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 №2

по дисциплине «Аппаратное и программное обеспечение   
роботизированных систем»

«Знакомство с симулятором Webots»

ПРОВЕРИЛ:

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

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Львутина Т.В.

**17-В-1**

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

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

Нижний Новгород

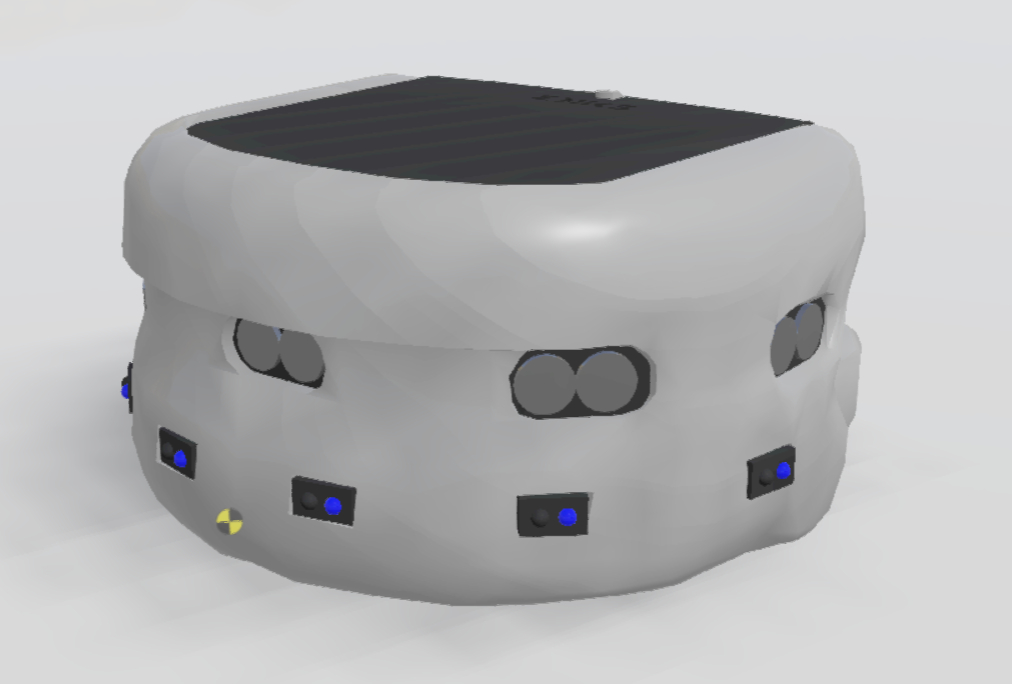
2020

**Цель:** познакомиться с симулятором Webots. Реализовать для конкретного робота (по вариантам) программу движения, а также создать новый мир и добавить в него объекты.

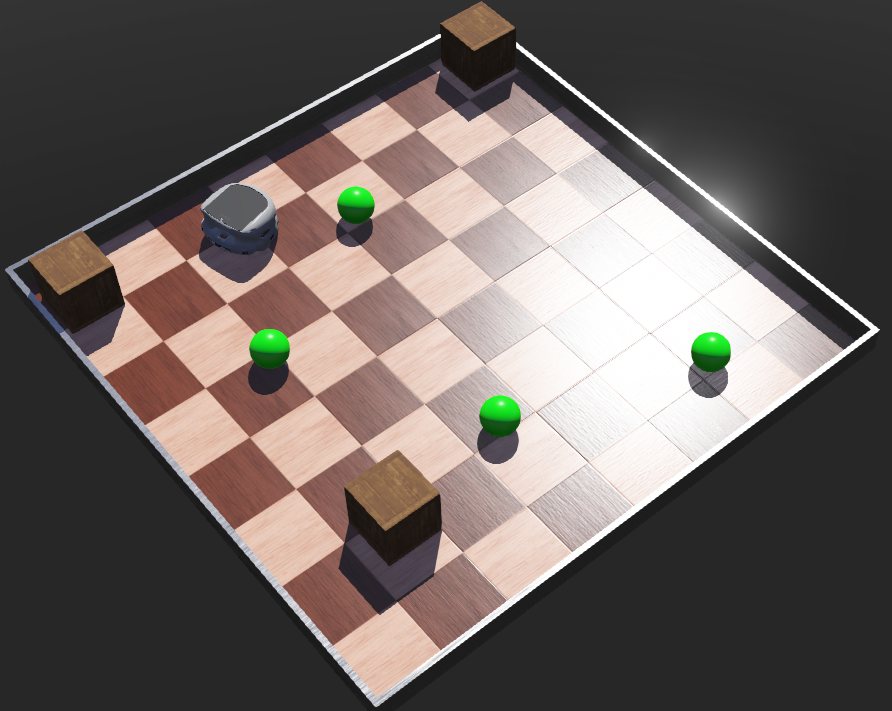
**Вариант:** **робот** K-Team's Khepera III**.**

