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»

ОТЧЕТ

по лабораторной работе №2

по дисциплине «Аппаратное и программное обеспечение

роботизированных систем»

РУКОВОДИТЕЛЬ:

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

СТУДЕНТ:

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

17-В-1

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

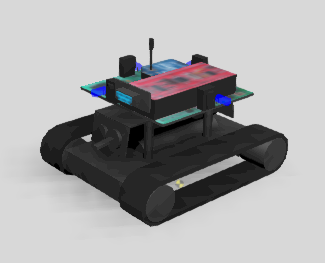
С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород

2020

Задача: привести в действие робота, но не по прямой линии.

Робот: Surveyor SRV – 1



Базовая станция подключается к ПК по USB и работает с любой ОС, поддерживающей Java. При этом с одного ПК можно контролировать несколько SRV-1 (это пригодится, если нужно охватить большую площадь - проходимость у бота неважная). Радиус действия радиоканала в помещении равен 100 м, а на открытой местности - 1000 м.

"Мозг" SRV-1 - 32-битный процессор ARM7TDMI, производительность которого составляет 60 MIPS. Габариты устройства - 100 x 70 x 120 мм, а весит оно всего 400 г (рама изготовлена из алюминия). Заряда литий-ионной батареи, по заверениям производителя, должно хватить на 4 часа дежурства.

Сообщается, что робот может работать и в автономном режиме, курсируя по заданному маршруту. ПО для таких операций, как обнаружение движений, слежение за объектом, избежание столкновений и базовая навигация, находится в разработке. Исходный код "прошивки" (firmware) робота, написанный на C, доступен как Open Source. Таким образом, даже если выпуск SRV-1 прекратится, энтузиасты смогут продлить его поддержку своими силами.

**Текст контроллера робота:**

from controller import Robot

TIME\_STEP = 164

robot = Robot()

ds = []

dsNames = ['ds0', 'ds1', 'ds2','ds3']

for i in range(4):

ds.append(robot.getDistanceSensor(dsNames[i]))

ds[i].enable(TIME\_STEP)

wheels = []

wheelsNames = ['wheel\_motor00', 'wheel\_motor01', 'wheel\_motor02', 'wheel\_motor03', 'wheel\_motor04', 'wheel\_motor05', 'wheel\_motor06', 'wheel\_motor07', 'wheel\_motor08', 'wheel\_motor09']

for i in range(10):

wheels.append(robot.getMotor(wheelsNames[i]))

wheels[0].setPosition(float('inf'))

wheels[0].setVelocity(0.0)

wheels[1].setPosition(float('inf'))

wheels[1].setVelocity(0.0)

wheels[2].setPosition(float('inf'))

wheels[2].setVelocity(0.0)

wheels[3].setPosition(float('inf'))

wheels[3].setVelocity(0.0)

wheels[4].setPosition(float('inf'))

wheels[4].setVelocity(0.0)

wheels[5].setPosition(float('inf'))

wheels[5].setVelocity(0.0)

wheels[6].setPosition(float('inf'))

wheels[6].setVelocity(0.0)

wheels[7].setPosition(float('inf'))

wheels[7].setVelocity(0.0)

wheels[8].setPosition(float('inf'))

wheels[8].setVelocity(0.0)

wheels[9].setPosition(float('inf'))

wheels[9].setVelocity(0.0)

avoidObstacleCounter = 0

while robot.step(TIME\_STEP) != -1:

leftSpeed = 10.0

rightSpeed = 10.0

if avoidObstacleCounter > 0:

avoidObstacleCounter -= 1

leftSpeed = 10.0

rightSpeed = -1.0

else:

for i in range(4):

if ds[i].getValue() < 950.0:

avoidObstacleCounter = 100

wheels[0].setVelocity(leftSpeed)

wheels[1].setVelocity(rightSpeed)

wheels[2].setVelocity(leftSpeed)

wheels[3].setVelocity(rightSpeed)

wheels[4].setVelocity(leftSpeed)

wheels[5].setVelocity(rightSpeed)

wheels[6].setVelocity(leftSpeed)

wheels[7].setVelocity(rightSpeed)

wheels[8].setVelocity(leftSpeed)

wheels[9].setVelocity(rightSpeed)

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