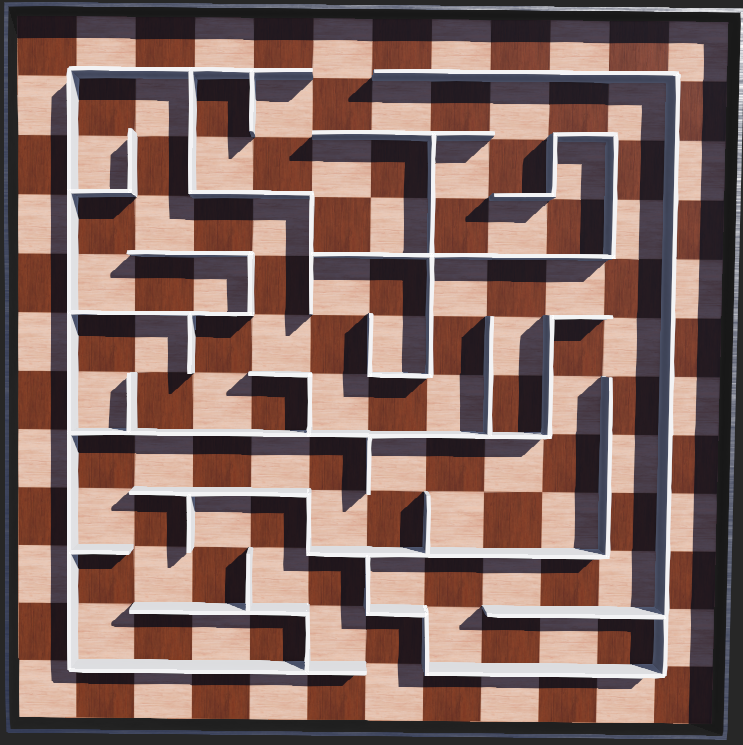
2020

**Цель:** написать программу для робота по вариантам в среде симуляции Webots.

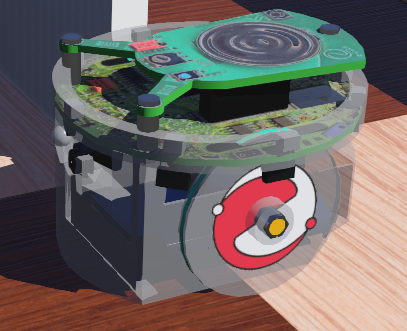
**Вариант:** **обход лабиринта:** робот должен найти выход из лабиринта.

**Ход работы:** для начала был создан небольшой лабиринт 10х10 при помощи сайта <http://www.mazegenerator.net/>





Для выполнения задания был взят робот «e-puck». <https://cyberbotics.com/doc/guide/epuck>



Алгоритм был разработан на базе правила «правой руки», взятого с сайта <https://myrobot.ru/articles/logo_mazesolving.php>

**Описание алгоритма работы робота:**

Движение роботов контролируется двумя датчиками расстояния. В зависимости от показаний датчиков выбирается направление движения робота. Начальное положение робота такое, что датчик справа направлен в стену. Далее, если робот «видит стену», то едет вперёд. Если стена закончилась, то выполняется небольшой алгоритм: робот проезжает немного вперёд, затем поворачивает направо на 90 градусов и едет вперёд до стены. Однако, он едет до стены не по показанию датчика, а по расстоянию, которое было вычислено опытным путём. Если же стена справа есть, но впереди есть препятствие, то робот поворачивается налево на 90 градусов.

Данный алгоритм является решением только для односвязных лабиринтов – это такой лабиринт, в котором нет замкнутых маршрутов.

**Код контроллера:**

from controller import Robot

MAX\_SPEED = 6.28 # максимальная скорость

robot = Robot()

timestep = int(robot.getBasicTimeStep())

leftMotor = robot.getMotor('left wheel motor') # подключение левого

rightMotor = robot.getMotor('right wheel motor') # и правого мотора

leftMotor.setPosition(float('inf'))

rightMotor.setPosition(float('inf'))

leftMotor.setVelocity(0.0)

rightMotor.setVelocity(0.0)

rightFor = robot.getDistanceSensor('ps0') # подключаем датчики расстояния:

rightFor.enable(timestep) # правый и передний правый,

right = robot.getDistanceSensor('ps2') # а также активируем их

right.enable(timestep)

# счётчики для выполнения определённых действий

cntForward=0 # для движения вперёд после потери препятствия

cntLeft=0 # для поворота налево

cntRight=0 # для поворота направо

cntForwardAfterRotation=0 # для движения вперёд после поворота

# коэффициенты скорости моторов

leftSpeed = 0.0

rightSpeed = 0.0

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

rf = rightFor.getValue() # получение данных

r = right.getValue() # с сенсоров

leftMotor.setVelocity(leftSpeed \* MAX\_SPEED) # задание скорости

rightMotor.setVelocity(rightSpeed \* MAX\_SPEED) # на моторах

# движение вперёд после потери препятствия для далнейшего поворота

if cntForward > 0:

leftSpeed = 0.4

rightSpeed = 0.4

cntForward-=1

continue

# выполнение поворота направо на 90 градусов

if cntRight > 0:

leftSpeed = 0.1

rightSpeed = -0.1

cntRight-=1

continue

# движение вперёд после поворота направо, чтобы найти стену справа

if cntForwardAfterRotation > 0:

leftSpeed = 0.2

rightSpeed = 0.2

cntForwardAfterRotation-=1

continue

# выполнение поворота налево на 90 градусов

if cntLeft > 0:

cntLeft-=1

continue

# проверка показаний датчиков для задания определённых действий

# если справа и спереди есть препятствие,

# то задаём коэффициенты скорости

# и количество итераций для поворота налево

if r > 100 and rf > 290:

leftSpeed = -0.1

rightSpeed = 0.1

cntLeft = 109

# если справа стена и спереди ничего не мешает, то ехать прямо

elif r > 100:

leftSpeed = 0.4

rightSpeed = 0.4

# если робот потерял стену справа,

# то задаём количество итераций,

# чтобы выполнить небольшой проезд вперёд,

# повернуть направо и ехать вперёд до

# обнаружение препятствия

elif r < 100:

cntForward = 24

cntRight = 110

cntForwardAfterRotation = 61

pass