**О роботе:**

Робот "Khepera III" - двухколесный робот производства [K-Team](https://www.k-team.com/mobile-robotics-products/old-products/khepera-iii) . Он монтируется несколькими датчиками, включая 8 датчиков расстояния.



**Мир.** Реализованный мир представлен ниже, а также к данному отчету приложен: видео – пример моделирования, а также сам созданный сим.мир в формате данного симулятора.



**Текст контроллера робота:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <webots/camera.h>

#include <webots/distance\_sensor.h>

#include <webots/motor.h>

#include <webots/robot.h>

#define MAX\_SENSOR\_NUMBER 16

#define RANGE (1024 / 2)

#define BOUND(x, a, b) (((x) < (a)) ? (a) : ((x) > (b)) ? (b) : (x))

static WbDeviceTag sensors[MAX\_SENSOR\_NUMBER], camera, left\_motor, right\_motor;

static double matrix[MAX\_SENSOR\_NUMBER][2];

static int num\_sensors;

static double range;

static int time\_step = 0;

static double max\_speed = 0.0;

static double speed\_unit = 1.0;

static int camera\_enabled;

static void initialize() {

wb\_robot\_init();

time\_step = wb\_robot\_get\_basic\_time\_step();

const char \*robot\_name = wb\_robot\_get\_name();

const char e\_puck\_name[] = "ps0";

const char khepera\_name[] = "ds0";

const char koala\_name[] = "ds0";

const char pioneer2\_name[] = "ds0";

char sensors\_name[5];

const double(\*temp\_matrix)[2];

camera\_enabled = 0;

range = RANGE;

const double e\_puck\_matrix[8][2] = {{150, -35}, {100, -15}, {80, -10}, {-10, -10},

{-10, -10}, {-10, 80}, {-30, 100}, {-20, 150}};

const double khepera3\_matrix[9][2] = {{-5000, -5000}, {-20000, 40000}, {-30000, 50000}, {-70000, 70000}, {70000, -60000},

{50000, -40000}, {40000, -20000}, {-5000, -5000}, {-10000, -10000}};

const double khepera\_matrix[8][2] = {{-2, 4}, {-3, 5}, {-7, 7}, {7, -6}, {5, -4}, {4, -2}, {-0.5, -0.5}, {-0.5, -0.5}};

const double pioneer2\_matrix[16][2] = {{-1, 15}, {-3, 13}, {-3, 8}, {-2, 7}, {-3, -4}, {-4, -2}, {-3, -2}, {-1, -1},

{-1, -1}, {-2, -3}, {-2, -4}, {4, -3}, {7, -5}, {7, -3}, {10, -2}, {11, -1}};

const double koala\_matrix[16][2] = {{17, -1}, {8, -2}, {4, -3}, {9, -2}, {5, -3}, {-4, -2}, {-4, -2}, {-2, -2},

{-2, -2}, {-2, -4}, {-2, -4}, {-4, 5}, {-3, 8}, {-3, 5}, {-2, 10}, {-1, 15}};

if (strncmp(robot\_name, "e-puck", 6) == 0) {

const double epuck\_max\_speed = 6.28;

const double epuck\_speed\_unit = 0.00628;

num\_sensors = 8;

sprintf(sensors\_name, "%s", e\_puck\_name);

temp\_matrix = e\_puck\_matrix;

max\_speed = epuck\_max\_speed;

speed\_unit = epuck\_speed\_unit;

if (strcmp(robot\_name, "e-puck camera") == 0)

camera\_enabled = 1;

} else if (strncmp(robot\_name, "Khepera III", 8) == 0) {

const double khepera3\_max\_speed = 19.1;

const double khepera3\_speed\_unit = 0.00053429;

num\_sensors = 9;

sprintf(sensors\_name, "%s", khepera\_name);

temp\_matrix = khepera3\_matrix;

range = 2000;

max\_speed = khepera3\_max\_speed;

speed\_unit = khepera3\_speed\_unit;

} else if (strncmp(robot\_name, "khepera", 7) == 0) {

const double khepera\_max\_speed = 1.0;

const double khepera\_speed\_unit = 1.0;

num\_sensors = 8;

sprintf(sensors\_name, "%s", khepera\_name);

temp\_matrix = khepera\_matrix;

max\_speed = khepera\_max\_speed;

speed\_unit = khepera\_speed\_unit;

} else if (strcmp(robot\_name, "koala") == 0) {

const double koala\_max\_speed = 10.0;

const double koala\_speed\_unit = 0.1;

num\_sensors = 16;

sprintf(sensors\_name, "%s", koala\_name);

temp\_matrix = koala\_matrix;

max\_speed = koala\_max\_speed;

speed\_unit = koala\_speed\_unit;

} else if (strcmp(robot\_name, "pioneer2") == 0) {

const double pioneer2\_max\_speed = 10.0;

const double pioneer2\_speed\_unit = 0.1;

num\_sensors = 16;

sprintf(sensors\_name, "%s", pioneer2\_name);

temp\_matrix = pioneer2\_matrix;

max\_speed = pioneer2\_max\_speed;

speed\_unit = pioneer2\_speed\_unit;

} else {

fprintf(stderr, "This controller doesn't support this robot\n");

exit(EXIT\_FAILURE);

}

int i;

for (i = 0; i < num\_sensors; i++) {

sensors[i] = wb\_robot\_get\_device(sensors\_name);

wb\_distance\_sensor\_enable(sensors[i], time\_step);

if ((i + 1) >= 10) {

sensors\_name[2] = '1';

sensors\_name[3]++;

if ((i + 1) == 10) {

sensors\_name[3] = '0';

sensors\_name[4] = '\0';

}

} else

sensors\_name[2]++;

int j;

for (j = 0; j < 2; j++)

matrix[i][j] = temp\_matrix[i][j];

}

left\_motor = wb\_robot\_get\_device("left wheel motor");

right\_motor = wb\_robot\_get\_device("right wheel motor");

wb\_motor\_set\_position(left\_motor, INFINITY);

wb\_motor\_set\_position(right\_motor, INFINITY);

wb\_motor\_set\_velocity(left\_motor, 0.0);

wb\_motor\_set\_velocity(right\_motor, 0.0);

if (camera\_enabled == 1) {

camera = wb\_robot\_get\_device("camera");

wb\_camera\_enable(camera, time\_step);

}

printf("The %s robot is initialized, it uses %d distance sensors\n", robot\_name, num\_sensors);

}

int main() {

initialize();

while (wb\_robot\_step(time\_step) != -1) {

int i, j;

double speed[2];

double sensors\_value[MAX\_SENSOR\_NUMBER];

if (camera\_enabled == 1)

wb\_camera\_get\_image(camera);

for (i = 0; i < num\_sensors; i++)

sensors\_value[i] = wb\_distance\_sensor\_get\_value(sensors[i]);

for (i = 0; i < 2; i++) {

speed[i] = 0.0;

for (j = 0; j < num\_sensors; j++) {

speed[i] += speed\_unit \* matrix[j][i] \* (1.0 - (sensors\_value[j] / range));

}

speed[i] = BOUND(speed[i], -max\_speed, max\_speed);

}

wb\_motor\_set\_velocity(left\_motor, speed[0]);

wb\_motor\_set\_velocity(right\_motor, speed[1]);

}

return 0;

**Выводы:** в рамках выполнения данной ЛР был опробован симулятор Webots, написан контроллер для робота, а также изучены некоторые главы документации и гайды.